The UPDATE statement is used to update data in a table. Apache Derby provides two types of updates (syntax) namely **searched** update and **positioned** update.

The searched UPDATE statement updates all the specified columns of a table.

Syntax

Following is the syntax of the UPDATE query −

ij> UPDATE table\_name

SET column\_name = value, column\_name = value, ...

WHERE conditions;

The WHERE clause can use the comparison operators such as =, !=, <, >, <=, and >=, as well as the BETWEEN and LIKE operators.

Example

Suppose you have a table Employee in the database with the 4 records as shown below −

ID |NAME |SALARY |LOCATION

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

1 |Amit |30000 |Hyderabad

2 |Kalyan |40000 |Vishakhapatnam

3 |Renuka |50000 |Delhi

4 |Archana |15000 |Mumbai

The following SQL UPDATE statement updates the location and salary of an employee whose name is Kaylan.

ij> UPDATE Employees SET Location = 'Chennai', Salary = 43000 WHERE Name =

'Kalyan';

1 rows inserted/updated/deleted

If you get the contents of the Employees table, you can observe the changes done by the UPDATE query.

ij> select \* from Employees;

ID |NAME |SALARY |LOCATION

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

1 |Amit |30000 |Hyderabad

2 |Kalyan |43000 |Chennai

3 |Renuka |50000 |Delhi

4 |Archana |15000 |Mumbai

4 rows selected

Update Data using JDBC program

This section explains how to update the existing records of a table in the Apache Derby database using JDBC application.

If you want to request the Derby network server using network client, make sure that the server is up and running. The class name for the Network client driver is org.apache.derby.jdbc.ClientDriver and the URL is jdbc:derby://localhost:1527/**DATABASE\_NAME**;create=true;user=**USER\_NAME**;passw ord=**PASSWORD**"

Follow the steps given below to update the existing records of a table in Apache Derby.

Step 1: Register the driver

To communicate with the database, first of all, you need to register the driver. The **forName()** method of the class **Class** accepts a String value representing a class name loads it in to the memory, which automatically registers it. Register the driver using this method.

Step 2: Get the connection

In general, the first step we do to communicate to the database is to connect with it. The Connection class represents the physical connection with a database server. You can create a connection object by invoking the **getConnection()** method of the **DriverManager** class. Create a connection using this method.

Step 3: Create a statement object

You need to create a **Statement** or **PreparedStatement or, CallableStatement** objects to send SQL statements to the database. You can create these using the methods **createStatement()**, **prepareStatement()** and, **prepareCall()** respectively. Create either of these objects using the appropriate method.

Step 4: Execute the query

After creating a statement, you need to execute it. The **Statement** class provides various methods to execute a query like the **execute()** method to execute a statement that returns more than one result set. The **executeUpdate()** method executes queries like INSERT, UPDATE, DELETE. The **executeQuery()** method returns data. Use either of these methods and execute the statement created previously.

Example

Following JDBC example demonstrates how to update the existing records of a table in Apache Derby using JDBC program. Here, we are connecting to a database named sampleDB (will create if it does not exist) using the embedded driver.

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

import java.sql.Statement;

public class UpdateData {

public static void main(String args[]) throws Exception {

//Registering the driver

Class.forName("org.apache.derby.jdbc.EmbeddedDriver");

//Getting the Connection object

String URL = "jdbc:derby:sampleDB;create=true";

Connection conn = DriverManager.getConnection(URL);

//Creating the Statement object

Statement stmt = conn.createStatement();

//Creating a table and populating it

String query = "CREATE TABLE Employees("

+ "Id INT NOT NULL GENERATED ALWAYS AS IDENTITY, "

+ "Name VARCHAR(255), Salary INT NOT NULL, "

+ "Location VARCHAR(255), "

+ "PRIMARY KEY (Id))";

String query = "INSERT INTO Employees("

+ "Name, Salary, Location) VALUES "

+ "('Amit', 30000, 'Hyderabad'), "

+ "('Kalyan', 40000, 'Vishakhapatnam'), "

+ "('Renuka', 50000, 'Delhi'), "

+ "('Archana', 15000, 'Mumbai'), "

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

+ "('Suchatra', 33000, 'Pune'), "

+ "('Rahul', 39000, 'Lucknow'), "

+ "('Trupti', 45000, 'Kochin')";

//Executing the query

String query = "UPDATE Employees SET Location = 'Chennai', Salary = 43000 WHERE

Name = 'Kalyan'";

int num = stmt.executeUpdate(query);

System.out.println("Number of records updated are: "+num);

}

}

Output

On executing the above program, you will get the following output −

Number of records updated are: 1