

A dark blue vertical bar runs down the left side of the page. A blue arrow points to the right from this bar, containing the text 'Big Data Lab'.

Big Data Lab

Hive Programs

Several thin, dark blue curved lines originate from the bottom left corner and sweep upwards and to the right, creating a sense of movement or data flow.

1.

- a. Create table “employees” with 6 attributes such as Id, Name, Age, Address, Salary, Department and load data into employees table from Employees.txt file.
- b. Display the list of corresponding employee id, name and their address.
- c. Write a query to group all the employees by their department and display the results.
- d. Create another table department with the attributes dno and dname. Now display the department id in which each employee works along with the employee id and their age.
- e. Drop the table employees

Ans.

1.a

```
create table employees(Id int, Name string, Age int, Address string, Salary float, DeptId int) row format delimited fields terminated by ',';

load data local inpath '/home/msrit/Downloads/Employees.txt' into TABLE employees;
```

1.b

```
SELECT Id, Name, Address from employees;
```

1.c

```
SELECT DeptId, count(*) FROM employees GROUP BY DeptId;
```

1.d

```
create table department(DeptId int, DeptName string) row format delimited fields terminated by ',';

load data local inpath '/home/msrit/Downloads/Departments.txt' into TABLE department;

select e.DeptId, e.Name, e.Id, e.Age from employees e join department d on e.DeptId = d.DeptId;
```

1.e

```
drop table employees;
```

2.

- a. Create table “employees” with 6 attributes such as Id, Name, Age, Address, Salary, Department and load data into employees table from Employees.txt file
- b. Display the total number of employees whose details are present in the employees table.
- c. Write a query to sort the employee details by their ‘id’ in descending order.
- d. Create a view ‘employee_view’ by taking id and name of employee from ‘employees’ table and display the contents of the view.
- e. Drop the view and the table

Ans.

2.a

```
create table employees(Id int, Name string, Age int, Address string, Salary float, DeptId int) row format delimited fields terminated by ',';
```

```
load data local inpath '/home/msrit/Downloads/Employees.txt' into TABLE employees;
```

2.b

```
select count(*) from employees;
```

2.c

```
SELECT * from employees SORT BY Id DESC;
```

2.d

```
create view employee_view as select Id, Name from employees;
```

```
SELECT * from employee_view;
```

2.e

```
drop view employee_view;
```

```
drop table employees;
```

3.

- a. Create table “employees” with 6 attributes such as Id, Name, Age, Address, Salary, Department and load data into employees table from Employees.txt file.
- b. Write a query to display the details of the employees from the ‘employees’ table ordered by the ‘department’ attribute using the order by clause.
- c. Create a table to static_part_employee with attributes id int, name String and partition it with the attribute salary int.
- d. Insert values into static_part_employee by overwriting the details from employees table and pass the value of salary as 25000 for partition.
- e. Display the static_part_employee table and then drop the same table

Ans.

3.a

```
create table employees(Id int, Name string, Age int, Address string, Salary float, DeptId int) row format delimited fields terminated by ',';
load data local inpath '/home/msrit/Downloads/Employees.txt' into TABLE employees;
```

3.b

```
SELECT * FROM employees ORDER BY department;
```

3.c

```
create table if not exists static_part_employee(Id int, name string) partitioned by (salary float) row format delimited fields terminated by ',';
```

3.d

```
insert overwrite table static_part_employee partition (salary = 25000) select Id, name
from employees where salary = 25000;
```

3.e

```
select * from static_part_employee;

drop table static_part_employee;
```

4.

- a. Create table “employees” with 6 attributes such as Id, Name, Age, Address, Salary, Department and load data into employees table from Employees.txt file.
- b. Create a table employee_rc with attributes (id int, name string, salary int) and store it as a rcfile.
- c. Insert values into the employee_ec by overwriting values from employee table.
- d. Display the employee and employee_rc table and also total salary given to all employees.
- e. Write a query to display names of employees whose salary is more than 25000.

Ans.

4.a.

```
create table employees(Id int, Name string, Age int, Address string, Salary float, DeptId int) row format delimited fields terminated by ',';
```

```
load data local inpath '/home/msrit/Downloads/Employees.txt' into TABLE employees;
```

4.b

```
create table employee_rc(Id int, name string, salary float) stored as rcfile;
```

4.c.

```
insert overwrite table employee_rc select Id, name, salary from employees;
```

4.d.

```
select * from employees;
```

```
select * from employee_rc;
```

```
select sum(salary) from employee_rc;
```

4.e.

```
select name from employees where salary>25000;
```

5.

- a. Create table “employees” with 6 attributes such as Id, Name, Age, Address, Salary, Department and load data into employees table from Employees.txt file.
- b. Create a table to dynamic_part_employee with attributes id int, name String and partition it with the attribute salary int.
- c. Insert values into dynamic_part_employee by overwriting the details from employees table and partition by salary. Display the dynamic_part_employee table.
- d. Write a query to display the average, minimum and maximum salary obtained by the employees.
- e. Write a query to display the number of employees under each department

Ans.

5.a.

```
create table employees(Id int, Name string, Age int, Address string, Salary float, DeptId int) row format delimited fields terminated by ',';
```

```
load data local inpath '/home/msrit/Downloads/Employees.txt' into TABLE employees;
```

5.b.

```
create table if not exists dynamic_part_employee(Id int, name string) partitioned by (salary float) row format delimited fields terminated by ',';
```

```
set hive.exec.dynamic.partition=true;
```

```
set hive.exec.dynamic.partition.mode=nonstrict;
```

5.c.

```
insert overwrite table dynamic_part_employee partition (salary) select Id, name, salary from employees;
```

```
select * from dynamic_part_employee;
```

5.d.

```
select avg(salary), min(salary), max(salary) from employees;
```

5.e.

```
SELECT DeptId, count(*) FROM employees GROUP BY DeptId;
```