Derby

Version 10.4

Derby Document build:
August 10, 2009, 1:12:00 PM (PDT)

Contents

Copyright.....7

License.....8

########.....12
#######.....12
#######.....12
#######.....12

SQL#########.....13
##########.....13
SQL#### 13
SQL92######.....13
SQL92###.....14
##.....14
###.....14
#####.....15
#####.....15
####.....15
##.....16
##.....16
####.....16
###.....16
###.....16
#####.....17
TriggerName.....17
#####.....17
#.....17
####.....17
ALTER TABLE #.....18
CALL (###) #.....22
CREATE #.....22
DECLARE GLOBAL TEMPORARY TABLE #.....34
DELETE #.....36
DROP#.....36
GRANT # 38
INSERT #.....39
LOCK TABLE #.....40
RENAME #.....41
REVOKE # 42
SET #.....44
SELECT #.....45
UPDATE#.....47
SQL #.....48
CONSTRAINT #.....48
FOR UPDATE #.....52
FROM #.....53
GROUP BY #.....53
HAVING #.....54
ORDER BY #.....54
WHERE #.....55
WHERE CURRENT OF #.....56
SQL#.....56

###.....58
TableExpression.....60
VALUES#.....60
#########.....61
###.....61
#######.....63
JOIN ##.....65
INNER JOIN ##.....65
LEFT OUTER JOIN##.....66
RIGHT OUTER JOIN ##.....67
SQL #####.....68
###.....68
########.....69
#####.....70
######.....71
#######.....71
(####).....72
ABS#ABSVAL##.....73
ACOS ##.....73
ASIN ##.....73
ATAN ##.....73
AVG##.....73
BIGINT##.....74
CASE #.....74
CAST ##.....75
CEIL###CEILING##.....78
CHAR ##.....78
Concatenation.....79
COS ##.....80
COUNT ##.....80
COUNT(*) ##.....80
CURRENT DATE ##.....81
CURRENT_DATE##.....81
CURRENT ISOLATION ##.....81
CURRENT SCHEMA##.....81
CURRENT TIME.....81
CURRENT_TIME##.....81
CURRENT TIMESTAMP##.....82
CURRENT_TIMESTAMP##.....82
CURRENT_USER##.....82
DATE##.....82
DAY ##.....83
DEGREES ##.....83
DOUBLE##.....83
EXP##.....83
FLOOR##.....84
HOUR##.....84
IDENTITY_VAL_LOCAL##.....84
INTEGER##.....86
LCASE####LOWER##.....86
LENGTH##.....86
LN####LOG##.....87
LOG10 ##.....87
LOCATE##.....87
LTRIM##.....88
MAX##.....88

MIN## ... 88
MINUTE ## ... 89
MOD## ... 89
MONTH## ... 89
NULLIF# ... 90
PI ## ... 90
RADIANS ## ... 90
RTRIM## ... 90
SECOND## ... 91
SESSION_USER## ... 91
SIN ## ... 91
SMALLINT## ... 91
SQRT## ... 92
SUBSTR## ... 92
SUM## ... 93
TAN ## ... 93
TIME ## ... 93
TIMESTAMP## ... 94
TRIM ## ... 94
UCASE#####UPPER## ... 95
USER## ... 95
VARCHAR## ... 96
XMLEXISTS ### ... 96
XMLPARSE### ... 97
XMLQUERY### ... 98
XMLSERIALIZE ### ... 99
YEAR## ... 100
######### ... 100
SYSCS_UTIL.SYSCS_CHECK_TABLE###### ... 100
SYSCS_UTIL.SYSCS_GET_DATABASE_PROPERTY###### ... 100
SYSCS_UTIL.SYSCS_GET_RUNTIMESTATISTICS ###### ... 101
########## ... 101
SYSCS_DIAG ######## ... 116
... 118
SQL### ... 134
Derby#######SQL-92### ... 138
Derby###### ... 146
SYSALIASES ##### ... 146
SYSCHECKS##### ... 146
SYSCOLPERMS ##### ... 147
SYSCOLUMNS##### ... 147
SYSCONGLOMERATES ##### ... 148
SYSCONSTRAINTS ##### ... 148
SYSDEPENDS ##### ... 148
SYSFILES##### ... 149
SYSFOREIGNKEYS##### ... 149
SYSKEYS##### ... 149
SYSROUTINEPERMS ##### ... 149
SYSSCHEMAS ##### ... 150
SYSSTATISTICS##### ... 150
SYSSTATEMENTS##### ... 151
SYSTABLEPERMS ##### ... 151
SYSTABLES ##### ... 152
SYSTRIGGERS##### ... 152
SYSVIEWS ##### ... 153
Derby#########SQL state ... 153

SQL#############.....153
JDBC ######.....185
java.sql####JDBC####### ##############.....185
java.sql.Driver########.....185
java.sql.DriverManager.getConnection ####.....186
java.sql.Connection ########.....188
java.sql.DatabaseMetaData ########.....188
java.sql.Statement########.....190
java.sql.CallableStatement ########.....190
java.sql.SQLException ###.....192
java.sql.PreparedStatement########.....192
java.sql.ResultSet ########.....193
java.sql.ResultSetMetaData ########.....194
java.sql.SQLWarning ###.....194
java.sql.SQLXML########.....194
SQL##java.sql.Types###.....194
JDBC 2.0 ###.....197
java.sql.Connection ########: JDBC 2.0##########.....197
java.sql.DatabaseMetaData ########: #####JDBC 2.0#####.....197
java.sql.PreparedStatement########: JDBC2.0##########.....197
java.sql.ResultSet.....197
java.sql.ResultSetMetaData ########: JDBC 2.0##########.....198
java.sql.Statement ########: #####JDBC 2.0#####.....198
java.sql.BatchUpdateException ###.....199
Connected Device Configuration###Foundation Profile###JDBC#####(JSR169).....199
JDBC 3.0###.....199
java.sql.Connection########: JDBC3.0########## 200
java.sql.DatabaseMetaData ########: JDBC 3.0##########.....200
java.sql.ParameterMetaData########:JDBC3.0########## 201
java.sql.PreparedStatement########: JDBC3.0##########.....201
java.sql.Savepoint ########.....201
java.sql.Statement########: JDBC 3.0##########.....202
JDBC 4.0###.....203
#####SQLException######.....204
java.sql.Connection########:JDBC4.0########.....204
java.sql.DatabaseMetaData########: JDBC4.0###.....204
java.sql.Statement########: JDBC 4.0########.....205
javax.sql.DataSource ########: JDBC 4.0###.....205
JDBC########.....205
######JDBC##########.....205
JDBC#######.....206
LIKE##JDBC#######.....206
fn#########JDBC#######.....206
#####JDBC#######.....209
#####JDBC#######.....210
#####JDBC#######.....210
##########JDBC#######.....210

########URL#########.....212
bootPassword=key##.....212
collation=collation ##.....212
create=true ##.....213
createFrom=Path##.....213
databaseName=nameofDatabase##.....213
dataEncryption=true##.....214

encryptionKey=key## ... 214
encryptionProvider=providerName## ... 215
encryptionAlgorithm=algorithm ## ... 215
logDevice=logDirectoryPath ## ... 216
newEncryptionKey=key ## ... 216
newBootPassword=newPassword ## ... 216
password=userPassword ... 216
restoreFrom=path## ... 217
rollForwardRecoveryFrom=path ## ... 217
shutdown=true## ... 217
territory=ll_CC ## ... 218
traceDirectory=path ## ... 218
traceFile=path ## ... 219
traceFileAppend=true ## ... 219
traceLevel=value ## ... 220
upgrade=true attribute ... 221
user=userName ## ... 221
ssl=sslMode ## ... 221
#################### ... 221

J2EE####:Java Transaction API#javax.sql ######## ... 222
JTA API ... 222
########## ... 222
javax.sql:JDBC######## ... 223

Derby API ... 224
#################### ... 224
JDBC###### ... 224
JDBC driver ... 224
########## ... 224
################## ... 225

######### ... 226

Derby##### ... 227
############ ... 227
DATE#TIME#TIMESTAMP### ... 227
######## ... 228
... 228
... 229
XML### ... 229

Trademarks ... 230

Apache Software FoundationDerby ############Apache Derby

Copyright

Related information

License

License

The Apache License, Version 2.0

that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

   You may add Your own copyright statement to Your modifications

and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

```
Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied.  See the License for the specific language governing
permissions and limitations under the License.
```

########

Derby##############*Derby*##

#######

##Derby #############Derby##########
#######Derby#SQL####Derby####JDBC####Derby##########Derby##########Derby#####SQL###

#######

######Derby## SQL####Java##################

#Derby############################Derby###############*Derby* ########
###############

#######

#################

- *SQL#########*

  Derby#SQL############ ############################

- *SQL###*

  SQL-92#####SQL#############

- *Derby#######SQL-92###*

  SQL-92#######Derby############################

- *Derby######*

  Derby####################

- *Derby#########SQL state*

  Derby#########################

- *JDBC ######*

  Derby#####JDBC#############JDBC 2.0 ################

- *########URL#########*

  Derby#JDBC########URL##########################

- *J2EE####:Java Transaction API#javax.sql ########*

  Java Transaction API#####Derby######################

- *Derby API*

  Derby###API#############

SQL#########

Derby#SQL-92#####################SQL-99###########

###################################SQL###########

###########

JDBC##########SQL##########Derby#SQL#############
SQL###############Unicode#####################

- ######SQL-92##
- ######################
- ######################################2######(################################)

  ##Java##################

  #:

  ```
  -- ###############-- #########

  VALUES 'Joe''s umbrella'
  -- ij###################
  VALUES 'He said, "hello!"'

  n = stmt.executeUpdate(
      "UPDATE aTable setStringcol = 'He said, \"hello!\"'");
  ```

- SQL#########################SELECT##########SELECT#Select#select#sELECT########
- SQL-92#########################(*SQL92###*##########)#########################
- Java############################
- *##*SelectExpression*.################*########*##########
  ###################################0####################
- %#_##LIKE#################################(###################)#######
- SQL-92###(--)###(/*)###############(*/)###

SQL####

##
SQL###################################

###SQL-92###*SQL92###*######.

SQL92######

###

########################(_)#0-9#################0-9######Unicode#####0-9##########Derby###

##

###

###########

```
"A.B"
```

#################

```
"A"."B"
```

###################################(####A##B########################)

SQL92###

SQL92######SQL-92##########################SQL-92########################128#######(#####################)################################ X0X11#######

Derby###SQL-92########################(SQL#######)

#

```
-- #######################
-- ANIDENTIFIER###
CREATE VIEW AnIdentifier (RECEIVED) AS VALUES 1
-- ########################
-- ###############
CREATE VIEW "ACaseSensitiveIdentifier" (RECEIVED) AS VALUES 1
```

#######*SQL92*#################################

#################

##

##

SQL###

###

- ####(CREATE TABLE #)
- ###############
- SELECT#########(#######)
- *TableExpression*######## (*TableExpression*###)

###############################SQL##############

```
SELECT c11 AS col1, c12 AS col2, c13 FROM t1 FOR UPDATE of c11,c13
```

###########c11#col1############ c11#FOR UPDATE ##################c12####################FOR UPDATE###########

##

```
[ { table-Name | correlation-Name } . ] SQL92Identifier
```

#

```
-- C.Country #
###########
--#####
SELECT C.Country
FROM APP.Countries C
```

###

####FROM###

############################SQL######### ####

```
SELECT c11 AS col1, c12 AS col2, c13 FROM t1 FOR UPDATE of c11,c13
```

##########c11#col1#################c11#FOR UPDATE##########
##########c12##############FOR UPDATE########

##

```
SQL92###
```

#

```
-- C#######
SELECT C.NAME
FROM SAMP.STAFF C
```

#####

##

##

```
SQL92###
```

#

```
-- FlightBooks ####################
RENAME TABLE FLIGHTAVAILABILITY TO FLIGHTAVAILABLE
```

#####

###

###################################*APP*##########(##############################)
SYS##########################

##############S

###

Syntax

```
SQL92###
```

#

```
-- SAMP.EMPLOYEE #################
SELECT COUNT(*) FROM SAMP.EMPLOYEE
-- ######################SYS##############
SELECT COUNT(*) FROM SYS.SysColumns
```

####

################################### #########CREATE
TABLE##############################

##

```
SQL92###
```

#

```
-- country ########
```

```
CREATE TABLE CONTINENT (COUNTRY VARCHAR(26) NOT NULL PRIMARY KEY,
COUNTRY_ISO_CODE CHAR(2), REGION VARCHAR(26))
```


########## ##########

##

```
[ #####. ] SQL92###
```


##########

##

```
[ #####. ] SQL92###
```

#

```
-- SAMP.PROJECT ##############
SELECT COUNT(*) FROM SAMP.PROJECT
```


##########

##

```
[ #####. ] SQL92###
```

Example

```
-- ##################
SELECT COUNT(*) FROM SAMP.EMP_RESUME
```


##########SYS##########

##

```
[ ##### . ] SQL92###
```

#

```
DROP INDEX APP.ORIGINDEX;
-- OrigIndex###############
CREATE INDEX ORIGINDEX ON FLIGHTS (ORIG_AIRPORT)
```


#########

##

```
SQL92###
```

#

```
-- country_fk2 #######
```

```
CREATE TABLE DETAILED_MAPS (COUNTRY_ISO_CODE CHAR(2)
CONSTRAINT country_fk2 REFERENCES COUNTRIES)
```

#####

##################SQL##################################JDBC API##*Derby* ######## ############# ###############SQL##################

##############

##

```
SQL92###
```

#

```
stmt.executeUpdate("UPDATE SAMP.STAFF SET COMM = " +
"COMM + 20 " + "WHERE CURRENT OF " + ResultSet.getCursorName());
```

TriggerName

################################

##

```
[ ##### . ] SQL92###
```

#

```
DROP TRIGGER TRIG1
```

#####

Derby##Derby#####################################
#########*Derby* ######## #############

##

```
SQL92###
```

Example

```
CALL SYSCS_UTIL.SYSCS_SET_DATABASE_PROPERTY(
    'derby.database.fullAccessUsers', 'Amber,FRED')
```

#

##
####CREATE INDEX##################JDBC################################SQL######################## BY##WHERE##*Select*##############
##

################################JDBC## ##

##

####

Derby ######################################SQL#########################(#######)############

#######################################(#####################################) ###Derby##################### #################################### ###################(################)###D

#########-UPDATE WHERE CURRENT###########################UPDATE WHERE CURRENT##########

###########################DDL##########

##

###Derby#ij##################

```
ij> CREATE TABLE mytable (mycol INT);
0 rows inserted/updated/deleted
ij> INSERT INTO mytable VALUES (1), (2), (3);
3 rows inserted/updated/deleted
-- #####ij#prepare##########
-- ##################
ij> prepare p1 AS 'INSERT INTO MyTable VALUES (4)';
-- p1#mytable######;
ij> execute p1;
1 row inserted/updated/deleted
-- Derby##################
ij> CREATE INDEX i1 ON mytable(mycol);
0 rows inserted/updated/deleted
-- #########p1#############
ij> execute p1;
1 row inserted/updated/deleted
-- Derby#####p1################
ij> DROP TABLE mytable;
0 rows inserted/updated/deleted
-- Derby######################
-- ###p1##############
-- ######p1#############
ij> CREATE TABLE mytable (mycol INT);
0 rows inserted/updated/deleted
ij> INSERT INTO mytable VALUES (1), (2), (3);
3 rows inserted/updated/deleted
ij> execute p1;
1 row inserted/updated/deleted
-- p1###########Derby##################
-- #####################
ij> DROP TABLE mytable;
0 rows inserted/updated/deleted
-- p1######
-- #######################
-- ########
ij> execute p1;
ERROR 42X05: Table/View 'MYTABLE' does not exist.
```

ALTER TABLE

The ALTER TABLE ##############

- #########
- ##########
- #########
- ############
- VARCHAR, CHAR VARYING, and CHARACTER VARYING ########
- #######################(###################)

- ##################
- ##Null############
- ############

##

```
ALTER TABLE ##
{
    ADD COLUMN ### |
    ADD CONSTRAINT# |
    DROP [ COLUMN ] ## [ CASCADE | RESTRICT ]
    DROP { PRIMARY KEY | FOREIGN KEY ### | UNIQUE
  ### | CHECK ### | CONSTRAINT ### }
    ALTER [ COLUMN ] ##### |
    LOCKSIZE { ROW | TABLE }
}
```

###

```
########
[ ##### ]*
[ [ WITH ] DEFAULT ##### ]
```

###############################

#####

```
## SET DATA TYPE VARCHAR(integer)
 |
column-name SET INCREMENT BY ###
 |
column-name RESTART WITH ###
 |
column-name [ NOT ] NULL
 |
column-name [ WITH ] DEFAULT ###
```

######SET INCREMENT BY
###IDENTI

RESTART WITH ##########################RESTART WITH##GENERATED
BY DEFAULT################################GENERATED
BY
DEFAULT###
WITH###

```
CREATE TABLE tauto(i INT GENERATED BY DEFAULT AS IDENTITY, k INT)
CREATE UNIQUE INDEX tautoInd ON tauto(i)
INSERT INTO tauto(k) values 1,2
```

###

```
INSERT INTO tauto VALUES (3,3)
INSERT INTO tauto VALUES (4,4)
INSERT INTO tauto VALUES (5,5)
```

###########1##5#####################################3#################################
WITH 6 #ALTER TABLE########

```
ALTER TABLE tauto ALTER COLUMN i RESTART WITH 6
```

ALTER TABLE
##########################SELECT###"*"###

####

#################CREATE TABLE####################ALTER TABLE ADD COLUMN#################################NOT NULL##ALTER TABLE###################

CREATE TABLE############################null####################NOT NULL########(SQLSTATE 42831)

Note:
#################UPDATE###################################UPDATE#####################

#####
ALTER TABLE ADD CONSTRAINT#####################################ALTER TABLE##

- ##########################Derby###Derby#####
- ###############null################################

  ALTER TABLE ADD UNIQUE#PRIMARY KEY#################################PRIMARY KEY#C####################PRIMARY KEY(C)###################null##############NOT NULL#################

##############CONSTRAINT ######### ADD TABLE ADD CONSTRAINT#####################################

####

ALTER TABLE DROP COLUMN#################

COLUMN###################

CASCADE#RESTRICT############################CASCADE###

RESTRICT##

CASCADE#################################

DROP COLUMN RESTRICT############################ ##DROP COLUMN RESTRICT#######

#########################

sqlAuthorization#######################(DERBY-1909##########))

############################CASCADE/ RESTRICT###

#####

ALTER TABLE DROP CONSTRAINT#####################################*SYS.SYSCONSTRAINTS*################

##(#################)

#####################################

- VARCHAR################VARCHAR##############CHARACTER VARYING#CHAR VARYING###########

  ######################################

  ###

- ############################

##(SQLSTATE 42837)################SET INCREMENT default#################

- ##Null######## ##Null#################

  NOT NULL################################NULL###################

  NOT NULL#################################PRIMARY KEY#UNIQUE#########################

- ##########

######

###NULL##

##################CREATE TABLE ##########

##########

####################################LOCKSIZE##########################(################################ ###################ALTER TABLE###LOCKSIZE##########

#

```
-- ####################
-- ####################NULL######
-- ########
ALTER TABLE CITIES ADD COLUMN REGION VARCHAR(26)
CONSTRAINT NEW_CONSTRAINT CHECK (REGION IS NOT NULL);

-- ################
-- ####################
ALTER TABLE SAMP.DEPARTMENT
ADD CONSTRAINT NEW_UNIQUE UNIQUE (DEPTNO);

-- Cities###############
-- ###############
-- ####################
-- ##########
ALTER TABLE CITIES ADD CONSTRAINT COUNTRY_FK
Foreign Key (COUNTRY) REFERENCES COUNTRIES (COUNTRY);

-- #############
-- ############
CREATE TABLE ACTIVITIES (CITY_ID INT NOT NULL,
SEASON CHAR(2), ACTIVITY VARCHAR(32) NOT NULL);
-- ##########null############
-- #################
ALTER TABLE Activities ADD PRIMARY KEY (city_id, activity);

-- ##############city_id########
ALTER TABLE Cities DROP COLUMN city_id RESTRICT;
-- ############city_id########
ALTER TABLE Cities DROP COLUMN city_id CASCADE;

-- CITIES#############
ALTER TABLE Cities DROP CONSTRAINT Cities_PK;
-- CITIES##############
ALTER TABLE Cities DROP CONSTRAINT COUNTRIES_FK;
-- ####1###DEPTNO#######
ALTER TABLE SAMP.EMP_ACT ADD COLUMN DEPTNO INT DEFAULT 1;
-- VARCHAR#########
ALTER TABLE SAMP.EMP_PHOTO ALTER PHOTO_FORMAT SET DATA TYPE VARCHAR(30);
-- #############
ALTER TABLE SAMP.SALES LOCKSIZE TABLE;

-- MANAGER##NOT NULL########
ALTER TABLE Employees ALTER COLUMN Manager NULL;
-- SSN##NOT NULL########
ALTER TABLE Employees ALTER COLUMN ssn NOT NULL;
```

```
-- SALARY##########
ALTER TABLE Employees ALTER COLUMN Salary DEFAULT 1000.0
```

####

ALTER TABLE ##ALTER TABLE#########

CALL (###)

CALL (###)################## ##################

##

```
CALL ### ( [ # [, #]* ] )
```

#

```
CREATE PROCEDURE SALES.TOTAL_REVENUE(IN S_MONTH INTEGER,
IN S_YEAR INTEGER, OUT TOTAL DECIMAL(10,2))
PARAMETER STYLE JAVA READS SQL DATA LANGUAGE JAVA EXTERNAL NAME
'com.acme.sales.calculateRevenueByMonth';
CALL SALES.TOTAL_REVENUE(?,?,?);
```

CREATE

##############################CREATE#######

CREATE FUNCTION #

CREATE FUNCTION############Java##########

##EXECUTE###

##

```
CREATE FUNCTION ### ( [ #####
   [, FunctionParameter] ] * ) RETURNS ###### [ ##### ] *
```

###

```
[ #####. ] SQL92###
```

###SYS############

#####

```
[ ### ] ####
```

######################

#####################

Note: CREATE FUNCTION##########################BLOB#CLOB#LONG VARCHAR#LONG VARCHAR FOR BIT DATA#XML####

######

```
## | ####
```

####################

##

```
TABLE( ### [, ### ]* )
```

######################Derby############ ####JDBC#*ResultSet*########
#########*Derby* ######## ##########Derby#######################

###

```
SQL92#########
```

##########################

Note: ##################XML#############

#####

```
 {
| LANGUAGE { JAVA }
| EXTERNAL NAME ###
| PARAMETER STYLE ########
| { NO SQL | CONTAINS SQL | READS SQL DATA }
| { RETURNS NULL ON NULL INPUT | CALLED ON NULL INPUT }
 }
```

LANGUAGE

JAVA- ###Java#######public static###############

EXTERNAL NAME *string*
######################Java##########################

```
####.#####
```

#################

########

```
JAVA | DERBY_JDBC_RESULT_SET
```

###Java#####SQL######################INOUT#OUT#####################################

Derby ######(######Long Varchar#BLOB#######)##########
############################

Derby########PARAMETER
STYLE###DERBY_JDBC_RESULT_SET############################JDBC#*ResultSet*####
#########PARAMETER STYLE###JAVA###

NO SQL, CONTAINS SQL, READS SQL DATA

######SQL###################SQL############

CONTAINS SQL
###############SQL######### ###################################
NO SQL
###SQL#########
READS SQL DATA
#############SQL############## ###################################
########

RETURNS NULL ON NULL INPUT or CALLED ON NULL INPUT
######null#########################null#######
RETURNS NULL ON NULL INPUT
#############null############################null######
CALLED ON NULL INPUT
#########null####################null###null#####

###

- **LANGUAGE**
- **PARAMETER STYLE**
- **EXTERNAL NAME**

Example

```
CREATE FUNCTION TO_DEGREES(RADIANS DOUBLE) RETURNS DOUBLE
PARAMETER STYLE JAVA NO SQL LANGUAGE JAVA
EXTERNAL NAME 'java.lang.Math.toDegrees'
```

CREATE INDEX #

CREATE INDEX################ ####################

##

```
CREATE [UNIQUE] INDEX ###
ON ## ( #### [ ASC | DESC ]
    [ , #### [ ASC | DESC ]] * )
```

The Derby###############16###

####128#############

#######CREATE INDEX##2##

Derby############################(*Derby*#################)#
###UNIQUE##################################

###############(###Derby######)
###

###Derby##############################ASC#########################
#####DESC#############Derby############# ##

#######################SYS#############

#####

##(####################)##########UNIQUE##
KEY###################################Derby###
##UNIQUE###UNIQUE

FLIGHTS_PK#################################

```
SELECT CONGLOMERATENAME FROM SYS.SYSCONGLOMERATES,
SYS.SYSCONSTRAINTS WHERE
SYS.SYSCONGLOMERATES.TABLEID = SYSCONSTRAINTS.TABLEID
AND CONSTRAINTNAME = 'FLIGHTS_PK'
```

```
CREATE INDEX OrigIndex ON Flights(orig_airport);
-- ##########################
-- ############
CREATE INDEX PAY_DESC ON SAMP.EMPLOYEE (SALARY);
-- ####################
call
 SYSCS_UTIL.SYSCS_SET_DATABASE_PROPERTY('derby.storage.pageSize','8192');
CREATE INDEX IXSALE ON SAMP.SALES (SALES);
call
 SYSCS_UTIL.SYSCS_SET_DATABASE_PROPERTY('derby.storage.pageSize',NULL);
```

############
Note:
###

####

SELECT#INSERT#UPDATE#UPDATE WHERE CURRENT#DELETE#DELETE##########CREATE INDEX##

CREATE PROCEDURE #

CREATE PROCEDURE####CALL PROCEDURE#######Java############

####################EXECUTE#####################################EXECUTE#############

##

```
CREATE PROCEDURE ### ( [ ######
   [, ######] ] * )
[ ###### ] *
```

###

```
[ #####. ] SQL92###
```

###SYS#############

#####

```
[ { IN | OUT | INOUT } ] [ ### ] ####
```

######IN#########################

#########################

Note: BLOB#CLOB#LONG VARCHAR#LONG VARCHAR FOR BIT DATA#XML#########CREATE PROCEDURE##############

#####

```
 {
| [ DYNAMIC ] RESULT SETS INTEGER
| LANGUAGE { JAVA }
| EXTERNAL NAME string
| PARAMETER STYLE JAVA
| { NO SQL | MODIFIES SQL DATA | CONTAINS SQL | READS SQL DATA }
 }
```

DYNAMIC RESULT SETS ##

##############################(0)###

LANGUAGE

JAVA- ###Java#######public static###############

EXTERNAL NAME ###
###############Java#############################

```
####.#####
```

####################

PARAMETER STYLE

JAVA -
###Java#SQL##########################INOUT###OUT#####################################

Derby ######(###Long Varchar, BLOB#######)##########
################################

NO SQL, CONTAINS SQL, READS SQL DATA, MODIFIES SQL DATA

###SQL######################SQL#############

CONTAINS SQL
############SQL###
NO SQL
########SQL##########
READS SQL DATA
#######SQL###
MODIFIES SQL DATA
##################SQL###################

###

- **LANGUAGE**
- **PARAMETER STYLE**
- **EXTERNAL NAME**

#

```
CREATE PROCEDURE SALES.TOTAL_REVENUE(IN S_MONTH INTEGER,
IN S_YEAR INTEGER, OUT TOTAL DECIMAL(10,2))
PARAMETER STYLE JAVA READS SQL DATA LANGUAGE JAVA EXTERNAL NAME
'com.acme.sales.calculateRevenueByMonth'
```

CREATE SCHEMA #

###

##

```
CREATE SCHEMA { [ ##### AUTHORIZATION #### ] | [ ##### ] |
[ AUTHORIZATION #### ] }
```

CREATE SCHEMA
############################128###

CREATE
SCHEMA####################derby.database.sqlAuthorization#true##################

```
AUTHORIZATION user-name
```

########
derby.database.sqlAuthorization#################*Derby*#################

CREATE SCHEMA##
######################anita###

```
CREATE SCHEMA FLIGHTS AUTHORIZATION anita
```

#################################

```
CREATE SCHEMA EMP
```

takumi##

```
CREATE SCHEMA AUTHORIZATION takumi
```

EMP#FLIGHTS#######availability###################

```
CREATE TABLE FLIGHTS.AVAILABILITY
 (FLIGHT_ID CHAR(6) NOT NULL, SEGMENT_NUMBER INT NOT NULL,
 FLIGHT_DATE DATE NOT NULL, ECONOMY_SEATS_TAKEN INT,
 BUSINESS_SEATS_TAKEN INT, FIRSTCLASS_SEATS_TAKEN INT,
 CONSTRAINT FLT_AVAIL_PK
 PRIMARY KEY (FLIGHT_ID, SEGMENT_NUMBER, FLIGHT_DATE))
```

```
CREATE TABLE EMP.AVAILABILITY
 (HOTEL_ID INT NOT NULL, BOOKING_DATE DATE NOT NULL, ROOMS_TAKEN INT,
 CONSTRAINT HOTELAVAIL_PK PRIMARY KEY (HOTEL_ID, BOOKING_DATE))
```

CREATE SYNONYM #

CREATE SYNONYM##
################################
######################SELECT#INSERT#UPDATE#DELETE#LOCK TABLE#################

###

###(SQLSTATE 01522) DML################################

#########(########)###(SQLSTATE 42916)

####################'SYS'################################Derby##############

##(SQLSTATE XCL51)

##

```
CREATE SYNONYM ### FOR { #### | ## }
```

###

#
CREATE SYNONYM SAMP.T1 FOR SAMP.TABLEWITHLONGNAME

CREATE TABLE #

CREATE
TABLE###

##

- INSERT
- SELECT
- REFERENCES
- TRIGGER
- UPDATE

################################

###########CONSTRAINT ###########

###NULL######################

SYSCS_UTIL.SYSCS_SET_DATABASE_PROPERTY###

###############*SYS*############

##

CREATE
TABLE###

```
CREATE TABLE ##
    ( {### | ####}
    [ , {### | ####} ] * )
  |
     [ ( ## [ , ## ] * ) ]
     AS ######
     WITH NO DATA
   }
```

#

```
CREATE TABLE HOTELAVAILABILITY
     (HOTEL_ID INT NOT NULL, BOOKING_DATE DATE NOT NULL,
 ROOMS_TAKEN INT DEFAULT 0, PRIMARY KEY (HOTEL_ID, BOOKING_DATE));
-- ###############2#################
```

```
PRIMARY KEY (hotel_id, booking_date))
-- #########INTEGER######
-- #################
CREATE TABLE PEOPLE
 (PERSON_ID INT NOT NULL GENERATED ALWAYS AS IDENTITY
 CONSTRAINT PEOPLE_PK PRIMARY KEY, PERSON VARCHAR(26));
-- #########SMALLINT######
-- ####5#####5#######
CREATE TABLE GROUPS
 (GROUP_ID SMALLINT NOT NULL GENERATED ALWAYS AS IDENTITY
 (START WITH 5, INCREMENT BY 5), ADDRESS VARCHAR(100), PHONE
 VARCHAR(15));
```

Note: ##################CREATE TABLE####CONSTRAINT #######

CREATE TABLE ... AS ...

#####CREATE
TABLE##################/
##

############################

WITH NO DATA##
WITH NO DATA#####################Derby##WITH
DATA###WITH
NO DATA############

#

```
-- ##########################
CREATE TABLE T3 AS SELECT * FROM T1 WITH NO DATA;
-- ########################################
CREATE TABLE T3 (A,B,C,D,E) AS SELECT * FROM T1 WITH NO DATA;
-- ################################################
CREATE TABLE T3 (A,B,C) AS SELECT V,DP,I FROM T1 WITH NO DATA;
-- ############################################################
CREATE TABLE T3 (X,Y) AS SELECT 2*I,2.0*F FROM T1 WITH NO DATA;
```

###:

```
########
    [ ##### ]*
    [ [ WITH ] DEFAULT #####
       |##### ]
    [ ##### ]*
```

##########################

##############CONSTRAINT ############

#####

##################################
##################################null###

```
#####:
          NULL
        | CURRENT { SCHEMA | SQLID }
        | USER | CURRENT_USER | SESSION_USER
        | DATE
        | TIME
        | TIMESTAMP
        | CURRENT DATE | CURRENT_DATE
        | CURRENT TIME | CURRENT_TIME
        | CURRENT TIMESTAMP | CURRENT_TIMESTAMP
        | literal
```

Derby###############################

##

- USER#CURRENT_USER#SESSION_USER##################8###########
- CURRENT SCHEMA#CURRENT SQLID##################128###########
- ################################
- ##10##

#####:

```
[ GENERATED { ALWAYS | BY DEFAULT } AS IDENTITY
[ ( START WITH ###
[ ,INCREMENT BY ###] ) ]  ]  ]
```

######

SMALLINT#INT#BIGINT##############Derby###
######################Derby##########################

IDENTITY##

- SMALLINT
- INT
- BIGINT

Derby###################GENERATED ALWAYS#######GENERATED BY DEFAULT###

GENERATED ALWAYS

GENERATED ALWAYS###GENERATED ALWAYS###DEFAULT###

```
create table greetings
 (i int generated always as identity, ch char(50));
insert into greetings values (DEFAULT, 'hello');
insert into greetings(ch) values ('bonjour');
```

GENERATED ALWAYS##

GENERATED BY DEFAULT

GENERATED BY DEFAULT###GENERATED ALWAYS#####################################

####################DEFAULT###

```
create table greetings
 (i int generated by default as identity, ch char(50));
-- "1"#####:
insert into greetings values (1, 'hi');
-- ###########
insert into greetings values (DEFAULT, 'salut');
-- ###########
insert into greetings(ch) values ('bonjour');
```

GENERATED ALWAYS#######GENERATED BY DEFAULT##############################hi###salut#########"1"###WITH###GENERATED BY DEFAULT################

#############1##########1### ####################START WITH#INCREMENT BY###Derby###########
################Derby############ ##0##############

##
Table 1. ########################

####	###	###
SMALLINT	32767 (*java.lang.Short.MAX_VALUE*)	-32768 (*java.lang.Short.MIN_VALUE*)
INT	2147483647 (*java.lang.Integer.MAX_VALUE*)	-2147483648 (*java.lang.Integer.MIN_VALUE*)
BIGINT	9223372036854775807 (*java.lang.Long.MAX_VALUE*)	-9223372036854775808 (*java.lang.Long.MIN_VALUE*)

##
#

IDENTITY_VAL_LOCAL##IDENTITY_VAL_LOCAL######

Note: ##

Derby##*SYS.SYSCOLUMNS*#########*AUTOINCREMEI*
##
Derby
##############*SYS.SYSCOLUMNS*###

######SQL####################SQL####################*ConnectionInfo*#########
######################################
####################SQL#######################
##########(##)#################
SQL####T1##############T1#####SQL###T2####################T1#T2####################
#########16#######

#

```
create table greetings
  (i int generated by default as identity (START WITH 2, INCREMENT BY 1),
  ch char(50));
-- 1######
insert into greetings values (1, 'hi');
-- ###########
insert into greetings values (DEFAULT, 'salut');
-- ###########
insert into greetings(ch) values ('bonjour');
```

CREATE TRIGGER #

##
###

##
##################################e-mail#####################

###

##
#########################TRIGGER################
##

######################################

#################*SYS*############

##

```
CREATE TRIGGER ####
{ AFTER | NO CASCADE BEFORE }
{ INSERT | DELETE | UPDATE [ OF ## [, ##]* ] }
ON ##
[ ### ]
[ FOR EACH { ROW | STATEMENT } ] [ MODE DB2SQL ]
######SQL#
```

##########

##################################

- ######################################
  ##(##############)
- ###################################
  ##(##############)

#########################

##(###############

- INSERT
- UPDATE
- DELETE

###

######################:###

################################SQL###################
#########SQL###################(###################)############

Derby###SQL#########
####################OLD/NEW AS #################

#########################

```
REFERENCING OLD AS DELETEDROW
```

######SQL###############

```
DELETE FROM HotelAvailability WHERE hotel_id = DELETEDROW.hotel_id
```

#####OLD#NEW##################*java.sql.ResultSet*#########
Note: ######################(############################)
#################OLD##########################NEW############

##################SQL################
##################OLD_TABLE/NEW_TABLE AS #################

####

```
REFERENCING OLD_TABLE AS DeletedHotels
```

#########(*DeletedHotels*)#######SQL############

```
DELETE FROM HotelAvailability WHERE hotel_id IN
    (SELECT hotel_id FROM DeletedHotels)
```

###*java.sql.ResultSet*#########
Note: #####(############################)###############
INSERT#########OLD###########DELETE#########NEW############

###
###

###########
###################### CREATE TRIGGER##FOR
EACH##

- #####

  ##
- #####

  ##

Note: ###################(####UPDATE T SET C = C##############)##

######SQL#

##########################SQL#######(#####################)

######SQL###############

- #####(?)############
- ##################################
- ##################################
- ##################################
- ##
- #############INSERT/UPDATE/DELETE#############
- ##################################

######SQL###

######SQL##########*Derby* ######## #############

#####

##########################Derby##############

- *###No Cascade Before*#################
- #####(##############)##########
- ########################
- *###After*############

######################(#################)#######################################

```
-- #####

CREATE TRIGGER t1 NO CASCADE BEFORE UPDATE ON x
  FOR EACH ROW MODE DB2SQL
  values app.notifyEmail('Jerry', 'Table x is about to be updated');

CREATE TRIGGER FLIGHTSDELETE
  AFTER DELETE ON FLIGHTS
  REFERENCING OLD_TABLE AS DELETEDFLIGHTS
  FOR EACH STATEMENT
  DELETE FROM FLIGHTAVAILABILITY WHERE FLIGHT_ID IN
  (SELECT FLIGHT_ID FROM DELETEDFLIGHTS);

CREATE TRIGGER FLIGHTSDELETE3
  AFTER DELETE ON FLIGHTS
  REFERENCING OLD AS OLD
  FOR EACH ROW
  DELETE FROM FLIGHTAVAILABILITY WHERE FLIGHT_ID = OLD.FLIGHT_ID;
```

Note: *Derby* ######## ####################

######

#########16#####

######

###

- CURRENT_DATE##
- CURRENT_TIME##
- CURRENT_TIMESTAMP##
- CURRENT_USER##

- SESSION_USER##
- USER##

###:

```
REFERENCING
{
{ OLD | NEW } [ ROW ] [ AS ] correlation-Name [ { OLD | NEW } [ ROW ] [
 AS ] correlation-Name ] |
{ OLD TABLE | NEW TABLE } [ AS ] Identifier [ { OLD TABLE | NEW TABLE }
 [AS] Identifier ]  |
{ OLD_TABLE | NEW_TABLE } [ AS ] Identifier [ { OLD_TABLE | NEW_TABLE }
 [AS] Identifier ]
}
```

Note: OLD_TABLE | NEW_TABLE#########DB2##################SQL#################
CREATE VIEW #

##

####################*SYS*############

############SELECT################SELECT##################**##########**##########SELEC

####################################SELECT################

##user2#########user1

SELECT##

##

```
CREATE VIEW ####
    [ ( #### [, ####] * ) ]
AS #####
```

##
###

```
CREATE VIEW SAMP.V1 (COL_SUM, COL_DIFF)
 AS SELECT COMM + BONUS, COMM - BONUS
 FROM SAMP.EMPLOYEE;

CREATE VIEW SAMP.VEMP_RES (RESUME)
 AS VALUES 'Delores M. Quintana', 'Heather A. Nicholls', 'Bruce Adamson';

CREATE VIEW SAMP.PROJ_COMBO
 (PROJNO, PRENDATE, PRSTAFF, MAJPROJ)
 AS SELECT PROJNO, PRENDATE, PRSTAFF, MAJPROJ
 FROM SAMP.PROJECT UNION ALL
SELECT PROJNO, EMSTDATE, EMPTIME, EMPNO
 FROM SAMP.EMP_ACT
 WHERE EMPNO IS NOT NULL;
```

####
###################### ########DML(data manipulation language)##############################
###
##########

```
CREATE TABLE T1 (C1 DOUBLE PRECISION);

CREATE FUNCTION SIN (DATA DOUBLE)
 RETURNS DOUBLE EXTERNAL NAME 'java.lang.Math.sin'
 LANGUAGE JAVA PARAMETER STYLE JAVA;

CREATE VIEW V1 (C1) AS SELECT SIN(C1) FROM T1;
```

##########################

```
SELECT * FROM V1
```

V1#######T1##### ##########SIN

DECLARE GLOBAL TEMPORARY TABLE

The DECLARE GLOBAL TEMPORARY TABLE statement defines a temporary table for the current connection.

