HIVEQL - SELECT-ORDER BY

This chapter explains how to use the ORDER BY clause in a SELECT statement. The ORDER BY clause is used to retrieve the details based on one column and sort the result set by ascending or descending order.

Syntax

Given below is the syntax of the ORDER BY clause:

```
SELECT [ALL | DISTINCT] select_expr, select_expr, ...
FROM table_reference
[WHERE where_condition]
[GROUP BY col_list]
[HAVING having_condition]
[ORDER BY col_list]]
[LIMIT number];
```

Example

Let us take an example for SELECT...ORDER BY clause. Assume employee table as given below, with the fields named Id, Name, Salary, Designation, and Dept. Generate a query to retrieve the employee details in order by using Department name.

```
+------+--------------+-------------+-------------------+--------+
| ID   | Name         | Salary      | Designation       | Dept   |
+------+--------------+-------------+-------------------+--------+
|1201  | Gopal        | 45000       | Technical manager | TP     |
|1202  | Manisha      | 45000       | Proofreader       | PR     |
|1203  | Masthanvali  | 40000       | Technical writer  | TP     |
|1204  | Krian        | 40000       | Hr Admin          | HR     |
|1205  | Kranthi      | 30000       | Op Admin          | Admin  |
+------+--------------+-------------+-------------------+--------+
```

The following query retrieves the employee details using the above scenario:

```
hive> SELECT Id, Name, Dept FROM employee ORDER BY DEPT;
```

On successful execution of the query, you get to see the following response:

```
+------+--------------+-------------+-------------------+--------+
| ID   | Name         | Salary      | Designation       | Dept   |
+------+--------------+-------------+-------------------+--------+
|1205  | Kranthi      | 30000       | Op Admin          | Admin  |
|1204  | Krian        | 40000       | Hr Admin          | HR     |
|1202  | Manisha      | 45000       | Proofreader       | PR     |
|1201  | Gopal        | 45000       | Technical manager | TP     |
|1203  | Masthanvali  | 40000       | Technical writer  | TP     |
+------+--------------+-------------+-------------------+--------+
```

JDBC Program

Here is the JDBC program to apply Order By clause for the given example.

```
import java.sql.SQLException;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.Statement;
import java.sql.DriverManager;

public class HiveQLOrderBy {
```

```
   private static String driverName = "org.apache.hadoop.hive.jdbc.HiveDriver";

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

      // Register driver and create driver instance
      Class.forName(driverName);

      // get connection
      Connection con = DriverManager.getConnection("jdbc:hive://localhost:10000/userdb",
"", "");

      // create statement
      Statement stmt = con.createStatement();

      // execute statement
      Resultset res = stmt.executeQuery("SELECT * FROM employee ORDER BY DEPT;");
      System.out.println(" ID \t Name \t Salary \t Designation \t Dept ");

      while (res.next()) {
         System.out.println(res.getInt(1) + " " + res.getString(2) + " " +
res.getDouble(3) + " " + res.getString(4) + " " + res.getString(5));
      }

      con.close();
   }
}
```

Save the program in a file named HiveQLOrderBy.java. Use the following commands to compile and execute this program.

```
$ javac HiveQLOrderBy.java
$ java HiveQLOrderBy
```

Output:

```
ID       Name           Salary      Designation          Dept
1205     Kranthi        30000       Op Admin             Admin
1204     Krian          40000       Hr Admin             HR
1202     Manisha        45000       Proofreader          PR
1201     Gopal          45000       Technical manager    TP
1203     Masthanvali    40000       Technical writer     TP
1204     Krian          40000       Hr Admin             HR
```