**Hive Queries: Order By, Group By, Distribute By, Cluster By Examples**

Hive provides[SQL](https://www.guru99.com/sql.html)type querying language for the ETL purpose on top of[Hadoop](https://www.guru99.com/bigdata-tutorials.html)file system.

Hive Query language (HiveQL) provides SQL type environment in Hive to work with tables, databases, queries.

We can have a different type of Clauses associated with Hive to perform different type data manipulations and querying. For better connectivity with different nodes outside the environment. HIVE provide JDBC connectivity as well.

Hive queries provides the following features:

* Data modeling such as Creation of databases, tables, etc.
* ETL functionalities such as Extraction, Transformation, and Loading data into tables
* Joins to merge different data tables
* User specific custom scripts for ease of code
* Faster querying tool on top of Hadoop

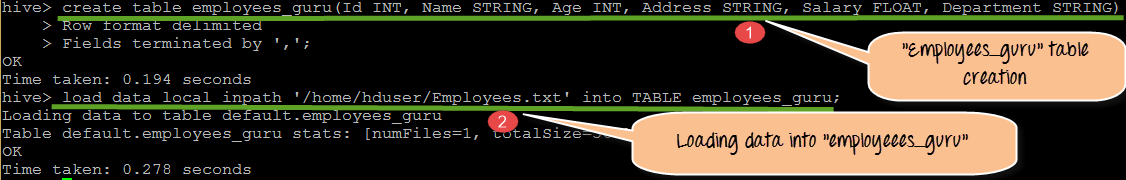
In this article, you will learn-

* [Order by query](https://www.guru99.com/hive-queries-implementation.html#1)
* [Group by query](https://www.guru99.com/hive-queries-implementation.html#2)
* [Sort by](https://www.guru99.com/hive-queries-implementation.html#3)
* [Cluster By](https://www.guru99.com/hive-queries-implementation.html#4)
* [Distribute By](https://www.guru99.com/hive-queries-implementation.html#5)

**Creating Table in Hive**

Before initiating with our main topic for this tutorial, first we will create a table to use it as references for the following tutorial.

Here in this tutorial, we are going to create table "employees\_guru" with 6 columns.

[](https://www.guru99.com/images/Hive/120415_1305_HiveQueries1.png)

From the above screen shot,

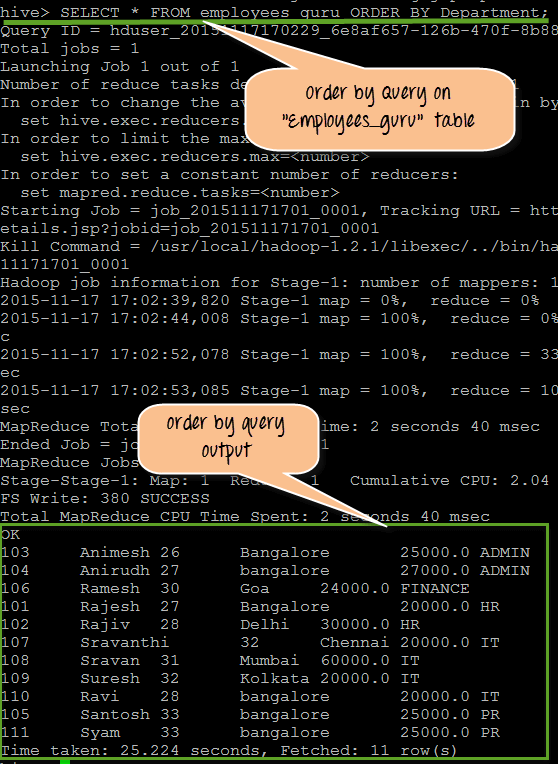
1. We are creating table "employees\_guru" with 6 column values such as Id, Name, Age, Address, Salary, Department, which belongs to the employees present in organization "guru."
2. Here in this step we are loading data into employees\_guru table. The data that we are going to load will be placed under Employees.txt file

**Order by query:**

The ORDER BY syntax in HiveQL is similar to the syntax of ORDER BY in SQL language.

Order by is the clause we use with "SELECT" statement in Hive queries, which helps sort data. Order by clause use columns on Hive tables for sorting particular column values mentioned with Order by. For whatever the column name we are defining the order by clause the query will selects and display results by ascending or descending order the particular column values.

If the mentioned order by field is a string, then it will display the result in lexicographical order. At the back end, it has to be passed on to a single reducer.