##
################################

##################

- ##########################
- ######################
- ################################
- ################################

##

```
DECLARE GLOBAL TEMPORARY TABLE ##
    { ### [ , ### ] * }
[ ON COMMIT {DELETE | PRESERVE} ROWS ]
NOT LOGGED [ON ROLLBACK DELETE ROWS]
```

##

#########SESSION############################(SQLSTATE 428EK) ################SESSION########
##

SESSION####################################
SESSION##############################

###

################CREATE TABLE########## DECLARE GLOBAL TEMPORARY TABLE##################

####
#################

- BIGINT
- CHAR
- DATE
- DECIMAL
- DOUBLE
- DOUBLE PRECISION
- FLOAT
- INTEGER
- NUMERIC
- REAL
- SMALLINT
- TIME
- TIMESTAMP
- VARCHAR

ON COMMIT

COMMIT####################

DELETE ROWS

##################################ON COMMIT###### ##ON ROLLBACK DELETE ROWS################################## ON COMMIT DELETE ROWS###################################(#################)#

PRESERVE ROWS

############

NOT LOGGED

##
ROLLBACK(####ROLLBACK TO SAVEPOINT)############(############)##############################(############)##########

ON ROLLBACK DELETE ROWS

###NOT LOGGED######NOT LOGGED [ON ROLLBACK DELETE ROWS]]####ROLLBACK####ROLLBACK TO SAVEPOINT###

#

```
set schema myapp;

create table t1(c11 int, c12 date);

declare global temporary table SESSION.t1(c11 int) not logged;
-- ####SESSION###############
-- SESSION###########
declare global temporary table t2(c21 int) not logged;
-- ####SESSION###############
-- #####SESSION##########

insert into SESSION.t1 values (1);
-- ########"myapp."############SESSION###########

select * from t1;
-- ##SESSION##############
-- ##select##"myapp.t1"##########
```

Note: ####SESSION############### ####SESSION############################ ##DB2 UDB#####DECLARE GLOBAL TEMPORARY TABLE####Derby########

- IDENTITY column-options
- IDENTITY attribute in copy-options
- AS (fullselect) DEFINITION ONLY
- NOT LOGGED ON ROLLBACK PRESERVE ROWS
- IN tablespace-name
- PARTITIONING KEY
- WITH REPLACE

Declared Global Temporary Tables#####

Derby#################### ############################Derby########

###################

- ALTER TABLE
- CREATE INDEX
- CREATE SYNONYM
- CREATE TRIGGER
- CREATE VIEW
- GRANT
- LOCK TABLE

- RENAME
- REVOKE

################

- SESSION########(###########)#############
- SESSION####################
- ########################
- ######SQL################
- ########
- #####
- ##########

##

####################
- BLOB
- CHAR FOR BIT DATA
- CLOB
- LONG VARCHAR
- LONG VARCHAR FOR BIT DATA
- VARCHAR FOR BIT DATA
- XML

DELETE

##

```
{
    DELETE FROM ##
        [WHERE #] |
    DELETE FROM ## WHERE CURRENT OF
}
```

########################WHERE ###################

###SELECT ###########

#

```
DELETE FROM SAMP.IN_TRAY

stmt.executeUpdate("DELETE FROM SAMP.IN_TRAY WHERE CURRENT OF " +
 resultSet.getCursorName());
```

####

############################(################)#################################CREATE#D INDEX#########################

###JDBC#*cl*

######################CREATE#DROP INDEX##########################

DROP#

############################Drop#######

DROP FUNCTION #

##

```
DROP FUNCTION ###
```

##
42704) ####################################

DROP INDEX #

DROP INDEX ###############

##

```
DROP INDEX ###
```

```
DROP INDEX OrigIndex

DROP INDEX DestIndex
```

####

####################################DROP INDEX
###

DROP PROCEDURE #

##

```
DROP PROCEDURE ###
```

############################
######################################(SQLSTATE 42704)
##############################

DROP SCHEMA #

DROP SCHEMA ################ ####################

APP####(##############)#*SYS*##############

##

```
DROP SCHEMA ##### RESTRICT
```

RESTRICT###RESTRICT###########

```
-- SAMP ###########
-- SAMP####################
-- ########################
DROP SCHEMA SAMP RESTRICT
```

DROP SYNONYM #

###############

##

```
DROP SYNONYM ###
```

DROP TABLE #

DROP TABLE ############

##

```
DROP TABLE ##
```

####

########(######################)###DROP
TABLE#################

############################(###)

DROP TRIGGER #

DROP TRIGGER removes the specified trigger.

##

```
DROP TRIGGER ####
```

```
DROP TRIGGER TRIG1
```

####

##############################(############################)

DROP VIEW #

##############

##

```
DROP VIEW ####
```

```
DROP VIEW AnIdentifier
```

####

##DROP VIEW##

GRANT

GRANT ##

#################

- #########
- ########
- ##########################
- ########################
- ##########
- ####################
- ############

GRANT#########derby.database.sqlAuthorization###true###############derby.databa

##
##########################CREATE##############

GRANT###

#######

```
GRANT ##### ON [TABLE] { ## | #### } TO #########
```

###############

```
GRANT EXECUTE ON { FUNCTION | PROCEDURE } ############## TO #########
```

###

```
  DELETE |
  INSERT |
  REFERENCES [###] |
  SELECT [###] |
  TRIGGER |
  UPDATE [###]
```

###

```
  ( #### {, ####}* )
```

ALL
PRIVILEGES###

DELETE#############################

INSERT#############################

REFERENCES#####################################REFERENCES#############################

SELECT##############SELECT################SELECT#################################

TRIGGER###############################

UPDATE###########UPDATE###WHERE########

#########

```
{ ##### | PUBLIC } [,{ ##### | PUBLIC } ] *
```

############################ PUBLIC
####################PUBLIC##
PUBLIC################################### ####t######SELECT####PUBLIC#harry########

#############

```
{
 ### | ######
}
```

#
t######SELECT###maria # harry#########################

```
GRANT SELECT ON TABLE t TO maria,harry
```

t######UPDATE#TRIGGER####anita#zhi#########################

```
GRANT UPDATE, TRIGGER ON TABLE t TO anita,zhi
```

s.v######SELECT############################

```
GRANT SELECT ON TABLE s.v to PUBLIC
```

p#############george#########################

```
GRANT EXECUTE ON PROCEDURE p TO george
```

INSERT

INSERT################################
INSERT##

##

```
INSERT INTO ##
    [ (#### [ , ####]* ) ]
   ###
```

#########################
- ###
- VALUES##
- ###VALUES#

  ################DEFAULT#####################DEFAULT#########################
- UNION#

#########################

```
INSERT INTO COUNTRIES
      VALUES ('Taiwan', 'TW', 'Asia')

-- DEPARTMENT#################
-- ########################
INSERT INTO DEPARTMENT (DEPTNO, DEPTNAME, ADMRDEPT)
 VALUES ('E31', 'ARCHITECTURE', 'E01')
-- 2###################DEPARTMENT###########
-- ########################
INSERT INTO DEPARTMENT (DEPTNO, DEPTNAME, ADMRDEPT)
 VALUES ('B11', 'PURCHASING', 'B01'),
 ('E41', 'DATABASE ADMINISTRATION', 'E01')
--  EMP_ACT########MA_EMP_ACT###
-- #########
-- MA_EMP_ACT#EMP_ACT#######
-- ########(PROJNO)#'MA'##################
CREATE TABLE MA_EMP_ACT
    (
 EMPNO CHAR(6)  NOT NULL,
 PROJNO CHAR(6)  NOT NULL,
 ACTNO SMALLINT  NOT NULL,
 EMPTIME DEC(5,2),
 EMSTDATE DATE,
 EMENDATE  DATE
    );

INSERT INTO MA_EMP_ACT
 SELECT * FROM EMP_ACT
 WHERE SUBSTR(PROJNO, 1, 2) = 'MA';
-- LOCATION#####DEFAULT#########
INSERT INTO DEPARTMENT
      VALUES ('E31', 'ARCHITECTURE', '00390', 'E01', DEFAULT)
```

####

INSERT
##########################(##################)#########################INSERT#############

LOCK TABLE

LOCK TABLE###################################
#################################

#########################

######################

- ############################(##############################)
- ############

##########################

##

```
LOCK TABLE ## IN { SHARE | EXCLUSIVE } MODE
```

###Flights#

###(*SQLState*
X0X02)

#
##############################Flights##########################

```
LOCK TABLE Flights IN SHARE MODE;
SELECT *
```

```
FROM Flights
WHERE orig_airport > 'OOO';
```

###UPDATE#########################
##UPDATE#########################
################################## ##########

```
LOCK TABLE FlightAvailability IN EXCLUSIVE MODE;
UPDATE FlightAvailability
SET economy_seats_taken = (economy_seats_taken + 2)
WHERE flight_id = 'AA1265' AND flight_date = DATE('2004-03-31');

UPDATE FlightAvailability
SET economy_seats_taken = (economy_seats_taken + 2)
WHERE flight_id = 'AA1265' AND flight_date = DATE('2004-04-11');

UPDATE FlightAvailability
SET economy_seats_taken = (economy_seats_taken + 2)
WHERE flight_id = 'AA1265' AND flight_date = DATE('2004-04-12');

UPDATE FlightAvailability
SET economy_seats_taken = (economy_seats_taken + 2)
WHERE flight_id = 'AA1265' AND flight_date = DATE('2004-04-15');
```

#:

```
LOCK TABLE Maps IN EXCLUSIVE MODE;
SELECT MAX(map_id) + 1 FROM Maps;
-- INSERT INTO Maps . . .
```

RENAME

#######RENAME #######

RENAME COLUMN #

RENAME COLUMN ##################

RENAME COLUMN#################################(*SYS##########*)

##

####################ALTER TABLE ###########

##

```
RENAME COLUMN ##.#### TO ####
```

#
employee#######manager######## supervisor################

```
RENAME COLUMN EMPLOYEE.MANAGER TO SUPERVISOR
```

ALTER TABLE#RENAME COLUMN##################
t#######c1##########NEWTYPE##################

```
    ALTER TABLE t ADD COLUMN c1_newtype NEWTYPE
    UPDATE t SET c1_newtype = c1
    ALTER TABLE t DROP COLUMN c1
    RENAME COLUMN t.c1_newtype TO c1
```

#######
Note: ##
Note: ############################RENAME COLUMN##########
Note: ##

RENAME INDEX #

############################ ####*SYS*####################

##

```
RENAME INDEX ### TO ####
```

```
RENAME INDEX DESTINDEX TO ARRIVALINDEX
```

####

###############RENAME INDEX #########

RENAME TABLE #

RENAME TABLE##############################(###*SYS*##########)

######################################

##

```
RENAME TABLE ## TO ###
```

##
##

```
RENAME TABLE SAMP.EMP_ACT TO EMPLOYEE_ACT
```

########ALTER TABLE ##########

####

##########################

############################RENAME TABLE########

REVOKE

REVOKE###

####################

- ###########
- ##########
- ##############################
- ##########################
- ##########
- ########################
- ########################

REVOKE#########derby.database.sqlAuthorization###true################derby.datab

#######################################

REVOKE###

########

```
REVOKE ##### ON [ TABLE ] { ## | #### } FROM ###########
```

#####################################

#################

```
REVOKE EXECUTE ON { FUNCTION | PROCEDURE } ############# FROM ###########
 RESTRICT
```

##########REVOKE#######RESTRICT#######RESTRICT#####################################

#####

```
ALL PRIVILEGES |
  ####
```

####

```
### {, ### }*
```

###

```
  DELETE |
  INSERT |
  REFERENCES [###] |
  SELECT [###] |
  TRIGGER |
  UPDATE [###]
```

###

```
  ( #### {, ####}* )
```

######ALL
PRIVILEGES###

DELETE#################################

INSERT#################################

REFERENCES###

SELECT###

TRIGGER#################################

UPDATE##########UPDATE###

###########

```
{ ##### | PUBLIC } [,{ authorization ID | PUBLIC } ] *
```

###PUBLIC###############PUBLIC##########
####harry##########SELECT#############harry#########PUBLIC########t########

####: ######################

#############

```
  {
  qualified-name [ signature ]
  }
```

###############

##Derby
################################*Derby* ######## ####"SQL standard
authorization"##########

####

REVOKE###############

############
   ###########SYSTABLEPERMS#######ID####################user2###user1.t1#SELECT#

   ##Derby
   ############SYSTABLEPERMS######################

####user2#SELECT * FROM
user1.t1#####v1#################v1##GRANTEE##user2#TABLEID##user1.t1###SYSTA

################REVOKE#################ID#####################################

############
SYSCOLPERMS#################ID#####################################user2##use
c13######

##Derby#######SYSCOLPERMS##########
c11 FROM
user1.t1#######v1#################GRANTEE##user2#TABLEID##user1.t1#TYPE##S##

REVOKE#####################################ID#####################################
#####################################user2.v1#################################

######
#t##SELECT####maria#harry############################

```
REVOKE SELECT ON TABLE t FROM maria,harry
```

#t##UPDATE#TRIGGER###anita#zhi############################

```
REVOKE UPDATE, TRIGGER ON TABLE t FROM anita,zhi
```

#s.v##SELECT############################

```
REVOKE SELECT ON TABLE s.v FROM PUBLIC
```

#s.v#c1#c2######UPDATE############################

```
REVOKE UPDATE (c1,c2) ON TABLE s.v FROM PUBLIC
```

p########EXECUTE####george############################

```
REVOKE EXECUTE ON PROCEDURE p FROM george RESTRICT
```

SET

Set####################

SET ISOLATION #

SET
ISOLATION###SERIALIZABLE#REPEATABLE
READ#READ COMMITTED#READ UNCOMMITTED###

##################################JDBC#*java.sql.Connection.setTransactionIsolation*###########

##############*Derby* ########
#######"############"##########JDBC#*java.sql.Connection.setTransactionIsolation*##########java.
#####

##

```
SET [ CURRENT ] ISOLATION [ = ]
{
UR | DIRTY READ | READ UNCOMMITTED
CS | READ COMMITTED | CURSOR STABILITY
RS |
RR | REPEATABLE READ | SERIALIZABLE
RESET
}
```

```
set isolation serializable;
```

SET SCHEMA

SET SCHEMA ##############################
##

SET SCHEMA##
########################CREATE SCHEMA ###########

SET SCHEMA ######################SET SCHEMA
##############################SET SCHEMA ##################

##

```
SET [CURRENT] SCHEMA [=]
{ #####|
USER | ? | '<#####>' } | SET CURRENT SQLID [=]
{
#####| USER | ? | '<#####>' }
```

#####128###(#######SYS#sYs#SYs#sys###

USER###*APP*###(#################################

? ######################SET
SCHEMA###*APP*##

```
-- ##########
-- HOTEL#################
SET SCHEMA HOTEL
SET SCHEMA hotel
SET CURRENT SCHEMA hotel
SET CURRENT SQLID hotel
SET SCHEMA = hotel
SET CURRENT SCHEMA = hotel
SET CURRENT SQLID = hotel
SET SCHEMA "HOTEL" -- ##########
SET SCHEMA 'HOTEL' -- ##########--####hotel#########
 --#############
SET SCHEMA = 'hotel'
 --SQLID#CURRENT#################
 --##########
SET SQLID hotel
 -- ################ID########
SET CURRENT SCHEMA USER
// Java#######set schema#######
PreparedStatement ps = conn.PrepareStatement("set schema ?");
ps.setString(1,"HOTEL");
ps.executeUpdate();
... do some work
ps.setString(1,"APP");
ps.executeUpdate();

ps.setString(1,"app"); //error - string is case sensitive
// no app will be found
ps.setNull(1, Types.VARCHAR); //error - null is not allowed
```

SELECT

##

```
###
[ORDER BY #]
[FOR UPDATE #]
WITH {RR|RS|CS|UR}
```

SELECT##########ORDER BY ##FOR UPDATE
#########
SELECT################SELECT#################(##########VALUES##UNION#INTERS

ORDER BY #####*ResultSet*##########FOR UPDATE
#########################SELECT###FOR EACH ONLY ##########FOR
EACH ONLY ##FOR READ ONLY########

SELECT###WITH {RR|RS|CS|UR}#########################

#

```
-- SAL+BONUS+COMM########TOTAL_PAY#########
-- ############
SELECT FIRSTNME, SALARY+BONUS+COMM AS TOTAL_PAY
     FROM EMPLOYEE
     ORDER BY TOTAL_PAY
-- FOR UPDATE##########
-- PROJECT###########(PRSTDATE)#####(PRENDATE)##########
-- ###############
SELECT PROJNO, PRSTDATE, PRENDATE
      FROM PROJECT
      FOR UPDATE OF PRSTDATE, PRENDATE
-- ############RR####
SELECT *
FROM Flights
WHERE flight_id BETWEEN 'AA1111' AND 'AA1112'
WITH RR
```

SELECT##*ResultSet*######
#####*ResultSet*##############Java############*ResultSets*######SQL#########################
UPDATE
#####SQL########SELECT###
Note: ORDER BY####SELECT#################ORDER
BY####################

############ResultSet###
##################SELECT#################
#####ResultSet######SELECT###:

- SELECT###ORDER BY############
- ########
- ###################
  - DISTINCT
  - ##
  - GROUP BY #
  - HAVING #
  - ORDER BY #
- ####FROM#################
  - #######
  - ######
  - ###*s*
  - ######

Note:
###ResultSet#########`ResultSet.CONCUR`
UPDATE##############(FOR UPDATE #########)

SQL############################JDBC
API###*Derby* ########
############################

####

SELECT##(######################)#######
CREATE INDEX#######SELECT################ DROP
INDEX##############################SELECT#########
##SELECT##(CREATE VIEW
#########)

SELECT###############UPDATE WHERE CURRENT####DELETE WHERE
CURRENT##SELECT#######*java.sql.Statement.close* ###SELECT#######UPDATE
WHERE CURRENT ### DELETE WHERE CURRENT#########

SELECT##SELECT############################

UPDATE#

##

```
{
    UPDATE ##
        SET ## = #
        [ , ## = #} ]*
        [WHERE#] |
    UPDATE ##
        SET ## = #
        [ , ## = # ]*
        WHERE CURRENT OF
}
```

#############

```
# | DEFAULT
```

####################WHERE#########################

###SELECT##FOR
UPDATE ##SELECT##FOR
UPDATE#######################

######DEFAULT##############################

#

```
-- 'E21'###(WORKDEPT)####################
-- ######EMPLOYEE##########(JOB)#NULL##
-- #######(SALARY, BONUS, COMM)#0########
UPDATE EMPLOYEE
  SET JOB=NULL, SALARY=0, BONUS=0, COMM=0
  WHERE WORKDEPT = 'E21' AND JOB <> 'MANAGER'

-- ########################
UPDATE EMPLOYEE
 SET JOB = 'MANAGER'
 WHERE JOB IS NULL;
// ##############(PRSTAFF)#1.5####
stmt.executeUpdate("UPDATE PROJECT SET PRSTAFF = "
"PRSTAFF + 1.5" +
"WHERE CURRENT OF" + ResultSet.getCursorName());

-- EMPLOYEE##########(EMPNO)#'000290'######(JOB)##
-- ######NULL######
UPDATE EMPLOYEE
  SET JOB = DEFAULT
  WHERE EMPNO = '000290'
```

####

##################################(##############)######WHERE##SET##################
INDEX#DROP INDEX##ALTER TABLE####################

###JD

####################CREATE INDEX#DROP INDEX##ALTER
TABLE####################

#####################################

#####################################

SQL

CONSTRAINT

CONSTRAINT###CREATE TABLE ##ALTER TABLE
###

##############

- #######

  ##(################)##################
- #######

  ##0###########

##############

- NOT NULL

  ######NULL###############(########################)
- PRIMARY KEY

  ################################NOT NULL###############

  Note: ALTER
  TABLE###ALTER
  TABLE ##########
- UNIQUE

  ######################NULL########
- FOREIGN KEY

  ######################NULL##################
- CHECK

  ##############

##############

- PRIMARY KEY

  ##############################NULL########
- UNIQUE

  ####################NOT NULL###############
- FOREIGN KEY

  ####################NULL##################

  Note:
  ##############################NULL########NULL########NULL########################
- CHECK

  ####################

###PRIMARY KEY#UNIQUE#CHECK#FOREIGN KEY###################(##########)###################

##

#########

############################

#############################NULL###################NULL################

#####NOT NULL################ALTER TABLE ADD PRIMARY KEY ####################NULL############NULL################################ALTER TABLE ############

###PRIMARY KEY#######################UNIQUE###############

######

##(######################

##

####(##############)##
####################################

###(################################

###NULL#

Note:
SQL-92##
NULL############

#######**DML**

###################################Derby###
###############Derby######################

#######(##########################)####################Derby########################
######################################Derby#########

Derby##

######

UNIQUE#PRIMARY KEY#FOREIGN KEY###############################(#####################)
UNIQUE###PRIMARY KEY##############FOREIGN KEY###############################UNIQUE###PRIMARY KEY###FOREIGN KEY###Derby################################

####################################(CREATE INDEX ############)################################

#####DROP INDEX##################################

######

##(#########)##########################

#######

#######################CREATE TABLE##########################

##

- ######## (?)
- #### (CURRENT_DATE#CURRENT_TIME#CURRENT_TIMESTAMP)
- ######
- ###### (###USER#SESSION_USER#CURRENT_USER)

########

############(CASCADE#RESTRICT#SET NULL###NO ACTION)########ON DELETE##/#ON UPDATE#########
##

####################################

##NO ACTION####RESTRICT###

####################RESTRICT#######Derby##

########NO ACTION#######Derby###

##NO ACTION######NO ACTION###

###

########NO ACTION#RESTRICT#CASCADE####SET NULL###SET NULL############null#####################

#########

NO ACTION##Derby###

RESTRICT###Derby###

CASCADE#################(####################)

SET NULL##############################(##)

###
#########RESTRICT#NO ACTION#####################
#######################################RESTRICT#NO ACTION###################

###

- ########RESTRICT####NO ACTION################################(###Derby###################)
- ########SET NULL###
- ########CASCADE#####################################
- ##

#

```
-- OUT_TRAY_PK################:
CREATE TABLE SAMP.OUT_TRAY
 (
 SENT TIMESTAMP,
 DESTINATION CHAR(8),
 SUBJECT CHAR(64) NOT NULL CONSTRAINT OUT_TRAY_PK PRIMARY KEY,
 NOTE_TEXT VARCHAR(3000)
   );

-- #################
```

```
-- #############
CREATE TABLE SAMP.SCHED
 (
 CLASS_CODE CHAR(7) NOT NULL,
 DAY SMALLINT NOT NULL,
 STARTING TIME,
 ENDING TIME,
 PRIMARY KEY (CLASS_CODE, DAY)
 );

-- ######################
-- ############################
-- #####
CREATE TABLE SAMP.EMP
 (
 EMPNO CHAR(6) NOT NULL CONSTRAINT EMP_PK PRIMARY KEY,
 FIRSTNME CHAR(12) NOT NULL,
 MIDINIT vARCHAR(12) NOT NULL,
 LASTNAME VARCHAR(15) NOT NULL,
 SALARY DECIMAL(9,2) CONSTRAINT SAL_CK CHECK (SALARY >= 10000),
 BONUS DECIMAL(9,2),
 TAX DECIMAL(9,2),
 CONSTRAINT BONUS_CK CHECK (BONUS > TAX)
 );

-- MEAL#########################
CREATE TABLE FLIGHTS
 (
 FLIGHT_ID CHAR(6) NOT NULL ,
 SEGMENT_NUMBER INTEGER NOT NULL ,
 ORIG_AIRPORT CHAR(3),
 DEPART_TIME TIME,
 DEST_AIRPORT CHAR(3),
 ARRIVE_TIME TIME,
 MEAL CHAR(1) CONSTRAINT MEAL_CONSTRAINT
 CHECK (MEAL IN ('B', 'L', 'D', 'S')),
 PRIMARY KEY (FLIGHT_ID, SEGMENT_NUMBER)
 );

CREATE TABLE METROPOLITAN
 (
 HOTEL_ID INT NOT NULL CONSTRAINT HOTELS_PK PRIMARY KEY,
 HOTEL_NAME VARCHAR(40) NOT NULL,
 CITY_ID INT CONSTRAINT METRO_FK REFERENCES CITIES
 );

-- ############################
-- ########
CREATE TABLE FLTAVAIL
 (
 FLIGHT_ID CHAR(6) NOT NULL,
 SEGMENT_NUMBER INT NOT NULL,
 FLIGHT_DATE DATE NOT NULL,
 ECONOMY_SEATS_TAKEN INT,
 BUSINESS_SEATS_TAKEN INT,
 FIRSTCLASS_SEATS_TAKEN INT,
 CONSTRAINT FLTAVAIL_PK PRIMARY KEY (FLIGHT_ID, SEGMENT_NUMBER),
 CONSTRAINT FLTS_FK
 FOREIGN KEY (FLIGHT_ID, SEGMENT_NUMBER)
 REFERENCES Flights (FLIGHT_ID, SEGMENT_NUMBER)
 );
-- ############
ALTER TABLE SAMP.PROJECT
ADD CONSTRAINT P_UC UNIQUE (PROJNAME);

-- #################
-- city_id##Cities#################
CREATE TABLE CONDOS
 (
 CONDO_ID INT NOT NULL CONSTRAINT hotels_PK PRIMARY KEY,
```

```
CONDO_NAME VARCHAR(40) NOT NULL,
CITY_ID INT CONSTRAINT city_foreign_key
REFERENCES Cities ON DELETE CASCADE ON UPDATE RESTRICT
);
```

##########

INSERT##UPDATE####################
DELETE##############################
##

#####

```
{
    NOT NULL |
    [ [CONSTRAINT ###]
    {
        CHECK (####) |
        {
            PRIMARY KEY |
            UNIQUE |
            REFERENCES #
        }
    }
}
```

######

```
[CONSTRAINT ###]
{
    CHECK (####) |
    {
        PRIMARY KEY ( #### [ , #### ]* ) |
        UNIQUE ( #### [ , #### ]* ) |
        FOREIGN KEY ( #### [ , #### ]* )
REFERENCES #
    }
}
```

####

```
REFERENCES ## [ ( #### [ , #### ]* ) ]
[ ON DELETE {NO ACTION | RESTRICT | CASCADE | SET NULL}]
   [ ON UPDATE {NO ACTION | RESTRICT }]
|
[ ON UPDATE {NO ACTION | RESTRICT }] [ ON DELETE
   {NO ACTION | RESTRICT | CASCADE | SET NULL}]
```

####

A #### ###########################

###############Derby########(############)######

FOR UPDATE

SELECT###FOR UPDATE############## ####################FOR UPDATE############SELECT##
##

##

```
FOR
{
    READ ONLY | FETCH ONLY |
    UPDATE [ OF #### [ , ####]* ]
}
```

##########FROM####################

Note: #####JDBC#ResultSet##########FOR UPDATE############# JDBC#ResultSet###################################JDBC Statement##########`ResultSet.CONCUR_UPDATABLE`#JDBC ResultSet#############

##*Derby*#################

```
SELECT RECEIVED, SOURCE, SUBJECT, NOTE_TEXT FROM SAMP.IN_TRAY FOR UPDATE
```

FROM

FROM##

##

```
FROM ## [ , ## ] *
```

```
SELECT Cities.city_id
FROM Cities
WHERE city_id < 5
-- #####
SELECT TABLENAME, ISINDEX
FROM SYS.SYSTABLES T, SYS.SYSCONGLOMERATES C
WHERE T.TABLEID = C.TABLEID
ORDER BY TABLENAME, ISINDEX
-- ########
SELECT *
FROM Flights, FlightAvailability
WHERE FlightAvailability.flight_id = Flights.flight_id
AND FlightAvailability.segment_number = Flights.segment_number
AND Flights.flight_id < 'AA1115'
-- #################
--######FROM###################
SELECT COUNTRIES.COUNTRY, CITIES.CITY_NAME, FLIGHTS.DEST_AIRPORT
FROM COUNTRIES LEFT OUTER JOIN CITIES
ON COUNTRIES.COUNTRY_ISO_CODE = CITIES.COUNTRY_ISO_CODE
LEFT OUTER JOIN FLIGHTS
ON Cities.AIRPORT = FLIGHTS.DEST_AIRPORT
```

GROUP BY

GROUP BY ##
############################## NULL##########################

########GROUP BY###############

##

```
GROUP BY ## [ , ## ] *
```

####GROUP BY###

GROUP BY###

```
-- airport######flights##
-- flying_times#######
SELECT AVG (flying_time), orig_airport
FROM Flights
GROUP BY orig_airport

SELECT MAX(city_name), region
```

```
FROM Cities, Countries
WHERE Cities.country_ISO_code = Countries.country_ISO_code
GROUP BY region
-- smallint#####
SELECT ID, AVG(SALARY)
FROM SAMP.STAFF
GROUP BY ID
-- AVGSALARY#EMPCOUNT###DEPTNO##AS#########
-- ###OTHERS########WORKDEPT##########
SELECT OTHERS.WORKDEPT AS DEPTNO,
AVG(OTHERS.SALARY) AS AVGSALARY,
COUNT(*) AS EMPCOUNT
FROM SAMP.EMPLOYEE OTHERS
GROUP BY OTHERS.WORKDEPT
```

HAVING

HAVING########GROUP BY############WHERE####################HAVING############################GROUP BY#######HAVING#################### SELECT###GROUP BY##

##

```
HAVING ####
```

#################(GROUP BY #########)#########################
##########SALARY##

```
-- SELECT COUNT(*)
-- FROM SAMP.STAFF
-- GROUP BY ID
-- HAVING SALARY > 15000
```

HAVING#####SELECT###############
##HAVING###

```
-- 2########################
SELECT SUM(ECONOMY_SEATS_TAKEN), AIRLINE_FULL
FROM FLIGHTAVAILABILITY, AIRLINES
WHERE SUBSTR(FLIGHTAVAILABILITY.FLIGHT_ID, 1, 2) = AIRLINE
GROUP BY AIRLINE_FULL
HAVING COUNT(*) > 1
```

ORDER BY

ORDER BY##SELECT############ ORDER BY###*ResultSet*###############

Syntax

```
ORDER BY { ## | ColumnPosition | Expression }
    [ ASC | DESC ]
    [ , ## | ### | #
        [ ASC | DESC ] ] *
```

##
######SELECT #####################
ORDER BY########SELECT#################
###
SELECT #####################
#####0###SELECT#################
#

##
#############CASE#########################

ASC

##############################ASC######

DESC

#################

####

- SELECT DISTINCT########SELECT##GROUP BY########ORDER BY####SELECT################
- ORDER BY######SELECT###ResultSet###########
  ######INTEGER################NULL#1####################NULL###################

#######

####CITIES#########COUNTRY##NATION#########

```
SELECT CITY_NAME, COUNTRY AS NATION
    FROM CITIES
    ORDER BY NATION
```

######
##############################

```
SELECT name, salary, bonus FROM employee
   ORDER BY salary+bonus
```

######salary#bonus##DECIMAL#######

######
######################################

```
SELECT i, len FROM measures
   ORDER BY sin(i)
```

WHERE

WHERE#############DELETE ##UPDATE ############
WHERE######################## ########################
###DELETE##########UPDATE############

##

```
WHERE ###
```

WHERE###

####################10###

#

```
-- ##################
-- #######
SELECT *
FROM FlightAvailability
WHERE business_seats_taken IS NULL
OR business_seats_taken = 0
-- EMP_ACT#EMPLOYEE##########
-- EMP_ACT#####EMPLOYEE########(LASTNAME)####
-- #########
SELECT SAMP.EMP_ACT.*, LASTNAME
  FROM SAMP.EMP_ACT, SAMP.EMPLOYEE
  WHERE EMP_ACT.EMPNO = EMPLOYEE.EMPNO
-- ################################
```

```
--###
-- ################(DINFO)###AS############
-- AVGSALARY#EMPCOUNT####WHERE########DEPTNO#
-- ################
SELECT THIS_EMP.EMPNO, THIS_EMP.SALARY, DINFO.AVGSALARY, DINFO.EMPCOUNT
 FROM EMPLOYEE THIS_EMP,
   (SELECT OTHERS.WORKDEPT AS DEPTNO,
           AVG(OTHERS.SALARY) AS AVGSALARY,
           COUNT(*) AS EMPCOUNT
    FROM EMPLOYEE OTHERS
    GROUP BY OTHERS.WORKDEPT
   )AS DINFO
 WHERE THIS_EMP.JOB = 'SALESREP'
    AND THIS_EMP.WORKDEPT = DINFO.DEPTNO
```

WHERE CURRENT OF

WHERE CURRENT OF##UPDATE#DELETE##############
################################ ###################SELECT ############

##

```
WHERE CURRENT OF #####
```

```
Statement s = conn.createStatement();
s.setCursorName("AirlinesResults");
ResultSet rs = conn.executeQuery(
    "SELECT Airline, basic_rate " +
    "FROM Airlines FOR UPDATE OF basic_rate");
Statement s2 = conn.createStatement();
s2.executeUpdate("UPDATE Airlines SET basic_rate = basic_rate " +
    "+ .25 WHERE CURRENT OF AirlinesResults");
```

SQL#

################################### ######################

##

##############

- ORDER BY #
- ###
- UPDATE# (SET###)
- VALUES#
- WHERE#

###

######SQL####################

#####
##########################
 # 2. #######

####	##
####	## ################################# FROM ## ######s#UPDATE##DML##WHERE############
##	##########################(#########)

####	##
NULL	NULL################## CAST##INSERT#VALUES#####UPDATE#SET############CAST##############
'	####### #########SQL############################## ################## ## ########################## ##
CAST #	NULL### CAST #############
########	################### ####################
#####	############################ ################ FROM##EXISTS#IN######################

Boolean#

#######################################
####WHERE###############SQL#####################SQL
####################

##
########################### ################

- BIGINT
- DECIMAL
- DOUBLE PRECISION
- INTEGER
- REAL
- SMALLINT

 # 3. ####

###	#################################
####	##
-#*#/####+#-##	## ##################################### ###+##############(###+4#4######)###-##- 1#########################
AVG	###############AVG##
SUM	###############SUM##
LENGTH	#################### LENGTH#############
LOWER	LCASE####LOWER#############
COUNT	############COUNT ###COUNT(*) #############

###
#######CHAR###VARCHAR#################CHAR###VARCHAR########

4.

