The WHERE clause is used in the SELECT, DELETE or, UPDATE statements to specify the rows on which the operation needs to be carried out. Usually, this clause is followed by a condition or expression which returns a Boolean value, the Select, delete or, update operations are performed only on the rows which satisfy the given condition.

ij> SELECT \* from table\_name WHERE condition;

or,

ij> DELETE from table\_name WHERE condition;

or,

ij> UPDATE table\_name SET column\_name = value WHERE condition;

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

Example

Let us assume we have a table named Employees in the database with 7 records as shown below −

ID |NAME |SALARY |LOCATION

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

1 |Amit |30000 |Hyderabad

2 |Kalyan |40000 |Vishakhapatnam

3 |Renuka |50000 |Delhi

4 |Archana |15000 |Mumbai

5 |Trupthi |45000 |Kochin

6 |Suchatra |33000 |Pune

7 |Rahul |39000 |Lucknow

The following SQL DELETE statement fetches the records of the employees whose salary is more than 35000 −

ij> SELECT \* FROM Employees WHERE Salary>35000;

This will produce the following output −

ID |NAME |SALARY |LOCATION

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

2 |Kalyan |40000 |Vishakhapatnam

3 |Renuka |50000 |Delhi

5 |Trupthi |45000 |Kochin

7 |Rahul |39000 |Lucknow

4 rows selected

Similarly, you can also delete and update records using this clause.

Following example updates the location of those whose salary is less than 30000.

ij> UPDATE Employees SET Location = 'Vijayawada' WHERE Salary<35000;

3 rows inserted/updated/deleted

If you verify the contents of the table, you can see the updated table as shown below −

ij> SELECT \* FROM Employees;

ID |NAME |SALARY |LOCATION

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

1 |Amit |30000 |Vijayawada

2 |Kalyan |40000 |Vishakhapatnam

3 |Renuka |50000 |Delhi

4 |Archana |15000 |Vijayawada

5 |Trupthi |45000 |Kochin

6 |Suchatra |33000 |Vijayawada

7 |Rahul |39000 |Lucknow

7 rows selected

Where clause JDBC example

This section teaches you how to use WHERE clause and perform CURD operations on 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 use WHERE clause and perform CURD operations on 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 results that returns data. Use either of these methods and execute the statement created previously.

Example

Following JDBC example demonstrates how to use WHERE clause and perform CURD operations on 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.Statement;

import java.sql.ResultSet;

public class WhereClauseExample {

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 = "SELECT \* FROM Employees WHERE Salary>35000";

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("Location: "+rs.getString("Location"));

System.out.println(" ");

}

}

}

Output

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

Id: 2

Name: Kalyan

Salary: 43000

Location: Chennai

Id: 3

Name: Renuka

Salary: 50000

Location: Delhi

Id: 5

Name: Trupthi

Salary: 45000

Location: Kochin

Id: 7

Name: Rahul

Salary: 39000

Location: Lucknow