[](https://www.guru99.com/images/Hive/120415_1305_HiveQueries2.png)

From the Above screen shot, we can observe the following

1. It is the query that performing on the "employees\_guru" table with the ORDER BY clause with Department as defined ORDER BY column name.

"Department" is String so it will display results based on lexicographical order.

1. This is actual output for the query. If we observe it properly, we can see that it get results displayed based on Department column such as ADMIN, Finance and so on in orderQuery to be perform.

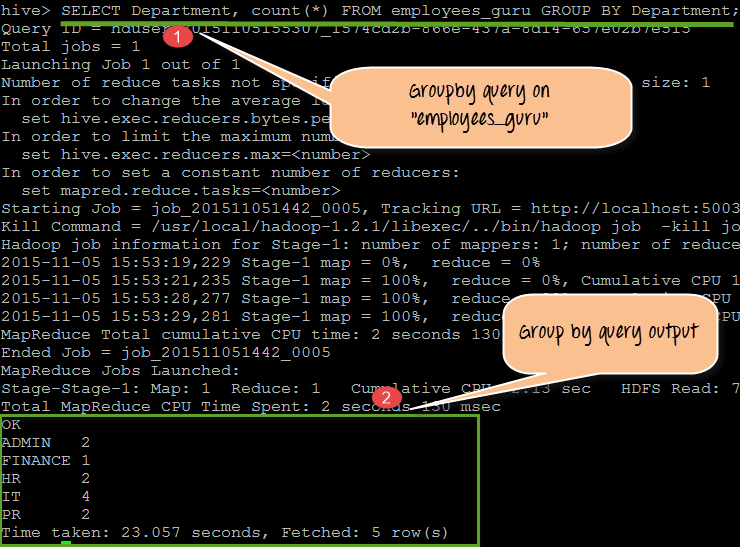
Query :

SELECT \* FROM employees\_guru ORDER BY Department;

**Group by query:**

Group by clause use columns on Hive tables for grouping particular column values mentioned with the group by. For whatever the column name we are defining a "groupby" clause the query will selects and display results by grouping the particular column values.

For example, in the below screen shot it's going to display the total count of employees present in each department. Here we have "Department" as Group by value.

[](https://www.guru99.com/images/Hive/120415_1305_HiveQueries3.png)

From the above screenshot, we will observe the following

1. It is the query that is performed on the "employees\_guru" table with the GROUP BY clause with Department as defined GROUP BY column name.
2. The output showing here is the department name, and the employees count in different departments. Here all the employees belong to the specific department is grouped by and displayed in the results. So the result is department name with the total number of employees present in each department.

Query:

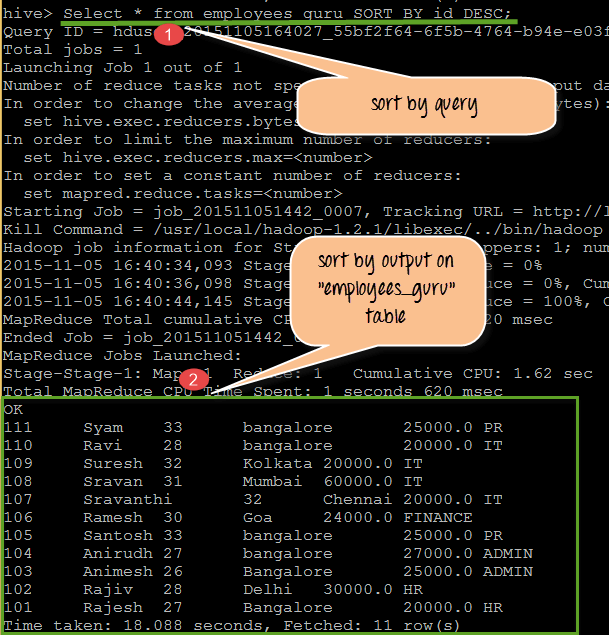
SELECT Department, count(\*) FROM employees\_guru GROUP BY Department;

**Sort by:**

Sort by clause performs on column names of Hive tables to sort the output. We can mention DESC for sorting the order in descending order and mention ASC for Ascending order of the sort.

In this sort by it will sort the rows before feeding to the reducer. Always sort by depends on column types.

For instance, if column types are numeric it will sort in numeric order if the columns types are string it will sort in lexicographical order.

[](https://www.guru99.com/images/Hive/120415_1305_HiveQueries4.png)

From the above screen shot we can observe the following:

1. It is the query that performing on the table "employees\_guru" with the SORT BY clause with "id" as define SORT BY column name. We used keyword DESC.
2. So the output displayed will be in descending order of "id".