####	##
##########CHAR#VARCHAR##	########%#_#####LIKE##########################
###	############"||"###Concatenation##########
#########	############################LTRIM###LCASE####LOWER###RTRIM###TRIM###SUBSTR######rrefsqlj29930##########
USER##	User#########################CURRENT_USER###SESSION_USER###########

##I###
##/########DATE#TIME#TIMESTAMP######### ##########/##########

5. ##I#####

####	##
CURRENT_DATE	###########CURRENT_DATE###########
CURRENT_TIME	###########CURRENT_TIME###########
CURRENT_TIMESTAMP	################CURRENT_TIMESTAMP############

###

####SELECT-FROM-WHERE##

##

```
SELECT [ DISTINCT | ALL ] #### [ ,
 ####]*
FROM #
[ WHERE #]
[ GROUP BY # ]
[ HAVING # ]
```

####:

```
{
    * |
    { ## | ### } .* |
    # [AS ####]
}
```

SELECT###########FROM##WHERE#################### DISTINCT#########################

##

FROM###WHERE#################

WHERE##

GROUP BY###GROUP BY####################

GROUP BY#######SELECT#################################SELECT#############################

```
-- #######(WORKDEPT)#######(SALARY)#
-- #EMPLOYEE##################
-- ####################
SELECT WORKDEPT, AVG(SALARY)
     FROM EMPLOYEE
     GROUP BY WORKDEPT
     ORDER BY 1
```

GROUP
BY###

HAVING######(WHERE#####)###H
BY#########################

Derby ################

- FROM #
- WHERE #
- GROUP BY (######GROUP BY)
- HAVING #
- SELECT #

#############

####FROM#######(#########################)#########VALUES######## #:

```
VALUES CURRENT_TIMESTAMP
```

VALUES###########

########*

*###FROM################

##.* # ###.* ######################FROM############

#####

AS###AS#################################

#################################

- JDBC#*ResultSetMetaData*#########
- FROM##
- ORDER BY#######################

```
-- ####ORDER BY#########
-- SELECT-FROM-WHERE######
SELECT CONSTRAINTNAME, COLUMNNAME
FROM SYS.SYSTABLES t, SYS.SYSCOLUMNS col,
SYS.SYSCONSTRAINTS cons, SYS.SYSCHECKS checks
WHERE t.TABLENAME = 'FLIGHTS' AND t.TABLEID = col.
REFERENCEID AND t.TABLEID = cons.TABLEID
AND cons.CONSTRAINTID = checks.CONSTRAINTID
ORDER BY CONSTRAINTNAME
-- ####DISTINCT##########
SELECT DISTINCT ACTNO
FROM EMP_ACT
-- ####################
-- EMPLOYEE#################################
-- ####(WORKDEPT)#BOSS##################(SALARY)#
-- ######
SELECT WORKDEPT AS DPT, MAX(SALARY) AS BOSS
 FROM EMPLOYEE EMP_COR
 GROUP BY WORKDEPT
 HAVING MAX(SALARY) < (SELECT AVG(SALARY)
    FROM EMPLOYEE
    WHERE NOT WORKDEPT = EMP_COR.WORKDEPT)
 ORDER BY BOSS
```

TableExpression

###################FROM ##############################

############################## ########################
##

FROM####################################FROM##############################

###AS#####################################
- VALUES#####################VALUES#######################
- ##

From##

#########################*Derby*##################

##

```
{
############ | ###
}
```

#

```
-- ############
SELECT E.EMPNO, E.LASTNAME, M.EMPNO, M.LASTNAME
 FROM EMPLOYEE E LEFT OUTER JOIN
    DEPARTMENT INNER JOIN EMPLOYEE M
  ON MGRNO = M.EMPNO
  ON E.WORKDEPT = DEPTNO
```

##############

```
{## | #### | #######}
    [ [ AS ] ###
        [ (#### [ , ####]* ) ] ] ]
```

########:

```
TABLE ###( [ [ ##### ] [, ##### ]* ] )
```

####################################

```


SELECT s.*
FROM TABLE( externalEmployees( 42 ) ) s
```

VALUES#

VALUES###################
VALUES###
- *ResultSet*####
- ##############
- INSERT###(INSERT########################VALUES########)

##

```
   VALUES ( # {, # }* )
       [ , ( # {, # }* ) ]* |
   VALUES # [ , # ]*
 }
```

###############

```
# | DEFAULT
```

##

VALUES##INSERT#########DEFAULT###################
##DEFAULT#########################
########################INSERT#######################################

#

```
-- 1#3#
VALUES (1),(2),(3)
-- 1#3#
VALUES 1, 2, 3
-- 3#1#
VALUES (1, 2, 3)
-- 2#3#
VALUES (1,21),(2,22),(3,23)
-- ########
VALUES ('orange', 'orange'), ('apple', 'red'),
('banana', 'yellow')
-- ###DEPARTMENT##################
-- ####################
INSERT INTO DEPARTMENT (DEPTNO, DEPTNAME, ADMRDEPT)
  VALUES ('B11', 'PURCHASING', 'B01'),
    ('E41', 'DATABASE ADMINISTRATION', 'E01')
-- MAJPROJ#################
INSERT INTO PROJECT (PROJNO, PROJNAME, DEPTNO, RESPEMP, PRSTDATE,
 MAJPROJ)
VALUES ('PL2101', 'ENSURE COMPAT PLAN', 'B01', '000020', CURRENT_DATE,
 DEFAULT)

-- #########
VALUES CURRENT_DATE
-- #########
VALUES (3*29, 26.0E0/3)
-- #############
values char(1)
```

#########

##############################

- (), ?, ## (#####), NULL, ###, ########, CAST
- LENGTH, CURRENT_DATE, CURRENT_TIME, CURRENT_TIMESTAMP, #########
- ###########+###-
- *, /, || (##)
- ###########+###-
- ### ##### EXISTS, IN, IS NULL, LIKE, BETWEEN, IS
- NOT
- AND
- OR

###

#

```
(3+4)*9
(age < 16 OR age > 65) AND employed = TRUE
```

###

####WHERE#################### ##################################
###CONSTRAINT ##########
WHERE################################WHERE##########

############################ ######SQL #########################

6. SQL

###	####	##
AND, OR, NOT	################## `(orig_airport = 'SFO') OR (dest_airport = 'GRU') -- #####`	`{ # AND #| # OR # | NOT # }`
#####	<, =, >, <=, >=, <> ######################## `DATE('1998-02-26') < DATE('1998-03-01') -- #####`	`# { < | = | > | <= | >= | <> } #`
IS NULL, IS NOT NULL	################## `WHERE MiddleName IS NULL`	`# IS [ NOT ] NULL`

LIKE	################################
	 %#####################(0####)#
	 _########################
	 ################################
	 `city LIKE 'Sant_'`
	 %#_##################ESCAPE#
	 `SELECT a FROM tabA WHERE a LIKE '%=_' ESCAPE '='`
	 Note: LIKE#######Derby##########
	 ########
	 ####**Derby**##################
		### `[ NOT ] LIKE` ############## `[ ESCAPE '#######']` ####### ##############

BETWEEN	################################ ######<=#>=################## `WHERE booking_date BETWEEN DATE('1998-02-26') AND DATE('1998-03-01')`	`# [ NOT ] BETWEEN # AND #` ##############
IN	################################ `WHERE booking_date NOT IN (SELECT booking_date FROM HotelBookings WHERE rooms_available = 0)`	`{ # [ NOT ] IN ###### | # [ NOT ] IN ( # [, # ]* ) }` ##############
EXISTS	################################	`[NOT] EXISTS #####` ####(*########

###	####	##
	WHERE EXISTS **(SELECT *** **FROM Flights** **WHERE dest_airport = 'SFO'** **AND orig_airport = 'GRU')**	
####	#####ALL#ANY#SOME######### (<,=, >, <=, >=, <>)###### ########################## ALL######################### **WHERE normal_rate < ALL** **(SELECT budget/550 FROM Groups)**	**#** **#####** **{** **ALL |** **ANY |** **SOME** **}** **#####** #SOME######

########

JDBC API#*PreparedStatement*###?####

JDBC API##########IN#INOUT#OUT##########SQL##IN########

New: Derby ##JDBC 3.0######*ParameterMetaData*###
#############Derby ######## ##############

##

#########

```
PreparedStatement ps2 = conn.prepareStatement(
    "UPDATE HotelAvailability SET rooms_available = " +
    "(rooms_available - ?) WHERE hotel_id = ? " +
    "AND booking_date BETWEEN ? AND ?");
-- ################
-- ################
ps2.setInt(1, numberRooms);
ps2.setInt(2, theHotel.hotelId);
ps2.setDate(3, arrival);
ps2.setDate(4, departure);
updateCount = ps2.executeUpdate();
```

#############

##

1. BETWEEN###
   ###

   ```
   WHERE ? BETWEEN DATE('1996-01-01') AND ?
   -- ##DATE########
   ```
2. BETWEEN##################################
   ############################

   ```
   WHERE DATE('1996-01-01') BETWEEN ? AND ?
   -- ##DATE########
   ```
3. IN###
   ##

   ```
   WHERE ? NOT IN (?, ?, 'Santiago')
   -- ##CHAR########
   ```

4. IN##IN########################## ##

```
WHERE FloatColumn IN (?, ?,
 ?)
-- ##FLOAT########
```

5. ######2########+#-#*#/#AND#OR#<#># =#<#>#<=#>=## ##############################

```
WHERE ? < CURRENT_TIMESTAMP
-- ##TIMESTAMP########
```

6. CAST##

```
CALL valueOf(CAST (? AS VARCHAR(10)))
```

7. LIKE#################################### ## (LIKE##CHAR#VARCHAR#################Concatenation##########)

```
WHERE ? LIKE 'Santi%'
--######java.lang.Integer.MAX_VALUE#
--CHAR########
```

8. ########?##||#################### ####"? || ?"#######||#########?##################||################################CHAR#### ##########CHAR FOR BIT DATA####VARCHAR FOR BIT DATA########################VARCHAR FOR BIT DATA###

```
SELECT BITcolumn || ?
FROM UserTable
-- ##BITcolumn#######CHAR FOR BIT DATA######
```

9. #####?## ##################

```
SELECT c1 IS NULL ? ? : c1
-- ########################
-- ############c1############
-- :########################
```

10. INSERT##values####select######################## ############################

```
INSERT INTO t VALUES (?)
-- ############
-- t#################
INSERT INTO t SELECT ?
FROM t2
-- ########
```

11. ##########?########################### #######

```
SELECT *
FROM tab1
WHERE ? = (SELECT x FROM tab2)

SELECT *
FROM tab1
WHERE ? = ANY (SELECT x FROM tab2)
-- ######################
-- tab2.x############
```

12. UPDATE###

```
UPDATE t2 SET c2 =? -- c2############
```

13. ##########-#+###################### #######

```
CREATE TABLE t1 (c11 INT, c12 SMALLINT, c13 DOUBLE, c14 CHAR(3))
SELECT * FROM t1 WHERE c11 BETWEEN -? AND +?
-- ############################INT######
```

```
-- (#####c11#INT##################INT########)
```

14. LENGTH##########################VARCHAR########

```
SELECT LENGTH(?)
```

15. ####

```
? = SOME (SELECT 1 FROM t)
-- ###################INTEGER########
1 = SOME (SELECT ? FROM t)
-- ###################INTEGER########
```

16. IS#####################################

###

JOIN

JOIN###FROM ##*TableExpression*#########2##############(WHERE###"WHERE t1.col1 = t2.col2"##)

##

```
JOIN#
```

JOIN###############

- INNER JOIN ##

  join####2##################INNER JOIN ##########

- LEFT OUTER JOIN##

  join####2##LEFT OUTER JOIN##########

- RIGHT OUTER JOIN ##

  join####2##RIGHT OUTER JOIN ##########

########################outer join#####WHERE #####################

JOIN#############

########################*Derby*##################

INNER JOIN

INNER JOIN######################JOIN #####

##

```
## [ INNER ] JOIN ## { ON ### }
```

ON########################

ON################################SELECT###############
######ON###############

```
SELECT *
FROM SAMP.EMPLOYEE INNER JOIN SAMP.STAFF
ON EMPLOYEE.SALARY < STAFF.SALARY
```

ON###(############)

```
-- EMP_ACT#EMPLOYEE###########
-- EMP_ACT##############
-- EMPLOYEE############(LASTNAME)########
-- #####
```

```
SELECT SAMP.EMP_ACT.*, LASTNAME
     FROM SAMP.EMP_ACT JOIN SAMP.EMPLOYEE
     ON EMP_ACT.EMPNO = EMPLOYEE.EMPNO
-- EMPLOYEE#DEPARTMENT##########
-- 1930#########(BIRTDATE)########
-- #####(EMPNO)#######(LASTNAME)#
-- ####(EMPLOYEE#####WORKDEPT##DEPARTMENT#####DEPTNO)#
-- ###(DEPTNAME)######
SELECT EMPNO, LASTNAME, WORKDEPT, DEPTNAME
     FROM SAMP.EMPLOYEE JOIN SAMP.DEPARTMENT
     ON WORKDEPT = DEPTNO
     AND YEAR(BIRTHDATE) < 1930

-- VALUES############(select####)#####"##"#####
-- ###################"R1"#"R2"###2##"x"#####
-- ##############
SELECT *
FROM (VALUES (3, 4), (1, 5), (2, 6))
AS VALUESTABLE1(C1, C2)
JOIN (VALUES (3, 2), (1, 2),
(0, 3)) AS VALUESTABLE2(c1, c2)
ON VALUESTABLE1.c1 = VALUESTABLE2.c1
-- This results in:
-- C1         |C2         |C1         |2
-- -----------------------------------------------
-- 3          |4          |3          |2
-- 1          |5          |1          |2


-- ############################

SELECT DEPTNO, DEPTNAME, EMPNO, LASTNAME
 FROM DEPARTMENT INNER JOIN EMPLOYEE
 ON MGRNO = EMPNO

-- #################
-- ######################
SELECT E.EMPNO, E.LASTNAME, M.EMPNO, M.LASTNAME
 FROM EMPLOYEE E INNER JOIN
 DEPARTMENT INNER JOIN EMPLOYEE M
        ON MGRNO = M.EMPNO
        ON E.WORKDEPT = DEPTNO
```

LEFT OUTER JOIN##

LEFT OUTER JOIN####join#######JOIN##########
##########(##)###############(##)##############NULL#########################

##

```
## LEFT [ OUTER ] JOIN ##
{
    ON ###
    }
```

ON###
ON###(################)

 # 1

```
--############

SELECT CITIES.COUNTRY, CITIES.CITY_NAME, REGION
FROM Countries
LEFT OUTER JOIN Cities
ON CITIES.COUNTRY_ISO_CODE = COUNTRIES.COUNTRY_ISO_CODE
WHERE REGION = 'Asia'
```

```
-- LEFT JOIN###################
-- #####

SELECT  COUNTRIES.COUNTRY, CITIES.CITY_NAME,REGION
FROM COUNTRIES
LEFT JOIN CITIES
ON CITIES.COUNTRY_ISO_CODE = COUNTRIES.COUNTRY_ISO_CODE
WHERE REGION = 'Asia'
```

 # 2

```
-- EMPLOYEE#DEPAETMENT##########
-- 1930########(BIRTHDATE)#######
-- ####(EMPNO)##
-- #####(LASTNAME)##
-- ####(EMPLOYEE#####WORKDEPT##DEPARTMENT#####DEPTNO)##
-- ###(DEPTNAME)######

SELECT EMPNO, LASTNAME, WORKDEPT, DEPTNAME
   FROM SAMP.EMPLOYEE LEFT OUTER JOIN SAMP.DEPARTMENT
   ON WORKDEPT = DEPTNO
   AND YEAR(BIRTHDATE) < 1930

-- ###################
-- ######################

SELECT DEPTNO, DEPTNAME, EMPNO, LASTNAME
      FROM DEPARTMENT LEFT OUTER JOIN EMPLOYEE
     ON MGRNO = EMPNO
```

RIGHT OUTER JOIN

RIGHT OUTER JOIN####JOIN##########JOIN########
#########(#)############(#)################################NULL#######A
LEFT OUTER JOIN B # B RIGHT OUTER JOIN A##################

##

```
## RIGHT [ OUTER ] JOIN ##
{
    ON ###
    }
```

ON#################SELECT#############
ON##(###############)

#1

```
-- ###########
-- ###############

SELECT COUNTRIES.COUNTRY, CITIES.CITY_NAME
FROM CITIES
RIGHT OUTER JOIN COUNTRIES
ON CITIES.COUNTRY_ISO_CODE = COUNTRIES.COUNTRY_ISO_CODE

-- ###########
-- ####################

SELECT COUNTRIES.COUNTRY, CITIES.CITY_NAME
FROM CITIES
RIGHT OUTER JOIN COUNTRIES
ON CITIES.COUNTRY_ISO_CODE = COUNTRIES.COUNTRY_ISO_CODE
WHERE Countries.region = 'Africa'
```

```
-- RIGHT JOIN##############
-- #############
SELECT COUNTRIES.COUNTRY, CITIES.CITY_NAME
FROM CITIES
RIGHT JOIN COUNTRIES
ON CITIES.COUNTRY_ISO_CODE = COUNTRIES.COUNTRY_ISO_CODE
WHERE Countries.region = 'Africa'
```

#2

```
-- ##########
-- ###FRON##################
-- ##################
-- ########################

SELECT E.EMPNO, E.LASTNAME, M.EMPNO, M.LASTNAME
 FROM EMPLOYEE E RIGHT OUTER JOIN
 DEPARTMENT RIGHT OUTER JOIN EMPLOYEE M
        ON MGRNO = M.EMPNO
        ON E.WORKDEPT = DEPTNO
```

SQL

###

#####################################

Syntax

```
{
    ( Query )
 |
    Query INTERSECT [ ALL | DISTINCT ]
 Query
 |
    Query EXCEPT [ ALL | DISTINCT ] Query |
    Query UNION [ ALL | DISTINCT ] Query |
### | VALUES#
}
```

##########################INTERSECT#EXCEPT#UNION##########################INTERSE

UNION#INTERSECT#EXCEPT ALL#########

ALL#DISTINCT################################## DISTINCT##########################ALL
ALL####################

##############L#######################R###(ALL#

- UNION: (L + R)#
- EXCEPT: (L – R)#0(#)########
- INTERSECT: L#R########

#

```
-- ###
SELECT *
FROM ORG

-- ####
SELECT *
FROM (SELECT CLASS_CODE FROM CL_SCHED) AS CS

-- ####
SELECT *
FROM (SELECT CLASS_CODE FROM CL_SCHED) AS CS (CLASS_CODE)
```

```
-- UNION###
-- ORG##########
-- DEPTNUMB#MANAGER########
-- (1,2)#(3,4)####
-- ###DEPTNUMB#MANAGER#smallint#######
SELECT DEPTNUMB, MANAGER
FROM ORG
UNION ALL
VALUES (1,2), (3,4)

-- ###
VALUES (1,2,3)

-- EMPLOYEE##########(WORKDEPT)#'E'##########
-- EMP_ACT##############(PROJNO)#
-- 'MA2100'#'MA2110'####'MA2112'####################
-- ####(EMPNO)######
SELECT EMPNO
     FROM EMPLOYEE
     WHERE WORKDEPT LIKE 'E%'
  UNION
  SELECT EMPNO
     FROM EMP_ACT
     WHERE PROJNO IN('MA2100', 'MA2110', 'MA2112')
-- #################EMPLOYEE###########'emp'#
-- EMP_ACT###########'emp_act'###"##"#####
-- ###############"##"######
-- ##################EMPNO###############
SELECT EMPNO, 'emp'
     FROM EMPLOYEE
     WHERE WORKDEPT LIKE 'E%'
  UNION
  SELECT EMPNO, 'emp_act' FROM EMP_ACT
     WHERE PROJNO IN('MA2100', 'MA2110', 'MA2112')
-- ##################
-- UNION ALL####################
SELECT EMPNO
      FROM EMPLOYEE
      WHERE WORKDEPT LIKE 'E%'
  UNION ALL
  SELECT EMPNO
     FROM EMP_ACT
     WHERE PROJNO IN('MA2100', 'MA2110', 'MA2112')
-- ##################
-- ####################
-- ####"new"##########
 SELECT EMPNO, 'emp'
     FROM EMPLOYEE
     WHERE WORKDEPT LIKE 'E%'
  UNION
  SELECT EMPNO, 'emp_act'
     FROM EMP_ACT
     WHERE PROJNO IN('MA2100', 'MA2110', 'MA2112')
  UNION
     VALUES ('NEWAAA', 'new'), ('NEWBBB', 'new')
```

########

################################
#####################################

############################

Sometimes also called an expression subquery.

##

```
(###)
```

Examples

```
-- avg #############
-- ###################
SELECT NAME, COMM
  FROM STAFF
  WHERE EXISTS
    (SELECT AVG(BONUS + 800)
       FROM EMPLOYEE
       WHERE COMM < 5000
       AND EMPLOYEE.LASTNAME = UPPER(STAFF.NAME)
)
-- VALUES###############
-- #####"##"######
-- ######"R1"#"R2"###########
-- "X"#########################
SELECT R1,R2
 FROM (VALUES('GROUP 1','GROUP 2')) AS X(R1,R2)
```

#####

####################

##

- FROM ####
- EXISTS#IN#####

FROM ##############################

EXISTS###*#######################

IN#######################

##

```
(###)
```

#

```
-- FROM################
SELECT VirtualFlightTable.flight_ID
FROM
    (SELECT flight_ID, orig_airport, dest_airport
    FROM Flights
    WHERE (orig_airport = 'SFO' OR dest_airport = 'SCL') )
AS VirtualFlightTable
-- FROM#############(values#)####
SELECT mycol1
FROM
    (VALUES (1, 2), (3, 4))
AS mytable (mycol1, mycol2)
-- EXISTS#########
SELECT *
FROM Flights
WHERE EXISTS
    (SELECT * FROM Flights WHERE dest_airport = 'SFO'
    AND orig_airport = 'GRU')
-- IN##########
SELECT flight_id, segment_number
FROM Flights
WHERE flight_id IN
    (SELECT flight_ID
    FROM Flights WHERE orig_airport = 'SFO'
    OR dest_airport = 'SCL')
-- #############
SELECT NAME, COMM
FROM STAFF
WHERE COMM >
```

```
(SELECT AVG(BONUS + 800)
     FROM EMPLOYEE
     WHERE COMM < 5000)
```

######

################SQL############### ###################################
#############SQL92###################### TIMESTAMPADD#TIMESTAMPDIFF#############JD

#######

Derby##############################

- ABS#ABSVAL##
- ACOS ##
- ASIN ##
- ATAN ##
- BIGINT##
- CAST ##
- CEIL###CEILING##
- CHAR ##
- Concatenation
- COS ##
- NULLIF#
- CURRENT_DATE##
- CURRENT ISOLATION ##
- CURRENT_TIME##
- CURRENT_TIMESTAMP##
- CURRENT_USER##
- DATE##
- DAY ##
- DEGREES ##
- DOUBLE##
- EXP##
- FLOOR##
- HOUR##
- IDENTITY_VAL_LOCAL##
- INTEGER##
- LENGTH##
- LN####LOG##
- LOG10 ##
- LOCATE##
- LCASE####LOWER##
- LTRIM##
- MINUTE ##
- MOD##
- MONTH##
- PI ##
- RADIANS ##
- RTRIM##
- SECOND##
- SESSION_USER##
- SIN ##
- SMALLINT##
- SQRT##
- SUBSTR##

- TAN ##
- TIME ##
- TIMESTAMP##
- TRIM ##
- UCASE#####UPPER##
- USER##
- VARCHAR##
- YEAR##

(####)

######(ANSI
SQL-
92#############################)###

###
7.

'	#####	#######
COUNT	X	X
MIN	'	X
MAX	'	X
AVG	'	X
SUM	'	X

######################
- ########
- HAVING#
- ################ORDER BY
  #(#####)#####################################ORDER BY #############

###################################(GROUP BY #############) (GROUP
BY #############HAVING#############)
##########*ResultSet*#######(##)######(##)#################################(#####################

```
-- not valid
SELECT MIN(flying_time), flight_id
FROM Flights
```

###########(##)###
####SUM###

```
SELECT c1
FROM t1
GROUP BY c1
HAVING c2 >
    (SELECT t2.x
    FROM t2
    WHERE t2.y = SUM(t1.c3))
```

################*ResultSet*###################

#################
- AVG##
- COUNT ##
- MAX##
- MIN##

- SUM##

ABS#ABSVAL##

ABS # ABSVAL############## ###################################### (DECIMAL, DOUBLE PRECISION, FLOAT, INTEGER# BIGINT# NUMERIC# REAL ###SMALLINT)#

##

```
ABS(##)
```

```
-- 3####
VALUES ABS(-3)
```

ACOS

ACOS##############################

##

- #####NULL##########NULL###
- #########1#############################(SQL state 22003)

################0##pi### ####################

Syntax

```
ACOS ( # )
```

ASIN

ASIN ##########################

##

- ###NULL########NULL###
- ###0################0###
- #######1#######################(SQL state 22003)

################-pi/2##pi/2############################

Syntax

```
ASIN ( # )
```

ATAN

ATAN ########################

################################ ###########################

- ###NULL##########NULL######
- ###0##################0######

################-pi/2##pi/2#########################

##

```
ATAN ( # )
```

AVG##

AVG#########################(## (####)##########)
AVG##############################

##

```
AVG ( [ DISTINCT | ALL ] # )
```

DISTICT##################
ALL###############ALL#DISTINCT#############ALL###
########1.0#1.0#1.0#1.0#2.0#############AVG(col)#AVG(DISTINCT
col)############

DISTINCT############################# ###############

```
SELECT AVG (DISTINCT flying_time), SUM (DISTINCT miles)
FROM Flights
```

#######SQL-
92###################SQL-
92##############################NULL####################

########################(##############) ##################*1*######

```
SELECT AVG(c1)
FROM (VALUES (1), (1), (1), (1), (2)) AS myTable (c1)
```

######################

```
SELECT AVG(CAST (c1 AS DOUBLE PRECISION))
FROM (VALUES (1), (1), (1), (1), (2)) AS myTable (c1)
```

BIGINT##

BIGINT#########################64############

##

```
BIGINT (### | ## )
```

###
    ######################## #################SQL##################
    ################## ####################big
    integer#########################
##
    ##################### #############big integer##########################
    ##

######big integer######null##########null###

EMPLOYEE#################big integer####EMPNO###########

```
SELECT BIGINT (EMPNO) FROM EMPLOYEE
```

CASE

Derby##########CASE#######

CASE####

######################CASE############ ############################

```
CASE
  WHEN ### THEN then#
  [ WHEN ### THEN then# ]...
```

```
  ELSE else#
END
```

Then##else###

```
-- 3####
VALUES CASE WHEN 1=1 THEN 3 ELSE 4 END
```

```
-- 7####
VALUES
   CASE
      WHEN 1 = 2 THEN 3
      WHEN 4 = 5 THEN 6
      ELSE 7
   END
```

CAST

CAST#############################(?)#NULL#########

CAST#################################

##

```
CAST ( [ # | NULL | ? ]
    AS ####)
```

###

CAST###SQL-92 ########
#######SQL################################
################################## ##################################
Y################################ ######################SMALLINT########
###################################SMALLINT###################
 # 8. SQL-92 ##########################

Types	SMALLINT	INTEGER	BIGINT	DECIMAL	REAL	DOUBLE	FLOAT	CHAR	VARCHAR	LONG VARCHAR	CHAR FOR BIT DATA	VARCHAR FOR BIT DATA	LONG VARCHAR FOR BIT DATA	CLOB	BLOB	DATE	TIME	TIMESTAMP	XML
SMALLINT	Y	Y	Y	Y	Y	Y	Y	Y	-	-	-	-	-	-	-	-	-	-	-
INTEGER	Y	Y	Y	Y	Y	Y	Y	Y	-	-	-	-	-	-	-	-	-	-	-
BIGINT	Y	Y	Y	Y	Y	Y	Y	Y	-	-	-	-	-	-	-	-	-	-	-
DECIMAL	Y	Y	Y	Y	Y	Y	Y	Y	-	-	-	-	-	-	-	-	-	-	-
REAL	Y	Y	Y	Y	Y	Y	Y	-	-	-	-	-	-	-	-	-	-	-	-
DOUBLE	Y	Y	Y	Y	Y	Y	Y	-	-	-	-	-	-	-	-	-	-	-	-
FLOAT	Y	Y	Y	Y	Y	Y	Y	-	-	-	-	-	-	-	-	-	-	-	-
CHAR	Y	Y	Y	Y	-	-	-	Y	Y	Y	-	-	-	Y	-	Y	Y	Y	-
VARCHAR	Y	Y	Y	Y	-	-	-	Y	Y	Y	-	-	-	Y	-	Y	Y	Y	-
LONG VARCHAR	-	-	-	-	-	-	-	Y	Y	Y	-	-	-	Y	-	-	-	-	-
CHAR FOR BIT DATA	-	-	-	-	-	-	-	-	-	-	Y	Y	Y	Y	Y	-	-	-	-
VARCHAR FOR BIT DATA	-	-	-	-	-	-	-	-	-	-	Y	Y	Y	Y	Y	-	-	-	-
LONG VARCHAR FOR BIT DATA	-	-	-	-	-	-	-	-	-	-	Y	Y	Y	Y	Y	-	-	-	-
CLOB	-	-	-	-	-	-	-	Y	Y	Y	-	-	-	Y	-	-	-	-	-
BLOB	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Y	-	-	-	-

Types	SMALLINT	INTEGER	BIGINT	DECIMAL	REAL	DOUBLE	FLOAT	CHAR	VARCHAR	LONG VARCHAR	CHAR FOR BIT DATA	VARCHAR FOR BIT DATA	LONG VARCHAR FOR BIT DATA	CLOB	BLOB	DATE	TIME	TIMESTAMP	XML
DATE	-	-	-	-	-	-	-	Y	Y	-	-	-	-	-	-	Y	-	-	-
TIME	-	-	-	-	-	-	-	Y	Y	-	-	-	-	-	-	-	Y	-	-
TIMESTAMP	-	-	-	-	-	-	-	Y	Y	-	-	-	-	-	-	Y	Y	Y	-
XML	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Y

########CAST##

####
#######Derby##SQL-92###################

- #
  - #### (SMALLINT, INTEGER, BIGINT, DECIMAL, NUMERIC)
  - #### (FLOAT, REAL, DOUBLE PRECISION)
- #
  - ### (CLOB, CHAR, VARCHAR, LONG VARCHAR)
  - #### (BLOB, CHAR FOR BIT DATA, VARCHAR FOR BIT DATA, LONG VARCHAR FOR BIT DATA)
- ##
  - DATE
  - TIME
  - TIMESTAMP

#########

###

########/######

##0###
########BLOB####/###########

#/#####

#/############TIMESTAMP#/##########
DATE#TIMESTAMP#############TIMESTAMP##TIME######00:00:00### ##TIME#TIMESTAMP####
TIMESTAMP#DATE##########TIME##################
TIMESTAMP#TIME##########DATE##################

```
SELECT CAST (miles AS INT)
FROM Flights
-- timestamp#########
INSERT INTO mytable (text_column)
VALUES (CAST (CURRENT_TIMESTAMP AS VARCHAR(100)))
-- NULL####################
SELECT airline
FROM Airlines
UNION ALL
VALUES (CAST (NULL AS CHAR(2)))
-- double#decimal######
SELECT CAST (FLYING_TIME AS DECIMAL(5,2))
FROM FLIGHTS
-- SMALLINT#BIGINT#####
VALUES CAST (CAST (12 as SMALLINT) as BIGINT)
```

XML####

############XML###XML####################XML##############XMLSERIALIZE
###########

CEIL###CEILING##

CEIL###CEILING#################################

###########################
- #####NULL#######NULL###
- ########################
- #####0#######0###
- #####0#####-1.0#############0###

##################(##########)#######################

Syntax

```
CEIL ( # )
```

```
CEILING ( # )
```

CHAR

CHAR########################

#################
- ##### ##################
- #### ############################
- #### #################
- ########### #######DOUBLE####REAL########
- #### #######SMALLINT#INTEGER#BIGINT########

################################CHAR########################NULL###################N
XML#######XMLSERIALIZE###########

##################

```
CHAR (### [, #] )
```

###
CHAR#VARCHAR#LONG VARCHAR#CLOB#################

#

##################0##254############

##
VARCHAR#CLOB)###############

###################

```
CHAR (### )
```

###

#######(SMALLINT#INTEGER#BIGINT)#######

#######SQL###10####n#################################

- #########SMALLINT#############6#################6#########6####################
- #########INTEGER#############11################11########11###################
- #########BIGINT#############20################20########20####################

###################

```
CHAR (### )
```

###

###############

- **##**: ####################10###
- **##**: ####################8###
- **#######**: ##########################26###

###################

```
CHAR (### )
```

###

##DECIMAL#####################

#######################

```
CHAR (###### )
```

#######

###########(DOUBLE#REAL)#########

CHAR#####EDLEVEL(smallint###############)###################

```
SELECT CHAR(EDLEVEL) FROM EMPLOYEE
```

EDLEVEL#18##########CHAR(6)##'18 '#######(18####4######)

Concatenation

######||###

###

##

```
{
    { ### || ### } |
    { #### || #### }
}
```

##########################CHAR############CHAR################VARCHAR###
CHAR#VARCHAR####################################

#########################

##############################CHAR FOR BIT DATA############CHAR FOR BIT DATA################VARCHAR FOR BIT DATA###

```
--####'supercalifragilisticexbealidocious(sp?)'###
VALUES 'supercalifragilistic' || 'exbealidocious' || '(sp?)'
-- NULL######
VALUES CAST (null AS VARCHAR(7))|| 'AString'
-- '130asdf'######
VALUES '130' || 'asdf'
```

COS

COS###############

############################## ###########################

- #####NULL###########NULL###

##

```
COS ( # )
```

COUNT

COUNT#################(## (####)##########) COUNT##################

##

```
COUNT ( [ DISTINCT | ALL ] # )
```

DISTINCT########################ALL##############################DISTINCT#ALL########
col)############

######## DISTINCT#####################################

```
-- #########
SELECT COUNT (DISTINCT flying_time), SUM (DISTINCT miles)
FROM Flights
```

##
####NULL##############################

COUNT############INTEGER###

```
-- ###############
-- ###############
SELECT COUNT (country), region
FROM Countries
GROUP BY region
HAVING COUNT (country) > 1
```

COUNT(*)

COUNT(*) ###############NULL############COUNT(*)###############

##

```
COUNT(*)
```

########INTEGER###

```
-- Flights###############
SELECT COUNT(*)
FROM Flights
```

CURRENT DATE

CURRENT DATE #CURRENT_DATE######

CURRENT_DATE##

CURRENT_DATE###

##

```
CURRENT_DATE
```

####

```
CURRENT DATE
```

```
-- ###############
SELECT * FROM Flightavailability where flight_date > CURRENT_DATE;
```

CURRENT ISOLATION

CURRENT ISOLATION###########char(2)###################
""(##)#"UR"#"CS"#"RS"####"RR"#

##

```
CURRENT ISOLATION
```

```
VALUES CURRENT ISOLATION
```

CURRENT SCHEMA##

CURRENT SCHEMA#################################

Note: CURRENT SCHEMA#CURRENT SQLID######

#########128#############

##

```
CURRENT SCHEMA

-- ####

CURRENT SQLID
```

```
-- name####################
CREATE TABLE mytable (id int, name VARCHAR(128) DEFAULT CURRENT SQLID)
-- ########################
INSERT INTO mytable(id) VALUES (1)
-- ##################
SELECT name FROM mytable WHERE name = CURRENT SCHEMA
```

CURRENT TIME

CURRENT TIME #CURRENT_TIME######

CURRENT_TIME##

CURRENT_TIME###

##

```
CURRENT_TIME
```

####

```
CURRENT TIME
```

```
VALUES CURRENT_TIME
-- ####

VALUES CURRENT TIME
```

CURRENT TIMESTAMP##

CURRENT TIMESTAMP #CURRENT_TIMESTAMP######

CURRENT_TIMESTAMP##

CURRENT_TIMESTAMP###

##

```
CURRENT_TIMESTAMP
```

###

```
CURRENT TIMESTAMP
```

```
VALUES CURRENT_TIMESTAMP
-- ###

VALUES CURRENT TIMESTAMP
```

CURRENT_USER##

CURRENT_USER ################### (##################DBMS##############)
################*APP*######

USER#SESSION_USER###########

######128############

##

```
CURRENT_USER
```

```
VALUES CURRENT_USER
```

DATE##

DATE###############

###############2,932,897########CLOB/LONG
VARCHAR/XML#########################7####### #####7######yyyynnn###########yyyy#####1

#########################

- ##
- ###################n######1#1#1###n-1###############
- #######7##############################

##

```
DATE ( # )
```

#####'1988/12/25'###############

```
VALUES DATE('1988-12-25')
```

DAY

DAY##################

###################CLOB#LONG VARCHAR#XML###1##31##############null###########

##

```
DAY ( # )
```

#

```
values day('2006-08-02');
```

###2######

DEGREES

DEGREES####################

###################################### #######################

##: ########################COS(RADIANS(90.0))####0.0##################

#####################

##

```
DEGREES ( # )
```

DOUBLE##

DOUBLE#########################

- ###############
- ####################

############

```
DOUBLE [PRECISION] (## )
```

##

#################

########################nul###############null#############null#####null#####################

###############

```
DOUBLE (#### )
```

####

###########CHAR####VARCHAR####################

####################
#####nul###############null#############null#####null###
##

EXP##

EXP###e#################

#######e#################### ########## ############

e##########

########################

##

```
EXP ( # )
```

FLOOR##

FLOOR#####################################

###########################

- #####NULL#########NULL###
- ########################
- #####0#########0###

################(########)##############################

##

```
FLOOR ( # )
```

HOUR##

HOUR################

###############CLOB#LONG
VARCHAR#XML####################################
#################null###############null#############null######null###

##

```
HOUR ( # )
```

#

TABLE1######################

```
SELECT * FROM TABLE1
WHERE HOUR(STARTING) BETWEEN 12 AND 17
```

IDENTITY_VAL_LOCAL##

Derby #IDENTITY_VAL_LOCAL###########

##:

```
IDENTITY_VAL_LOCAL ( )
```

IDENTITY_VAL_LOCAL###################VALUES####INSERT####################

IDENTITY_VAL_LOCAL#####################################DECIMAL (31,0)###

IDENTITY_VAL_LOCAL####################INSERT##################INSERT###########V
#######Derby###########
#################INSERT##################null######

##########

- ######VALUES#######INSERT#
- VALUES########INSERT#
- select###INSERT#

##########INSERT###IDENTITY_VAL_LOCAL()##########

#:

```
ij> create table t1(c1 int generated always as identity, c2 int);
0 rows inserted/updated/deleted
ij> insert into t1(c2) values (8);
1 row inserted/updated/deleted
ij> values IDENTITY_VAL_LOCAL();
1
------------------------------
1
1 row selected
ij> select IDENTITY_VAL_LOCAL()+1, IDENTITY_VAL_LOCAL()-1 from t1;
1                              |2
-----------------------------------------------------------------------
2                              |0
1 row selected
ij> insert into t1(c2) values (IDENTITY_VAL_LOCAL());
1 row inserted/updated/deleted
ij> select * from t1;
C1             |C2
------------------------------
1              |8
2              |1
2 rows selected
ij> values IDENTITY_VAL_LOCAL();
1
------------------------------
2
1 row selected
ij> insert into t1(c2) values (8), (9);
2 rows inserted/updated/deleted
ij> -- ######################
values IDENTITY_VAL_LOCAL();
1
------------------------------
2
1 row selected
ij> select * from t1;
C1             |C2
------------------------------
1              |8
2              |1
3              |8
4              |9
4 rows selected
ij> insert into t1(c2) select c1 from t1;
4 rows inserted/updated/deleted
-- select######################
ij> values IDENTITY_VAL_LOCAL();
1
------------------------------
2
1 row selected
ij> select * from t1;
C1             |C2
------------------------------
1              |8
2              |1
3              |8
4              |9
5              |1
6              |2
```

```
7              |3
8              |4
8 rows selected
```

INTEGER##

INTEGER#################################

##

```
INT[EGER] (## | #### )
```

##

- ################# ##
- ##

####

- ############################
- ######################SQL###################### ######################
- ##

##################null###############null############null######null###

EMPLOYEE#####(SALARY)######(EDLEVEL)###

```
SELECT INTEGER (SALARY / EDLEVEL), SALARY, EDLEVEL, EMPNO
FROM EMPLOYEE
ORDER BY 1 DESC
```

LCASE####LOWER##

LCASE###LOWER#####################################

##

```
LCASE ### LOWER ( ### )
```

##########CHAR#VARCHAR####LONG
VARCHAR######(########)####################

##########CHAR####LONG VARCHAR##################CHAR#####LONG
VARCHAR###################VARCHAR###

############################

######null#################null###

```
-- 'asd1#w'######
VALUES LOWER('aSD1#w')

SELECT LOWER(flight_id) FROM Flights
```

LENGTH##

LENGTH#####################################

##

##

```
LENGTH ( { #### | ##### } )
```

```
-- 20####
VALUES LENGTH('supercalifragilistic')
-- 1####
```

```
VALUES LENGTH(X'FF')
-- 4###
VALUES LENGTH(1234567890)
```

LN####LOG##

LN####LOG##########(e#######)######

###0#####################################

- #####NULL############NULL###
- #####0############################(SQL state 22003)#######

####################

##

```
LN ( # )
```

```
LOG ( # )
```

LOG10

LOG10###10#############

###0#################### ###

- #####NULL##########NULL###
- #####0############################(SQL state 22003)#######

####################

Syntax

```
LOG10 ( # )
```

LOCATE##

LOCATE###LOCATE#####################

##

```
LOCATE(###, ### [, ####] )
```

LOCATE###################################

- ################# ########
- ################## ##########
- ##LOCATE#############

LOCATE##########LOCATE###1########
############################LOCATE#0###################(")#########################
#########NULL########NULL######

```
-- 'love'#2##########2####
           VALUES LOCATE('love', 'clover')
```

```
-- 'clover'##'stove'#########0####
           VALUES LOCATE('stove', 'clover')
```

```
-- 5####(#####4)
           VALUES LOCATE('iss', 'Mississippi', 4)
```

```
-- ########1####
           VALUES LOCATE('', 'ABC')
```

```
-- ''##'AAA'#########0####
            VALUES LOCATE('AAA', '')

-- 3####
            VALUES LOCATE('', '', 3)
```

LTRIM##

LTRIM#############################

##

```
LTRIM(###)
```

A ##########CHAR#VARCHAR####LONG VARCHAR#########################

#############LTRIM#NULL######

```
-- 'asdf  '######
VALUES LTRIM(' asdf  ')
```

MAX##

MAX######################(## (####)##########)
MAX################################ (CHAR#VARCHAR#DATE#TIME#CHAR FOR BIT DATA#######)

##

```
MAX ( [ DISTINCT | ALL ] # )
```

DISTINCT#ALL##MAX############
###########DISTINCT############ ####################

```
SELECT COUNT (DISTINCT flying_time), MAX (DISTINCT miles)
FROM Flights
```

##
########################## ################################
(###*java.lang.Integer#int*########INTEGER########)
####NULL##########################

CHAR#VARCHAR####################MAX############## #####'z'#'z
'###

########MAX##################(###################)

```
-- FlightAvailability###############
SELECT MAX (flight_date) FROM FlightAvailability
-- ######################
-- ##10############
SELECT MAX(flying_time), orig_airport
FROM Flights
GROUP BY orig_airport
HAVING MAX(flying_time) > 10
```

MIN##

MIN######################## (## (####)##########)
MIN#######################(####CHAR#VARCHAR#DATE#TIME########)

##

```
MIN ( [ DISTINCT | ALL ] # )
```

DISTINCT#ALL##MIN############
########################### ################

```
SELECT COUNT (DISTINCT flying_time), MIN (DISTINCT miles)
FROM Flights
```

###################################

####################CHAR#VARCHAR###MIN############################
###'z'#'z
'###

#########################(###################)

```
-- ##########
SELECT DISTINCT flying_time, MIN(DISTINCT miles) from Flights
-- #####
SELECT COUNT(DISTINCT flying_time), MIN(DISTINCT miles) from Flights
-- #############
SELECT MIN (flight_date) FROM FlightAvailability;
```

MINUTE

MINUTE################

#################CLOB#LONG
VARCHAR#XML################################### ####0##59###########null###############

##

```
MINUTE ( # )
```

#

#flights######departure_time##6:00##6:30AM################

```
SELECT * FROM flights WHERE HOUR(departure_time) = 6 and
 MINUTE(departure_time) < 31;
```

MOD##

MOD#####################(#)######

##

```
mod(###, ###)
```

#######

- ######SMALLINT###SMALLIINT#
- ######INTEGER#######INTEGER####SMALLINT###INTEGER#
- ######BIGINT#######BIGINT####INTEGER####SMALLINT###BIGINT#

