This chapter teaches you how to create and drop procedures in Derby.

Creating a procedure

You can create a procedure using the CREATE PROCEDURE statement.

Syntax

Following is the syntax of the CREATE PROCEDURE statement.

CREATE PROCEDURE procedure\_name (parameter\_type parameter\_name1, parameter\_type

parameter\_name2 . . . .) parameter\_style;

Example

Suppose, we have created a table in Derby as shown below.

CREATE TABLE Emp ( Id INT NOT NULL GENERATED ALWAYS AS IDENTITY,

Name VARCHAR(255),

Salary INT NOT NULL,

Location VARCHAR(255),

Phone\_Number BIGINT

);

And inserted values in it as follows −

INSERT INTO Employees(Name, Salary, Location) VALUES

('Amit', 30000, 'Hyderabad'),

('Kalyan', 40000, 'Vishakhapatnam'),

('Renuka', 50000, 'Delhi'),

('Archana', 15000, 'Mumbai'),

('Trupthi', 45000, 'Kochin')";

Following example creates a procedure named Update\_Procedure which accepts JAVA parameters.

ij> CREATE PROCEDURE Update\_Procedure(IN id INTEGER, IN name VARCHAR(10))

PARAMETER STYLE JAVA READS SQL DATA LANGUAGE JAVA EXTERNAL NAME

'ProcedureExample.testProc';

> 0 rows inserted/updated/deleted

Where ProcedureExample class looks like −

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

public class ProcedureExample {

public static void testProc(int salary, String name) throws Exception {

String connectionURL = "jdbc:derby:MYDATABASE;create=true";

Connection conn = DriverManager.getConnection(connectionURL);

String query = "UPDATE Employees SET SALARY = ? WHERE NAME = ?";

PreparedStatement pstmt = conn.prepareStatement(query);

pstmt.setInt(1, salary);

pstmt.setString (2, name);

pstmt.executeUpdate();

}

}

You can verify the list of procedures using **SHOW PROCEDURES** query.

ij> SHOW PROCEDURES;

PROCEDURE\_SCHEM |PROCEDURE\_NAME |REMARKS

------------------------------------------------------------------------

APP |UPDATE\_PROCEDURE |ProcedureExample.te&

SALES |EXAMPLE\_ PROCEDURE |com.example.sales.c&

SQLJ |INSTALL\_JAR |org.apache.derby.ca&

SQLJ |REMOVE\_JAR |org.apache.derby.ca&

SQLJ |REPLACE\_JAR |org.apache.derby.ca&

SYSCS\_UTIL |SYSCS\_BACKUP\_DATABASE |org.apache.derby.ca&

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

Here you can observe the newly created procedure.

Dropping a procedure

You can drop a procedure using the DROP PROCEDURE statement.

Syntax

Following is the syntax of the DROP PROCEDURE statement.

DROP PROCEDURE procedure\_name;

Example

Following example drops a procedure named Update\_Procedure created above.

ij> DROP PROCEDURE Update\_Procedure;

> 0 rows inserted/updated/deleted