The SELECT statement is used to retrieve data from a table. This returns the data in the form of a table known as result set.

Syntax

Following is the syntax of the SELECT statement −

ij> SELECT column\_name, column\_name, ... FROM table\_name;

Or,

Ij>SELECT \* from table\_name

Example

Let us suppose we have a table named Employees in the database as shown below −

ij> CREATE TABLE Employees (

Id INT NOT NULL GENERATED ALWAYS AS IDENTITY,

Name VARCHAR(255),

Salary INT NOT NULL,

Location VARCHAR(255),

PRIMARY KEY (Id)

);

> > > > > > > 0 rows inserted/updated/deleted

And, inserted four records in it as shown below −

ij> INSERT INTO Employees (Name, Salary, Location) VALUES

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

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

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

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

> > > > 4 rows inserted/updated/deleted

The following SQL statement retrieves the name, age and salary details of all the employees in the table;

ij> SELECT Id, Name, Salary FROM Employees;

The output of this query is −

ID |NAME |SALARY

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

1 |Amit |30000

2 |Kalyan |40000

3 |Renuka |50000

4 |Archana |15000

4 rows selected

If you want to get all the records of this table at once, use \* instead of the names of the columns.

ij> select \* from Employees;

This will produce the following result −

ID |NAME |SALARY |LOCATION

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

1 |Amit |30000 |Hyderabad

2 |Kalyan |40000 |Vishakhapatnam

3 |Renuka |50000 |Delhi

4 |Archana |15000 |Mumbai

4 rows selected

Retrieve Data using JDBC program

This section teaches you how to Retrieve data from 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 Retrieve data from 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 to results that returns data etc. Use either of these methods and execute the statement created previously.

Example

Following JDBC example demonstrates how to Retrieve data from 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.

The **executeQuery()** method returns a **ResultSet** object which holds the result of the statement. Initially the result set pointer will be at the first record, you can print the contents of the ResultSet object using its **next()** and **getXXX()** methods.

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

public class RetrieveData {

public static void main(String args[]) throws SQLException,

ClassNotFoundException {

//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

4Statement 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 = "SELECT Id, Name, Salary FROM Employees";

ResultSet rs = stmt.executeQuery(query);

while(rs.next()) {

System.out.println("Id: "+rs.getString("Id"));

System.out.println("Name: "+rs.getString("Name"));

System.out.println("Salary: "+rs.getString("Salary"));

System.out.println(" ");

}

}

}

Output

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

Id: 1

Name: Amit

Salary: 30000

Id: 2

Name: Kalyan

Salary: 43000

Id: 3

Name: Renuka

Salary: 50000

Id: 4

Name: Archana

Salary: 15000

Id: 5

Name: Trupthi

Salary: 45000

Id: 6

Name: Suchatra

Salary: 33000

Id: 7

Name: Rahul

Salary: 39000