####NULL##################NULL#######NULL###

MONTH##

MONTH################

###############CLOB#LONG
VARCHAR#XML################################### #######1##12##############null##########

##

```
MONTH ( # )
```

#

EMPLOYEE#########(BIRTHDATE)#12###########

```
SELECT * FROM EMPLOYEE
WHERE MONTH(BIRTHDATE) = 12
```

NULLIF#

Derby########NULLIF#######

NULLIF####

```
NULLIF ( L, R )
```

NULLIF##CASE########## ####

```
NULLIF(V1,V2)
```

##CASE#######

```
CASE WHEN V1=V2 THEN NULL ELSE V1 END
```

PI

PI###pi#############

###pi################

##########################

##

```
PI (  )
```

RADIANS

RADIANS####################

########################

Attention: ###################

######################

##

```
RADIANS ( # )
```

RTRIM##

RTRIM####################

##

```
RTRIM(###)
```

A ####CHAR#VARCHAR#LONG VARCHAR############################

######null###RTRIM####null###

```
-- ' asdf'######
```

```
VALUES RTRIM(' asdf  ')
-- 'asdf'######
VALUES RTRIM('asdf  ')
```

SECOND##

SECOND#################

##############CLOB#LONG VARCHAR#XML###0##59#########null################

##

```
SECOND ( # )
```

RECEIVED##################2005-12-25-17.12.30.000000############

```
SECOND(RECEIVED)
```

#####30##########

SESSION_USER##

SESSION_USER########################## ##############*APP*######

USER#CURRENT_USER# SESSION_USER #########

##

```
SESSION_USER
```

```
VALUES SESSION_USER
```

SIN

SIN###############

############################## ###########################

- #####NULL##########NULL###
- ######(0)##########0###

######################

##

```
SIN ( # )
```

SMALLINT##

SMALLINT###################################

##

```
SMALLINT ( ## | ### )
```

##

###############
##
############################## ####################

###

##########
##########SQL##########
########## ##########
##########

##########null##########null##########null######null###

#
32767.99##########

```
VALUES SMALLINT (32767.99)
```

###32767###

1##########

```
VALUES SMALLINT (1)
```

###1###

SQRT##

##
SQRT##########

Note: ##########SQRT##########

##

```
SQRT(#######)
```

```
-- ##########
VALUES SQRT(3421E+09)

-- INTEGER##########
SELECT SQRT(myDoubleColumn) FROM MyTable

VALUES SQRT (CAST(25 AS FLOAT));
```

SUBSTR##

SUBSTR##########
##########VARCHAR########VARCHAR FOR BIT DATA###
##########

##

```
SUBSTR({ ### },
   #### [, #### ] )
```

########## ##########1###
##0#########Derby#1##########

#########CHAR#VARCHAR#LONG
VARCHAR#####(########)##########

##########

##########(######1######)##########

##########SUBSTR##########SUBSTR##########
##########SUBSTR##########

#
hello##########

```
VALUES SUBSTR('hello', 2)
```

###'ello'######

hello##

```
VALUES SUBSTR('hello',1,2)
```

###'he'######

SUM##

SUM ##########################(## (####)##########)
SUM##########################

##

```
SUM ( [ DISTINCT | ALL ] # )
```

DISTINCT###ALL#########################ALL#DISTINCT##########ALL##################
col)############

######DISTINCT############################

```
SELECT AVG (DISTINCT flying_time), SUM (DISTINCT miles)
FROM Flights
```

###NULL###

############################(##################)

```
-- #################
SELECT SUM (economy_seats) FROM Airlines;

-- SUM###############
-- (#################)
SELECT SUM (economy_seats_taken + business_seats_taken +
 firstclass_seats_taken)
as seats_taken FROM FLIGHTAVAILABILITY;
```

TAN

TAN###################

########################## ##################################
- ###NULL########NULL###
- ####(0)############

######################

##

```
TAN ( # )
```

TIME

TIME################

##############CLOB#LONG
VARCHAR#XML#################################
################null################null########### ###null########null###

######################
- ###################

- ##################################
- ############################

##

```
TIME ( # )
```

```
values time(current_timestamp)
```

#########5:03#######17:03:00###

TIMESTAMP##

TIMESTAMP##########################

##################################
- ############################CLOB#LONG VARCHAR#XML#################14##########
- ##
########################
- ##0
- ##
- ##14########################0######

Syntax

```
TIMESTAMP ( # [, # ] )
```

#
records_table############(1998-12-
25####)#################(17:12:30####)########## ########################

```
SELECT TIMESTAMP(col2, col3) FROM records_table
```

####1998-12-25-17:12:30.0######

```
VALUES TIMESTAMP('1998-12-25', '17.12.30');
1
--------------------------
1998-12-25 17:12:30.0
```

TRIM

TRIM########################/
##

##x

```
TRIM( [ trimOperands ] trimSource)
```

```
trimOperands  ::= { trimType [ trimCharacter ]  FROM | trimCharacter FROM
 }
trimType      ::= { LEADING | TRAILING | BOTH }
trimCharacter ::= ###
trimSource    ::= ###
```

trimType################BOTH########*trimCharacter*####################('
')########*trimCharacter*#########################

- ##############
- NULL

trimCharacter#trimSource#NULL#########TRIM######NULl#########TRIM###################

- trimType#LEADING######trimSource######trimChar##########

- trimType#TRAILING######trimSource#######trimChar##########
- trimType#BOTH######trimSource####*#*###trimChar##########

trimSource######CHAR####VARCHAR###TRIM#########VARCHAR###########TRIM#########CL

#

```
-- 'derby'######(#########)
VALUES TRIM('  derby ')

-- 'derby'###### (#########)
VALUES TRIM(BOTH ' ' FROM '  derby ')

-- 'derby '###### (###########)
VALUES TRIM(LEADING ' ' FROM '  derby ')

-- '  derby'###### (#############)
VALUES TRIM(TRAILING ' ' FROM '  derby ')

-- NULL######
VALUES TRIM(cast (null as char(1)) FROM '  derby ')

-- NULL######
VALUES TRIM(' ' FROM cast(null as varchar(30)))

-- ' derb'######(############)
VALUES TRIM('y' FROM ' derby')

-- trimCharacter#########################
VALUES TRIM('by' FROM ' derby')
```

UCASE#####UPPER##

UCASE###UPPER#######################################

##

```
UCASE###UPPER ( ### )
```

#######CHAR###########CHAR##################VARCHAR###
Note: UPPER#LOWER##*territory=ll_CC*
############

############################

#
#######
aSD1#w######################

```
VALUES UPPER('aSD1#w')
```

####ASD1#W###

USER##

USER#####################################*APP*######

USER#CURRENT_USER#SESSION_USER######

##

```
USER
```

```
VALUES USER
```

VARCHAR##

VARCHAR#####################

#################

```
VARCHAR (#### )
```

####
#######32,672#################

################

```
VARCHAR (### )
```

###
##################################

EMPLOYEE####"Dolores Quintana"######(CHAR(8)#JOB)##################

```
SELECT VARCHAR(JOB)
FROM EMPLOYEE
WHERE LASTNAME = 'QUINTANA'
```

XMLEXISTS

XMLEXISTS##SQL##XML###############SQL/XML######

XMLEXISTS###############XML#####DerbyXML####

##

```
XMLEXISTS ( xquery#######
    PASSING BY REF XML## [ BY REF ] )
```

xquery#######
##(######)###########Derb
xquery########Apache
Xalan#########XPath############Derby##XML#############Apache
Xalan######Xalan#XQuery##############Derby########## ##Xalan#######################
XPath###XQuery################################http://www.w3.org/TR/
xpath####http://www.w3.org/TR/xquery/

XML##
XML###########SQL/XML############# XML###################
Derby#########XML##
#####DerbyXMLQUERY###################
#########Derby##########

BY REF
###############Derby#####################
BY
REF###############XMLEXISTS###SQL/
XML############

###################
XMLEXISTS########xquery########XML#################SQL######
XMLEXISTS#########

UNKNOWN
XML###null#########

TRUE
######xml##################################

FALSE

######xml#########################

XMLEXISTS##XMLQUERY##############

XMLEXISTS#######SQL#################################XMLEXISTS###################XML

#
x_table#######xcol###XML####age###20#student#########################

```
SELECT id, XMLEXISTS('//student[@age=20]' PASSING BY REF xcol)
    FROM x_table
```

x_table####xcol###XML###null##########/roster/
student############ID##############

```
SELECT id FROM x_table WHERE XMLEXISTS('/roster/student' PASSING BY REF
 xcol)
```

x_table#######xcol###XML########XML#####################age###25#####student#####
################

```
CREATE TABLE x_table ( id INT, xcol XML CHECK (XMLEXISTS ('//student[@age
 < 25]' PASSING BY REF xcol)) )
```

######

Derby##XML#########Apache Xerces####JAXP
parser#Apache Xalan#Java###################### JAXP
parser####Xalan###################XMLEXISTS#################

XMLPARSE###

XMLPARSE#############DerbyXML####SQL/XML######

##############################Derby#XML################ ################XML###XMLEXIST

##

```
XMLPARSE (DOCUMENT ##### PRESERVE WHITESPACE)
```

DOCUMENT

#################Derby########XML###############
Derby#################XML###############
####Derby#JAXP#########################
JAXP#################XML###############
##########XML###########JAXP##########
Derby############SQLException###############

#####
CHAR#VARCHAR#LONGVARCHAR#CLOB###SQL#####################################
########CAST############CAST#######################Derby##########

PRESERVE WHITESPACE

###############Derby#####XML#########################
PRESERVE
WHITESPACE#########################SQL/
XML########Derby###############

####XML#####################http://www.w3.org/TR/REC-xml/#sec-well-formed .

####:
SQL/
XML########XMLPARSE#######################Derby#XMLPARSE#######################

#
#####x_table##xcolXML#####XML##########:

```
INSERT INTO x_table VALUES
    (1,
    XMLPARSE(DOCUMENT '
        <roster>
          <student age="18">AB</student>
          <student age="23">BC</student>
          <student>NOAGE</student>
        </roster>'
      PRESERVE WHITESPACE)
    )
```

JDBC####x_table##xcolXML######XML##################

```
INSERT INTO x_table VALUES
    (2,
    XMLPARSE (DOCUMENT CAST (? AS CLOB) PRESERVE WHITESPACE)
    )
```

######setCharacterStream()##################JDBC###setXXX#####################

######

Derby#XML############Apache Xerces####JAXP########Apache Xalan#Java#################

XMLQUERY###

XMLQUERY#SQL##XML###############SQL/XML######

XMLQUERY###################XML###########DerbyXML####

##

```
XMLQUERY ( xquery#######
    PASSING BY REF xml##
    [ RETURNING SEQUENCE [ BY REF ] ]
    EMPTY ON EMPTY
    )
```

xquery#######
##(######)#########
###xquery#########Apache
Xalan#######XPath############Derby######XML###########Apache
Xalan#######Xalan#XQuery##############Derby################ ##Xalan################
XPath###XQuery############Web#############http://www.w3.org/TR/
xpath#http://www.w3.org/TR/xquery/

xml##
###XML################SQL/XML############# xml##################
Derby##########XML## #####Derby#XMLQUERY####

BY REF
################Derby####################BY REF#################################
XML############

RETURNING SEQUENCE
################Derby#XMLQUERY###############XML##########SEQUENCE###########
XML############

EMPTY ON EMPTY
#################XMLQUERY################################XMLQUERY################

XMLQUERY########XML######
###XML##XMLQUERY###########XML##########

XML##
###XMLSERIALIZE ###################################

#
x_table####xcol###XML#######age###20#####students################

```
SELECT ID,
    XMLSERIALIZE(
        XMLQUERY('//student[@age>20]' PASSING BY REF xcol EMPTY ON EMPTY)
    AS VARCHAR(50))
FROM x_table
```

########XMLQUERY##########################x_table##############

x_table###xcol###XML#######BC#########################

```
SELECT ID,
    XMLSERIALIZE(
        XMLQUERY('string(//student[text() = "BC"]/@age)' PASSING BY REF
 xcol EMPTY ON EMPTY)
    AS VARCHAR(50))
FROM x_table
WHERE
    XMLEXISTS('//student[text() = "BC"]' PASSING BY REF xcol)
```

############x_table##BC#################

######

Derby#XML#############Apache Xerces####JAXP######Apache Xalan#Java###################

XMLSERIALIZE

XMLSERIALIZE#XML################SQL/XML#########Derby
XML##############

##: ####SQL/XML############
Derby#XMLSERIALIZE########################XMLSERIALIZE#########XML################
####[xString]#####XML##############################

```
INSERT INTO x_table (id, xcol)
    VALUES (3, XMLPARSE(DOCUMENT '[xString]' PRESERVE WHITESPACE));

SELECT id, XMLSERIALIZE(xcol AS VARCHAR(100))
    FROM x_table WHERE id = 3;
```

####XMLSERIALIZE##########[xString]################
XMLSERIALIZE#####SQL/XML#########################
XMLSERIALIZE##

##############XMLSERIALIZE###########XMLSERIALIZE####################JDBC####getX
Derby#XML##############

##

```
XMLSERIALIZE ( xml## AS ##### )
```

xml##
####Derby
XML############XMLQUERY######XML###############xml#####################

#####
CHAR#VARCHAR#LONG VARCHAR####CLOB###SQL##################
###################Derby##########

#
x_table####xcolXML####################

```
SELECT ID,
    XMLSERIALIZE(
    xcol AS CLOB)
FROM x_table
```

JDBC##################JDBC#getCharacterStream()#getString()###############

XMLQUERY#######################

```
SELECT ID,
    XMLSERIALIZE(
        XMLQUERY('//student[@age>20]'
           PASSING BY REF xcol EMPTY ON EMPTY)
    AS VARCHAR(50))
FROM x_table
```

######

Derby##XML##########Apache Xerces####JAXP######Apache Xalan#Java#####################

YEAR##

YEAR###
#######1##9999########## ###null##############null###########
#####null#######null###

##

```
YEAR ( # )
```

Example

PROJECT#################(PRSTDATE)####(PRENDATE)###################

```
SELECT * FROM PROJECT
WHERE YEAR(PRSTDATE) = YEAR(PRENDATE)
```

#########

########Derby#################

SYSCS_UTIL.SYSCS_CHECK_TABLE######

SYSCS_UTIL.SYSCS_CHECK_TABLE###
##########################

##

```
SMALLINT SYSCS_UTIL.SYSCS_CHECK_TABLE(IN SCHEMANAME VARCHAR(128),
IN TABLENAME VARCHAR(128))
```

SCHEMANAME####TABLENAME#null################

#

```
VALUES SYSCS_UTIL.SYSCS_CHECK_TABLE('SALES', 'ORDERS');
```

SYSCS_UTIL.SYSCS_GET_DATABASE_PROPERTY######

SYSCS_UTIL.SYSCS_GET_DATABASE_PROPERTY###############KEY###############

##

```
VARCHAR(32762) SYSCS_UTIL.SYSCS_GET_DATABASE_PROPERTY(IN KEY
 VARCHAR(128))
```

KEY#null###############

#

```
VALUES SYSCS_UTIL.SYSCS_GET_DATABASE_PROPERTY('key_value_string');
```

SYSCS_UTIL.SYSCS_GET_RUNTIMESTATISTICS

SYSCS_UTIL.SYSCS_GET_RUNTIMESTATISTICS####java.sql.ResultSet#########################
########################## ################## ###################################
SYSCS_UTIL.SYSCS_GET_RUNTIMESTATISTICS#############SELECT#INSERT#UPDATE##DML#

##

```
VARCHAR(32762) SYSCS_UTIL.SYSCS_GET_RUNTIMESTATISTICS()
```

#

```
VALUES SYSCS_UTIL.SYSCS_GET_RUNTIMESTATISTICS()
```

SYSCS_UTIL.SYSCS_GET_USER_ACCESS ######

SYSCS_UTIL.SYSCS_GET_USER_ACCESS##############################

##derby.database.defaultConn

##

```
SYSCS_UTIL.SYSCS_GET_USER_ACCESS (USERNAME VARCHAR(128)) RETURNS
 VARCHAR(128)
```

USERNAME
Derby###########ID######VARCHAR(128)#########

##########fullAccess#readOnlyAccess#noAccess########

noAccess###*derby.database.fullAccessUsers*#########*derby.*

########Derby##############

#

```
VALUES SYSCS_UTIL.SYSCS_GET_USER_ACCESS ('BRUNNER')
```

############

###################################SQL#########################Derby########

SYSCS_UTIL.SYSCS_BACKUP_DATABASE#######

SYSCS_UTIL.SYSCS_BACKUP_DATABASE##

##

```
SYSCS_UTIL.SYSCS_BACKUP_DATABASE(IN BACKUPDIR VARCHAR())
```

################

BACKUPDIR
#######VARCHAR(32672)###JV
######################

JDBC##

#############c:/backupdir#################

```
CallableStatement cs = conn.prepareCall
("CALL SYSCS_UTIL.SYSCS_BACKUP_DATABASE(?)");
cs.setString(1, "c:/backupdir");
cs.execute();
cs.close();
```

SQL##

#############c:/backupdir#################

```
CALL SYSCS_UTIL.SYSCS_BACKUP_DATABASE('c:/backupdir');
```

SYSCS_UTIL.SYSCS_BACKUP_DATABASE_NOWAIT #######

SYSCS_UTIL.SYSCS_BACKUP_DATABASE_NOWAIT###

###SYSCS_UTIL.SYSCS_BACKUP_DATABASE_NOWAI

##

```
SYSCS_UTIL.SYSCS_BACKUP_DATABASE_NOWAIT(IN BACKUPDIR VARCHAR())
```

###############

BACKUPDIR
#######VARCHAR(32672)###

JDBC##

#############c:/backupdir#################

```
CallableStatement cs = conn.prepareCall
("CALL SYSCS_UTIL.SYSCS_BACKUP_DATABASE_NOWAIT(?)");
cs.setString(1, "c:/backupdir");
cs.execute();
cs.close();
```

SQL##

#############c:/backupdir#################

```
CALL SYSCS_UTIL.SYSCS_BACKUP_DATABASE_NOWAIT('c:/backupdir');
```

SYSCS_UTIL.SYSCS_BACKUP_DATABASE_AND_ENABLE_LOG_ARCHIVE_MODE#######

SYSCS_UTIL.SYSCS_BACKUP_DATABASE_AND_ENABLE_LOG_ARCHIVE_MODE#####################

##

```
SYSCS_UTIL.SYSCS_BACKUP_DATABASE_AND_ENABLE_LOG_ARCHIVE_MODE
(IN BACKUPDIR VARCHAR(32672), IN SMALLINT DELETE_ARCHIVED_LOG_FILES)
```

###############

BACKUPDIR
#######VARCHAR(32672)################################### ##############JVM#user
DELETE_ARCHIVED_LOG_FILES
#####DELETE_ARCHIVED_LOG_FILES###

JDBC##

#############c:/backupdir#################

```
CallableStatement cs = conn.prepareCall
```

```
("CALL SYSCS_UTIL.SYSCS_BACKUP_DATABASE_AND_ENABLE_LOG_ARCHIVE_MODE(?,
 ?)");
cs.setString(1, "c:/backupdir");
cs.setInt(2, 0);
cs.execute();
```

SQL##

############c:/
backupdir###

```
SYSCS_UTIL.SYSCS_BACKUP_DATABASE_AND_ENABLE_LOG_ARCHIVE_MODE('c:/
backupdir', 0)
```

############c:/
backupdir###

```
SYSCS_UTIL.SYSCS_BACKUP_DATABASE_AND_ENABLE_LOG_ARCHIVE_MODE('c:/
backupdir', 1)
```

SYSCS_UTIL.SYSCS_BACKUP_DATABASE_AND_ENABLE_LOG_ARCHIVE_MODE_NOWAIT#######

SYSCS_UTIL.SYSCS_BACKUP_DATABASE_AND_ENABLE_LOG_ARCHIVE_MODE_NOWAIT############
###

##

```
SYSCS_UTIL.SYSCS_BACKUP_DATABASE_AND_ENABLE_LOG_ARCHIVE_MODE_NOWAIT
(IN BACKUPDIR VARCHAR(32672),
IN SMALLINT DELETE_ARCHIVED_LOG_FILES)
```

################

BACKUPDIR
#######VARCHAR(32672)################################### #########JVM#user.dir#
######################
DELETE_ARCHIVED_LOG_FILES
#####DELETE_ARCHIVED_LOG_FILES###

JDBC##

############c:/backupdir#####################################

```
CallableStatement cs = conn.prepareCall
("CALL
 SYSCS_UTIL.SYSCS_BACKUP_DATABASE_AND_ENABLE_LOG_ARCHIVE_MODE_NOWAIT(?,
 ?)");
cs.setString(1, "c:/backupdir");
cs.setInt(2, 0);
cs.execute();
```

SQL##

##############c:/
backupdir###

```
SYSCS_UTIL.SYSCS_BACKUP_DATABASE_AND_ENABLE_LOG_ARCHIVE_MODE_NOWAIT('c:/
backupdir', 0)
```

##############c:/
backupdir###

```
SYSCS_UTIL.SYSCS_BACKUP_DATABASE_AND_ENABLE_LOG_ARCHIVE_MODE_NOWAIT('c:/
backupdir', 1)
```

SYSCS_UTIL.SYSCS_EMPTY_STATEMENT_CACHE #######

########SYSCS_UTIL.SYSCS_EMPTY_STATEMENT_CACHE################

##

```
SYSCS_UTIL.SYSCS_EMPTY_STATEMENT_CACHE()
```

#

```
CALL SYSCS_UTIL.SYSCS_EMPTY_STATEMENT_CACHE()
```

SYSCS_UTIL.SYSCS_CHECKPOINT_DATABASE######

SYSCS_UTIL.SYSCS_CHECKPOINT_DATABASE##

##

```
SYSCS_UTIL.SYSCS_CHECKPOINT_DATABASE()
```

################

JDBC##

```
CallableStatement cs = conn.prepareCall
("CALL SYSCS_UTIL.SYSCS_CHECKPOINT_DATABASE()");
cs.execute();
cs.close();
```

SQL##

```
CALL SYSCS_UTIL.SYSCS_CHECKPOINT_DATABASE();
```

SYSCS_UTIL.SYSCS_COMPRESS_TABLE######

SYSCS_UTIL.SYSCS_COMPRESS_TABLE##
##Derby##########(OS
SYSCS_UTIL.SYSCS_COMPRESS_TABLE##################(OS)########

SYSCS_UTIL.SYSCS_COMPRESS_TABLE##

##

```
SYSCS_UTIL.SYSCS_COMPRESS_TABLE (IN SCHEMANAME VARCHAR(128),
IN TABLENAME VARCHAR(128), IN SEQUENTIAL SMALLINT)
```

SCHEMANAME
##VARCHAR(128)##
TABLENAME
##VARCHAR(128)##"Fred"#####SQL######
################
SEQUENTIAL
####SMALLINT##################################0#################################

SQL ##

SEQUENTIAL######US#######CUSTOMER###########

```
call SYSCS_UTIL.SYSCS_COMPRESS_TABLE('US', 'CUSTOMER', 1)
```

Java ##

SEQUENTIAL######US#######CUSTOMER###########

```
CallableStatement cs = conn.prepareCall
("CALL SYSCS_UTIL.SYSCS_COMPRESS_TABLE(?, ?, ?)");
cs.setString(1, "US");
cs.setString(2, "CUSTOMER");
cs.setShort(3, (short) 1);
cs.execute();
```

SEQUENTIAL##########Derby########################## SEQUENTIAL##########
#############Derby##
(##########################)##############################COMMIT#########(OS)#######

SEQUENTIAL##########Derby##########################SEQUENTIAL##########

SYSCS_UTIL.SYSCS_COMPRESS_TABLE##COMMIT######################(OS)#######
################################COMMIT###
###: #############SYSCS_UTIL.SYSCS_COMPRESS_TABLE##################

Note:
###
##################
SYSCS_UTIL.SYSCS_INPLACE_COMPRESS_TABLE#######

SYSCS_UTIL.SYSCS_INPLACE_COMPRESS_TABLE###
###
####Derby#############(OS)###(OS)#########
SYSCS_UTIL.SYSCS_INPLACE_COMPRESS_TABLE##################(OS)#############

####################SQL###############
#######PURGE_ROWS#DEFRAGMENT_ROWS### TRUNCATE_END#####SYSCS_UTIL.SYSCS_COMPRESS_

##

```
SYSCS_UTIL.SYSCS_INPLACE_COMPRESS_TABLE(
  IN SCHEMANAME VARCHAR(128),
  IN TABLENAME VARCHAR(128),
  IN PURGE_ROWS SMALLINT,
  IN DEFRAGMENT_ROWS SMALLINT,
  IN TRUNCATE_END SMALLINT )
```

SCHEMANAME
#####VARCHAR(128)###
TABLENAME
#####VARCHAR(128)##"Fred"########SQL#
################
PURGE_ROWS
PURGE_ROWS#0###
###
DEFRAGMENT_ROWS
DEFRAGMENT_ROWS#0## ##############
TRUNCATE_END########DEFRAGMENT_ROWS############## DEFRAGMENT_ROWS######
TRUNCATE_END
TRUNCATE_END#0################################(OS)###########PURGE_ROWS#DEFR

SQL##
US##########CUSTOMER##########################

```
call SYSCS_UTIL.SYSCS_INPLACE_COMPRESS_TABLE('US', 'CUSTOMER', 1, 1, 1);
```

###

```
call SYSCS_UTIL.SYSCS_INPLACE_COMPRESS_TABLE('US', 'CUSTOMER', 0, 0, 1);
```

Java##
US##########CUSTOMER##########################

```
CallableStatement cs = conn.prepareCall
("CALL SYSCS_UTIL.SYSCS_INPLACE_COMPRESS_TABLE(?, ?, ?, ?, ?)");
cs.setString(1, "US");
cs.setString(2, "CUSTOMER");
cs.setShort(3, (short) 1);
cs.setShort(4, (short) 1);
```

```
cs.setShort(5, (short) 1);
cs.execute();
```

##

```
CallableStatement cs = conn.prepareCall
("CALL SYSCS_UTIL.SYSCS_INPLACE_COMPRESS_TABLE(?, ?, ?, ?, ?)");
cs.setString(1, "US");
cs.setString(2, "CUSTOMER");
cs.setShort(3, (short) 0);
cs.setShort(4, (short) 0);
cs.setShort(5, (short) 1);
cs.execute();
```

###:

SYSCS_UTIL.SYSCS_INPLACE_COMPRESS_TABLE############################

Note:

##*Derby* ####################

SYSCS_UTIL.SYSCS_DISABLE_LOG_ARCHIVE_MODE#######

SYSCS_UTIL.SYSCS_DISABLE_LOG_ARCHIVE_MODE#####################################DEL

##

```
SYSCS_UTIL.SYSCS_DISABLE_LOG_ARCHIVE_MODE(IN SMALLINT
 DELETE_ARCHIVED_LOG_FILES)
```

#################

DELETE_ARCHIVED_LOG_FILES

##DELETE_ARCHIVED_LOG_FILES################################ ##############################

JDBC##

###

```
CallableStatement cs = conn.prepareCall
("CALL SYSCS_UTIL.SYSCS_DISABLE_LOG_ARCHIVE_MODE(?)");
cs.setInt(1, 1);
cs.execute();
cs.close();
```

SQL##

###

```
CALL SYSCS_UTIL.SYSCS_DISABLE_LOG_ARCHIVE_MODE
 DELETE_ARCHIVED_LOG_FILES(0);
```

###

```
CALL SYSCS_UTIL.SYSCS_DISABLE_LOG_ARCHIVE_MODE
 DELETE_ARCHIVED_LOG_FILES(1);
```

SYSCS_UTIL.SYSCS_EXPORT_TABLE #######

SYSCS_UTIL.SYSCS_EXPORT_TABLE################(OS)#################################

SYSCS_UTIL.SYSCS_EXPORT_TABLE#####################################(OS)#############

##

##############################

##

```
SYSCS_UTIL.SYSCS_EXPORT_TABLE (IN SCHEMANAME  VARCHAR(128),
IN TABLENAME VARCHAR(128), IN FILENAME VARCHAR(32672),
IN COLUMNDELIMITER CHAR(1), IN CHARACTERDELIMITER CHAR(1),
IN CODESET VARCHAR(128))
```

###################

SCHEMANAME
An input argument of type VARCHAR(128) that specifies the schema name of the table. Passing a NULL value will use the default schema name.

TABLENAME
#######VARCHAR(128)##

FILENAME
##
(32672)##############

COLUMNDELIMITER
#######CHAR(1)##

CHARACTERDELIMITER
#######CHAR(1)##
#########################(")###

CODESET
#######VARCHAR(128)###Java############
#################JVM#################

##

###

###################*Derby*
###

#

#####SAMPLE#########STAFF###myfile.del#########################

```
CALL SYSCS_UTIL.SYSCS_EXPORT_TABLE (null, 'STAFF', 'myfile.del', null,
 null, null);
```

SYSCS_UTIL.SYSCS_EXPORT_TABLE_LOBS_TO_EXTFILE #######

SYSCS_UTIL.SYSCS_EXPORT_TABLE_LOBS_TO_EXTFILE##################################LO

###

##############################

##

```
SYSCS_UTIL.SYSCS_EXPORT_TABLE_LOBS_TO_EXTFILE (
    IN SCHEMANAME VARCHAR(128),
    IN TABLENAME VARCHAR(128),
    IN FILENAME VARCHAR(32672),
    IN COLUMNDELIMITER CHAR(1),
    IN CHARACTERDELIMITER CHAR(1),
    IN CODESET VARCHAR(128)
    IN LOBSFILENAME VARCHAR(32672)
    )
```

###

SCHEMANAME
#############################NULL#############SCHEMANAME########VARCHAR
(128)##################

TABLENAME
###TA
(128)#############

FILENAME
##
##NULL#################FILENAME#######
(32672)################
COLUMNDELIMITER
##NULL#############
COLUMNDELIMITER######CHAR (1)###############
CHARACTERDELIMITER
##NULL#############
CHARACTERDELIMITER########CHAR (1)################
CODESET
#######################################Java###
(128)################
LOBSFILENAME
##LOB#################################
NULL#################
LOBSFILENAME#######VARCHAR (32672)################

##############

###

####################Derby
##

######LOB######################

####################STAFF####staff.del#########pictures.dat###LOB#################

```
CALL SYSCS_UTIL.SYSCS_EXPORT_TABLE_LOBS_TO_EXTFILE(
    'APP', 'STAFF', 'c:#data#staff.del', ',' ,'"',
    'UTF-8', 'c:#data#pictures.dat');
```

SYSCS_UTIL.SYSCS_EXPORT_QUERY#######

The SYSCS_UTIL.SYSCS_EXPORT_QUERY system procedure exports the results of a SELECT statement to an operating system file.

###

##############################

##

```
SYSCS_UTIL.SYSCS_EXPORT_QUERY(IN SELECTSTATEMENT VARCHAR(32672),
IN FILENAME VARCHAR(32672), IN COLUMNDELIMITER CHAR(1),
IN CHARACTERDELIMITER CHAR(1), IN CODESET VARCHAR(128))
```

###############

SELECTSTATEMENT
#######VARCHAR(32672)#########################(###)#########################
FILENAME
##
FILENAME######VARCHAR (32672)###############
COLUMNDELIMITER
#######CHAR(1)###(,
CHARACTERDELIMITER
#######CHAR(1)###
CODESET
#######VARCHAR(128)################################
##########Java################################

##
##########JVM#####################

###

#######################Derby
###

#

#####SAMPLE#########STAFF####myfile.del#########################

```
CALL SYSCS_UTIL.SYSCS_EXPORT_QUERY('select * from staff where dept =20',
'c:/output/awards.del', null, null, null);
```

SYSCS_UTIL.SYSCS_EXPORT_QUERY_LOBS_TO_EXTFILE #######

SYSCS_UTIL.SYSCS_EXPORT_QUERY_LOBS_TO_EXTFILE##########################LOB#######

##

##############################

##

```
SYSCS_UTIL.SYSCS_EXPORT_QUERY_LOBS_TO_EXTFILE (
    IN SELECTSTATEMENT VARCHAR(32672),
    IN FILENAME VARCHAR(32672),
    IN COLUMNDELIMITER CHAR(1),
    IN CHARACTERDELIMITER CHAR(1),
    IN CODESET VARCHAR(128)
    IN LOBSFILENAME VARCHAR(32672)
    )
```

##

SELECTSTATEMENT

##########################
NULL################SELECTSTATEMENT######VARCHAR
(32672)###############

FILENAME

##
(32672)###############

COLUMNDELIMITER

##
NULL#######################COLUMNDELIMITER######CHAR
(1)##################

CHARACTERDELIMITER

##
NULL#########################CHARACTERDELIMITER######CHAR
(1)##################

CODESET

##Java#####################################
NULL##########JVM#####################CODESET######VARCHAR
(128)###############

LOBSFILENAME

large
object#####################################lob#####################################
#####################################
NULL################LOBSFILENAME######VARCHAR
(32672)###############

###

####################Derby
###

LOB####################################
#############STAFF####20##################staff.del#lob####pictures.dat############

```
CALL SYSCS_UTIL.SYSCS_EXPORT_QUERY_LOBS_TO_EXTFILE(
    'SELECT * FROM STAFF WHERE dept=20',
    'c:#data#staff.del', ',' ,'"',
    'UTF-8','c:#data#pictures.dat');
```

SYSCS_UTIL.SYSCS_IMPORT_DATA######

SYSCS_UTIL.SYSCS_IMPORT_DATA###
##

##

```
SYSCS_UTIL.SYSCS_IMPORT_DATA (IN SCHEMANAME VARCHAR(128),
IN TABLENAME VARCHAR(128), IN INSERTCOLUMNS VARCHAR(32672),
IN COLUMNINDEXES VARCHAR(32672), IN FILENAME VARCHAR(32672),
IN COLUMNDELIMITER CHAR(1), IN CHARACTERDELIMITER CHAR(1),
IN CODESET VARCHAR(128), IN REPLACE SMALLINT)
```

###############

SCHEMANAME
#######VARCHAR(128)##
TABLENAME
#######VARCHAR
(128)###
INSERTCOLUMNS
#######VARCHAR (32762)##########################(#########)########
COLUMNINDEXES
#######VARCHAR
(32762)####################(1#################)#####################################
FILENAME
#######VARCHAR(32672)#########################
###
COLUMNDELIMITER
#######CHAR(1)###
CHARACTERDELIMITER
#######CHAR(1)###
CODESET
#######VARCHAR(128)#######################################Java#####################
##################################(utf-
8)############################JVM#################
REPLACE
#######SMALLINT###########REPLACE#########INSERT#########REPLACE#############
##################################REPLACE#########INSERT#########################

###################

###

###############Derby
###

#
######data.del################################staff###########

```
CALL SYSCS_UTIL.SYSCS_IMPORT_DATA
(NULL, 'STAFF', null, '1,3,4', 'data.del', null, null, null,0)
```

SYSCS_UTIL.SYSCS_IMPORT_DATA_LOBS_FROM_EXTFILE #######

SYSCS_UTIL.SYSCS_IMPORT_DATA_LOBS_FROM_EXTFILE##################################
################LOB############################

##

```
SYSCS_UTIL.SYSCS_IMPORT_DATA_LOBS_FROM_EXTFILE (
   IN SCHEMANAME VARCHAR(128),
   IN TABLENAME VARCHAR(128),
   IN INSERTCOLUMNS VARCHAR(32672),
   IN COLUMNINDEXES VARCHAR(32672),
   IN FILENAME VARCHAR(32672),
   IN COLUMNDELIMITER CHAR(1),
   IN CHARACTERDELIMITER CHAR(1),
   IN CODESET VARCHAR(128),
   IN REPLACE SMALLINT)
   )
```

##############LOB###################################

SCHEMANAME
##############################NULL###############
SCHEMANAME#######VARCHAR (128)#############
TABLENAME
###################### ######################
############################NULL################
TABLENAME#######VARCHAR (128)#############
INSERTCOLUMNS
################################## ################NULL###############
INSERTCOLUMNS#######VARCHAR (32672)#############
COLUMNINDEXES
########################(1#######)###############
###############NULL#############COLUMNINDEXES#######VARCHAR
(32762)#############
FILENAME
##
NULL############### fileName#######VARCHAR (32672)#############
COLUMNDELIMITER
##NULL#########
COLUMNDELIMITER#######CHAR (1)#############
CHARACTERDELIMITER
##NULL#########
CHARACTERDELIMITER####### CHAR (1)#############
CODESET
##################################Java#########################
##############################(UTF-8)#########
####JVM########################NULL##############
CODESET#######VARCHAR (128)#############
REPLACE
########0##################REPLACE#############0##############INSERT##########
REPLACE## ##############################
######NULL################ REPLACE#######SMALLINT#############

######################

###

##############################LOB#########
##################LOB######lobsFileName.Offset.length/##############

- Offset######################

- length#########LOB##########

###############*Derby*
##

##################LOB###################
#######STAFF############################
STAFF##############LOB#######
##########staff.del###################staff.del#######LOB#########################
########################(")###########(')##############
##########STAFF###############

```
CALL SYSCS_UTIL.SYSCS_IMPORT_DATA_LOBS_FROM_EXTFILE
    (null, 'STAFF', 'NAME,DEPT,SALARY,PICTURE', '2,3,4,6',
+    'c:#data#staff.del', ',','""','UTF-8',  0);
```

SYSCS_UTIL.SYSCS_IMPORT_TABLE######

SYSCS_UTIL.SYSCS_IMPORT_TABLE###################################
##

##

```
SYSCS_UTIL.SYSCS_IMPORT_TABLE (IN SCHEMANAME VARCHAR(128),
IN TABLENAME VARCHAR(128), IN FILENAME VARCHAR(32672),
IN COLUMNDELIMITER CHAR(1), IN CHARACTERDELIMITER CHAR(1),
IN CODESET VARCHAR(128), IN REPLACE SMALLINT)
```

###############

SCHEMANAME
#######VARCHAR(128)##
TABLENAME
#######VARCHAR (128)#########################
##
FILENAME
#######VARCHAR(32672)##############################
##
COLUMNDELIMITER
#######CHAR(1)################## ################################
########################(,)###
CHARACTERDELIMITER
#######CHAR(1)################## ################################
########################(")###
CODESET
#######VARCHAR(128)##Java################
8)######## ##################JVM##########################
REPLACE
#######SMALLINT###########REPLACE################INSERT#########
REPLACE## ####################INSER
##
##

###

###############*Derby*
##

#
#######myfile.del###################STAFF#####################################(%)##

```
CALL SYSCS_UTIL.SYSCS_IMPORT_TABLE
(null, 'STAFF', 'c:/output/myfile.del', ';', '%', null,0);
```

############Derby ########################

SYSCS_UTIL.SYSCS_IMPORT_TABLE_LOBS_FROM_EXTFILE #######

SYSCS_UTIL.SYSCS_IMPORT_TABLE_LOBS_FROM_EXTFILE##############################LOB
#############LOB##########################

##

```
SYSCS_UTIL.SYSCS_IMPORT_TABLE_LOBS_FROM_EXTFILE (
   IN SCHEMANAME VARCHAR(128),
   IN TABLENAME VARCHAR(128),
   IN FILENAME VARCHAR(32672),
   IN COLUMNDELIMITER CHAR(1),
   IN CHARACTERDELIMITER CHAR(1),
   IN CODESET VARCHAR(128),
   IN REPLACE SMALLINT)
   )
```

###############LOB###################################

SCHEMANAME
############################NULL###########SCHEMANAME######VARCHAR
(128)#############
TABLENAME
###NULL########
(128)#############
FILENAME
###
COLUMNDELIMITER
###NULL###########COLUMNDEL
(1)#############
CHARACTERDELIMITER
###NULL##########
(1)#############
CODESET
###################################Java###
8)############JVM##################NULL###########CODESET######VARCHAR
(128)#############
REPLACE
0#############REPLACE##################0#########INSERT################REPLAC

###

###

############################LOB########### Derby##################################LO
#######

- Offset######################
- length#LOB##################

######################Derby
##

LOB##

###################STAFF##staff.del###

```
CALL SYSCS_UTIL.SYSCS_IMPORT_TABLE_LOBS_FROM_EXTFILE(
    'APP','STAFF','c:\data\staff.del',',','"','UTF-8',0);
```

