if you are running first time then pull the image otherwise skip next two line

```
--------------------
docker pull macio232/hadoop-pseudo-distributed-mode

docker run -p 9870:9870 -p 8088:8088 -it --name=myHadoop
macio232/hadoop-pseudo-distributed-mode

--------------------

docker container start -i  myHadoop
ls
mkdir test
```

another terminal: docker cp Documents/MajidMac/Documents/CSE/student_results.csv myHadoop:\test\

```
hive
CREATE DATABASE IF NOT EXISTS education_db;


CREATE TABLE IF NOT EXISTS education_db.student_results (
   student_id INT,
   subject_code STRING,
   marks INT,
   grade STRING
)
PARTITIONED BY (
   exam_year INT,
   exam_session STRING
)
STORED AS PARQUET
LOCATION '/Test/Result';
```

Container terminal: hdfs dfs -ls /

```
hdfs dfs -rm -r /Test
hdfs dfs -ls /Test/
```

Again in hive

```
CREATE TABLE IF NOT EXISTS education_db.result_tmp (
   student_id INT,
   subject_code STRING,
   marks INT,
   grade STRING,
   exam_year INT,
   exam_session STRING
)
ROW FORMAT DELIMITED
```

```
FIELDS TERMINATED BY ','
LOCATION '/Test/Result_Temp';


show tables from education_db;

LOAD DATA LOCAL INPATH '/test/student_results.csv' INTO TABLE
education_db.result_tmp;

select * from education_db.result_tmp;

SET hive.exec.dynamic.partition=true;
SET hive.exec.dynamic.partition.mode=nonstrict;


INSERT OVERWRITE TABLE education_db.student_results PARTITION(exam_year,
exam_session) SELECT student_id, subject_code , marks , grade, exam_year,
exam_session from education_db.result_tmp WHERE exam_year IS NOT NULL AND
exam_session IS NOT NULL;

MSCK REPAIR TABLE education_db.student_results;
select * from education_db.student_results;

SHOW PARTITIONS education_db.student_results;

Drop table education_db.result_tmp;
DROP DATABASE education_db CASCADE;


select * from education_db.student_results where exam_year=2025 and
exam_session='Fall';
```

Q3 — Add a new Partition: Write Hive commands to add a partition (exam_year=2025, exam_session='Spring') to the student_results table

```
------------------------------------------------------------------------------------------------------------------------------------------------
```

```
ALTER TABLE education_db.student_results
ADD PARTITION (exam_year=2020, exam_session='Fall')
LOCATION '/Test/Result/exam_year=2020/exam_session=Fall';


SHOW PARTITIONS education_db.student_results;

MSCK REPAIR TABLE education_db.student_results;
```

Q4 — Data Insertion into Specific Partition: Insert 4 new records for the above partition. After that verify the insertion.


```
INSERT INTO TABLE education_db.student_results
PARTITION (exam_year=2020, exam_session='Fall')
VALUES
(1071,'CSE101',92,'A+'),
(1072,'CSE102',78,'B+'),
(1073,'CSE103',85,'A'),
(1074,'CSE104',67,'B');

MSCK REPAIR TABLE education_db.student_results;

select * from education_db.student_results where exam_year=2020 and
exam_session='Fall';


SELECT *
FROM education_db.student_results
WHERE exam_year=2025
  AND exam_session='Fall';


SELECT *
FROM education_db.student_results
WHERE exam_year=2025
  AND exam_session='Fall';
```

Q5 — Drop Partition: Write Hive commands to drop the partition (exam_year=2022, exam_session='Fall') from the student_results table.


```
ALTER TABLE education_db.student_results
DROP PARTITION (exam_year=2020, exam_session='Fall');

SHOW PARTITIONS education_db.student_results;

MSCK REPAIR TABLE education_db.student_results;
```

"We need two more tables(students, courses) to execute next queries, So we will create that tables now"

============================ students, courses table creation
======================================

```
CREATE TABLE IF NOT EXISTS education_db.students (
   student_id INT,
   student_name STRING,
   dob STRING,
   department STRING
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE
LOCATION '/Test/Student';

LOAD DATA LOCAL INPATH '/test/students.csv' INTO TABLE education_db.students;
select * from education_db.students;


CREATE TABLE IF NOT EXISTS education_db.courses (
   course_id INT,
   course_name STRING,
   credits INT,
   department STRING
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE
LOCATION '/Test/Course';

LOAD DATA LOCAL INPATH '/test/courses.csv' INTO TABLE education_db.courses;

select * from education_db.courses;
```

================================== students table created
=====================================

Q6 — Query with Partition Filtering: Write a Hive query to display records of students who appeared in the exam_year=2023 and exam_session='Fall'.

```
SELECT
   sr.student_id,
   s.student_name,
   s.dob,
   s.department,
   sr.subject_code,
   sr.marks,
   sr.grade,
    sr.exam_year,
    sr.exam_session
FROM education_db.student_results sr
JOIN education_db.students s
ON sr.student_id=s.student_id
WHERE
    sr.exam_year=2025 AND sr.exam_session='Fall';
```

Q8 — Join Query: Write a Hive query to display student_name, course_name, marks for all students in the Computer Science department.

```
SELECT
    s.student_name,
    c.course_name,
    sr.marks
FROM education_db.student_results sr
JOIN education_db.students s
ON sr.student_id = s.student_id
JOIN education_db.courses c
ON sr.subject_code = c.course_name
WHERE s.department = 'Computer Science';
```

Q9 — Aggregation: Find the average marks per department for the exam_year 2025.

```
SELECT
    s.department,
    AVG(sr.marks) AS avg_marks
FROM education_db.student_results sr
JOIN education_db.students s
    ON sr.student_id = s.student_id
WHERE sr.exam_year = 2025
GROUP BY s.department;
```

Q10 — Top Scorer Query: Write a Hive query to find the student(s) with the highest marks in the Spring 2025 session.

Find the maximum marks in Spring 2025

```
WITH max_marks_cte AS (
    SELECT MAX(marks) AS max_marks
    FROM education_db.student_results
    WHERE exam_year = 2025 AND exam_session = 'Fall'
)
```

```
SELECT
    sr.student_id,
    s.student_name,
    sr.subject_code,
    sr.marks
FROM education_db.student_results sr
JOIN max_marks_cte mm
    ON sr.marks = mm.max_marks
JOIN education_db.students s
    ON sr.student_id = s.student_id
WHERE sr.exam_year = 2025 AND sr.exam_session = 'Fall';
```

Necessary Commands

```
--------------------
DROP TABLE education_db.students;
DROP DATABASE education_db; // table gula delete kore nite hobe age
SHOW TABLES IN education_db;
hdfs dfs -rm -r /TeacherDetail/Result/exam_year=__HIVE_DEFAULT_PARTITION__
```

Debugging command

```
hdfs dfs -ls -R /Test/Result | grep exam_year=2020
hdfs dfs -rm -r /Test/Result/exam_year=2020
```