## Hive DDL Queries

**1. Employee Table (Hive DDL)**

```
-- i. Create database
CREATE DATABASE Employee_DB;
-- ii. Create table partitioned by department
USE Employee_DB;
CREATE TABLE employee (
  empid INT,
  ename STRING,
  designation STRING
)
PARTITIONED BY (department STRING);
-- iii. Show structure
DESCRIBE employee;
-- iv. Rename column ename to emp_name
ALTER TABLE employee CHANGE ename emp_name STRING;
-- v. Add column salary
ALTER TABLE employee ADD COLUMNS (salary FLOAT);
-- vi. Rename table
ALTER TABLE employee RENAME TO employee_new;
-- vii. Delete table
DROP TABLE employee_new;
```

**2. Student Table (Hive DDL)**

```
-- i. Create database
CREATE DATABASE Student_DB;
-- ii. Create table partitioned by class
USE Student_DB;
CREATE TABLE student (
  division STRING,
  roll_no INT,
  sname STRING
)
PARTITIONED BY (class STRING);
-- iii. Show structure
DESCRIBE student;
```

```
-- iv. Rename column sname to student_name
ALTER TABLE student CHANGE sname student_name STRING;
-- v. Add column percentage
ALTER TABLE student ADD COLUMNS (percentage FLOAT);
-- vi. Rename table
ALTER TABLE student RENAME TO student_new;
-- vii. Delete table
DROP TABLE student_new;
```

**Hive DML Queries**

**3. Employee Table (DML)**

```
-- i. Create database
CREATE DATABASE Employee_DB;

-- ii. Create table
USE Employee_DB;
CREATE TABLE employee (
  empid INT,
  ename STRING,
  designation STRING,
  department STRING,
  salary FLOAT
);

-- iii. Insert records
INSERT INTO TABLE employee VALUES
(101, 'John', 'Manager', 'HR', 50000),
(102, 'Sara', 'Executive', 'Sales', 30000),
(103, 'Mike', 'Analyst', 'IT', 35000),
(104, 'Anna', 'Lead', 'Marketing', 42000),
(105, 'Bob', 'Clerk', 'Admin', 25000);

-- iv. Update salary
UPDATE employee SET salary = 40000 WHERE empid = 101;
```

```
-- v. Delete record
DELETE FROM employee WHERE empid = 101;
```

**4. Student Table (DML)**

```
-- i. Create database
CREATE DATABASE Student_DB;

-- ii. Create table
USE Student_DB;
CREATE TABLE student (
  division STRING,
  roll_no INT,
  sname STRING,
  percentage FLOAT
)
PARTITIONED BY (class STRING);

-- iii. Insert records
INSERT INTO TABLE student PARTITION (class='TY') VALUES
('A', 1, 'Neha', 75),
('B', 2, 'Amit', 55),
('A', 3, 'Priya', 67),
('B', 4, 'Rohit', 59),
('C', 5, 'Sneha', 82);

-- iv. Update percentage
UPDATE student SET percentage = 80 WHERE roll_no = 4;

-- v. Delete record
DELETE FROM student WHERE roll_no = 4;
```

**Hive Data Retrieval**

**5. Employee Table**

```
-- Create DB and table + Insert (similar to previous steps)
-- iv. Retrieve employees with salary > 20000 and from "sales"
SELECT * FROM employee WHERE salary > 20000 AND department = 'sales';
```

```
-- v. Average salary per department
SELECT department, AVG(salary) AS avg_salary FROM employee GROUP BY department;

-- vi. Departments with avg salary > 20000
SELECT department FROM (
  SELECT department, AVG(salary) AS avg_salary FROM employee GROUP BY department
) AS dept_avg
WHERE avg_salary > 20000;
```

**6. Student Table**

```
-- Create DB and table + Insert (similar to previous steps)
-- iv. Students with percentage > 60 and from “TY”
SELECT * FROM student WHERE percentage > 60 AND class = 'TY';

-- v. Average percentage in TY
SELECT class, AVG(percentage) AS avg_percent FROM student WHERE class = 'TY' GROUP BY class;

-- vi. Classes with avg percentage > 60
SELECT class FROM (
  SELECT class, AVG(percentage) AS avg_percent FROM student GROUP BY class
) AS class_avg
WHERE avg_percent > 60;
```

**Pig Operators**

**7. Basic Pig Script**

```
employee = LOAD 'hdfs:/data/employee.csv' USING PigStorage(',')
  AS (empid:int, ename:chararray, department:chararray, salary:float);

-- Transform
employee_trans = FOREACH employee GENERATE empid, ename, salary;

-- Filter
employee_filtered = FILTER employee BY salary > 20000;
```

```
DUMP employee_trans;
DUMP employee_filtered;
```

**8. Grouping & Sorting**

```
student_info = LOAD 'student_info.csv' USING PigStorage(',')
  AS (exam_no:int, name:chararray, class:chararray, department:chararray);

result_info = LOAD 'result_info.csv' USING PigStorage(',')
  AS (exam_no:int, percentage:float, grade:chararray);

joined = JOIN student_info BY exam_no, result_info BY exam_no;
DUMP joined;
```

**10. Outer Joins**

```
left_join = JOIN student_info BY exam_no LEFT OUTER, result_info BY exam_no;
right_join = JOIN student_info BY exam_no RIGHT OUTER, result_info BY exam_no;
full_join = JOIN student_info BY exam_no FULL OUTER, result_info BY exam_no;
```

**11. Top Rankers**

```
sorted = ORDER result_info BY percentage DESC;
top_three = LIMIT sorted 3;
DUMP top_three;
```

**12. Sampling & Splitting**

```
sampled = SAMPLE result_info 0.2;
DUMP sampled;

high = FILTER result_info BY percentage >= 60;
low = FILTER result_info BY percentage < 60;
```