SYSCS_UTIL.SYSCS_FREEZE_DATABASE######

SYSCS_UTIL.SYSCS_FREEZE_DATABASE#####################################

##

```
SYSCS_UTIL.SYSCS_FREEZE_DATABASE()
```

###############

#

```
String backupdirectory = "c:/mybackups/" + JCalendar.getToday();
CallableStatement cs = conn.prepareCall
("CALL SYSCS_UTIL.SYSCS_FREEZE_DATABASE()");
cs.execute();
cs.close();
// "backupdirectory"####################
// ########################
CallableStatement cs = conn.prepareCall
("CALL SYSCS_UTIL.SYSCS_UNFREEZE_DATABASE()");
cs.execute();
cs.close();
```

SYSCS_UTIL.SYSCS_UNFREEZE_DATABASE #######

#########SYSCS_UTIL.SYSCS_UNFREEZE_DATABASE#####################

##

```
SYSCS_UTIL.SYSCS_UNFREEZE_DATABASE()
```

###############

#

```
String backupdirectory = "c:/mybackups/" + JCalendar.getToday();
CallableStatement cs = conn.prepareCall
("CALL SYSCS_UTIL.SYSCS_FREEZE_DATABASE()");
cs.execute();
cs.close();
// "backupdirectory"##################
// ##########################
now unfreeze the database once backup has completed:
CallableStatement cs = conn.prepareCall
("CALL SYSCS_UTIL.SYSCS_UNFREEZE_DATABASE()");
cs.execute();
cs.close();
```

SYSCS_UTIL.SYSCS_RELOAD_SECURITY_POLICY #######

SYSCS_UTIL.SYSCS_RELOAD_SECURITY_POLICY################################Java#######
####################*Derby*
##*Derby* ########
#####Derby##################################

##

```
SYSCS_UTIL.SYSCS_RELOAD_SECURITY_POLICY()
```

################

#

```

CallableStatement cs = conn.prepareCall
("CALL SYSCS_UTIL.SYSCS_RELOAD_SECURITY_POLICY()");
cs.execute();
cs.close();
```

SYSCS_UTIL.SYSCS_SET_DATABASE_PROPERTY######

SYSCS_UTIL.SYSCS_SET_DATABASE_PROPERTY##################################/
######

"VALUE"#null########"KEY"###############"VALUE"########"VALUE"#null####"KEY"##########

##

```
SYSCS_UTIL.SYSCS_GET_DATABASE_PROPERTY(IN KEY VARCHAR(128),
IN VALUE VARCHAR(32672))
```

###############

JDBC##

derby.locks.deadlockTimeout###10##########

```
CallableStatement cs = conn.prepareCall
("CALL SYSCS_UTIL.SYSCS_SET_DATABASE_PROPERTY(?, ?)");
cs.setString(1, "derby.locks.deadlockTimeout");
cs.setString(2, "10");
cs.execute();
cs.close();
```

SQL##

derby.locks.deadlockTimeout###10##########

```
CALL SYSCS_UTIL.SYSCS_SET_DATABASE_PROPERTY
('derby.locks.deadlockTimeout', '10');
```

SYSCS_UTIL.SYSCS_SET_RUNTIMESTATISTICS######

SYSCS_UTIL.SYSCS_SET_RUNTIMESTATISTICS()#######################on#off##########
##################runtimestatistics###on########off########Derby##########commit###
runtimestatistics###off#########0##########
runtimestatistics###on#########0##################

###next()#0#

##

```
SYSCS_UTIL.SYSCS_SET_RUNTIMESTATISTICS(IN SMALLINT ENABLE)
```

#

```
-- ######
-- ###RUNTIMESTATISTIC#on#####
CALL SYSCS_UTIL.SYSCS_SET_RUNTIMESTATISTICS(1);
-- ############
-- ############
-- #################
CALL SYSCS_UTIL.SYSCS_SET_RUNTIMESTATISTICS(0);
```

SYSCS_UTIL.SYSCS_SET_STATISTICS_TIMING######

SYSCS_UTIL.SYSCS_SET_STATISTICS_TIMING####################on#off##############
########off### runtimestatistics#####on############on######
runtimestatistics###off#########on###############

#####on#########0################# #####off#########0#############

#####on########Derby##################################SYSCS_UTIL.SYSCS_GET_RUNTIMES
#####off###SYSCS_UTIL.SYSCS_GET_RUNTIMESTATISTICS####################0######

##

```
SYSCS_UTIL.SYSCS_SET_STATISTICS_TIMING(IN SMALLINT ENABLE)
```

#

runtimestatistics###############on#####

```
CALL SYSCS_UTIL.SYSCS_SET_RUNTIMESTATISTICS(1);
CALL SYSCS_UTIL.SYSCS_SET_STATISTICS_TIMING(1);
```

SYSCS_UTIL.SYSCS_SET_USER_ACCESS#######

SYSCS_UTIL.SYSCS_SET_USER_ACCESS############################

##

```
SYSCS_UTIL.SYSCS_SET_USER_ACCESS USERNAME VARCHAR(128),
 CONNECTION_PERMISSION VARCHAR(128))
```

USERNAME

VARCHAR(128)############Derby##########ID#######

CONNECTION_PERMISSION

###CONNECTION_PERMISSION###########

fullAccess

##derby.database.fullAccessUsers######

readOnlyAccess

##derby.database.readOnlyAccessUse

null

##null#############

#

```
CALL SYSCS_UTIL.SYSCS_SET_USER_ACCESS ('BRUNNER', 'readOnlyAccess')
```

################null######################

```
CALL SYSCS_UTIL.SYSCS_SET_USER_ACCESS ('ISABEL', null)
```

SYSCS_DIAG

Derby##

Derby################
###

Derby############### ################################
#####

Derby##############################0######################SQL###################

######Derby#################

 # 9. Derby#########

####	####
SYSCS_DIAG.ERROR_LOG_READER	###
SYSCS_DIAG.ERROR_MESSAGES	#
SYSCS_DIAG.LOCK_TABLE	#
SYSCS_DIAG.SPACE_TABLE	###
SYSCS_DIAG.STATEMENT_CACHE	#
SYSCS_DIAG.STATEMENT_DURATION	###

####	####
SYSCS_DIAG.TRANSACTION_TABLE	#

####: DDL####################Derby#########

SYSCS_DIAG.ERROR_LOG_READER #####

SYSCS_DIAG.ERROR_LOG_READER#########derby.log#########################SQL######

##

SYSCS_DIAG.ERROR_LOG_READER###########SQL##################

#######:

```
SELECT *
    FROM TABLE (SYSCS_DIAG.ERROR_LOG_READER())
    AS T1
```

###T1############################

SYSCS_DIAG.ERROR_LOG_READER###

For example:

```
SELECT *
    FROM TABLE (SYSCS_DIAG.ERROR_LOG_READER('myderbyerrors.log'))
    AS T1
```

###: ####Derby################################ ########Derby################*Derby*######

SYSCS_DIAG.ERROR_MESSAGES ###

SYSCS_DIAG.ERROR_MESSAGES#########SQLState#################
Derby####################
######SYSCS_DIAG.ERROR_MESSAGES################

#######

```
SELECT * FROM SYSCS_DIAG.ERROR_MESSAGES
```

SYSCS_DIAG.LOCK_TABLE ###

SYSCS_DIAG.LOCK_TABLE###########Derby##################
SYSCS_DIAG.LOCK_TABLE####################

#######

```
SELECT * FROM SYSCS_DIAG.LOCK_TABLE
```

SYSCS_DIAG.LOCK_TABLE####################################
##
#####################################

SYSCS_DIAG.SPACE_TABLE #####

SYSCS_DIAG.SPACE_TABLE###################################
##

SYSCS_DIAG.SPACE_TABLE###########SQL################
############################## ##

##

```
SELECT T2.*
    FROM
        SYS.SYSTABLES systabs,
```

```
        TABLE (SYSCS_DIAG.SPACE_TABLE(systabs.tablename)) AS T2
    WHERE systabs.tabletype = 'T'
```

###T2#########################

#############Java#######################
##

#######

```
SELECT *
    FROM TABLE (SYSCS_DIAG.SPACE_TABLE('MYSCHEMA', 'MYTABLE'))
    AS T2
```

SYSCS_DIAG.STATEMENT_CACHE ###

SYSCS_DIAG.STATEMENT_CACHE#####SQL#################
SYSCS_DIAG.STATEMENT_CACHE################

#######

```
SELECT * FROM SYSCS_DIAG.STATEMENT_CACHE
```

SYSCS_DIAG.STATEMENT_DURATION #####

SYSCS_DIAG.STATEMENT_DURATION##########derby.log##################SQL#########

######################JDBC###################################

SYSCS_DIAG.STATEMENT_DURATION###########SQL##################

#######

```
SELECT *
    FROM TABLE (SYSCS_DIAG.STATEMENT_DURATION())
    AS T1
```

###T1#########################

####:
##################ID###ID##########

SYSCS_DIAG.STATEMENT_DURATION############################
######################Java#####################

#######

```
SELECT *
    FROM TABLE (SYSCS_DIAG.STATEMENT_DURATION('somederby.log'))
    AS T1
```

###: ####Derby################################## ########Derby################*Derby*####
############SYSCS_DIAG.STATEMENT_DURATION################

SYSCS_DIAG.TRANSACTION_TABLE ###

SYSCS_DIAG.TRANSACTION_TABLE###################################SYSCS_DIAG.TRANS

#######

```
SELECT * FROM SYSCS_DIAG.TRANSACTION_TABLE
```

###SYSCS_DIAG.TRANSACTION_TABLE###

####

#####Derby##################

#########

####################SQL###

##(NULL)##################################
######NULL##

############CREATE TABLE###################

###

Derby#######

######:
#######################################
- ##
  - SMALLINT (2 bytes)
  - INTEGER (4 bytes)
  - BIGINT (8 bytes)
- ############
  - REAL (4 bytes)
  - DOUBLE PRECISION (8 bytes)
  - FLOAT (DOUBLE PRECISION####REAL###)
- ####
  - DECIMAL (########)
  - NUMERIC (DECIMAL###)

#############:

#############Derby###########INTEGER###############################Derby############
 # 10. ###########

###########	#####
DOUBLE PRECISION	DOUBLE PRECISION
REAL	DOUBLE PRECISION
DECIMAL	DECIMAL
BIGINT	BIGINT
INTEGER	INTEGER
SMALLINT	INTEGER

#:

```
-- ###############
VALUES 1 + 1.0e0
-- ##########
VALUES 1 + 1.0
-- ########
VALUES CAST (1 AS INT) + CAST (1 AS INT)
```

################:

###
#########

```
create table mytable (r REAL, d DOUBLE PRECISION);
0 rows inserted/updated/deleted
INSERT INTO mytable (r, d) values (3.4028236E38, 3.4028235E38);
ERROR X0X41: The number '3.4028236E38' is outside the range for
the data type REAL.
```

##########INTEGER###

```
INSERT INTO mytable(integer_column) values (1.09e0);
1 row inserted/updated/deleted
SELECT integer_column
FROM mytable;
I
--------------
1
```

##

##

```
ij> insert into mytable (decimal_column)
VALUES (55555555556666666666);
ERROR X0Y21: The number '55555555556666666666' is outside the
range of the target DECIMAL/NUMERIC(5,2) datatype.
```

##

```
INSERT INTO mytable (int_column) values 2147483648;
ERROR 22003: The resulting value is outside the range for the
data type INTEGER.
```

Note: NUMERIC#################Derby##########
##############:

SQL#########(###################)####(#########)#######################
###################################

####################:

- *lp##################*
- *rp##################*
- *ls###################*
- *rs###################*

#####################################

- ###

  ls + rs
- ###

  31 - lp + ls - rs
- *AVG()*

  max(max(ls, rs), 4)
- #####

  max(ls, rs)

####################27###

```
11.0/1111.33
// 31 - 3 + 1 - 2 = 27
```

###################################

- ###

  lp + rp
- ###

  2 * (p - s) + s
- ###

  lp - ls + rp + max(ls + rp - rs + 1, 4)
- #####

max(lp - ls, rp - rs) + 1 + max(ls, rs)

####################

11. Derby#######

#####Derby#########################"Y"###############

Types	SMALLINT	INTEGER	BIGINT	DECIMAL	REAL	DOUBLE	FLOAT	CHAR	VARCHAR	LONG VARCHAR	CHAR FOR BIT DATA	VARCHAR FOR BIT DATA	LONG VARCHAR FOR BIT DATA	CLOB	BLOB	DATE	TIME	TIMESTAMP	XML
SMALL INT	Y	Y	Y	Y	Y	Y	Y	-	-	-	-	-	-	-	-	-	-	-	-
INTEGER	Y	Y	Y	Y	Y	Y	Y	-	-	-	-	-	-	-	-	-	-	-	-
BIGINT	Y	Y	Y	Y	Y	Y	Y	-	-	-	-	-	-	-	-	-	-	-	-
DECIMAL	Y	Y	Y	Y	Y	Y	Y	-	-	-	-	-	-	-	-	-	-	-	-
REAL	Y	Y	Y	Y	Y	Y	Y	-	-	-	-	-	-	-	-	-	-	-	-
DOUBLE	Y	Y	Y	Y	Y	Y	Y	-	-	-	-	-	-	-	-	-	-	-	-
FLOAT	Y	Y	Y	Y	Y	Y	Y	-	-	-	-	-	-	-	-	-	-	-	-
CHAR	-	-	-	-	-	-	-	Y	Y	Y	-	-	-	Y	-	Y	Y	Y	-
VARCHAR	-	-	-	-	-	-	-	Y	Y	Y	-	-	-	Y	-	Y	Y	Y	-
LONG VARCHAR	-	-	-	-	-	-	-	Y	Y	Y	-	-	-	Y	-	-	-	-	-
CHAR FOR BIT DATA	-	-	-	-	-	-	-	-	-	-	Y	Y	Y	-	-	-	-	-	-
VARCHAR FOR BIT DATA	-	-	-	-	-	-	-	-	-	-	Y	Y	Y	-	-	-	-	-	-

Types	SMALLINT	INTEGER	BIGINT	DECIMAL	REAL	DOUBLE	FLOAT	CHAR	VARCHAR	LONG VARCHAR	CHAR FOR BIT DATA	VARCHAR FOR BIT DATA	LONG VARCHAR FOR BIT DATA	CLOB	BLOB	DATE	TIME	TIMESTAMP	XML
LONG VARCHAR FOR BIT DATA	-	-	-	-	-	-	-	-	-	-	Y	Y	Y	-	-	-	-	-	-
CLOB	-	-	-	-	-	-	-	Y	Y	Y	-	-	-	Y	-	-	-	-	-
BLOB	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Y	-	-	-	-
DATE	-	-	-	-	-	-	-	Y	Y	-	-	-	-	-	-	Y	-	-	-
TIME	-	-	-	-	-	-	-	Y	Y	-	-	-	-	-	-	-	Y	-	-
TIME STAMP	-	-	-	-	-	-	-	Y	Y	-	-	-	-	-	-	-	-	Y	-
XML	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Y

 # 12. Derby#######

#####Derby################################ "Y"##################

Types	SMALLINT	INTEGER	BIGINT	DECIMAL	REAL	DOUBLE	FLOAT	CHAR	VARCHAR	LONG VARCHAR	CHAR FOR BIT DATA	VARCHAR FOR BIT DATA	LONG VARCHAR FOR BIT DATA	CLOB	BLOB	DATE	TIME	TIMESTAMP	XML
SMALL INT	Y	Y	Y	Y	Y	Y	Y	-	-	-	-	-	-	-	-	-	-	-	-
INTEGER	Y	Y	Y	Y	Y	Y	Y	-	-	-	-	-	-	-	-	-	-	-	-
BIGINT	Y	Y	Y	Y	Y	Y	Y	-	-	-	-	-	-	-	-	-	-	-	-
DECIMAL	Y	Y	Y	Y	Y	Y	Y	-	-	-	-	-	-	-	-	-	-	-	-
REAL	Y	Y	Y	Y	Y	Y	Y	-	-	-	-	-	-	-	-	-	-	-	-
DOUBLE	Y	Y	Y	Y	Y	Y	Y	-	-	-	-	-	-	-	-	-	-	-	-
FLOAT	Y	Y	Y	Y	Y	Y	Y	-	-	-	-	-	-	-	-	-	-	-	-
CHAR	-	-	-	-	-	-	-	Y	Y	-	-	-	-	-	-	Y	Y	Y	-
VARCHAR	-	-	-	-	-	-	-	Y	Y	-	-	-	-	-	-	Y	Y	Y	-
LONG VARCHAR	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CHAR FOR BIT DATA	-	-	-	-	-	-	-	-	-	-	Y	Y	-	-	-	-	-	-	-
VARCHAR FOR BIT DATA	-	-	-	-	-	-	-	-	-	-	Y	Y	-	-	-	-	-	-	-
LONG VARCHAR FOR BIT DATA	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CLOB	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
BLOB	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Types	SMALLINT	INTEGER	BIGINT	DECIMAL	REAL	DOUBLE	FLOAT	CHAR	VARCHAR	LONG VARCHAR	CHAR FOR BIT DATA	VARCHAR FOR BIT DATA	LONG VARCHAR FOR BIT DATA	CLOB	BLOB	DATE	TIME	TIMESTAMP	XML
DATE	-	-	-	-	-	-	-	Y	Y	-	-	-	-	-	-	Y	-	-	-
TIME	-	-	-	-	-	-	-	Y	Y	-	-	-	-	-	-	-	Y	-	-
TIME STAMP	-	-	-	-	-	-	-	Y	Y	-	-	-	-	-	-	-	-	Y	-
XML	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

BIGINT ####

BIGINT##########8#################

##

```
BIGINT
```

##########Java##

java.lang.Long

JDBC#######(java.sql.Types)

BIGINT

###

-9223372036854775808 (*java.lang.Long.MIN_VALUE*)

###

9223372036854775807 (*java.lang.Long.MAX_VALUE*)

##

##

```
9223372036854775807
```

BLOB####

BLOB(##############)####2,147,483,647################################BLOB###############

BLOB#####################1024#1024*1024#1024*1024*1024####K#M#G####################

Note: BLOB####Length#############

##

```
{ BLOB | BINARY LARGE OBJECT } [ ( length [{K |M |G }] ) ]
```

#####

###BLOB####2GB(2,147,483,647)###

##########Java##

java.sql.Blob

JDBC#######

BLOB

############BLOB##############*java.sql.ResultSet#getBlob*##########

######

java.sql.Blob#########java.sql.Clob###########

```
create table pictures(name varchar(32) not null primary key, pic
 blob(16M));

--###############
select length(pic), name from pictures where name like '%logo%';
--###########(blob###)
select a.name as double_one, b.name as double_two
from pictures as a, pictures as b
where a.name < b.name
and a.pic = b.pic
order by 1,2;
```

CHAR ####

CHAR##################

Syntax

```
CHAR[ACTER] [(##)]
```

######################1###

########Java##

java.lang.String

JDBC#######(java.sql.Types)

CHAR

Derby##############################Derby ##

CHAR#VARCHAR###

#########CHAR###

##############

CHAR#########*java.lang.Integer.MAX_VALUE*###

```
-- ###################
```

```
-- ######################
VALUES 'hello this is Joe''s string'
```

CHAR FOR BIT DATA ####

CHAR FOR BIT DATA###

##

```
{ CHAR | CHARACTER }[(##)] FOR BIT DATA
```

##############################

CHAR FOR BIT DATA########1############254######

JDBC #####(java.sql.Types)

BINARY

CHAR FOR BIT DATA##0x20################

CHAR FOR BIT DATA#VARCHAR FOR BIT DATA###
(######DBMS#########################SQL-92###############

VARCHAR FOR BIT DATA#CHAR FOR BIT DATA######(#######)#VARCHAR FOR BIT DATA######

```
CREATE TABLE t (b CHAR(2) FOR BIT DATA);
INSERT INTO t VALUES (X'DE');
SELECT *
FROM t;
-- ############
B
-----
de20
```

CLOB####

CLOB(###############)#######2,147,483,647###############CLOB#############################

CLOB################K#M#G#########################1024#1024*1024#1024*1024*1024########

CLOB####(######)############

##

```
{CLOB |CHARACTER LARGE OBJECT} [ ( ## [{K |M |G}] ) ]
```

#####

############CLOB####2##(2,147,483,647)#########

########Java##

java.sql.Clob

JDBC #######(java.sql.Types)

CLOB

CLOB###############*java.sql.ResultSet*###*getClob*##############

######

java.sql.Blob#########java.sql.Clob############################

```
import java.sql.*;
```

```
public class clob
{
 public static void main(String[] args) {
  try {
   String url = "jdbc:derby:clobberyclob;create=true";

   Class.forName("org.apache.derby.jdbc.EmbeddedDriver").newInstance();
   Connection conn = DriverManager.getConnection(url);

            Statement s = conn.createStatement();
            s.executeUpdate("CREATE TABLE documents (id INT, text CLOB(64
 K))");
            conn.commit();

            // --- ##########
            java.io.File file = new java.io.File("asciifile.txt");
            int fileLength = (int) file.length();

            // - ################
            java.io.InputStream fin = new java.io.FileInputStream(file);
            PreparedStatement ps = conn.prepareStatement("INSERT
            INTO documents VALUES (?, ?)");
            ps.setInt(1, 1477);

            // - ####################
            ps.setAsciiStream(2, fin, fileLength);
            ps.execute();
            conn.commit();

            // --- #####
            ResultSet rs = s.executeQuery("SELECT text FROM documents
             WHERE id = 1477");
            while (rs.next()) {
                java.sql.Clob aclob = rs.getClob(1);
                java.io.InputStream ip = rs.getAsciiStream(1);
                int c = ip.read();
                while (c > 0) {
                    System.out.print((char)c);
                    c = ip.read();
                }
                System.out.print("\n");
                // ...
            }
        } catch (Exception e) {
            System.out.println("Error! "+e);
        }
    }
}
```

DATE####

DATE####*java.sql.Date*###############

##

```
DATE
```

########Java##

java.sql.Date

JDBC######(java.sql.Types)

DATE

################################

java.sql.Date##########################SQL#######################Derby#DATE##########

```
yyyy-mm-dd
mm/dd/yyyy
```

```
dd.mm.yyyy
```

####################*java.sql.Date*######

####4##

####Derby#DB2###

#

```
VALUES DATE('1994-02-23')

VALUES '1993-09-01'
```

DECIMAL ####

DECIMAL#########################
#######(############)####(########)########
################################

##

```
{ DECIMAL | DEC } [(## [, ### ])]
```

###1##31#########################

####################0#########################5######

DECIMAL#################################
DECIMAL################Derby##########

####

```
-- ####################
values cast (1.798765 AS decimal(5,2));
1
--------
1.79
-- ###############
values cast (1798765 AS decimal(5,2));
1
--------
ERROR 22003: The resulting value is outside the range
for the data type DECIMAL/NUMERIC(5,2).
```

##

#################### ##########

###DECIMAL###

###########Java##

java.math.BigDecimal

JDBC ####### (java.sql.Types)

DECIMAL

```
VALUES 123.456

VALUES 0.001
```

BIGINT###############DECIMAL#########

DOUBLE ####

DOUBLE#DOUBLE PRECISION######

##

```
DOUBLE
```

DOUBLE PRECISION ####

DOUBLE PRECISION#IEEE###############8#############

##

```
DOUBLE PRECISION
```

####

```
DOUBLE
```

DOUBLE#DOUBLE PRECISION#################

##

DOUBLE##########

- DOUBLE####: -1.79769E+308
- DOUBLE####: 1.79769E+308
- DOUBLE#######: 2.225E-307
- DOUBLE#######: -2.225E-307

########Java#####`java.lang.Double`##########

############################## ######################**#####**
####################

##########30############

```
-- ######################
values 01234567890123456789012345678901e0;
```

##########Java##

java.lang.Double

JDBC ####### (java.sql.Types)

DOUBLE

##

#################**##########**

#

```
3421E+09
425.43E9
9E-10
4356267544.32333E+30
```

FLOAT####

FLOAT##########REAL####DOUBLE PRECISION#########

##

```
FLOAT [ (##) ]
```

######53#####DOUBLE PRECISION######
###23########FLOAT#REAL############24########FLOAT#DOUBLE PRECISION###############0##

JDBC ####### (java.sql.Types)

REAL or DOUBLE

##

##24##############FLOAT##DOUBLE############

##23##############FLOAT##REAL############

INTEGER ####

INTEGER####4##############

##

```
{ INTEGER | INT }
```

##########Java##

java.lang.Integer

JDBC####### (java.sql.Types)

INTEGER

###

-2147483648 (*java.lang.Integer.MIN_VALUE*)

###

2147483647 (*java.lang.Integer.MAX_VALUE*)

##

##########################

```
3453
425
```

LONG VARCHAR#

LONG
VARCHAR#######32,700###VARCHAR

##

```
LONG VARCHAR
```

##########Java##

java.lang.String

JDBC###### (java.sql.Types)

LONGVARCHAR

Java###SQL##########LONG VARCHAR#####Java##########

LONG VARCHAR FOR BIT DATA####

LONG VARCHAR FOR BIT
DATA######32,700###VARCHAR
FOR BIT DATA######

##

```
LONG VARCHAR FOR BIT DATA
```

JDBC ######(java.sql.Types)

LONGVARBINARY

NUMERIC####

NUMERIC#DECIMAL############## DECIMAL ############

##

```
NUMERIC [(## [, ### ])]
```

##########Java##

java.math.BigDecimal

####JDBC#######(java.sql.Types)

NUMERIC

```
123.456
.001
```

REAL ####

REAL###IEEE###############4###############

##

```
REAL
```

##########Java##

java.lang.Float

JDBC#######(java.sql.Types)

REAL

##

REAL#####:

- REAL####: -3.402E+38
- REAL####: 3.402E+38
- REAL#######: 1.175E-37
- REAL#######: -1.175E-37

#######Java##`java.lang.Float`############

################################ ##################**#####**
###

###############30###############

```
-- ########################
values 01234567890123456789012345678901e0;
```

###

###################**##########**

#####DOUBLE PRECISION######REAL#######CAST#########

SMALLINT ####

SMALLINT##2################

Syntax

```
SMALLINT
```

##########Java##

java.lang.Short

JDBC####### (java.sql.Types)

SMALLINT

###

-32768 (*java.lang.Short.MIN_VALUE*)

###

32767 (*java.lang.Short.MAX_VALUE*)

##

#################### ##########

#############INTEGER####BIGINT######

TIME####

TIME####################

##

```
TIME
```

###########Java##

java.sql.Time

JDBC ###### (java.sql.Types)

TIME

###############CAST####################

SQL###/
##########*java.sql.Time*######################Derby#####################

```
hh:mm[:ss]
hh.mm[.ss]
hh[:mm] {AM | PM}
```

###############*java.sql.Time*######

###

Derby###

#

```
VALUES TIME('15:09:02')
VALUES '15:09:02'
```

TIMESTAMP####

TIMESTAMP#################################9############

##

```
TIMESTAMP
```

###########Java##

java.sql.Timestamp

JDBC ###### (java.sql.Types)

TIMESTAMP

################################

Derby##TIMESTAMP###########

```
yyyy-mm-dd hh:mm:ss[.nnnnnn]
yyyy-mm-dd-hh.mm.ss[.nnnnnn]
```

##################*java.sql.Timestamp*######

####4#########################1######2####################2#########################

Derby
##

#

```
VALUES '1960-01-01 23:03:20'
VALUES TIMESTAMP('1962-09-23 03:23:34.234')
VALUES TIMESTAMP('1960-01-01 23:03:20')
```

VARCHAR####

VARCHAR#########################

##

```
{ VARCHAR | CHAR VARYING | CHARACTER VARYING }(##)
```

################ ##############*java.lang.Integer.MAX_VALUE*#################

VARCHAR#########32,672#####

###########Java##

java.lang.String

JDBC####### (java.sql.Types)

VARCHAR

Derby #############VARCHAR############
Derby#VARCHAR###
##################################
VARCHAR###

######CHAR#VARCHAR###

########VARCHAR####CHAR###

VARCHAR FOR BIT DATA ####

VARCHAR FOR BIT
DATA###

##

```
{ VARCHAR | CHAR VARYING | CHARACTER VARYING } (##) FOR BIT DATA
```

################################

CHAR FOR BIT DATA######VARCHAR FOR BIT
DATA######################32,672#############

JDBC###### (java.sql.Types)

VARBINARY

VARCHAR FOR BIT DATA##################CHAR FOR BIT
DATA######VARCHAR FOR BIT DATA######################

#######VARCHAR FOR BIT DATA#CHAR FOR BIT
DATA################VARCHAR FOR BIT DATA######

############VARCHAR FOR BIT DATA###CHAR FOR BIT DATA########

XML data type

The XML data type is used for Extensible Markup Language (XML) documents.

XML#################

- SQL/XML############XML(DOCUMENT(ANY))########
- #########XML(DOCUMENT(ANY))#######XML(SEQUENCE)######

Note:
XML##JAXP######Xalan##########################

Derby##JDBC###SQL/
XML##################XML#############JDBC##########XML################
#######XMLPARSE#XMLSERIALIZE#####XML#########SQL#####XML####Java################

##

```
XML
```

###########Java##

##

XML#######Java###java.sql.SQLXML###########java.sql.SQLXML##Derby##############

JDBC ###### (java.sql.Types)

##

XML##############SQLXML###########SQLXML#Derby##############

Derby#########XML################SQL##XMLSERIALIZE############
#######

```
SELECT XMLSERIALIZE (xcol as CLOB) FROM myXmlTable
```

####################getXXX#########XML####################CLOB#######

Derby########JDBC####XML###########SQL###XMLPARSE############
#######

```
INSERT INTO myXmlTable(xcol) VALUES XMLPARSE(
    DOCUMENT CAST (? AS CLOB) PRESERVE WHITESPACE)
```

#################setXXX#############
#####PreparedStatement.setString###PreparedStatement.setCharacterStream####################

SQL###

#####Derby##################SQL-92###############
###############(')####################Derby##########
###SQL92######
ADD
ALL
ALLOCATE
ALTER
AND
ANY
ARE
AS
ASC
ASSERTION

AT
AUTHORIZATION
AVG
BEGIN
BETWEEN
BIGINT
BIT
BOOLEAN
BOTH
BY
CALL
CASCADE
CASCADED
CASE
CAST
CHAR
CHARACTER
CHECK
CLOSE
COALESCE
COLLATE
COLLATION
COLUMN
COMMIT
CONNECT
CONNECTION
CONSTRAINT
CONSTRAINTS
CONTINUE
CONVERT
CORRESPONDING
CREATE
CURRENT
CURRENT_DATE
CURRENT_TIME
CURRENT_TIMESTAMP
CURRENT_USER
CURSOR
DEALLOCATE
DEC
DECIMAL
DECLARE
DEFAULT
DEFERRABLE
DEFERRED
DELETE
DESC
DESCRIBE
DIAGNOSTICS
DISCONNECT
DISTINCT
DOUBLE
DROP
ELSE
END
END-EXEC

ESCAPE
EXCEPT
EXCEPTION
EXEC
EXECUTE
EXISTS
EXPLAIN
EXTERNAL
FALSE
FETCH
FIRST
FLOAT
FOR
FOREIGN
FOUND
FROM
FULL
FUNCTION
GET
GETCURRENTCONNECTION
GLOBAL
GO
GOTO
GRANT
GROUP
HAVING
HOUR
IDENTITY
IMMEDIATE
IN
INDICATOR
INITIALLY
INNER
INOUT
INPUT
INSENSITIVE
INSERT
INT
INTEGER
INTERSECT
INTO
IS
ISOLATION
JOIN
KEY
LAST
LEFT
LIKE
LOWER
LTRIM
MATCH
MAX
MIN
MINUTE
NATIONAL
NATURAL

NCHAR
NVARCHAR
NEXT
NO
NOT
NULL
NULLIF
NUMERIC
OF
ON
ONLY
OPEN
OPTION
OR
ORDER
OUTER
OUTPUT
OVERLAPS
PAD
PARTIAL
PREPARE
PRESERVE
PRIMARY
PRIOR
PRIVILEGES
PROCEDURE
PUBLIC
READ
REAL
REFERENCES
RELATIVE
RESTRICT
REVOKE
RIGHT
ROLLBACK
ROWS
RTRIM
SCHEMA
SCROLL
SECOND
SELECT
SESSION_USER
SET
SMALLINT
SOME
SPACE
SQL
SQLCODE
SQLERROR
SQLSTATE
SUBSTR
SUBSTRING
SUM
SYSTEM_USER
TABLE
TEMPORARY

TIMEZONE_HOUR
TIMEZONE_MINUTE
TO
TRANSACTION
TRANSLATE
TRANSLATION
TRIM
TRUE
UNION
UNIQUE
UNKNOWN
UPDATE
UPPER
USER
USING
VALUES
VARCHAR
VARYING
VIEW
WHENEVER
WHERE
WITH
WORK
WRITE
XML
XMLEXISTS
XMLPARSE
XMLQUERY
XMLSERIALIZE
YEAR

Derby#######SQL-92###

SQL-92#######4###########

- SQL92E

  ###
- SQL92T

  ### FIPS 127-2########NIST###
- SQL92I

  ##
- SQL92F

  ##

######
#####Derby#######SQL-92##############
 # 13. #####SQL-92###: ######

##	######	Derby
SMALLINT	SQL92E	Yes
INTEGER	SQL92E	Yes
DECIMAL(p,s)	SQL92E	Yes

##	######	Derby
NUMERIC(p,s)	SQL92E	Yes
REAL	SQL92E	Yes
FLOAT(p)	SQL92E	Yes
DOUBLE PRECISION	SQL92E	Yes
CHAR(n)	SQL92E	Yes

######
 # 14. SQL-92########:########

##	######	Derby
+, *, -, /, unary +, unary -	SQL92E	Yes

######
 # 15. SQL-92########:######

##	######	Derby
<, >, <= ,>=, <>, =	SQL92E	Yes

######
 # 16. SQL-92########:######

##	######	Derby
BETWEEN, LIKE, NULL	SQL92E	Yes

######
 # 17. SQL-92########:######

##	######	Derby
IN, ALL/SOME, EXISTS	SQL92E	Yes

#######
 # 18. SQL-92########:#######

##	######	Derby
#	SQL92E	Yes
###	SQL92E	Yes
##	SQL92E	Yes

####
 # 19. SQL-92########:####

##	######	Derby
###	SQL92E	Yes
##########	SQL92E	Yes

##(######)

 # 20. SQL-92########: ##(######)

##	######	Derby
NOT NULL	SQL92E	Yes (SYSCONSTRAINTS########)
UNIQUE/PRIMARY KEY	SQL92E	Yes
FOREIGN KEY	SQL92E	Yes
CHECK	SQL92E	Yes
View WITH CHECK OPTION	SQL92E	No ##########

####

 # 21. SQL-92########:####

##	######	Derby
DECLARE, OPEN, FETCH, CLOSE	SQL92E	Yes JDBC############
UPDATE, DELETE CURRENT	SQL92E	Yes

###SQL1

 # 22. SQL-92########: ###SQL1

##	######	Derby
ALLOCATE / DEALLOCATE / GET / SET DESCRIPTOR	SQL92T	Yes JDBC##############
PREPARE / EXECUTE / EXECUTE IMMEDIATE	SQL92T	Yes JDBC##############
DECLARE, OPEN, FETCH, CLOSE, UPDATE, DELETE dynamic cursor	SQL92T	Yes JDBC##############
DESCRIBE output	SQL92T	JDBC#####

##########

 # 23. SQL-92######## ##########

##	######	Derby
TABLES	SQL92T	*SYS.SYSTABLES, SYS.SYSVIEWS, SYS.SYSCOLUMNS*
VIEWS	SQL92T	*SYS.SYSTABLES, SYS.SYSVIEWS, SYS.SYSCOLUMNS*

##	######	Derby
COLUMNS	SQL92T	*SYS.SYSTABLES, SYS.SYSVIEWS, SYS.SYSCOLUMNS*

##########

 # 24. SQL-92########: ###########

##	######	Derby
CREATE / DROP TABLE	SQL92T	Yes
CREATE / DROP VIEW	SQL92T	Yes
GRANT / REVOKE	SQL92T	Yes
ALTER TABLE ADD COLUMN	SQL92T	Yes
ALTER TABLE DROP COLUMN	SQL92T	Yes

####

 # 25. SQL-92########: ####

##	######	Derby
INNER JOIN	SQL92T	Yes
natural join	SQL92T	No
LEFT, RIGHT OUTER JOIN	SQL92T	Yes
join condition	SQL92T	Yes
named columns join	SQL92T	Yes

#######

 # 26. SQL-92########: #######

##	######	Derby
###DATE, TIME, TIMESTAMP, INTERVAL	SQL92T	Yes ###INTERVAL####
#####	SQL92T	Yes
#####	SQL92T	Yes Java#########
#####	SQL92T	Yes
##: OVERLAPS	SQL92T	Yes Java#########

VARCHAR ####

 # 27. SQL-92########: VARCHAR

##	######	Derby
LENGTH	SQL92T	Yes

##	######	Derby
## (||)	SQL92T	Yes

##########
 # 28. SQL-92########: ##########

##	######	Derby
READ WRITE / READ ONLY	SQL92T	JDBC####################
RU, RC, RR, SER	SQL92T	Yes

###############
 # 29. SQL-92########: ###############

##	######	Derby
SCHEMATA ###	SQL92T	*SYS.SYSSCHEMAS*

###
 # 30. SQL-92########: ###

##	######	Derby
TABLE_PRIVILEGES	SQL92T	No
COLUMNS_PRIVILEGES	SQL92T	No
USAGE_PRIVILEGES	SQL92T	No

#####
 # 31. SQL-92########: #####

##	######	Derby
UNION relaxation	SQL92I	Yes
EXCEPT	SQL92I	Yes
INTERSECT	SQL92I	Yes
CORRESPONDING	SQL92I	No

########
 # 32. SQL-92########: ########

##	######	Derby
CREATE SCHEMA	SQL92I	######

#####
 # 33. SQL-92########: #####

##	######	Derby
SET SESSION AUTHORIZATION	SQL92I	SET SCHEMA###

##	######	Derby
CURRENT_USER	SQL92I	Yes
SESSION_USER	SQL92I	Yes
SYSTEM_USER	SQL92I	No

####
 # 34. SQL-92########:####

##	######	Derby
TABLE CONSTRAINTS	SQL92I	*SYS.SYSCONSTRAINTS*
REFERENTIAL CONSTRAINTS	SQL92I	*SYS.SYSFOREIGNKEYS*
CHECK CONSTRAINTS	SQL92I	*SYS.SYSCHECKS*

#########
 # 35. SQL-92########: #########

##	######	Derby
SQL_FEATURES	SQL92I/FIPS 127-2	JDBC#*DatabaseMetaData*###
SQL_SIZING	SQL92I/FIPS 127-2	JDBC#*DatabaseMetaData*###

######
 # 36. SQL-92########: ######

##	######	Derby
TIME#TIMESTAMP###	SQL92F	Yes

########
 # 37. SQL-92########: ########

##	######	Derby
POSITION#	SQL92F	Java#########LOCATE###
UPPER/LOWER ##	SQL92F	Yes

##
 # 38. SQL-92########: ##

##	######	Derby
##########	SQL92E	Yes
###########	SQL92E	Yes
#########	SQL92E	Yes
##	SQL92E	Yes
Where#####	SQL92E	Yes
Group by	SQL92E	Yes

##	######	Derby
Having	SQL92E	Yes
####	SQL92E	Yes
Order by	SQL92E	Yes
###	SQL92E	Yes
Select *	SQL92E	Yes
SQLCODE	SQL92E	No SQL-92#########
SQLSTATE	SQL92E	Yes
#####UNION#INTERSECT#	SQL92T	Yes
########	SQL92T	Yes
#########	SQL92T	Yes
#######	SQL92T	JDBC#*SQLExceptions*###
########	SQL92T	Yes
select#################*	SQL92T	Yes
#######	SQL92T	Yes
###############	SQL92T	No
############	SQL92T	No (JDBC################)
########	SQL92T	CASCADE#SET NULL#RESTRICT#NO ACTION
CAST##	SQL92T	Yes
INSERT#	SQL92T	Yes
#######	SQL92T	Yes
########	SQL92T	Yes
Domain###	SQL92I	No
CASE#	SQL92I	########
###########	SQL92I	########
LIKE#####	SQL92I	Yes
UNIQUE###	SQL92I	No
######	SQL92I	*SYS.SYSDEPENDS*
##########	SQL92I	JDBC#*DatabaseMetaData*###Derby########
###########	SQL92I	########JDBC########
####SQL#######	SQL92I	No
#######	SQL92I	Yes
#####	SQL92I	Yes
########	SQL92I	No
#########	SQL92I	No

