The insert query inserts data: **new records**, into the table.

Syntax

Following is the basic syntax of the INSERT statement −

ij>INSERT INTO table\_name VALUES (column\_name1, column\_name2, ...);

where column1, column2 are the column values in the row that is to be inserted.

Example

The following SQL INSERT statement inserts a new row in the Student table, where it inserts values in the columns **id, age, first name** and, **last name**.

SQL> INSERT INTO Student VALUES (101, 20, 'Zara', 'Ali');

Syntax 2

Or, you can insert two specific columns by mentioning the column names, as given below −

ij>INSERT INTO table\_name VALUES (column\_name1, column\_name2, ...) VALUES

(value1, value2, ...);

**Note** − Apache Derby automatically calculates values for generated columns. For example, there is no need to pass values for the id column in the student table created earlier in this tutorial. In case your table has generated columns, use **syntax2**.

Example

ij> INSERT INTO Student(Age, First\_Name, Last\_Name) VALUES (21, 'Sucharitha' , 'Tyagi');

1 row inserted/updated/deleted

And, you can also insert two rows using one statement as follows −

ij>INSERT INTO Student(Age, First\_Name, Last\_Name) VALUES (20, 'Amit',

'Bhattacharya'), (22, 'Rahul', 'Desai');

2 rows inserted/updated/deleted

You can verify the contents of the table using the SELECT command (we will discuss this command later in this tutorial).

Syntax 3

You can use another query in the insert statement as −

INSERT INTO table\_Name Query

Example

Suppose, we have a table named **First\_Year** in the database as shown below with similar columns as in Student table −

ID |AGE |FIRST\_NAME |LAST\_NAME

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

1 |20 |Raju |Pendyala

2 |21 |Bhargav |Prayaga

3 |22 |Deepthi |Yerramilli

You can insert values in this table to the student table using the above syntax as −

ij> INSERT INTO Student (Age, First\_Name, Last\_Name)

SELECT Age, First\_Name, Last\_Name FROM First\_Year;

> 3 rows inserted/updated/deleted

After executing all the above insert statements, the Student table will be as follows −

ID |AGE |FIRST\_NAME |LAST\_NAME

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

1 |21 |Sucharitha |Tyagi

2 |20 |Amit |Bhattacharya

3 |22 |Rahul |Desai

4 |20 |Raju |Pendyala

5 |21 |Bhargav |Prayaga

6 |22 |Deepthi |Yerramilli

Insert Data using JDBC program

This section teaches you how to insert data in to a table in 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 insert data into 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 any 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 to results that returns data etc. Use either of these methods and execute the statement created previously.

Example

Following JDBC example demonstrates how to insert data into 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 InsertData {

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

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))";

//Executing the query

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')";

stmt.execute(query);

System.out.println("Values inserted");

}

}

Output

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

Values inserted