Query:

SELECT \* from employees\_guru SORT BY Id DESC;

**Cluster By:**

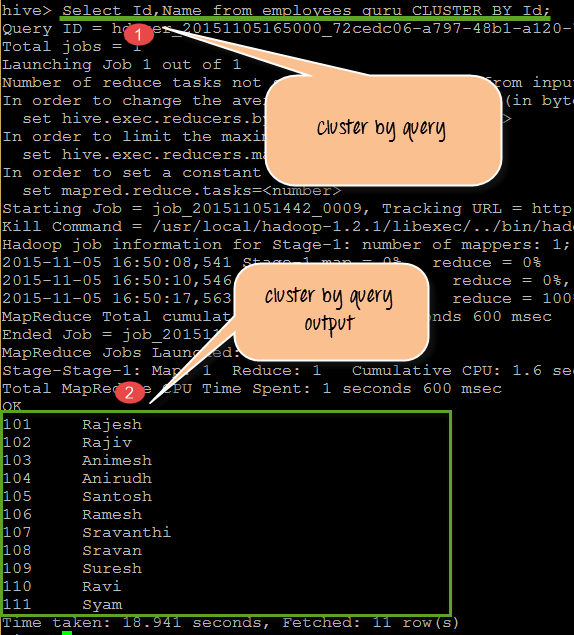
Cluster By used as an alternative for both Distribute BY and Sort BY clauses in Hive-QL.

Cluster BY clause used on tables present in Hive. Hive uses the columns in Cluster by to distribute the rows among reducers. Cluster BY columns will go to the multiple reducers.

* It ensures sorting orders of values present in multiple reducers

For example, Cluster By clause mentioned on the Id column name of the table employees\_guru table. The output when executing this query will give results to multiple reducers at the back end. But as front end it is an alternative clause for both Sort By and Distribute By.

This is actually back end process when we perform a query with sort by, group by, and cluster by in terms of Map reduce framework. So if we want to store results into multiple reducers, we go with Cluster By.

[](https://www.guru99.com/images/Hive/120415_1305_HiveQueries5.png)

From the above screen shot we are getting the following observations:

1. It is the query that performs CLUSTER BY clause on Id field value. Here it's going to get a sort on Id values.
2. It displays the Id and Names present in the guru\_employees sort ordered by

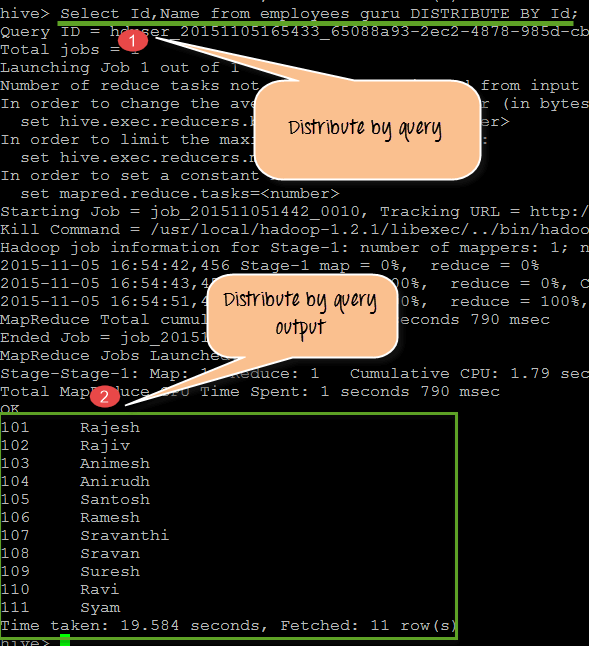
Query:

SELECT Id, Name from employees\_guru CLUSTER BY Id;

**Distribute By:**

Distribute BY clause used on tables present in Hive. Hive uses the columns in Distribute by to distribute the rows among reducers. All Distribute BY columns will go to the same reducer.

* It ensures each of N reducers gets non-overlapping ranges of column
* It doesn't sort the output of each reducer

[](https://www.guru99.com/images/Hive/120415_1305_HiveQueries6.png)

From the above screenshot, we can observe the following

1. DISTRIBUTE BY Clause performing on Id of "empoloyees\_guru" table
2. Output showing Id, Name. At back end, it will go to the same reducer

Query:

SELECT Id, Name from employees\_guru DISTRIBUTE BY Id;