##	######	Derby
##########	SQL92I	### (JDBC2.0##########insensitive######
##########	SQL92I	######
########	SQL92I	Java##########
##########	SQL92I	Java##########
##########	SQL92I	Yes
#####null##	SQL92I	Yes
#####	SQL92I	Yes (ADD/DROP CONSTRAINT)
FOR BIT DATA #	SQL92F	Yes
Assertion ##	SQL92F	No
###	SQL92F	DECLARE GLOBAL TEMPORARY TABLE#########
#####SQL	SQL92F	No
###values#	SQL92F	Yes
#####	SQL92F	Yes
####FROM####	SQL92F	Yes
##################	SQL92F	Yes
Indicator####	SQL92F	JDBC########
#########	SQL92F	No
###SQL######	SQL92F	No
#########	SQL92F	Yes
#########	SQL92F	No
###	SQL92F	No
CHECK########	SQL92F	No ###Java################
Union join	SQL92F	No
Collation#translation	SQL92F	Java##########
########	SQL92F	RESTRICT###NO ACTION# ##############
ALTER domain	SQL92F	No
INSERT ###	SQL92F	No
####MATCH#	SQL92F	No
####CHECK########	SQL92F	No############
Session##	SQL92F	JDBC###
####	SQL92F	JDBC###
########	SQL92F	Yes
insensitive#####	SQL92F	Yes JDBC 2.0######

##	######	Derby
#######	SQL92F	######
##########	SQL92F	No
#########	SQL92F	No
#########	SQL92F	No

Derby######

Derby#############

################################

##########*SYS*###*SYS*##########

####################Java# *java.sql.DatabaseMetaData*#####################

SYSALIASES

######################

##	#	##	###	##
ALIASID	CHAR	36	##	alias#######
ALIAS	VARCHAR	128	##	alias
SCHEMAID	CHAR	36	#	##########
JAVACLASSN	LONGVARCHAR	255	##	Java#####
ALIASTYPE	CHAR	1	##	*'F'* (##)*'P'* (###)
NAMESPACE	CHAR	1	##	*'F'* (##)*'P'* (###)
SYSTEMALIAS	BOOLEAN	'	##	# (################alias) *false* (########alias)
ALIASINFO	org.apache.derby. catalog.AliasInfo: ########API########	'	#	alias########################
SPECIFICNAM	VARCHAR	128	##	############

SYSCHECKS#####

########################

##	#	##	###	##
CONSTRAINTID	CHAR	36	##	######
CHECKDEFINITION	LONG VARCHAR	'	##	###########
REFERENCEDCOLUM	org.apache.derby.ca ReferencedColumns ########API#####	'	##	###############

SYSCOLPERMS

SYSCOLPERMS ########################

##########GRANTEE, TABLEID, TYPE, GRANTOR#########SYSCOLPERMS##########SYSCOLPERMS##############

- ### (GRANTEE, TABLEID, TYPE, GRANTOR)
- #### (COLPERMSID)
- #### (TABLEID references SYS.SYSTABLES)

##	#	##	Null#	##
COLPERMSID	CHAR	36	##	##
GRANTEE	VARCH	30	##	###################
GRANTOR	VARCH	30	##	##
TABLEID	CHAR	36	##	##################
TYPE	CHAR	1	##	################## SELECT##s##UPDATE##u##REFERENCES##r ##################### SELECT##S##UPDATE##U##REFERENCES##
COLUMNS	org.apa	'	##	############### ######API############

SYSCOLUMNS#####

######################

##	#	##	###	##
REFERENCEID	CHAR	36	##	#####(*SYSTABLES.TABLEID*######)
COLUMNNAME	CHAR	128	##	########
COLUMNNUMBER	INT	4	##	######
COLUMNDATATYPE	org.apache.derby.c TypeDescriptor ########API#####	'	##	##################################
COLUMNDEFAULT	*java.io.Serializable*	'	#	################################ TABLE#ALTER TABLE##########
COLUMNDEFAULTID	CHAR	36	#	#######
AUTOINCREMENT COLUMNVALUE	BIGINT	'	#	##################
AUTOINCREMENT COLUMNSTART	BIGINT	'	#	###########(################
AUTOINCREMENT COLUMNINC	BIGINT	'	#	###########(###################

SYSCONGLOMERATES

##

##	#	##	###	##
SCHEMAID	CHAR	36	##	################
TABLEID	CHAR	36	##	#####(*SYSTABLES.TABLEID*#####
CONGLOMERATENUMBE	BIGINT	8	##	########(##########)##########
CONGLOMERATENAME	VARCHAR	128	#	##########################
ISINDEX	BOOLEAN	1	##	######
DESCRIPTOR	org.apache.derby. catalog.IndexDesc ##########API#	'	#	############
ISCONSTRAINT	BOOLEAN	1	#	######################
CONGLOMERATEID	CHAR	36	##	################

SYSCONSTRAINTS

##################(###################################)###############

##	#	##	###	##
CONSTRAINTID	CHAR	36	##	######
TABLEID	CHAR	36	##	#####(*SYSTABLES.TABLEID*###)
CONSTRAINTNAME	VARCHAR	128	##	### (####################)
TYPE	CHAR	1	##	*P* (###)#*U* (###)#*C* (####)# #### *F* (####)
SCHEMAID	CHAR	36	##	################(*SYSSCHEMAS.SCHEMAI*
STATE	CHAR	1	##	####*E*#####*D*
REFERENCECOUNT	INTEGER	1	##	##

SYSDEPENDS

SYSDEPENDS###################################

########################## ################### ######################

- ######################
- ##############################

##	#	##	###	##
DEPENDENTID	CHAR	36	##	##########
DEPENDENTFINDER	org.apache.derby.ca DependableFinder: #	1	##	########################
PROVIDERID	CHAR	36	##	############

##	#	##	###	##
PROVIDERFINDER	org.apache.derby.ca DependableFinder #	1	false	##############################

SYSFILES#####

############jar###########

##	#	##	###	##
FILEID	CHAR	36	##	jar##########
SCHEMAID	CHAR	36	##	jar##############(*SYSSCHEMAS. SCHEMAID######*)
FILENAME	VARCHAR	128	##	jar#####SQL#
GENERATIONID	BIGINT	'	##	################jar##############

SYSFOREIGNKEYS#####

#########################

Derby
#####################################SYSFOREIGNKEYS.CONGLOMERATEID######

##	#	##	###	##
CONSTRAINTID	CHAR	36	false	unique identifier for the foreign key constraint (join with *SYSCONSTRAINTS. CONSTRAINTID*)
CONGLOMERATEID	CHAR	36	*##*	##################(*SYSCONGLOMERATES CONGLOMERATEID######*)
KEYCONSTRAINTID	CHAR	36	*##*	##############################(*SYSKE CONSTRAINTID*)
DELETERULE	CHAR	1	*##*	NO ACTION (##)*##R#*RESTRICT*##S#*CASCADE NULL*##U*
UPDATERULE	CHAR	1	*##*	*NO ACTION(##)##R#restrict##S*

SYSKEYS#####

################################ Derby############
####################*SYSKEYS.CONGLOMERATEID######*

##	#	##	###	##
CONSTRAINTID	CHAR	36	##	#########
CONGLOMERATEID	CHAR	36	##	################

SYSROUTINEPERMS

SYSROUTINEPERMS####################

###########SYSROUTINEPERMS####EXECUTE###########SYSROUTINEPERMS############

- ### (GRANTEE, ALIASID, GRANTOR)
- #### (ROUTINEPERMSID)
- #### (ALIASID#SYS.SYSALIASES######)

######SYSTABLEPERMS##############

##	#	##	###	##
ROUTINEPERMSID	CHAR	36	##	##
GRANTEE	VARCH	30	##	##################
GRANTOR	VARCH	30	##	###################################
ALIASID	CHAR	36	##	############PERMTYPE#'E'##ALIASID#SYS.S
GRANTOPTION	CHAR	1	##	GRANTEE####################Y####N#

SYSSCHEMAS

############################

##	#	##	###	##
SCHEMAID	CHAR	36	##	##########
SCHEMANAME	VARCHAR	128	##	#####
AUTHORIZATIONID	VARCHAR	128	##	##############

SYSSTATISTICS#####

############################

##	#	##	###	##
STATID	CHAR	36	##	#########
REFERENCEID	CHAR	36	##	###############(SYSCONGLOMERATES. CONGLOMERATEID###)
TABLEID	CHAR	36	##	#########
CREATIONTIMESTA	TIMESTAM	'	##	#####/######
TYPE	CHAR	1	##	#####
VALID	BOOLEAN	'	##	########
COLCOUNT	INTEGER	'	##	#####
STATISTICS	org.apache. derby.catalo Statistics: ########A	'	#	####

SYSSTATEMENTS#####

######################

##	##	##	###	##
STMTID	CHAR	36	##	########
STMTNAME	VARCHAR	128	##	####
SCHEMAID	CHAR	36	##	#########
TYPE	CHAR	1	##	##'S'
VALID	BOOLEAN	'	##	####TRUE#####FALSE
TEXT	LONG VARCHAR	'	##	######
LASTCOMPILED	TIMESTAMP	'	#	##########
COMPILATION SCHEMAID	CHAR	36	##	###########
USINGTEXT	LONG VARCHAR	'	#	CREATE STATEMENT#ALTER STATEMENT##USING######

SYSTABLEPERMS

SYSTABLEPERMS############################

##########GRANTEE, TABLEID, GRANTOR##########SYSTABLEPERMS##########SYSTABLEPERMS##############

- ### (GRANTEE, TABLEID, GRANTOR)
- #### (TABLEPERMSID)
- #### (TABLEID references SYS.SYSTABLES)

##SYSTABLEPERMS###############

##	#	##	Null#	##
TABLEPERMSID	CHAR	36	##	##
GRANTEE	VARCH	30	##	###################
GRANTOR	VARCH	30	##	##
TABLEID	CHAR	36	##	######################
SELECTPRIV	CHAR	1	##	SELECT#####################'y'(##### 'Y'(##########) 'N'(#######)###
DELETEPRIV	CHAR	1	##	DELETE################## ######'y'(################) 'Y'(##########) 'N'(#######)###
INSERTPRIV	CHAR	1	##	INSERT################## ######'y'(################) 'Y'(##########) 'N'(#######)###

##	#	##	Null#	##
UPDATEPRIV	CHAR	1	##	UPDATE################## ######'y'(################) 'Y'(##########) 'N'(#######)###
REFERENCEPRIV	CHAR	1	##	REFERENCE################## ######'y'(################) 'Y'(##########) 'N'(#######)###
TRIGGERPRIV	CHAR	1	##	TRIGGER##########################'y'(#### 'Y'(##########) 'N'(#######)###

SYSTABLES

##########################

##	#	##	###	##
TABLEID	CHAR	36	##	############
TABLENAME	VARCHAR	128	##	########
TABLETYPE	CHAR	1	##	*'S'* (#####)#*'T'* (####)#### *'V'* (###)
SCHEMAID	CHAR	36	##	##############
LOCK GRANUL	CHAR	1	##	############## ***'T'*** (########) ***'R'*** (######## ##)

SYSTRIGGERS#####

##############

##	#	##	###	##
TRIGGERID	CHAR	36	##	##########
TRIGGERNAME	VARCHAR	128	##	######
SCHEMAID	CHAR	36	##	############(*SYSSCHEMAS.SCHEMAID*###)
CREATIONTIMESTAMI	TIMESTAMP	'	##	##########
EVENT	CHAR	1	##	'U'#### 'D'####'I'#######
FIRINGTIME	CHAR	1	##	'B'### 'A'######
TYPE	CHAR	1	##	'R'###'S'######
STATE	CHAR	1	##	'E'####'D'#######

##	#	##	###	##
TABLEID	CHAR	36	##	###############
WHENSTMTID	CHAR	36	#	WHEN##########(###############)
ACTIONSTMTID	CHAR	36	#	######SQL################## (*SYSSTATEMENTS.STMTID*###)
REFERENCEDCOLUM	org.apache.derby. ReferencedColum ########API####	'	#	UPDATE#############
TRIGGERDEFINITION	LONG VARCHAR	'	#	####SQL######
REFERENCINGOLD	BOOLEAN	'	true	whether or not the OLDREFERENCINGNA if non-null, refers to the OLD row or table
REFERENCINGNEW	BOOLEAN	'	#	#########NEWREFERENCINGNAME###
OLDREFERENCINGNA	VARCHAR	128	#	REFERENCING OLD AS############
NEWREFERENCINGNA	VARCHAR	128	#	REFERENCING NEW AS############

######SQL#######SQL##########*SYSSTATEMENTS########ACTIONSTMTID#WHENSTMTID##SY*

SYSVIEWS

##########################

##	#	##	###	##
TABLEID	CHAR	36	##	########## (TABLEID##########SYS
VIEWDEFINITION	LONG VARCHAR	'	##	##########
CHECKOPTION	CHAR	1	##	'*N*' (########################)
COMPILATION SCHEMAID	CHAR	36	##	###############

Derby#########SQL state

Derby###############JDBC##*SQLException######### ##########################SQLException#*
Derby###*SQLException*##*SQLState*#########*X*#####################*SQLState*################

JDBC####################0A####*SQLState*#*SQLException*######
################JDK
1.6##########################*java.sql.SQLFeatureNotSupportedException*######
####################Derby#################

Derby#####*SQLState*###################Derby#*nextException*#########*SQLExceptions*#########
#########SQL-92####################Derby############

SQLExceptions#########*Derby* ######## #############

SQL##########

##########*SQLStates*#######X#######Derby#########

 # 39. ## 01:##

SQLSTATE	##
01001	##
01003	######## NULL ##############
01006	####### <#####>###########
0100E	XX ######################
01500	#<##>###<###>###########
01501	### <####gt; ###########
01502	#<##>##### <####>###########
01503	# <##> ## <##> ######################
01504	######### <###> #########
01505	# <##> ###############
01522	############ '<#####>' ############### '<#####>' #########
01J01	###### '<#######>' ##################################
01J02	###################################
01J04	# '<##>' ####### '<####>' # java.io.Serializable ### java.sql.SQLData ############### ################################ 1 ######################
01J05	###################### ########################## ############################### ########################
01J06	ResultSet ######### ######### ResultSet ##################
01J07	################## ResultSetHoldability # ResultSet.CLOSE_CURSORS_AT_COMMIT ##########
01J08	ResultSet # <######> ############ ResultSet # <######> ########
01J10	##
01J12	########################## ############## ############## SYSIBM.SQLCAMESSAGE ###################### ###################
01J13	###### (<#>) ###################################
01J14	############# SQL ############

 # 40. ## 07: ##SQL####

SQLSTATE	##
07000	################ 1 ###################
07004	###### <###> # <###> ###################### CallableStatement.registerOutParameter ##############
07009	################

41. ## 08:

SQLSTATE	##
08000	##########################
08000	################################## ##################################
08001	##
08001	#### ID ##
08001	##
08001	### Derby DataSource ###### <###> ############
08001	<###>: ### <#####> ##### <####> ############ <##> ############
08001	SocketException: '<###>'
08001	#### '<###>' ##############
08001	#### ID ### (<#>) # 1 ## <#> #######
08001	######## (<#>) # 1 ## <#> #######
08001	#### ID # NULL ############
08001	###### NULL ############
08001	####### '<#######>' ##
08003	#######
08003	####### PooledConnection #### getConnection() ######
08003	######### LOB ##############
08003	##############################
08004	##########: <###>
08004	################ ##: <#####>#
08004	###### <#######> #######################
08004	##################
08004	User '<*##ID*>' cannot shut down database '<#######>'. Only the database owner can perform this operation.
08004	User '<*##ID*>' cannot (re)encrypt database '<#######>'. Only the database owner can perform this operation.
08004	User '<*##ID*>' cannot hard upgrade database '<#######>'. Only the database owner can perform this operation.
08006	############################## #########################
08006	###### '<#######>' ###############

42. ## 0A:

SQLSTATE	##
0A000	######################: <###>#
0A000	DRDA #### <#####> ################## ##########
0A000	JDBC ######################
0A000	JDBC #### <#####> ################## ##################
0A000	resultSetHoldability ###### <###> #############
0A000	cancel() ##################
0A000	############# '<######>' #############
0A000	#### '<#####>' #############

43. ## 21:

SQLSTATE	##
21000	###################

44. ## 22:

SQLSTATE	##
22001	<#> '<#>' ### <#> ##########################
22003	######### <#####> #######
22003	# (<#>) #### '<#>' ########
22003	##### 10 ####31 ######
22003	'<#####>' ## <#####> ############################
22004	## (<#>) ############ (<#####>) ########
22005	# '<##>' #### '<##>' ###################################
22005	######################
22005	Unicode ###### EBCDIC ##################
22005	###### JDBC #### #: <##>#columnCount: <#>#columnIndex: <#>#
22005	###### <######> ####### JDBC ####
22005	###### Java SQL # <#####> ###
22005	# '<#####>' ######## '<#####>' ################
22007	##################
22007	########################
22008	'<##>' # <###> ##############
2200L	XML ############ DOCUMENT ##############
2200M	### XML DOCUMENT: <#########>
2200V	<####> ################################## DOCUMENT ##############
2200W	XQuery ##############: 1 ##########################

SQLSTATE	##
22011	SUBSTR ### 2 ##### 3 ############
22012	############
22013	#### '<#>' ###############
22014	LOCATE ################################ ########## '<#####>' ### ########## '<#######>' ### ########## '<####>' ###
22015	########################<#>##########
22015	'<###>' ###################### # 1 ####### '<##>' ### # 2 ####### '<##>' ### # 3 ##### (####) ## '<##>' ###
22018	# <##> #########################
22019	#################: <######># ##############1 ############## NULL ## 2 ############
22020	Invalid trim string, '<###>'. The trim string must be exactly one character or NULL. It cannot be more than one character.
22025	################## "_" ### "%" ###### ##
22027	#### TRIM() ###################### LTRIM() ### RTRIM() #########################
22028	############ <#> ########
22501	NULL # ESCAPE #########################

 # 45. ## 23: ####

SQLSTATE	##
23502	# '<##> # NULL ############
23503	# '<##>' ## <#>' ####### <###> ####### '<###>' ########## ######################
23505	############################ '<#>' ####### '<#>' ###
23513	# '<##>' # INSERT ### UPDATE ################ '<###>' ###########

 # 46. ## 24: ##########

SQLSTATE	##
24000	######### - ##########
24501	########################

 # 47. Class 25: ##############

SQLSTATE	##
25000	################
25001	##
25501	##

SQLSTATE	##
25502	############################SQL ################
25503	############################ DDL ##########
25505	##

 # 48. ## 28: #######

SQLSTATE	##
28502	##### "<#####>" ######

 # 49. Class 2D: ##############

SQLSTATE	##
2D521	setAutoCommit(true) #########################
2D521	############## COMMIT ### ROLLBACK ######

 # 50. ## 38: #######

SQLSTATE	##
38000	############ '<##>' ##########
38001	####### SQL ################
38002	############################## MODIFIES SQL DATA ################
38004	############################## READS SQL DATA ################

 # 51. ## 39: ##############

SQLSTATE	##
39004	######## '<#>' ################ NULL ##############

 # 52. ## 3B: ###SAVEPOINT

SQLSTATE	##
3B001	SAVEPOINT#<*SAVEPOINT*#> ##################################
3B002	##################
3B501	######## SAVEPOINT #####################
3B502	RELEASE ### ROLLBACK TO SAVEPOINT #########################

 # 53. ## 40: ###############

SQLSTATE	##
40001	###################################:\n<#######># ## XID : <###########>#
40XC0	########### ##

SQLSTATE	##
40XD0	################
40XD1	##########################
40XD2	##### <#####> ##########################
40XL1	#####################
40XL2	##################### lockTable ###: <#####>
40XT0	RawStore ###################
40XT1	###########################
40XT2	SAVEPOINT ###################
40XT4	############################## ###################
40XT5	########################
40XT6	################################ ######################
40XT7	################################

 # 54. ## 42: ################

SQLSTATE	##
42000	###################################
42500	#### '<*##ID*>' ### '<#####>'.'<##>' #### <####> ##############
42501	#### '<*##ID*>' ### '<#####>'.'<##>'#### GRANT ######## <####> ##############
42502	#### '<*##ID*>' ### '<#####>'.'<##>' ## '<##>' #### <####> ##############
42503	#### '<*##ID*>' ### '<#####>'.'<##>' ## '<##>' #### GRANT ######## <####> ##############
42504	#### '<*##ID*>' ##<#######> '<#####>'.'<##>' ###################
42505	#### '<*##ID*>' ##<#######> '<#####>'.'<##>' #### GRANT ########################
42506	#### '<*##ID*>' # <#######> '<#####>'.'<##>' ############
42507	#### '<*##ID*>' ###### '<#####>' #################
42508	#### '<*##ID*>' ###### '<#####>' ######### ##############################
42509	################################### '<#######>' ############
4250A	#### '<*##ID*>' ######## '<#####>'.'<#######>' #### <###> ##############
4250B	################# '<#>=<#>'#
4250C	#### '<*##ID*>' ###
4250D	######## '<####>' ########## '<*##ID*>'#
4250E	#####: ################# <*##ID*> ID #######
42X24	Column <*columnName*> is referenced in the HAVING clause but is not in the GROUP BY list.

SQLSTATE	##
42X25	'<###>' ####'*<1>*' ##########
42X26	# '<##>' ####### '<####>' ################## ########## public ##################
42X28	### '<##>' ##### '<#####>' ##############
42X29	### '<##>' ##### '<#####>' ##############
42X30	#### '<#####>' ######### ############################
42X31	# '<##>' ##### '<#####>' # FOR UPDATE ##########
42X32	############ '<##>' ##################
42X33	############## '<##>' ######
42X34	###### ? ############ ##########
42X35	'<#>' ########## ? ##################
42X36	'<###>' #####? ##########################
42X37	## '<###>' #####'<#>' ##########
42X38	'SELECT *' ##EXISTS ### NOT EXISTS ############
42X39	######################
42X40	NOT ########################## NOT ########TRUE#FALSE#### UNKNOWN ###############
42X41	FROM ################### '<###### #>' ##### (######### '<#>' ##########)#
42X42	# '<##>' # FOR UPDATE ###############################
42X43	###/###### '<####>' ######## ResultSetMetaData # NULL #### ######################ResultSetMetaData # NULL ############
42X44	######### '<#>' ##########
42X45	<#> # <#> ##### <#> ##########
42X46	'<###>' ################# ########################
42X47	'<###>' ###################### ########################
42X48	# '<#>' # <#> ##############
42X49	# '<#>' ##################
42X50	######### <#####>.<#>(<#>) ## public ### (###) ##
42X51	### '<####>' ################# ########## public ##################
42X52	Java ####### '<#>' ############ ('<#####>') ##################
42X53	LIKE ###"CHAR" ### "VARCHAR" ################# # '<#>' #########
42X54	Java #### '<#####>' ####### ? ######## ##########
42X55	## '<##>' # '<#>' ##############

SQLSTATE	##
42X56	##############"{0}" ##############################
42X57	##### '<##>' # getColumnCount() ##### '<#>' ####### #### 1 #####
42X58	<##>########################
42X59	# VALUES ##########################
42X60	# '<##>' # insertMode ########## '<#>' #########
42X61	# '<#>' # '<#>' # <#> ##########
42X62	'<#>' # '<#####>' ##############
42X63	USING ############### ###############
42X64	############useStatistics ########## '<#>' ######### ##### TRUE ### FALSE #####
42X65	## '<##>' ########
42X66	## '<##>' ##CREATE INDEX ###################
42X68	### '<####>' ######### '<######>' ######### ############# public ######################## public ############
42X69	Java ####### '<#>' ############# ('<######>') ##################
42X70	The number of columns in the table column list does not match the number of columns in the underlying query expression in the table definition for '<#*gt;*'.
42X71	Invalid data type '<#####>' for column '<##>'.
42X72	### "{1}" ########## '<######>' ######### ############# public ### (###) ######################## public ############
42X73	###### <#>.<#>(<#>) ################ (####################
42X74	### CALL ##########
42X75	###### <#>(<#>) ################### ################################
42X76	################ 1 ### '<##>' # NULL ##### ######## NULL ##############
42X77	### '<###>' ##########
42X78	# '<##>' ##############
42X79	## '<##>' #####################
42X80	VALUES ######### 1 ###################### ################
42X82	USING ############## #### ResultSet ##########
42X83	# '<##>' ######NULL ########## NULL #############
42X84	## '<###>' ########## '<##>' ######### ##################################
42X85	## '<###>' ### '<##>' ##################
42X86	ALTER TABLE ######## # '<##>' ### '<###>' #######
42X87	'<#>' ####### 1 ##### (THEN ### ELSE) # "?" ##############
42X88	

SQLSTATE	##
	##################### ##########TRUE#FALSE#### UNKNOWN ###############
42X89	# '<#>' # '<#>' ################ ######################
42X90	# '<##>' ######################
42X91	### '<###>' ##CREATE TABLE ###################
42X92	## '<##>' ######################
42X93	# '<##>' ########## '<##>' ################
42X94	<#> '<#>' ########
42X96	################ jar #### '<#####>' #########
42X98	######################
42X99	Parameters are not allowed in a TABLE definition.
42Y00	### '<####>' ##org.apache.derby.iapi.db.AggregateDefinition ###############################
42Y01	## '<###>' ######
42Y03	'<#>' ######################
42Y04	###################EXTERNAL NAME '<##>' ###################### ############## <full java path>.<method name> ###
42Y05	'<##>' ################
42Y07	#### '<#####>' #######
42Y08	#########################
42Y09	void ######CALL ################
42Y10	INSERT ###################### 1 ##### ? ############### ########### 1 ############################
42Y11	'<###>' ###############
42Y12	JOIN # ON ### '<####>' #### ##################
42Y13	## '<##>' ##CREATE VIEW ##################
42Y16	public ####### '<#####>' #### '<####>' ######### ####################public ##################
42Y22	### <#####> ### <#> ############
42Y23	# <##> ####### JDBC ###########
42Y24	## '<####>' ######### (#############)
42Y25	'<##>' ######### ##############################
42Y26	#### GROUP BY #############
42Y27	############################
42Y29	############## SELECT ########## 1 ############ SELECT ######### 1 ##
42Y30	########## SELECT ########## 1 ############ SELECT #### GROUP BY ###

SQLSTATE	##
42Y32	# {2} ####### "{1}" # Aggregator ### "{0}" ##com.ibm.db2j.aggregates.Aggregator ################
42Y33	### <####> ###1 ################
42Y34	## '<##>' ### '<##>' ################
42Y35	### '<##>' ###### SELECT ######### 1 ##
42Y36	### '<##>' ###### GROUP BY ### SELECT ##
42Y37	'<#>' # Java ################################
42Y38	insertMode = replace ######## '<##>' # SELECT #########################
42Y39	'<#>' ### deterministic ##############CHECK CONSTRAINT ################
42Y40	'<#>' ###### '<####>' # UPDATE OF #############
42Y41	'<#>' ############EXECUTE STATEMENT ###################
42Y42	### '<#####>' ##<#> ################
42Y43	### '<#####>' ##'<##>' ###################
42Y44	FROM ###################### '<##>' ######### ############## (#/########) # '<##>' ###
42Y45	VTI '<#>' ####### VTI ############################ WHEN ############################
42Y46	FROM ################### # '<##>' ### '<##>' #######
42Y48	"FROM ################### ###### '<###>' ## '<##>' ##############################
42Y49	######### '<##>' ################
42Y50	# '<##>' ##
42Y55	'<#>' ################ '<#>' ###############
42Y56	# '<##>' ######################### '<########>' ######### ##########################: "hash" ### "nestedloop"#
42Y58	# '<#>' ################## '<#>' ############ NumberFormatException ########
42Y59	hashInitialCapacity ################ '<#>' ######### ## 0 ################
42Y60	hashLoadFactor ################ '<#>' ######### ###0.0 ######1.0 ##############
42Y61	hashMaxCapacity ################ '<#>' ######### ## 0 ################
42Y62	#########'<####>' ### '<#>' #########

SQLSTATE	##
42Y63	## ###### '<##>' ############################## "index" ###################### '<##>' #########################
42Y64	'<#>' # bulkFetch ####### bulkFetch ##### 1 ###
42Y65	bulkFetch ##'<#####>' ############
42Y66	bulkFetch ###################
42Y67	#### '<#####>' ###########
42Y69	################################# ######## 2 ##### 1 ## (##############)##2 ###################
42Y70	###################### ###
42Y71	################ '<###>' ###########
42Y82	####################### '<#>' ##DROP STATEMENT ############## #############
42Y83	#### NULL ##### <####> ############## NULL #################
42Y84	'<#>' # DEFAULT ###########
42Y85	DEFAULT #######VALUES ### INSERT ################ VALUES ############
42Y90	FOR UPDATE #######################
42Y91	USING ############## EXECUTE STATEMENT ##########
42Y92	<####> ######<#> ####/###########
42Y93	### REFERENCING ##: ####/######### 1 ##############
42Y94	AND ### OR ################## AND ### OR ########TRUE#FALSE#### UNKNOWN ###############
42Y95	######## '<######>' ######### '<######>' ### '<####>' ################
42Y97	# "<###>'## '<##>' ############
42Z02	######## DISTINCT ################
42Z07	#### ON ############
42Z08	'<#>' ########## (<#>) ################################
42Z15	# '<##>' ################## ############
42Z16	# VARCHAR ######################
42Z17	# '<##>' ################## ##########################
42Z18	# '<##>' ######## '<###>' ###### ##########################ALTER TABLE ######################
42Z19	# '<##>' ###### 1 ######## '<###>' ############ ############################ALTER TABLE ######################

SQLSTATE	##
42Z20	# '<##>' # NULL ############## ######################NULL ###############
42Z21	# '<##>' # ID ################## ############
42Z22	ID # '<##>' ################## ID ########BIGINT#INT ### SMALLINT #####
42Z23	ID # '<##>' ############
42Z24	# '<##>' ## '<##>' #### ID # ################
42Z25	##### ID ###### ### \= NULL ######################
42Z26	ID ######## '<##>' # NULL ## #######
42Z27	NULL ### '<##>' ##ID ########################
42Z50	#####: <#> ##############
42Z53	#####: ##### <#> ######## Activation ##########
42Z60	<#> ############### <######> ### '<#>' ##############
42Z70	XML #################XMLPARSE ############
42Z71	########## XML ##########XMLSERIALIZE ############
42Z72	SQL/XML ##### '<#####>' ## <###>## <###> #######
42Z73	XMLSERIALIZE ##############: '<######>'#
42Z74	XML ##################: '<#######>'#
42Z75	XML ##########################
42Z76	### XML ################
42Z77	######### "XML" ############'<#>' #########
42Z79	XMLPARSE #################CAST ############
42Z90	### '<####>' ####### ResultSet #######
42Z91	###
42Z92	########
42Z93	## '<###>' # '<###>' ############################
42Z97	# '<##>' ############## '<###>' #######
42Z99	######## 16 ########64K ##############
42Z9A	###############
42Z9B	################BLOB ### CLOB ########### '<#>' ## '<#>'#
42Z9D	BEFORE #######SQL ########################
42Z9D	'<#>' ######## '<####>' ###############
42Z9E	## '<###>' ##<#> ##########
42Z9F	# <##> ######## (<##>) ####### ### <#> ###
42ZA0	################ ######################## ##
42ZA1	######## SQL ######: '<###>'#

SQLSTATE	##
42ZA2	Operand of LIKE predicate with type <#> and collation <#> is not compatable with LIKE pattern operand with type <#> and collation <#>.
42ZA3	The table will have collation type <#> which is different than the collation of the schema <#> hence this operation is not supported .

55. ## 57: DRDA ##########:

SQLSTATE	##
57017	########## <######> ############### <######> ############ ##########

56. ## 58: DRDA ##########:

SQLSTATE	##
58009	##############: VCM#VCS ###### 0 ########## 1 ###### ##########
58009	##################################
58009	##############: ########## <#> ################ <#> ######## ##########
58009	##############: DDM ################# 4 ######## ##########
58009	##############: ### ID ###################################### ##########
58009	##############: ### ID ############DSS #### 0 ########## ##########
58009	##############: DSS ######################### ID ## ##############
58009	##############: InputStream (###### #<#>) ########################## ##########
58009	##############: ### FDOCA LID ### ##########
58009	##############: SECTKN ########## ##########
58009	##############: NVCM#NVCS ########## 1 ###### ##########
58009	##############: SCLDTA ### <##> # RDBNAM ######### ##########
58009	SocketException: '<###>'
58009	#############: <###>#
58009	################################ ########################
58009	########################<#> ############################ <#> ######### ##########
58009	JVM ######### LOB ######################## ##########
58009	##############: SCLDTA ### <##> # RDBNAM ######### ##########
58009	##############: SCLDTA ### <##> # PKGID ######### ##########
58009	##############: PKGNAMCSN ### <##> # SQLAM <##> ####### ##########

SQLSTATE	##
58009	################################: <###>
58010	######################## ### <#> ####### <#> ##################################
58014	DDM #### 0x<#> ############ ##########
58015	DDM ###### 0x<#> ############ ##########
58016	DDM ###### 0x<#> ############ ##########
58017	DDM ####### 0x<#> ############ ## ##########

57. ## X0:

SQLSTATE	##
X0A00	######## '<##>' # 2 ########## GROUP BY ##### HAVING ###################### ###### 1 ######################
X0X02	# '<##>' # '<###> ############
X0X03	######################################
X0X05	####### '<##>' ########
X0X07	JAR #### '<#####>' # derby.database.classpath '<#####>' ##############
X0X0E	################### '<###>' ################
X0X0F	Column name '<*columnName*>' listed in auto-generated column selection array not found in the insert table.
X0X10	USING ################## ResultSets ##########
X0X11	USING ##############################
X0X13	#### '<#####>' # JAR #### '<#####>' ########
X0X57	# '<#>' # Java ## SQL ################# SQL ######## ## Java ##################################
X0X60	'<#####>' ###################
X0X61	## '<##>' ## '<###>.<#####>' ## '<##>' ############## '<##>' ######## ##### '<#>' ########## '<#>' ### ################## '<####>' ### ######################
X0X62	# '<##>' ### '<###>' ############### ########## '<###>' ###################### ################## '<####>' ### ######################
X0X63	IOException '<#>' #########
X0X67	# '<#>' ##########################CREATE INDEX#ORDER BY#GROUP BY#UNION#INTERSECT#EXCEPT ### DISTINCT ################
X0X81	<#> '<#>' ########
X0X85	'<###>' ################ '<###>' ############
X0X86	0 ##ResultSet.absolute(int row) ################
X0X87	####################ResultSet.relative(int row) ##############

SQLSTATE	##
X0X95	###### '<#######>' ########## '<########>' ## ResultSet ######
X0X99	## '<###>' ########
X0Y16	'<#>' ########## ########## DROP TABLE ##########
X0Y23	###### '<#######>' ########## '<########>' #########VIEW '<####>' ##################
X0Y24	###### '<#######>' ########## '<########>' #########STATEMENT '<#>' ##################
X0Y25	###### '<#######>' ########## '<###>' #########<#> '<#>' ##################
X0Y26	## '<###>' ### '<##>' ##################
X0Y28	## '<###>' ####### '<##>' ################ ##############################
X0Y32	<#> '<#>' #### <#> '<#>' #######
X0Y38	# '<##>' ########## '<###>' #########
X0Y41	### <##> ########## '<###>' ###### #### <##> ###
X0Y42	## '<###>' #####: #####################
X0Y43	## '<###>' #####: <###> (<#>) ######## (<#>) ##########
X0Y44	## '<###>' #####: # '<##>' ###
X0Y45	1 ################################## '<###>' ## <##> ###########################
X0Y46	## '<###>' #####: ### <##> ########
X0Y54	#### '<#####>' ##################
X0Y55	################ 1 ############### # '<#####>.<##>' #### '<##>' # <#> ######### <#> #### ######################
X0Y56	'<#>' ####### '<##>' ##########
X0Y57	NULL ######## '<##>' ######################## 1 ########## NULL ######################
X0Y58	####### '<##>' ###################################### ###1 ###################
X0Y59	# '<##>' ###################################### <###> ##################: <##>#
X0Y63	# '<##>' ############################/#### NULL ############ ######################## NULL ############
X0Y66	##
X0Y67	##
X0Y68	<#> '<#>' ##########
X0Y69	<#####> ##### <#> ##############

SQLSTATE	##
X0Y70	#### <##> ########## <#####> ## INSERT#UPDATE ### DELETE #########
X0Y71	SET ISOLATION ################### <####> ##################
X0Y72	'<#>' ########## (<#>) ###############################
X0Y77	## set transaction isolation ################
X0Y78	Statement.executeQuery() ##############################
X0Y78	################<#>.executeQuery() ######### ############ <#>.execute() ##########
X0Y78	<#>.executeQuery() ########################## ############<#>.executeUpdate() ##########
X0Y79	Statement.executeUpdate() ##ResultSet ######################
X0Y80	ALTER # '<##>' ######## # '<##>' # NULL ############
X0Y83	##: ################################ ID <*id*> ######### <##> ################ ##################################

58. ## XBCA:

SQLSTATE	##
XBCA0	## <##> #### <#####> ########################## #########################

59. ## XBCM:

SQLSTATE	##
XBCM1	######## <####> ###### Java ##################
XBCM2	######## <####> ################
XBCM3	#### <#####>() ########## <####> ########
XBCM4	Java ########################: ######## <####> # <#>#

60. ## XBCX:

SQLSTATE	##
XBCX0	################ ######################
XBCX1	###################ENCRYPT # DECRYPT ###############
XBCX2	############################ ############## <#> ############
XBCX5	########## NULL ##############
XBCX6	###
XBCX7	######################## ################: old_boot_password, new_boot_password#
XBCX8	##################################

SQLSTATE	##
XBCX9	################################
XBCXA	###############
XBCXB	########## '<#>' ####################### "NoPadding" ##########
XBCXC	######### '<#######>' ######## ########## '<######>' ################################
XBCXD	##################################
XBCXE	##################################
XBCXF	############### '<####>' #########
XBCXG	######### '<#######>' ########
XBCXH	encryptionAlgorithm '<#######>' ############ ###########algorithm/feedbackMode/NoPadding ###
XBCXI	########## '<###>' ############ ###################### CBC#CFB#OFB#### ECB ###
XBCXJ	##########1.2.1 #### Java Cryptography Extension (JCE) ############### JCE 1.2.1 ################################
XBCXK	################################## ################################
XBCXL	################## ## ######################
XBCXM	#######################
XBCXN	############## 1 ########## 16 ############ 0 ## 9#a ## f#### A ## F ###
XBCXO	##
XBCXP	##
XBCXQ	################################
XBCXR	##
XBCXS	##
XBCXT	##
XBCXU	############################: <#####>
XBCXV	##: <#####>

61. ## XBM:

SQLSTATE	##
XBM01	################ ######################
XBM02	<#> ################## ######### Derby ##########################
XBM05	<*value*> ##########################
XBM06	########## ###

SQLSTATE	##
XBM07	########## ############ 8 ################
XBM08	<#> StorageFactory ### <#> ################
XBM0G	################ Java 2 ########JCE ##
XBM0H	######## <########> #########
XBM0I	######## <########> #########
XBM0J	######## <########> ##########
XBM0K	######## <#######> ##############
XBM0L	################ <####> ############ <#########> ################
XBM0M	########## <####> ######################
XBM0N	java.sql.DriverManager ## JDBC ################ ######################
XBM0P	################### ##################
XBM0Q	#### <#####> ######### ##
XBM0R	#### <#####> #########
XBM0S	#### '<#####>' # '<#####>' ###########
XBM0T	######## <#######> ###############
XBM0U	ID <####> ##################
XBM0V	ID <####> ############ <####> #################
XBM0W	ID <####> ############ <####> ######################
XBM0X	############ '<#>' ####################: ln[_CO[_variant]]\nln=### 2 ###### ISO-639 ######CO= ### 2 ###### ISO-3166 ######java.util.Locale ####
XBM03	Supplied value '<#>' for collation attribute is invalid, expecting UCS_BASIC or TERRITORY_BASED.
XBM0Y	###################### <########> ######### ##############################
XBM0Z	#### '<#####>' # '<#####>' ########## ##############################

62. ## XCL:

SQLSTATE	##
XCL01	############## ######## <###> #########
XCL05	Activation ################## <###> #########
XCL07	#### '<#####>' ############ ##########################
XCL08	#### '<#####>' #########
XCL09	PreparedStatement ###### '<#####>' ##### Activation ########
XCL10	

SQLSTATE	##
	PreparedStatement ######################### JDBC ##################################
XCL12	# '<#####>' ####### '<#####>' ##################
XCL13	######## '<#######>' ####### ######################## '<#>' ###
XCL14	### '<###>' ####### ## ResultSet ####### '<#>' ###
XCL15	###### '<######>' #### compareTo() ############### ClassCastException ######## compareTo() ########### '<####>' ###
XCL16	ResultSet ############ ####### '<##>' ######### ############################
XCL16	ResultSet ############ ##########################
XCL17	#########################
XCL18	############ 2 ##############
XCL19	# '<##>' ### '<##>' ############
XCL20	######### '<#######>' ################ '<#######>' ##################
XCL21	############################## (CREATE#DROP#### ALTER) ############ ########## ####SQL ################ Java ##
XCL22	###### <#######> # IN ############OUT ################
XCL23	SQL ### '<#>' ##registerOutParameter() ###################
XCL24	###### <######> ################registerOutParameter() ################### ################### <#> ############
XCL25	###### <######> ## <#> ###################### <#> ########## 2 ########
XCL26	###### <######> ################
XCL27	###################
XCL30	InputStream ## '<#>' ############IOException ##########
XCL31	##################
XCL33	############## <##> ######################## (######################## SET NULL ################)
XCL34	############## <##> ######################## (## (###############CASCADE ######))
XCL35	############## <##> ######################## (## SET NULL ###)
XCL36	########## <#> ########## (## (NO ACTION#RESTRICT ### CASCADE)#)
XCL37	########## <#> ########## (#################### CASCADE ################)

SQLSTATE	##
XCL38	########## <###> ########## (## (NO ACTION#RESTRICT ### CASCADE)#)
XCL39	########## CASCADE ######## (########SET NULL#NO ACTION ### RESTRICT ############)
XCL40	########## CASCADE ######## (## ################ 1 ## CASCADE ############ CASCADE ############################)
XCL41	########## CASCADE ######## (####################################SET NULL ##)
XCL42	CASCADE
XCL43	SET NULL
XCL44	RESTRICT
XCL45	NO ACTION
XCL46	SET DEFAULT
XCL47	'<#>' ################### <#######> ####### <#######> ###################
XCL48	TRUNCATE TABLE # '<#>' ##################/######################################
XCL49	'<#>' ###### DELETE #### (<#>) ################ TRUNCATE TABLE #########
XCL50	################################### ######################### '<#######>' ##################### '<#######>' ###
XCL51	#########SESSION ################
XCL52	##############################
XCL53	Stream is closed

 # 63. ## XCW: ################

SQLSTATE	##
XCW00	'<#>' ## '<#>' #######################

 # 64. ## XCX: ################

SQLSTATE	##
XCXA0	### ID#
XCXB0	###############: '<#####>'#
XCXC0	### ID ####
XCXE0	##

 # 65. ## XCY: Derby ############

SQLSTATE	##
XCY00	########## '<#>'='<#>'#
XCY02	############################'<#>'='<#>'#
XCY03	######### '<######>' ###########
XCY04	######################### ################### -- DERBY-PROPERTIES propertyName = value [, propertyName = value]*

66. ## XCZ: org.apache.derby.database.UserUtility

SQLSTATE	##
XCZ00	##### '<###>'#
XCZ01	####### '##ID'#
XCZ02	######### '<#>'='<#>'#

67. ## XD00:

SQLSTATE	##
XD003	###################### DependableFinder = '<#>'# ####: "<#>"#
XD004	##############

68. ## XIE: #####/#########

SQLSTATE	##
XIE01	### NULL ####
XIE03	# <###> ## <##> ###########################
XIE04	################: <#####>
XIE05	######### NULL ############
XIE06	######### NULL ####
XIE07	###
XIE08	############: <##>#
XIE09	########: <#>#
XIE0B	### '<##>' ### <#> ########/##############################
XIE0D	# <###> ##################
XIE0E	# <###> ######### endOfFile ########
XIE0I	############### IOException ########
XIE0J	##########1 ############
XIE0K	############################
XIE0M	# '<##>' ########
XIE0N	An invalid hexadecimal string '<*16*####>' detected in the import file.
XIE0P	Lob data file <#####> referenced in the import file not found.

