EXP NO: 11	Implement a program using Hive External Table
DATE:	

AIM: -

BACKGROUND THEORY: -

PROCEDURE: -

- Switch to superuser mode using sudo su.
- Start Hive
- Upload the data to HDFS
- Load Data into the Table
- Create an External Table in Hive
- Query the External Table
- Run the Hive Script hive -f employee_analysis.hql

CODING: -

- sudo su
- -- Step 1: Start Hive
    - hive
- -- Step 2: Create a Hive Table
    - CREATE TABLE employees (
    - id INT,
    - name STRING,
    - age INT,
    - department STRING
    -) ROW FORMAT DELIMITED
    - FIELDS TERMINATED BY ','
    - STORED AS TEXTFILE;
- -- Step 3: Load Data into the Table
    - LOAD DATA LOCAL INPATH '/path/to/your/employees.csv' INTO TABLE employees;
- -- Step 4: Create a View
    - CREATE VIEW sales_employees AS
    - SELECT id, name, age
    - FROM employees
    - WHERE department = 'Sales';

- -- Step 5: Query the View
    - SELECT * FROM sales_employees;
- -- Step 6: Update the View (Optional)
    - DROP VIEW sales_employees;
    - CREATE VIEW sales_employees AS
    - SELECT id, name, age, department
    - FROM employees
    - WHERE department = 'Sales';
- -- Step 7: Drop the View (Optional)
    - DROP VIEW sales_employees;
- -- Select all data from the external table
- SELECT * FROM employee_info;
-
- -- Filter the data
- SELECT * FROM employee_info WHERE department = 'Engineering';
- jdbc:hive2://> select sum(salary) from employee;
- jdbc:hive2://> select sum(distinct salary) from employee;
- jdbc:hive2://> select age,sum(salary) from employee group by age;
- hive> select max(Salary) from employee_data;
- hive> select min(Salary) from employee_data;

OUTPUT: -

department	total_employees	avg_age	min_age	max_age	avg_salary
Engineering	2	29.5	29	30	61000
Marketing	2	34.0	34	34	73500
Sales	2	37.0	29	45	81500