SQLSTATE	##
XIE0Q	Lob data file name cannot be null.
XIE0R	Import error on line <###> of file <#####>: <##>

69. ## XJ:

SQLSTATE	##
XJ004	###### '<#######>' #########
XJ008	##
XJ009	CallableStatement ##*<value>*
XJ010	autoCommit ###############savepoint #########
XJ011	######### NULL ############
XJ012	'<#>' ###############
XJ013	############## ID #######
XJ014	######################
XJ015	Derby ##################
XJ016	#### '<#####>' ########################
XJ017	savepoint ##########################
XJ018	### NULL ############
XJ020	######## TYPE '<##>' ##########java.sql.Types ###################### NULL ####
XJ021	##############
XJ022	##############: '<##>'#
XJ023	#####################################
XJ025	#########################
XJ028	URL '<*url*#>' ##############
XJ030	##############################
XJ040	###### '<#######>' ############################
XJ041	###### '<#######*gt;*' ################################
XJ042	'<#>' ######## '<#######>' ##############
XJ044	'<#>' ##########
XJ045	##### (##) ################ '<####>' # Connection.setTransactionIsolationLevel() ######## ##############java.sql.Connection.TRANSACTION_SERIALIZABLE#java.sql.Connection.T java.sql.Connection.TRANSACTION_READ_UNCOMMITTED ###
XJ049	###################
XJ04B	###################################
XJ04C	CallableStatement ##########################

SQLSTATE	##
XJ056	######AUTOCOMMIT ON #########
XJ057	Cannot commit a global transaction using the Connection, commit processing must go thru XAResource interface.
XJ058	Cannot rollback a global transaction using the Connection, commit processing must go thru XAResource interface.
XJ059	##
XJ05B	JDBC ## '<###>' ## '<#>' ########## '<#>' ###
XJ05C	################### ResultSet.HOLD_CURSORS_OVER_COMMIT #########
XJ061	'<#####>' ##########################
XJ062	ResultSet.setFetchSize(int rows) ############## '<#>'#
XJ063	Statement.setMaxRows(int maxRows) ############## '<#>'# ####### >= 0 ##########
XJ064	setFetchDirection(int direction) ############## '<#>'#
XJ065	Statement.setFetchSize(int rows) ############## '<#>'#
XJ066	Statement.setMaxFieldSize(int max) ############## '<#>'#
XJ067	SQL ########## NULL ###
XJ068	#########executeBatch # clearBatch ##########
XJ069	SetXXX ######USING ######################
XJ070	########## '<##>' # BLOB ### CLOB ###########
XJ071	###### '<##>' # BLOB ### CLOB ###########
XJ072	NULL ####### searchStr # BLOB ### CLOB ##############
XJ073	## BLOB ### CLOB ############### BLOB/CLOB ###
XJ074	Statement.setQueryTimeout(int seconds) ############## '<#>'#
XJ076	#### '<####>' # BLOB/CLOB ###########
XJ077	getBytes/getSubString ##### BLOB/CLOB ##########/######################
XJ078	##### '<#>' ############ BLOB/CLOB ##########
XJ079	####### '<#>' # BLOB/CLOB ###########
XJ080	USING ########## <#> #### <#> ##############
XJ081	##### create/restore/recovery ##########
XJ081	###### '<#######>' ####### '<#>' ##### '<#####>' ########
XJ085	##
XJ086	########################## ResultSet ########## CONCUR_READ_ONLY ##############
XJ087	Sum of position('<##>') and length('<##>') is greater than the size of the LOB.
XJ088	##########: ######### wasNull() ##########

SQLSTATE	##
XJ090	#########: ###### NULL ###
XJ091	#####: ######## <####> # OUT ######## INOUT ##############
XJ093	BLOB/CLOB ### <#> ####### ### <#> ##########
XJ094	######################
XJ095	######################
XJ096	########## <#> # <######> ##################
XJ097	##
XJ098	########## <#> ######
XJ099	Reader/Stream ######################
XJ100	registerOutParameter ################ setter ############ ################
XJ102	################
XJ103	### NULL ############
XJ104	##########: <#>#
XJ105	DES ################## <#>######## <#> ###
XJ106	################
XJ107	############
XJ108	##################
XJ110	#### NULL ############
XJ111	##### NULL ############
XJ112	######################################
XJ113	#### <#####> ##########: <###>
XJ114	######## '<#####>' ##
XJ115	######### <#> ##################
XJ116	################ <#> ######
XJ117	######### J2EE ##################
XJ118	############################
XJ121	######################
XJ122	####### updateXXX ################
XJ123	##
XJ124	##########
XJ125	############### ResultSet ###### (TYPE_SCROLL_SENSITIVE #### TYPE_SCROLL_INSENSITIVE #) ##################
XJ126	###################################
XJ128	'<#>' ###############
XJ200	######### <#> #######
XJ202	######## '<#####>' ###

SQLSTATE	##
XJ203	##### '<#####>' ############
XJ204	######### <######> ##################
XJ206	SQL #### '<#>' ############
XJ207	executeQuery ######################
XJ208	################## ############################## 1 ############## ############################ getNextException() ##########
XJ209	###################################
XJ210	##
XJ211	############################ ################
XJ212	#######: <####>
XJ213	traceLevel ###########################
XJ214	CLOB ### BLOB #### free() ####### IO ##########
XJ215	free() ################ java.sql.Clob/java.sql.Blob ##################
XJ216	The length of this BLOB/CLOB is not available yet. When a BLOB or CLOB is accessed as a stream, the length is not available until the entire stream has been processed.
XJ217	The locator that was supplied for this LOB/BLOB is invalid

70. ## XK:

SQLSTATE	##
XK000	The security policy could not be reloaded: <##>

71. ## XN:

SQLSTATE	##
XN001	##########################
XN008	##########################
XN009	BLOB/CLOB ##################################
XN010	######### NULL ############
XN011	########## <#> ######### 1 ## <#> #########
XN012	<#######> ##########XA ###### <#######> ################### <#######> ###
XN013	##############
XN014	################: IOException ############### #<#> ### ######## 0x0 ########## #####: <##>#
XN015	################: ##### InputStream #### (###### #<#>) #### InputStream ##############
XN016	################: ############################ #<#> ### #####: <##>#

SQLSTATE	##
XN017	################: ########################## #<#> ### ######## 0x0 ##########
XN018	################: ##### Reader #### (###### #<#>) #### Reader ##############
XN019	<#> ################## <#> #######

72. ## XSAI: ### - access.protocol.interface

SQLSTATE	##
XSAI2	###################### (<#>) ########
XSAI3	######################

73. ## XSAM: ### - AccessManager

SQLSTATE	##
XSAM0	'<#>' ##########################
XSAM2	#######conglom id '<*conglomID*>' ##################################
XSAM3	conglom id '<*conglomID*>' ##################################
XSAM4	'<####>' ################
XSAM5	##########next() ##################################
XSAM6	##### <#####> ##### <#####> ###### <######> ############

74. ## XSAS: Store -

SQLSTATE	##
XSAS0	##
XSAS1	##################################
XSAS3	##################################
XSAS6	######################

75. ## XSAX: ### - access.protocol.XA statement

SQLSTATE	##
XSAX0	XA ########
XSAX1	################### Xid ##############################

76. ## XSCB: ### - BTree

SQLSTATE	##
XSCB0	################
XSCB1	##### <#####> #########
XSCB2	

SQLSTATE	##
	######### <#######> ##btree # 2 #### createConglomerate() ##############################
XSCB3	###################
XSCB4	################ (####next() ##########) ##btree ########################## ########### (<#>) ###
XSCB5	btree ########## UNDO ###################
XSCB6	##: ################btree # 2 ################## ################## derby.storage.pageSize ###/### derby.storage.pageReservedSpace ##########
XSCB7	btree ####################current_rh # NULL = <#>###### NULL = <#>#
XSCB8	btree ################## <#> ############
XSCB9	##############

 # 77. ## XSCG0: ########

SQLSTATE	##
XSCG0	##################

 # 78. ## XSCH: ###

SQLSTATE	##
XSCH0	################
XSCH1	##### <#####> #########
XSCH4	##############################
XSCH5	################ <#> ##### <#> ################
XSCH6	######### (##### ID <#####*ID*>) ###########
XSCH7	##################
XSCH8	#######################

 # 79. ## XSDA: RawStore - Data.Generic statement

SQLSTATE	##
XSDA1	###########################
XSDA2	###################
XSDA3	##: ################################## ################## derby.storage.pageSize ###/### derby.storage.pageReservedSpace ##########
XSDA4	################
XSDA5	###########################
XSDA6	### <##> # NULL ############################

SQLSTATE	##
XSDA7	### <####> ############ SQLData ##
XSDA8	### <####> ############ SQLData ######################
XSDA9	### <####> ############ SQLData ###########################
XSDAA	########### <#>##
XSDAB	NULL ################
XSDAC	##
XSDAD	1 ##############################
XSDAE	########### ID ###################
XSDAF	###################################
XSDAG	###
XSDAI	############## <###> ########################
XSDAJ	########### SQLData #######################
XSDAK	######### <#> ###################
XSDAL	######### <#> ###########################
XSDAM	### <####> # SQLData ###################### ######################
XSDAN	### <####> # SQLData ###################### ############################

 # 80. ## XSDB: RawStore - Data.Generic transaction

SQLSTATE	##
XSDB0	######### <###> ###############
XSDB1	### <###> ##################
XSDB2	##### <#####> ####################: <#>
XSDB3	########################<#> ######## <#> ###
XSDB4	### <###> ###### <#######> ################## <#######> ##
XSDB5	###### <###> ###################################
XSDB6	Derby ################## <#######> ################
XSDB7	##: Derby (###### <#>) ############################Derby (###### <#>) ####### <#######> ############## ############# Derby #########1 ## 1 ###### ##
XSDB8	##: Derby (###### <#>) ############################Derby (###### <#>) ####### <#######> ############## ############# Derby #########1 ## 1 ###### Derby # 2 ## db2j.database.forceDatabaseLock=true ##################db.lck ########################## ###################### Derby

SQLSTATE	##
	## ################################## db.lck ################################## VM ######################
XSDB9	########## <#####> ##########
XSDBA	###### <######> #############

 # 81. ## XSDF: RawStore - Data.Filesystem statement

SQLSTATE	##
XSDF0	#### <#####> ######################
XSDF1	########## <#####> #######
XSDF2	########## <#####> ################################ ##: <#>##
XSDF3	##### <######> #########
XSDF4	################ <#####> ##############################: <#>##
XSDF6	####### <###> #########
XSDF7	########################: <#>
XSDF8	######## <###> #########
XSDFB	###################################
XSDFD	### <###> # 2 ############################1 ##########################2 ########################## ################: <#><#>
XSDFF	################################
XSDFH	################################### <#####> ######################
XSDFI	## ############################## ## ## ######### <###> ################

 # 82. ## XSDG: RawStore - Data.Filesystem database

SQLSTATE	##
XSDG0	### <###> ##################
XSDG1	### <###> ##
XSDG2	### <###> ########################=<#>############=<#>########: <#>
XSDG3	##### <######> ##################
XSDG5	createFinished ############################
XSDG6	################################ <#> ######### ##
XSDG7	############## <#######> ############ ##################

SQLSTATE	##
XSDG8	############## '<#######>' # '<#######>' ########## ##############################

 # 83. ## XSLA: RawStore - Log.Generic database exceptions

SQLSTATE	##
XSLA0	############ <#> ############
XSLA1	########################## (###### <######>) ######### ##############################
XSLA2	###
XSLA3	##################################
XSLA4	########################## ################## ##
XSLA5	################################ <##########> ##################
XSLA6	##################
XSLA7	############ <#######> ##########
XSLA8	######## <#> ############<#> # <#> ################
XSLAA	######################################
XSLAB	####### <#######> #########logDevice ##
XSLAC	<#> ##
XSLAD	####### <#######> ######## <#> ################### ################ <#> ########## <#> ####
XSLAE	<#> ########################
XSLAF	##
XSLAH	######################
XSLAI	############################
XSLAJ	##
XSLAK	################## <#> #######
XSLAL	########### <#> ################ <#> ####### ####### <#######> (## <#>) ############
XSLAM	IOException ####{1} ########################
XSLAN	<#> ## ################ <#######*versionNumber*> ######################
XSLAO	####### <#> ##################
XSLAP	###### (<#>) ###### <#######> ### ########################
XSLAQ	####### <#######> ################

SQLSTATE	##
XSLAR	############## '<#######>' # '<#>' ########## ##############################
XSLAS	########################## <#######> ######### ###
XSLAT	########## '<#######>' ####### ################################### logDevice #########################

 # 84. ## XSLB: RawStore - Log.Generic statement exceptions

SQLSTATE	##
XSLB1	########## <#########> ##
XSLB2	########## <#########> ##################################
XSLB4	truncationLWM <#> #########
XSLB5	######## <#> #### truncationLWM ###### <#># ###### <#> ## <#> ###
XSLB6	0 ### -ve ##########################
XSLB8	<#> ############# <#> ###############
XSLB9	##################################

 # 85. ## XSRS: RawStore - protocol.Interface statement

SQLSTATE	##
XSRS0	##################################
XSRS1	####### <#> ##########################
XSRS4	##### (########) <#> ## <#> ######################
XSRS5	##### (########) <##> ## <##> ######################
XSRS6	############## <########> #########
XSRS7	#########################
XSRS8	##
XSRS9	#### <#####*recordName*> ##########
XSRSA	## ###
XSRSB	##
XSRSC	####### <########*directoryLocation*> ##################################

 # 86. ## XSTA2: XACT_TRANSACTION_ACTIVE

SQLSTATE	##
XSTA2	##

 # 87. ## XSTB: RawStore - Transactions.Basic system

SQLSTATE	##
XSTB0	##########################
XSTB2	##
XSTB3	(###############) ########## NULL #######################
XSTB5	######################################
XSTB6	1 ##

 # 88. ## XXXXX: No SQLSTATE

SQLSTATE	##
XXXXX	######################

JDBC

Derby######JDBC###############

######JDBC API###########Derby###################API###
#######Java#####################(Sun##############4###)

#####Derby####JDBC API###############JDBC
2.0#3.0#4.0#API######################

########################*Derby* ######## #############

##JDBC#####Sun########JDBC################ Derby###JVM############################JDBC
##############################
######/
################JDBC#############JDBC#################Derby###JVM################JD

Derby#####JDBC#####Derby##############################JDBC################
JDBC###################"Feature not
implemented"#########XJZZZ###*SQLState#SQLException*#######
################Derby######################

java.sql####JDBC#######

###########*java.sql*####################Derby#############

- *java.sql.Driver#########*
- *java.sql.DriverManager.getConnection ####*
- *java.sql.Driver.getPropertyInfo ####*
- *java.sql.Connection ########*
- *java.sql.DatabaseMetaData ########*
- *java.sql.Statement########*
- *java.sql.PreparedStatement########*
- *java.sql.CallableStatement ########*
- *java.sql.ResultSet ########*
- *java.sql.ResultSetMetaData ########*
- *java.sql.SQLException ###*
- *java.sql.SQLWarning ###*
- *SQL##java.sql.Types###*

java.sql.Driver########

Derby##############org.apache.derby.jdbc.EmbeddedDriver##########
############################
java.sql.Driver##############################DriverManager#########

- ###########JDK1.6###
- *Class.forName("org.apache.derby.jdbc.EmbeddedDriver")*

  #################JVM################################
- *new org.apache.derby.jdbc.EmbeddedDriver()*

  Class.forName("org.apache.derby.jdbc.EmbeddedDriver")#################################
- *Class c = org.apache.derby.jdbc.EmbeddedDriver.class*

  ###Class.forName("org.apache.derby.jdbc.EmbeddedDriver")#################################
- *##########jdbc.drivers*

  ###

```
java -Djdbc.drivers=org.apache.derby.jdbc.EmbeddedDriver
    applicationClass
```

jdbc:derby:##############DriverManager################org.apache.derby.jdbc.EmbeddedDriver####

*jdbc:derby:##########*Derby*##############DriverManager########*(*java.sql.Driver*)*#####*
####getDriver#getConnection###########(*java.sql.Connection*)*###########*

java.sql.Driver.getPropertyInfo ####

*################*JDBC*#######DriverPropertyInfo#################*

```
java.sql.DriverManager.getDriver("jdbc:derby:").
    getPropertyInfo(URL, Prop)
```

org.apache.derby.jdbc.EmbeddedDriver##################################DriverPropertyInfo#######

#######DriverPropertyInfo############
Derby*####################*URL*############*
#########databaseName=nameofDatabase,#################################

###Derby*####################toursDB#flightsDB*###A:
dbs/tours94*############*
getPropertyInfo########databaseName####################
*##DriverPropertyInfo#########*choices*#######toursDB#flightsDB*#A:/dbs/
tours94*####################*
############################(*#########*)*###############*shutdown###true*############*

java.sql.Driver.getPropertyInfo###################Derby ########
####8##################################

java.sql.DriverManager.getConnection

JDBC
API*#################################Connection#############Connection#################*
JDBC*#########*URL(uniform resource locator)*##################*

*DriverManager.getConnection#########*URL*####Properties####################Properties######*

*###*Derby*###*
*#####*Derby*###*
*#########################*Derby*########APP########*Derby*##################*
###################################

Derby#########URL##

Derby#########URL##############URL#####################################

##########################*Derby* ########
#####1##########################

##########################URL##

################################URL#########

```
jdbc:derby: [#########:][#######][;##]*
```

- *jdbc:derby:*

  JDBC##*derby*##Derby##########################
  ############*derby*###################
- *#########:*

  ######################Derby#########################jar#################################
  ######### ############

  - directory
  - classpath:
    ##
  - jar ####################

  jar: ########:###################

  ```
  (#########)
  ```

  ####################jar#zip####################

  ########################URL####*Derby* ######## ##############
- *#######*

  ################################

  ####################/####################*Derby* ######## #####1#####
  "######-####################"##########
- *##*

  0#############URL################Derby#########URL############

####SQL##
Derby##########JDBC##############SQL##############################

```
jdbc:default:connection
```

Derby#########URL###

########URL#######################
Derby###*DriverManager.getConnection#Properties########*

#######Derby####*#########*URL*#########*##############

###URL

```
import java.util.Properties;

Connection conn = DriverManager.getConnection(
    "jdbc:derby:sampleDB;create=true");
/* ###Properties##########*/
Properties myProps = new Properties();
myProps.put("create", "true");
Connection conn = DriverManager.getConnection(
    "jdbc:derby:sampleDB", myProps);

/* ################*/
```

```
Connection conn = DriverManager.getConnection(
    "jdbc:derby:sampleDB", "dba", "password");
```

Note:
##(Derby#####################
##########################)

java.sql.Connection

Derby###*Connection*##################JDBC#################################*Connection*##

############################Connection##################JDBC#########################

java.sql.Connection.setTransactionIsolation ####

Derby############################*java.sql.Connection.TRANSACTION_SERIALIZABLE*#
java.sql.Connection.TRANSACTION_REPEATABLE_READ#
java.sql.Connection.TRANSACTION_READ_COMMITTED
###java.sql.Connection.TRANSACTION_READ_UNCOMMITTED ###

##########TRANSACTION_READ_COMMITTED ###

#########setTransactionIsolation###
##########################Derby ######## ################################

java.sql.Connection.setReadOnly ####

java.sql.Connection.setReadOnly########

############Derby #############Connection.setReadOnly
########################

java.sql.Connection.isReadOnly ####

##############################isReadOnly#######################Connections#setReadOnly#####
#########)Connections######readOnlyAccess#############
###################################

###########

Derby ##################################"#############"#########

- *createArrayOf(java.lang.String, java.lang.Object[])*
- *createNClob( )*
- *createSQLXML( )*
- *createStruct(java.lang.String, java.lang.Object[])*
- *getTypeMap( )*
- *prepareStatement(java.lang.String, int[])*
- *prepareStatement(java.lang.String, java.lang.String[])*
- *setTypeMap(java.util.Map)*

java.sql.DatabaseMetaData

######Derby###`java.sql.DatabaseMetaData`#############

DatabaseMetaData #####

##################DatabaseMetaData##############################

####JDBC#################################*DatabaseMetaData*#############################
###############

java.sql.DatabaseMetaData.getProcedureColumns####

Derby#Java#############Derby###SQL####Java###############
getProcedureColumns#######Derby#############

######Java## Derby##CREATE PROCEDURE#########Java#############

getProcedureColumns#ResultSet#####################################

getProcedureColumns###

JDBC API############################

- *catalog*

  Derby#########*null*#########
- *schemaPattern*

  Java################
- *procedureNamePattern*

  ##########################
- *column-Name-Pattern*

  ##CREATE PROCEDURE###Java#################################%###########

getProcedureColumns#################

getProcedureColumns####ResultSet###API#################API#############################

- PROCEDURE_CAT

  Derby#####null####
- PROCEDURE_SCHEM

  Java#############
- PROCEDURE_NAME

  #########
- COLUMN_NAME

  ########(column-Name-Pattern##########)
- COLUMN_TYPE

  short#############################*DatabaseMetaData.procedureColumnIn*#################
- TYPE_NAME

  Derby########
- NULLABLE

  ##############*DatabaseMetaData.procedureNoNulls*##############*DatabaseMetaData.procedure*
- REMARKS

  a String describing the java type of the method parameter
- COLUMN_DEF

  ###############(null############)
- SQL_DATA_TYPE

  #########JDBC###########
- SQL_DATETIME_SUB

  #########JDBC###########
- CHAR_OCTET_LENGTH

  ################(#########NULL###)
- ORDINAL_POSITION

  ######/#########1##########
- IS_NULLABLE

  ########################(YES#####NULL##########NO########)

- SPECIFIC_NAME

  ######################
- METHOD_ID

  Derby#######
- PARAMETER_ID

  Derby#######

java.sql.DatabaseMetaData.getBestRowIdentifier####

java.sql.DatabaseMetaData.getBestRowIdentifier##

java.sql.DatabaseMetaData.getBestRowIdentifier#########################

- *#####*
- *################*
- *######*

Note:
java.sql.DatabaseMetaData.getBestRowIdentifier##

java.sql.Statement########

JDBC 1.2#######*java.sql.Statement#setEscapeProcessing*######Derby############
############################

- *cancel( )*
- *execute(java.lang.String, int[])*
- *execute(java.lang.String, String[])*
- *executeUpdate(java.lang.String, int[])*
- *executeUpdate(java.lang.String, String[])*

ResultSet ######

####SELECT###################*ResultSet*############*ResultSet*########################*Resul*

####java.sql.Statement#java.sql.PreparedStatement#executeQuery##############################

##################################ResultSet#########################
######################GROUP BY#ORDER
BY#DISTINCT#INTERSECT#EXCEPT#UNION###################

Statement################ResultSet############

ResultSet###

java.sql.CallableStatement

Derby###JDBC 1.2####*CallableStatement*############

- *getBoolean()*
- *getByte()*
- *getBytes()*
- *getDate()*
- *getDouble()*
- *getFloat()*
- *getInt()*
- *getLong()*
- *getObject()*
- *getShort()*
- *getString()*
- *getTime()*
- *getTimestamp()*

- *registerOutParamter()*
- *wasNull()*

CallableStatements#OUT##

Derby###OUT#######CALL#########################

```
CallableStatement cs = conn.prepareCall(
   "? = CALL getDriverType(cast (? as INT))"
cs.registerOutParameter(1, Types.INTEGER);
cs.setInt(2, 35);
cs.executeUpdate();
```

Note: ####CALL#########*? =*###########

##############################

CallableStatements#INOUT##

Java#########INOUT#########################(#########################)
#######SQL##############

#########:

```
CallableStatement call = conn.prepareCall(
    "{CALL doubleMyInt(?)}");
// ##################
// inout############
call.registerOutParameter(1, Types.INTEGER);
call.setInt(1,10);
call.execute();
int retval = call.getInt(1);
```

#####*doubleIt*#####int###############################

```
public static void doubleMyInt(int[] i) {
    i[0] *=2;
    /* Derby###############*/
}
```

Note: ############################
 # 89. INOUT#######

JDBC##	**###########**	**#######**
BIGINT	long[]	long
BINARY	byte[][]	byte[]
BIT	boolean[]	boolean
DATE	*java.sql.Date[]*	*java.sql.Date*
DOUBLE	double[]	double
FLOAT	double[]	double
INTEGER	int[]	int
LONGVARBINARY	byte[][]	byte[]
REAL	float[]	float
SMALLINT	short[]	short
TIME	*java.sql.Time[]*	*java.sql.Time*
TIMESTAMP	*java.sql.Timestamp[]*	*java.sql.Timestamp*
VARBINARY	byte[][]	byte[]

JDBC##	##########	#######
OTHER	*yourType[]*	*yourType*
JAVA_OBJECT (Java2/JDBC 2.0########)	*yourType[]*	*yourType*

######################INOUT###################################

java.sql.SQLException

Derby###*SQLExceptions#getMessage()#getSQLState()#getErrorCode()*###################
####Derby##*nextException*##########*SQLExceptions*##########
Derby##########SQL-92#####################
SQLExceptions##########*Derby* ########
####5###"##########Derby#SQLExceptions###"##########

java.sql.PreparedStatement########

Derby###JDBC
1.2##############################*setXXX*##########*setObject(Value,JDBCTypeCode)*##########

#################*setString*###########

######*PreparedStatement#setCursorName*##

###############

setXXXStream#####################################

JDBC##IN#####Java###JDBC##################
Derby##JDBC 1.2############################ ###############

- *setBinaryStream*

  ####################

- *setAsciiStream*

  ASCII###############

- *setUnicodeStream*

  Unicode###############

JDBC 2.0#JDBC 3.0######################Derby##########JDK 1.5#######################
##############################Java##################*java.io.InputStream*################

JDBC#######################################
 # 90. ########JDBC#####

######	####Java##	AsciiStream	UnicodeStream	BinaryStream
CLOB	java.sql.Clob	x	x	'
CHAR	'	x	x	'
VARCHAR	'	x	x	'
LONGVARCHAR	'	X	X	'
BINARY	'	x	x	x
BLOB	java.sql.Blob	x	x	x
VARBINARY	'	x	x	x

######	####Java##	AsciiStream	UnicodeStream	BinaryStream
LONGVARBINARY	'	x	x	X

Note:

- #########X#################################SQL##java.sql.Types###############
- ##################LONG VARCHAR###LONG VARCHAR FOR BIT DATA## LONG VARCHAR###LONG VARCHAR FOR BIT DATA##############################
- ##

#####java.io.File#########LONG VARCHAR################

```
Statement s = conn.createStatement();
s.executeUpdate("CREATE TABLE atable (a INT, b LONG VARCHAR)");
conn.commit();
java.io.File file = new java.io.File("derby.txt");
int fileLength = (int) file.length();
// ###############
java.io.InputStream fin = new java.io.FileInputStream(file);
PreparedStatement ps = conn.prepareStatement(
    "INSERT INTO atable VALUES (?, ?)");
ps.setInt(1, 1);
// ######################
ps.setAsciiStream(2, fin, fileLength);
ps.execute();
conn.commit();
```

java.sql.ResultSet

ResultSet##*ResultSet*######################

##*ResultSet*##########################

Derby*###getXXX*######JDBC 1.2#########################

JDBC*##ResultSet.getBigDecimal*######################Derby*###java.math.BigDecimal.ROUND_HA*

###############

#############OutputStream########getBinaryStream#################

########getXXXStream###################*ResultSet*#################### ##################

##

#####LONG VARCHAR##################

```
// #################
ResultSet rs = s.executeQuery("SELECT b FROM atable");
while (rs.next()) {
    // java.io.InputStream#############
    java.io.InputStream ip = rs.getAsciiStream(1);
    // ###########--###########// ###########
    int c;
    int columnSize = 0;
    byte[] buff = new byte[128];
    for (;;) {
        int size = ip.read(buff);
        if (size == -1)
            break;
        columnSize += size;
        String chunk = new String(buff, 0, size);
        System.out.print(chunk);
    }
}
rs.close();
```

```
s.close();
conn.commit();
```

java.sql.ResultSetMetaData

Derby##*ResultSets*##

#####	#
isDefinitelyWritable	false
isReadOnly	false
isWritable	false

java.sql.SQLWarning

Derby##*create*##########################sum()#

######################Derby#####*derby.log*############## All other informational messages are written to the Derby system's *derby.log* file.

java.sql.SQLXML########

JDBC 4.0###SQL#XML#####java.sql.SQLXML##################Derby###XML##SQL###############JDB

Derby###java.sql.SQLXML########################XML########################XML#########

###Derby###XML#####JDBC###################

SQL##java.sql.Types###

Derby##java.sql.Types#SQL###############

#####*java.sql.Types*#SQL#####################
 # 91. SQL##java.sql.Types###

java.sql.Types	**SQL#**
BIGINT	BIGINT
BINARY	CHAR FOR BIT DATA
BIT[1]	CHAR FOR BIT DATA
BLOB	BLOB (JDBC 2.0##)
CHAR	CHAR
CLOB	CLOB (JDBC 2.0##)
DATE	DATE
DECIMAL	DECIMAL
DOUBLE	DOUBLE PRECISION
FLOAT	DOUBLE PRECISION[2]
INTEGER	INTEGER

java.sql.Types	SQL#
LONGVARBINARY	LONG VARCHAR FOR BIT DATA
LONGVARCHAR	LONG VARCHAR
NULL	##################
NUMERIC	DECIMAL
REAL	REAL
SMALLINT	SMALLINT
SQLXML[3]	XML
TIME	TIME
TIMESTAMP	TIMESTAMP
VARBINARY	VARCHAR FOR BIT DATA
VARCHAR	VARCHAR

Notes:

1. BIT#JDBC 2.0##############
2. FLOAT#################DOUBLE PRECISION#####################DOUBLE########
3. SQLXML#JDBC 4.0#############Derby###SQLXML#SQL#XML######## ####Derby#java.sql.Ty ###XML#####

java.sql.Blob#########java.sql.Clob###########

JDBC 2.0##*java.sql.Blob*#SQL#BLOB(binary large object)#######*java.sql.Clob*#SQL#CLOB(character large object)########

java.sql.Blob#java.sql.Clob###(large object)##########################Derby#################################BLOB##############

###Derby###########BLOB#CLOB##########
##Derby#########################

- **BLOB###**Derby##*java.sql.Blob*#########*java.sql.PreparedStatement########: JDBC2.0##########*###*java.sql.ResultSet*#BLOB################## *CallableStatement#getBlob*###############
- **CLOB###**Derby##java.sql.Clob#########*java.sql.PreparedStatement########: JDBC2.0##########*#*java.sql.ResultSet*#CLOB################## *CallableStatement#getClob*###############

java.sql.Blob#java.sql.Clob#################

- ###SQL#BLOB########LONG VARCHAR FOR BIT DATA#BINARY###VARCHAR FOR BIT DATA########
- ###SQL#CLOB########LONG VARCHAR#CHAR###VARCHAR########
- *java.sql.ResultSet#########getBlob####getClob####################BLOB#CLOB#######*
- LOB#########################

#######BLOB###

Derby#########(1##2###)##################ASCII##(1#####1###)############################ ###########CLOB###########

BLOB#CLOB#(LOB#)###

- LOB####(=)####(!=# <>. #########
- LOB##################<# <=# >=# >=############
- LOB######LOB#############

- LOB######DISTINCT#GROUP BY#ORDER BY##########
- #########LOB#################

Derby###*CallableStatement*#set###get############JDBC 2.0#########################

####:java.sql.Blob#java.sql.Clob################################java.sql.Blob###java.sql.Clob####

 # 92. #####JDBC 2.0 java.sql.Blob#####

###	*#####*	*#####*
InputStrea	*getBinaryStream()*	'
byte[]	*getBytes(long pos, int length)*	*pos* < 1#####*pos*#length#########*length* <= 0################
long	*length()*	'
long	*position(byte[] pattern, long start)*	*pattern* == null######*start* < 1######*pattern*#####0################
long	*position(Blob pattern, long start)*	*pattern* == null#####*start* < 1#####*pattern*####0######*pattern*##########################

 # 93. #####JDBC 2.0 java.sql.Clob #####

###	*#####*	*######*
InputStrear	*getAsciiStream()*	'
Reader	*getCharacterStream()*	'
String	*getSubString(long pos, int length)*	*pos* < 1#####*pos*# *Clob*############length <= 0################
long	*length()*	'
long	*position(Clob searchstr, long start)*	*searchStr* == null####*start* < 1#####*searchStr* ####0#####*searchStr*################
long	*position(String searchstr, long start)*	*searchStr* == null#####start < 1#####pattern###################

java.sql.Blob#java.sql.Clob###########:

Derby######(#####)###*java.sql.Blob#java.sql.Clob*############################## *#######*Derby###########*java.sql.Blob#java.sql.Clob*############################## *####getBlob/getClob###java.sql.Blob/java.sql.Clob*#########################

###java.sql.Blob/ java.sql.Clob##############(###############)#########################(#####getXXXStream##### *java.sql.Blob/ java.sql.Clob*###

ResultSet getXXX###

- *getBlob*
- *getClob*
- *getAsciiStream*
- *getBinaryStream*

- *getUnicodeStream*

##
#######

```
ResultSet rs = s.executeQuery("SELECT text FROM CLOBS WHERE i = 1");
while (rs.next()) {
    aclob=rs.getClob(1);
    ip = rs.getAsciiStream(1);
}
```

##

Clob#########################

JDBC 2.0

#########Derby##########JDBC2.0###################

java.sql.Connection ########: JDBC 2.0##########

 # 94. Connecction#######JDBC 2.0#####

###	#####
Statement	*createStatement(int resultSetType, int resultSetConcurrency)*
PreparedStatement	*prepareStatement(String sql, int resultSetType, int resultSetConcurrency)*
CallableStatement	*prepareCall(String sql, int resultSetType, int resultSetConcurrency*

##########
###############ResultSet.TYPE_FORWARD_ONLY#ResultSet.TYPE_SCROLL_INSENSITIVE#######
TYPE_SCROLL_SENSITIVE########Derby#SQLWarning######TYPE_SCROLL_INSENSITIVE#Resul

###########ResultSet.CONCUR_READ_ONLY#ResultSet.CONCUR_UPDATABLE################

java.sql.DatabaseMetaData ########: #####JDBC 2.0#####

Derby#################JDBC 2.0#################

java.sql.PreparedStatement########: JDBC2.0##########

 # 95. java.sql.PreparedStatement######JDBC 2.0#####

###	#####	#######
void	*addBatch()*	'
ResultSetMetaData	*getMetaData()*	'
void	*setBlob(int i, Blob x)*	'
void	*setClob(int i, Clob x)*	'

java.sql.ResultSet

 # 96. ResultSet#######JDBC 2.0#####

###	#####	#######
boolean	*absolute(int row)*	'
void	*afterLast()*	'
void	*beforeFirst()*	'
void	*beforeFirst()*	'
void	*deleteRow()*	#######ResultSet###############ResultSet#close######
boolean	*first()*	'
Blob	*getBlob(int columnIndex)*	java.sql.Blob#########java.sql.Clob##########
Blob	*getBlob(String column-Name)*	
Clob	*getClob(int columnIndex)*	
Clob	*getClob(String column-Name)*	
int	*getConcurrency()*	Statement######*CONCUR_READ_ONLY*#########R ######Statement######*CONCUR_UPDATABLE*########
int	*getFetchDirection()*	'
int	*getFetchSize()*	Always returns 1.
int	*getRow()*	'
void	*insertRow()*	'
boolean	*isAfterLast()*	'
boolean	*isBeforeFirst*	'
boolean	*isFirst()*	'
boolean	*isLast()*	'
boolean	*last()*	'
void	*moveToCurrentRow()*	'
void	*moveToInsertRow()*	'
boolean	*previous()*	'
boolean	*rowDeleted()*	##################################*false*###############
boolean	*rowInserted()*	*##false######
boolean	*rowUpdated()*	##################################*false*###############
boolean	*relative(int rows)*	'
void	*setFetchDirection(int direction)*	'
void	*setFetchSize(int rows)*	A fetch size of 1 is the only size supported.
void	*updateRow()*	#########ResultSet################ ResultSet#close###

java.sql.ResultSetMetaData ########: JDBC 2.0##########

Derby#################JDBC 2.0#################

java.sql.Statement ########: #####JDBC 2.0#####

 # 97. java.sql.Statement######JDBC2.0#####

###	#####	#######
void	*addBatch(String sql)*	'
void	*clearBatch()*	'
int[]	*executeBatch()*	'
int	*getFetchDirection()*	########################
int	*getFetchSize()*	########################
int	*getMaxFieldSize()*	'
void	*getMaxRows()*	'
void	*setEscapeProcessing(boolean enable)*	'
void	*setFetchDirection(int direction)*	########################
void	*setFetchSize(int rows)*	########################
void	*setMaxFieldSize(int max)*	*Blobs#Clobs#######*
void	*setMaxRows()*	'

java.sql.BatchUpdateException

######################

Connected Device Configuration###Foundation Profile###JDBC#####(JSR169)

Derby##JSR169#####Connected Device Configuration###Foundation Profile########JDBC API###############JDBC 3.0###############JSR169###################D

JSR169##################`org.apache.derby.jdbc.EmbeddedSimpleDataSource`##########
`org.apache.derby.jdbc.EmbeddedDataSource`################*Derby* ######## ################

####Derby##JSR169#######################:

- #########DECIMAL##################`getString()`#`setString()`#####getXXX###setXXX
  2.0#JDBC 3.0##DECIMAL##################JSR169############
- `CONTAINS SQL`#`READS SQL DATA`#`MODIFIES SQL DATA`#######JDBC#####Java###############JSR169##########
- ############API(`jdbc:default:connection`)#JSR169##########
  #####`jdbc:default:connection`####################################
- ############
- ############
- ######################
- DriverManager#########DriverManager.getConnection()############

JDBC 3.0###

JDBC 3.0####API#######################Derby###################
Note: ##########Java2#1.4##############

###########

- DatabaseMetaData###############java.sql.DatabaseMetaData ########: JDBC 3.0###########################
- ###########java.sql.ParameterMetaData########:JDBC3.0###########java.sql.PreparedStatem
- ##############java.sql.Statement#########java.sql.DatabaseMetaData ########: JDBC 3.0#########################
- ########java.sql.Connection########: JDBC3.0#########################
- HOLD#####java.sql.DatabaseMetaData ########: JDBC 3.0###########################

java.sql.Connection########: JDBC3.0############

 # 98. Connection#######JDBC 3.0#####

###	#####	#######
Savepoi	*setSavepoint (String name)*	##
Savepoi	*setSavepoint ()*	##
void	*releaseSavepoint (Savepoint savepoint)*	#####################################
void	*rollback(Savepoint savepoint)*	############################
Prepare	*prepareStatement(String sql, int autoGeneratedKeys)*	*sql*#INSERT#######*autoGeneratedKeys*##########
Prepare	*prepareStatement(String sql, int [] columnIndexes)*	*sql*#INSERT#######columnIndexes################
Prepare	*prepareStatement(String sql, String [] columnNames)*	*sql*#INSERT#######columnNames##################

- **Autogenerated keys**

java.sql.DatabaseMetaData ########: JDBC 3.0###########

 # 99. DatabaseMetaData######JDBC 3.0#####

###	#####	#######
boolean	*supportsSavepoints()*	'
int	*getDatabaseMajorVersion()*	'
int	*getDatabaseMinorVersion()*	'
int	*getJDBCMajorVersion()*	'
int	*getJDBCMinorVersion()*	'
int	*getSQLStateType()*	'
boolean	*supportsNamedParameters()*	'
boolean	*supportsMultipleOpenResults()*	'
boolean	*supportsGetGeneratedKeys()*	'
boolean	*supportsResultSetHoldability(int holdability)*	'
int	*getResultSetHoldability()*	*ResultSet.HOLD_CURSORS_OVER_COMMIT####

java.sql.ParameterMetaData########:JDBC3.0##########

ParameterMetaData#JDBC 3.0###*PreparedStatement.getParameterMetaData* #####*PreparedStatement*#####################ParameterMetaData###############*java.sql.Prepa JDBC3.0##########*##########

ParameterMetaData#####################
 # 100. ParameterMetaData#JDBC 3.0##########

###	#####	#######
int	*getParameterCount()*	'
int	*isNullable(int param)*	'
boolean	*isSigned(int param)*	'
int	*getPrecision(int param)*	'
int	*getScale(int param)*	'
int	*getParameterType(int param)*	'
String	*getParamterTypeName (int param)*	'
String	*getParamterClassName (int param)*	'
int	*getParameterMode (int param)*	'

java.sql.PreparedStatement########: JDBC3.0##########

PreparedStatement.getParameterMetaData#####*PreparedStatement*###############*ParameterMetaDa*
 # 101. PreparedStatement######JDBC 3.0#####

###	#####	#######
ParameterMetaData	*getParameterMetaData()*	'

java.sql.Savepoint

Savepoint#########JDBC 3.0###

###################

JDBC 3.0#API###*Connection.setSavepoint*###
Connection.rollback###*java.sql.Connection########: JDBC3.0##########*##########

######################svpt1###svpt1#############

```
conn.setAutoCommit(false); // ################################
Statement stmt = conn.createStatement();
int rows = stmt.executeUpdate("INSERT INTO TABLE1 (COL1) VALUES(1)");
// ##########
set savepoint
```

```
Savepoint svpt1 = conn.setSavepoint("S1");
rows = stmt.executeUpdate("INSERT INTO TABLE1 (COL1) VALUES (2)");
...
conn.rollback(svpt1);
...
conn.commit();
```

##########

Connection.releaseSavepoint###

##

##

###########

###*executeBatch*##########################

############################(######################)############(###################)#######

############

Derby###############################

Derby#####################################

 # 102. JDBC 3.0##################

###	#####	#######
int	*getSavepointId()*	############################### ################*SQLException*#######
String	*getSavepointName()*	##################### ################*SQLException*#######

java.sql.Statement########: JDBC 3.0##########

 # 103. JDBC 3.0######Statement#####

###	#####	#######
ResultSet	*getGeneratedKeys()*	##############################IDENTITY_VAL_LO
boolean	*execute(String sql, int autoGeneratedKeys)*	*sql*#INSERT########*autoGeneratedKeys*##########
boolean	*execute(String sql, int [] columnIndexes)*	*sql*#INSERT########columnIndexes###############
boolean	*execute(String sql, String [] columnNames)*	*sql*#INSERT########columnNames################
int	*executeUpdate(String sql, int autoGeneratedKeys)*	*sql*#INSERT########*autoGeneratedKeys*##########
int	*executeUpdate(String sql, int [] columnIndexes)*	*sql*#INSERT########columnIndexes######### #############INSERT#########################
int	*executeUpdate(String sql, String [] columnNames)*	*sql*#INSERT########columnNames################

Autogenerated keys

JDBC 3.0##
Derby###
JDBC
3.0##*Statement.getGeneratedKeys*###*Result*
##*getGeneratedKeys*######*ResultSet*########*ResultSet.getMetaData*#########IDENTITY_VAL_LOCA

####*Connection.prepareStatement*######*Statement.execute*#####*Statement.executeUpdate*#########

- ############################`Statement.RETURN_GENERATED_KEYS`###
- ##Derby
  #########(########Derby####################)
- ##Derby#########(#####Derby########

#

###########TABLE1############

```
CREATE TABLE TABLE1 (C11 int, C12 int GENERATED ALWAYS AS IDENTITY)
```

#########################TABLE1######C12 ######*ResultSet*#######

1:

```
Statement stmt = conn.createStatement();
stmt.execute(
    "INSERT INTO TABLE1 (C11) VALUES (1)",
    Statement.RETURN_GENERATED_KEYS);
ResultSet rs = stmt.getGeneratedKeys();
```

2:

```
Statement stmt = conn.createStatement();
String [] colNames = new String [] { "C12" };
stmt.execute(
    "INSERT INTO TABLE1 (C11) VALUES (1)",
    colNames);
ResultSet rs = stmt.getGeneratedKeys();
```

3:

```
Statement stmt = conn.createStatement();
int [] colIndexes = new int [] { 2 };
stmt.execute(
    "INSERT INTO TABLE1 (C11) VALUES (1)",
    colIndexes);
ResultSet rs = stmt.getGeneratedKeys();
```

##############################
Statement.getGeneratedKeys#######null#*ResultSet*######

JDBC 4.0###

JDBC 4.0 ####API################## #####Derby##################

Note: ########JDK 1.6###########

################

- **DataSources** JDBC
  4.0###################Derby######`javax.sql.DataSource`############*javax.sql.DataSou*
  ########: JDBC 4.0###########

- **JDBC##########**
  ########JDBC##
  JDBC
  4.0#########################Class.forName()#####################################Driv
- **SQLExceptions.** JDBC 4.0
  #########SQLException#####################SQLException################
- **####** JDBC
  4.0########JDBC###Connection#
  Derby#################Derby####################
- **######** JDBC
  4.0###*javax.sql.PooledConnection*######
- **Streaming API** JDBC
  4.0##CallableStatement#PreparedStatement#ResultSet#################################
- **#######** ############################ `javax.sql.Connection`#
  `javax.sql.DatabaseMetaData`
  ###`javax.sql.Statement`java.sql.Connection########:JDBC4.0#########java.sql.DatabaseM
  JDBC4.0#######java.sql.Statement########: JDBC 4.0#########

#####SQLException######

#########JDK 1.6############## Derby
##############JDBC4.0#######SQLException#########################
########################javadoc##########

- **java.sql.SQLClientInfoException**
- **java.sql.SQLDataException**
- **java.sql.SQLFeatureNotSupportedException**
- **java.sql.SQLIntegrityConstraintViolationException**
- **java.sql.SQLInvalidAuthorizationSpecException**
- **java.sql.SQLSyntaxErrorException**
- **java.sql.SQLTransactionRollbackException**
- **java.sql.SQLTransientConnectionException**

java.sql.Connection########:JDBC4.0########

JDBC 4.0 ##################

- **LOB###** -
  ########*createBlob()*#*createClob()*###############Blob#Clob#############################
- ***#####*** - *isValid*#######Connection####################

java.sql.DatabaseMetaData########: JDBC4.0###

Derby###JDBC 4.0##############################

- ***############*** -
  JDBC4.0###
  autoCommitFailureClosesAllResultSets# providesQueryObjectGenerator#
  getClientInfoProperties### supportsStoredFunctionsUsingCallSyntax###
- ***######*** -
  ###############*getColumns*################*IS_AUTOINCREMENT*#YES###
- ***#######*** - JDBC
  4.0###
  ############*getFunctions*###*getFunctionColumns*###
  ###############*getProcedures*###*getProcedureColumns*######
- ***########*** -
  getProcedureColumns###javado
  COLUMN_DEF# SQL_DATA_TYPE# SQL_DATETIME_SUB#

CHAR_OCTET_LENGTH# ORDINAL_POSITION#
IS_NULLABLE###SPECIFIC_NAME###

- ********* - JDBC 4.0*##getSchemas*###

java.sql.Statement########: JDBC 4.0########

Derby#Statement###JDBC 4.0###################################

- ******** - JDBC 4.0##
  ###isPoolable#setPoolable###
- ****** - JDBC 4.0*##isClosed*###############################

javax.sql.DataSource ########: JDBC 4.0###

Derby###JDBC 4.0###DataSource###############JDK 1.6##################DataSource#########

- **org.apache.derby.jdbc.EmbeddedConnectionPoolDataSource40**
- **org.apache.derby.jdbc.EmbeddedDataSource40**
- **org.apache.derby.jdbc.EmbeddedXADataSource40**
- **org.apache.derby.jdbc.ClientConnectionPoolDataSource40**
- **org.apache.derby.jdbc.ClientDataSource40**
- **org.apache.derby.jdbc.ClientXADataSource40**

JDBC########

JDBC##DBMS#######SQL## ##########JDB

JDBC##

```
{##### }
```

Derby#####JDBC#####################################

- *######JDBC##########*

  #############CallableStatements#########
- *JDBC#######*

  ##########################
- *LIKE##JDBC#######*

  *#############*LIKE*##########################*
- *fn#########JDBC#######*

  ############################
- *#####JDBC#######*

  ##########################
- *#####JDBC#######*

  ##########################
- *##########JDBC#######*

  ##############################

#####JDBC###################
Note:
Derby*##Connection.nativeSQL*################SQL###################SQL################

######JDBC##########

#####CallableStatement####java.sql.Statement#java.sql.PreparedStatement #########

##

```
{call statement }
```

```
-- Java##########
{ call TOURS.BOOK_TOUR(?, ?) }
```

JDBC#######

Derby####JDBC############SQL############

##

```
{d 'yyyy-mm-dd'}
```

#####

```
DATE('yyyy-mm-dd')
```

```
VALUES {d '1999-01-09'}
```

LIKE##JDBC#######

SQL#LIKE###%(#######)#_(#######)############JDBC##################################LIKE

##

```
WHERE ### [ NOT ]
 LIKE
    ###########
    { ESCAPE '#######' }
```

```
-- "%"###################
SELECT a FROM tabA WHERE a LIKE '$%%' {escape '$'}
-- "_"###################
SELECT a FROM tabA WHERE a LIKE '%=_' {escape '='}
```

Note: LIKE###############?#######?####################

##############################(###########16###########)##############################

JDBC######################LIKE##########################

fn#########JDBC#######

JDBC#########fn########################

##

```
{fn ######}
```

#################################

abs

#########

```
abs(##)
```

JDBC########{fn abs(##)}########ABSOLUTE(##)######
#########ABS#ABSVAL##########

acos

###############

```
acos(#)
```

JDBC########{fn acos(#)}########ACOS(#)################ACOS##########

asin

###############

```
asin(#)
```

JDBC########{fn asin(#)}########ASIN(#)################ASIN##########

atan

###############

```
atan(#)
```

The JDBC########{fn
atan(#)}########ATAN(#)################ATAN##########

ceiling

##############################

```
ceiling(#)
```

JDBC########{fn
ceiling(#)}########CEILING(#)###############CEIL###CEILING##########

concat

#############

```
concat(###, ###)
```

###NULL######JDBC########{fn
concat (###, ###)########{ ### || ### }##########Concatenation##########

cos

###############

```
cos(#)
```

JDBC########{fn cos(#)}########COS(#)##########COS ##########

degrees

####################

```
degrees(#)
```

JDBC########{fn degrees(#)}########DEGREES(#)#########DEGREES
##########

exp

e#################

```
exp(#)
```

JDBC########{fn exp(#)}#########EXP(#)#########EXP##########

floor

##############################

```
floor(#)
```

JDBC########{fn floor(#)}########FLOOR(#)###### ###FLOOR##########

locate

##########
##########

```
locate(###,### [, ####] )
```

JDBC########{fn locate(###,### [, ####]
)}########LOCATE(CharacterExpression, CharacterExpression [, StartPosition]
)#########LOCATE##########

log

##########(##e#####)######

```
log(#)
```

JDBC########{fn log(#)}########LOG(#)#########LN####LOG##########

log10

######10##############

```
log10(#)
```

JDBC########{fn log10(#)}#########LOG10(#)#########LOG10 ##########

mod

##################(##)###### ############################

```
mod(##, ##)
```

###MOD##########

pi

pi###########

```
pi()
```

JDBC########{fn pi()}########PI()#########PI ##########

radians

####################

```
radians(#)
```

JDBC########{fn radians(#)}########RADIANS(#)#########RADIANS
##########

sin

##############

```
sin(#)
```

JDBC########{fn sin(#)}########SIN(#)#########SIN##########

sqrt

################

```
sqrt(#######)
```

JDBC########{fn sqrt
(#######)}########SQRT(#######)#########SQRT###########

substring

############################ ##########1######

```
substring(###, ####, ##)
```

tan

###########

```
tan(#)
```

JDBC########{fn tan(#)}########TAN(#)#########TAN##########

TIMESTAMPADD

TIMESTAMPADD###

TIMESTAMPADD#JDBC##########JDBC####################

```
TIMESTAMPADD( #####, ###, ######## )
```

######TIMESTAMPADD###00:00:00.0########

WHERE###

TIMESTAMPDIFF

TIMESTAMPDIFF###

TIMESTAMPDIFF#JDBC##########JDBC####################

```
TIMESTAMPDIFF( #####, ########1, ########2 )
```

######TIMESTAMPDIFF###00:00:00.0#######

WHERE###

TIMESTAMPADD ### TIMESTAMPDIFF###########

TIMESTAMPADD###TIMESTAMPDIFF###

- SQL_TSI_DAY
- SQL_TSI_FRAC_SECOND
- SQL_TSI_HOUR
- SQL_TSI_MINUTE
- SQL_TSI_MONTH
- SQL_TSI_QUARTER
- SQL_TSI_SECOND
- SQL_TSI_WEEK
- SQL_TSI_YEAR

########TIMESTAMPADD#TIMESTAMPDIFF######

########################

```
{fn TIMESTAMPADD( SQL_TSI_MONTH, 1, CURRENT_TIMESTAMP)}
```

##########2008#1#1###############

```
{fn TIMESTAMPDIFF(SQL_TSI_WEEK, CURRENT_TIMESTAMP,
  timestamp('2008-01-01-12.00.00.000000'))}
```

#####JDBC#######

Derby######(#######)#JDBC################################SQL##########

###########JOIN ##########

##

```
{oj #### [#### ]* }
```

####

```
#### [#### ]*
```

```
-- ####
SELECT *
FROM
{oj Countries LEFT OUTER JOIN Cities ON
   (Countries.country_ISO_code=Cities.country_ISO_code)}
-- ######
SELECT *
FROM
{oj Countries JOIN Cities ON
 (Countries.country_ISO_code=Cities.country_ISO_code)}
-- ################
-- FROM#################
SELECT E.EMPNO, E.LASTNAME, M.EMPNO, M.LASTNAME
FROM {oj EMPLOYEE E INNER JOIN DEPARTMENT
INNER JOIN EMPLOYEE M ON MGRNO = M.EMPNO ON E.WORKDEPT = DEPTNO};
```

#####JDBC#######

Derby####JDBC###############SQL#########Derby##8##(6#######2#)#ISO##########

##

```
{t 'hh:mm:ss'}
```

#######

```
TIME 'hh:mm:ss'
```

#

```
VALUES {t '20:00:03'}
```

#####JDBC#######

Derby####JDBC###############SQL#########

##

```
{d 'yyyy-mm-dd'}
```

#####

```
DATE 'yyyy-mm-dd'
```

#

```
VALUES {d '1995-12-19'}
```

##########JDBC#######

Derby#########JDBC####################SQL#########
###Derby##23##(17#######3######3#)#ISO###############

##

```
{ts 'yyyy-mm-dd hh:mm:ss.f...'}
```

#####

```
TIMESTAMP 'yyyy-mm-dd hh:mm:ss.f...'
```

###################(.f...)########

```
VALUES {ts '1999-01-09 20:11:11.123455'}
```

########URL#########

Derby##########URL######### JDBC###########

###Derby######

###Derby#########URL###########

Note: ##

bootPassword=key##

##
##################
- #################
- ########################
- ##################

8######################

############

##################bootPassword=key####create=true
####dataEncryption=true####################

######################bootPassword=key###dataEncryption=true################
##################SQL##########*Derby* ########
####################SQL#######################

##

#

```
-- ####################
jdbc:derby:newDB;create=true;dataEncryption=true;
    bootPassword=cseveryPlace
-- ############################
jdbc:derby:salesdb;dataEncryption=true;bootPassword=cseveryPlace
-- ####################
jdbc:derby:encryptedDB;bootPassword=cseveryPlace
```

collation=collation

##

collation###
collation#########TERRITORY_BASED#UCS_BASIC###

####:
collation###

Derby#################collation###TERRITORY_BASED##########territory##############

##############territory############Derby#java.util.Locale.getDefault########Java#

Note:
collation###

#

#####MexicanDB###########URL######territory#####################################collat

```
jdbc:derby:MexicanDB;create=true;territory=es_MX;collation=TERRITORY_BASED
```

Derby#################*Derby* ######## #####Derby####################

create=true

##

Derby#########URL###

###################################*SQLWarning*########

JDBC###
##########URL#*create=true*###
###

#########

#####################################(*user=userName* ##########
###SQL#############(*Derby* ########
###################SQL#####################)###
##APP#####################(SET
SCHEMA ##########)

############

(########URL###########)*databaseName*#######*databaseName=nameofDatabase*##############

###*territory=ll_CC*############
Note: *create=true*##########################*SQLWarning*#######

```
jdbc:derby:sampleDB;create=true

jdbc:derby:;databaseName=newDB;create=true;
```

createFrom=Path##

##

######URL#*createFrom=path*###
##derby.system.home## ##derby.system.home###

########################logDevice###*createFrom=path*###*cr*

#####################*Derby* ##

###########

#####*rollforwardrecoveryFrom#restoreFrom#create*##################

```
URL: jdbc:derby:wombat;createFrom=d:/backup/wombat
```

databaseName=nameofDatabase##

##

###

######URL(###Properties######)###########

- *jdbc:derby:toursDB*
- *jdbc:derby:;databaseName=toursDB*
- *jdbc:derby:*(########*Properties*#########*databaseName*#####*toursDB*######)

##URL###URL#*tour*

```
jdbc:derby:toursDB;databaseName=flightsDB
```

############################Derby###*getPropertyInfo*###
############*java.sql.Driver.getPropertyInfo* ####################

############

############################

```
jdbc:derby:;databaseName=newDB;create=true
```

dataEncryption=true##

##

###Derby
######## ###################################

############

dataEncryption####*bootPassword=key*####newEncryptionKey=key#####################
#####encryptionProvider=providerName#encryptionAlgorithm=algorithm########

######SQL###*#########*##############*Derby*
*######## ###################*SQL#######################

#

```
 -- ################
jdbc:derby:encryptedDB;create=true;dataEncryption=true;
    bootPassword=cLo4u922sc23aPe
 -- ##########################
   jdbc:derby:salesdb;dataEncryption=true;bootPassword=cLo4u922sc23aPe
```

encryptionKey=key##

##
####################
- ############
- ##########################
- ###################

############################

############

##############encryptionKey#create=true#dataEncryption=true#################

##############################encryptionKey###dataEncryption=true#################
###SQL##########################################*####################Derby*
*######## ###################*SQL#######################

###encryptionAlgorithm########

Derby################DES/CBC/NoPadding###

#

####################JDBC URL##:

```
jdbc:derby:newDB;create=true;dataEncryption=true;
   encryptionAlgorithm=DES/CBC/NoPadding;encryptionKey=6162636465666768
```

############################JDBC URL##:

```
jdbc:derby:salesdb;dataEncryption=true;encryptionKey=6162636465666768
```

#################JDBC URL##:

```
jdbc:derby:encryptedDB;encryptionKey=6162636465666768
```

encryptionProvider=providerName##

##

################################Derby ########
################################

##################JVM#####################################

##########

encryptionProvider###bootPassword=key#dataEncryption=true#####################
##encryptionAlgorithm=algorithm ##################

###SQL################################
##########################Derby ########
####################SQL#########################

#

```
-- ######################
jdbc:derby:encryptedDB;create=true;dataEncryption=true;
    encryptionProvider=com.sun.crypto.provider.SunJCE;
    encryptionAlgorithm=DESede/CBC/NoPadding;
    bootPassword=cLo4u922sc23aPe
-- ####################
    jdbc:derby:salesdb;dataEncryption=true;
    encryptionProvider=com.sun.crypto.provider.SunJCE;
    encryptionAlgorithm=DESede/CBC/NoPadding;
    bootPassword=cLo4u922sc23aPe
```

encryptionAlgorithm=algorithm

##

################

Java############################

algorithmName/feedbackMode/padding

Derby##padding####*NoPadding*#########

##########################DES/CBC/NoPadding######

############Derby ######## ################################

##########

encryptionAlgorithm####bootPassword=key###dataEncryption=true
####################
####encryptionProvider=providerName##

#####SQL###################*#########*################################ *#########Derby*
*######## ####################*SQL#########################

#

```
 -- ####################
    jdbc:derby:encryptedDB;create=true;dataEncryption=true;
    encryptionProvider=com.sun.crypto.provider.SunJCE;
    encryptionAlgorithm=DESede/CBC/NoPadding;
    bootPassword=cLo4u922sc23aPe
-- ##################
```

```
jdbc:derby:salesdb;dataEncryption=true;
encryptionProvider=com.sun.crypto.provider.SunJCE;
encryptionAlgorithm=DESede/CBC/NoPadding;
bootPassword=cLo4u922sc23aPe
```

Note: ##################################Derby#########

logDevice=logDirectoryPath

##

logDirectoryPath###*logDirectoryPath*#########

####################Derby
##########logDevice=logDirectoryPath#################

##########

create# *createFrom*#*restoreFrom*####*rollForwardRecoveryFrom*########

```
jdbc:derby:newDB;create=true;logDevice=d:/newDBlog
```

newEncryptionKey=key

##

###
######## ###################################

##########

newEncryptionKey###*encryptionKey=key*#####################

newEncryptionKey#################################

*######################*SQL*##################*#########*#####################*
*#############Derby ######## ###############*SQL*####################*

#

```
-- ####################
jdbc:derby:salesdb;encryptionKey=6162636465666768;newEncryptionKey=6862636465666768
```

newBootPassword=newPassword

##

###
###############Derby
#######################################

##########

newBootPassword###*bootPassword=key*#####################

newBootPassword#################################

*#####*SQL*#################################*#########*####################*
*#########Derby ######## ####################*SQL*####################*

#

```
-- ######################
jdbc:derby:salesdb;bootPassword=abc1234xyz;newBootPassword=new1234xyz
```

password=userPassword

##

####################

##########

user=userName#############

```
jdbc:derby:toursDB;user=jack;password=upTheHill
```

restoreFrom=path##

##

######URL#####*restoreFrom=path*###################################derby.system.home######

##logDevice###*restoreFrom=path*################

####################*Derby* ##

##########

######*createFrom#rollforwardrecoveryFrom*#####*create*##############

```
URL: jdbc:derby:wombat;restoreFrom=d:/backup/wombat
```

rollForwardRecoveryFrom=path

##

####URL#
rollForwardRecoveryFrom=path###

###

###

###################*Derby* ###################################

##########

#####*createFrom#restoreFrom*###*create*##############

```
URL: jdbc:derby:wombat;rollForwardRecoveryFrom=d:/backup/wombat
```

shutdown=true##

##

databaseName##########################(#####################################)
###SQL##################*Derby*
######## ###################SQL#####################

databaseName##############Derby##############

##################Derby#################################

################Derby############################JDBC#############JVM#######JVM######
Derby#####################################Derby###############*Derby*
######## #####1#########################

###

##
Note: *###shutdown=true#####DriverManager###################*

```
-- ##########
jdbc:derby:;shutdown=true
-- salesDB######(##########)
jdbc:derby:salesDB;shutdown=true
```

territory=ll_CC

##

territory#########################

##########ll_CC######ll#2#########CC #2##########

#################ISO-639######
 # 104. #######

#####	##
de	German
en	English
es	Spanish
ja	Japanese

ISO-639############*http://www.ics.uci.edu/pub/ietf/http/related/iso639.txt*##########

#####2##########ISO-3166######
 # 105. ######

####	##
DE	Germany
US	United States
ES	Spain
MX	Mexico
JP	Japan

ISO-3166####*http://www.chemie.fu-berlin.de/diverse/doc/ISO_3166.html*######

##########

territory#########################

##

```
jdbc:derby:MexicanDB;create=true;territory=es_MX
```

collation
##territory###############Unicode##

traceDirectory=path

##

Derby#############JDBC###
traceFile=path################################

###############Derby###

#################*Derby* ## ####################*tra*

###########

##########################

#

```
-- #############################
jdbc:derby://localhost:1527/mydb;traceDirectory=/home/mydir/mydbtracedir
-- ######################
jdbc:derby://localhost:1527/mydb;traceDirectory=/home/mydir/
mydbtracedir;traceFile=trace.out
-- #################
jdbc:derby://localhost:1527/mydb;traceDirectory=/home/mydir/
mydbtracedir;traceFileAppend=true
```

traceFile=path

##

Derby#############JDBC###*derby.system.home*#
######## #################################)

traceFile=path#traceFileAppend=true##*traceFile=*

#################*Derby* ##
############## *traceDirectory=path# traceLevel=value*##########

###########

##########################

#

```
-- ####################
jdbc:derby://localhost:1527/mydb;create=true;traceFile=trace.out
```

traceFileAppend=true

##

Derby###############JDBC##################################*traceFile=path*###############

#################*Derby*
###*traceDirectory=path#traceLevel*

###########

######traceFile=path####traceDirectory=path###

#

```
-- ######################
-- ################
jdbc:derby://localhost:1527/mydb;traceFile=trace.out;traceFileAppend=true
-- #####################
-- Derby#################################
-- ##########
jdbc:derby://localhost:1527/
mydb;traceDirectory=mytracedir;traceFileAppend=true
```

traceLevel=value

##

################Derby##################################*value*##########################

##############*Derby* #######################################
#################### *traceFile=path*# *traceFileAppend=true*###
traceDirectory=path##########

########

#############################

 # 106. Available tracing levels and values

########	16##	10##
org.apache.derby.jdbc.ClientDataSource.TRACE_NONE	0x0	0
org.apache.derby.jdbc.ClientDataSource.TRACE_CONNECTION_CALL	0x1	1
org.apache.derby.jdbc.ClientDataSource.TRACE_STATEMENT_CALLS	0x2	2
org.apache.derby.jdbc.ClientDataSource.TRACE_RESULT_SET_CALL	0x4	4
org.apache.derby.jdbc.ClientDataSource.TRACE_DRIVER_CONFIGUR	0x10	16
org.apache.derby.jdbc.ClientDataSource.TRACE_CONNECTS	0x20	32
org.apache.derby.jdbc.ClientDataSource.TRACE_PROTOCOL_FLOWS	0x40	64
org.apache.derby.jdbc.ClientDataSource.TRACE_RESULT_SET_META	0x80	128
org.apache.derby.jdbc.ClientDataSource.TRACE_PARAMETER_META	0x100	256
org.apache.derby.jdbc.ClientDataSource.TRACE_DIAGNOSTICS	0x200	512
org.apache.derby.jdbc.ClientDataSource.TRACE_XA_CALLS	0x800	2048
org.apache.derby.jdbc.ClientDataSource.TRACE_ALL	0xFFF	-1

################################

- `ij`#########10##TRACE_PROTOC (64)#TRACE_CONNECTION_CALLS (1)############65#######
- JDBC##############################
  - ######OR###(|)##

    ```
    TRACE_PROTOCOL_FLOWS | TRACE_CONNECTION_CALLS
    ```

  - ############(~)##

    ```
    ~TRACE_PROTOCOL_FLOWS
    ```

###########

############*traceFile=path*######*traceDirectory=path*################

#

```
-- ####################
jdbc:derby://localhost:1527/
mydb;create=true;traceFile=trace.out;traceLevel=65
```

upgrade=true attribute

##

#######Derby#################################Derby###

#####Derby ########
##
##################Derby ######## ###################################

############SQL##################*##########*##############################
*########Derby ######## ###################*SQL######################

##########

(#########URL##############)###########*databaseName=nameofDatabase*#####################

collation#### *territory=ll_CC*#########################

```
jdbc:derby:sampleDB;upgrade=true
```

```
jdbc:derby:;databaseName=sampleDB;upgrade=true;
```

user=userName

##

##########

password=userPassword#################

##########URL###jill#######*toursDB*#######

```
jdbc:derby:toursDB;user=jill;password=toFetchAPail
```

ssl=sslMode

##

#######SSL##########
sslMode#basic#peerAuthentication####off(##)################## ###*Derby*
############*SSL/TLS*############################

##########

#########################

#

mydb#basic SSL##########

```
jdbc:derby://localhost/mydb;ssl=basic
```

####################

############*databaseName*##############

Derby###
##*SQLException*#########

```
jdbc:derby:mydb
```

J2EE####:Java Transaction API#javax.sql

J2EE#Java 2 Platform, Enterprise Edition###
J2EE##Java 2 Platform, Standard Edition (J2SE)#####Enterprise Java Beans (EJB)#Java Server Pages (JSP)########XML###############
J2EE###

Derby#######J2EE##########J2EE##############JNDI#####################################

J2EE##############*http://java.sun.com/javaee/reference/*
#######J2EE##############

J2EE############################J2EE#################################

- JNDI#####

  ##URL#############################JDBC###
- ###########

  #########((Derby)###############
  ######################################
  ############################/
  ##
  #####JDBC##########*javax.sql.ConnectionPoolDataSource#javax.sql.PooledConnection*#######
- XA #####

  XA######################
  ###2##################*javax.sql.XAxxx*#########*java.transaction.xa*######XA##########
  XA########*X/Open CAE Specification-Distributed Transaction Processing: The XA Specification*, X/Open Document No. XO/CAE/91/300##ISBN 1 872630 24 3##########
  JTA#API##java.transaction.xa###############(*javax.sql.XAConnection*,*javax.sql.XADataSource*,*jav*
  and *javax.transaction.xa.XAException*)###################

#####JDBC###
Note: ################################Derby###################*Derby* ######## #####6#####Derby#J2EE########################

JTA API

JTA API#
java.transaction.xa################################API#Derby##############

- *javax.transaction.xa.XAResource*
- *javax.transaction.xa.Xid*
- *javax.transaction.xa.XAException*

##########

#################

XAResource.prepare## ###############

XAConnection###########

###################XAConnection#####################################XAConnection#########
###################XAConnection####################XAConnection###############

###XAConnection#######################

Note: ####(######/###)####XA##########################XA DataSource#################

javax.sql:JDBC########

#####Derby#####JDBC#####################J2EE###########

##########################JDK#API#################*http://java.sun.com/javase/reference/api.jsp*#######

- *javax.sql.DataSource*

  ############################DataSource####################Java Naming and Directory (JNDI) API##########
- *javax.sql.ConnectionPoolDataSource# javax.sql.PooledConnection*

  ################/ ##

  Derby#`ConnectionPoolDataSource`###`PooledConnection`########################## ##
- *javax.sql.XAConnection*

  ###*XAConnection######XAResource###############Connection*#########################
- *javax.sql.XADataSource*

  XADataSource##########XAConnections######ConnectionPoolDataSource###

###Derby##*XADataSource#DataSource#ConnectionPoolDataSource*################# Derby#####################

- *setCreateDatabase(String create)*

  ##################################"create"###########
- *setShutdownDatabase(String shutdown)*

  ##"shutdown"###########

Note: ##################################

Derby API

Derby#*javadoc*###############API#################Javadoc#HTML##########

######API#############Derby###Derby####API####*java.sql*#Javadoc##JDBC API################ Derby####JDBC#########JDBC ################

######API###java class###JDBC######JDBC API###

####################

#######org.apache.derby.tools###########

- *org.apache.derby.tools.ij*

  SQL##############################/ ################################*Derby* ##########################

- *org.apache.derby.tools.sysinfo*

  ############################JVM#Derby########## *Derby* ##########################

- *org.apache.derby.tools.dblook*

  ####################DDL##########*Derby* ##########################

JDBC######

JDBC driver

######Derby#JDBC#######:

- *org.apache.derby.jdbc.EmbeddedDriver*

  #########Derby#############JDBC###########

- *org.apache.derby.jdbc.ClientDriver*

  #########Derby#################/##############

Derby ######## ##############

##########

*######javax.sql.DataSource######*API#####Derby####### ########*Derby* ######## ##############

##JDK1.5#######################################

######:

- *org.apache.derby.jdbc.EmbeddedDataSource* and *org.apache.derby.jdbc.EmbeddedDataSource40*
- *org.apache.derby.jdbc.EmbeddedConnectionPoolDataSource* and *org.apache.derby.jdbc.EmbeddedConnectionPoolDataSource40*
- *org.apache.derby.jdbc.EmbeddedXADataSource* and *org.apache.derby.jdbc.EmbeddedXADataSource40*

Client-server environment

- *org.apache.derby.jdbc.ClientDataSource* and *org.apache.derby.jdbc.ClientDataSource40*

- *org.apache.derby.jdbc.ClientConnectionPoolDataSource* and *org.apache.derby.jdbc.ClientConnectionPoolDataSource40*
- *org.apache.derby.jdbc.ClientXADataSource* and *org.apache.derby.jdbc.ClientXADataSource40*

##################

- *org.apache.derby.authentication.UserAuthenticator*
- ###########Derby######### ##*Derby* ######## #####7#########################

##########

##################

##	Derby#####(derby.territory)
###(###)	zh_CN
###(###)	zh_TW
####	cs
#####	fr
####	de_DE
######	hu
#####	it
###	ja_JP
###	ko_KR
######	pl
######(#####)	pt_BR
####	ru
#####	es

Derby#####

#####Derby#############

#############

 # 107. ############
######Derby##########################

#	##
######	1,012
########	5,000
#########	90
#######	32,767 ###########
SQL####################	#######
########	1,012
WHERE##HAVING########	#######
GROUP BY########	32,677
ORDER BY########	1,012
#########	#######
###################	#######
##################	#######
##############	#######
##################	#######
##############	#######
###############	#######
#########	#######
##########	16

DATE#TIME#TIMESTAMP###

#######Derby############################
Table 108. DATE#TIME#TIMESTAMP###

#	##
DATE####	0001-01-01
DATE####	9999-12-31
TIME####	00:00:00
TIME####	24:00:00
TIMESTAMP####	0001-01-01-00.00.00.000000

#	##
TIMESTAMP####	9999-12-31-23.59.59.999999

#########

 # 109. ########
######Derby####################

###	######
###	128
###	128
#####	128
#########	128
##########	128
#########	128
##########	128
#####	128
##########	128
###########	128
###########	128
############	128
######	128
############	128
####################	128

#####

Derby##############

 # 110. #####

#	##
INTEGER####	-2,147,483,648
INTEGER####	2,147,483,647
BIGINT####	-9,223,372,036,854,775,808
BIGINT####	9,223,372,036,854,775,807
SMALLINT####	-32,768
SMALLINT####	32,767
decimal#######	31
DOUBLE####	-1.79769E+308

#	##
DOUBLE####	1.79769E+308
##DOUBLE####	2.225E-307
##DOUBLE####	-2.225E-307
REAL####	-3.402E+38
REAL####	3.402E+38
##REAL####	1.175E-37
##REAL####	-1.175E-37

#####

 # 111. ######
######Derby##################

#	##
CHAR###	254##
VARCHAR###	32,672##
LONG VARCHAR###	32,700##
CLOB###	2,147,483,647##
BLOB###	2,147,483,647##
#######	32,672
###########	2,147,483,647
#########	2,147,483,647
16######	16,336
DOUBLE######	30 characters

XML###

######Derby##XML##############

Table 112. XML###

##	##
XML###	2,147,483,647##
XML######	Apache Xerces####JAXP######Apache Xalan##########

Trademarks

The following terms are trademarks or registered trademarks of other companies and have been used in at least one of the documents in the Apache Derby documentation library:

Cloudscape, DB2, DB2 Universal Database, DRDA, and IBM are trademarks of International Business Machines Corporation in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.