DEPARTMENT OF COMPUTER SCIENCE

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

BHARATHIAR UNIVERSITY

COIMBATORE-641046

NAME OF THE CANDIDATE: ______________________________

REGISTER NUMBER: ______________________________

M.Sc. DATA SCIENCE

SEMESTER -III

BIGDATA ANALYTICS LAB

24DS3C1

OCTOBER -2025

DEPARTMENT OF COMPUTER SCIENCE SCHOOL OF

COMPUTER SCIENCE AND ENGINEERING

BHARATHIAR UNIVERSITY

COIMBATORE-641046

CERTIFICATE

This is to certify that the bonafide record work **BIGDATA ANALYTICS PRACTICAL LAB – 24DS3C1** was done and submitted by Mr. / Ms. ______________________________with the Register Number ____________________in partial fulfillment of the requirements for the Degree, Master of Science in Data Science at Department of Computer Science, Bharathiar University, Coimbatore – 641 046, during the period OCTOBER 2025.

Submitted for Practical Examination on ______________________.

__________________________	__________________________
Staff - In-Charge	Head of the Department

__________________________	__________________________
Internal Examiner	External Examiner

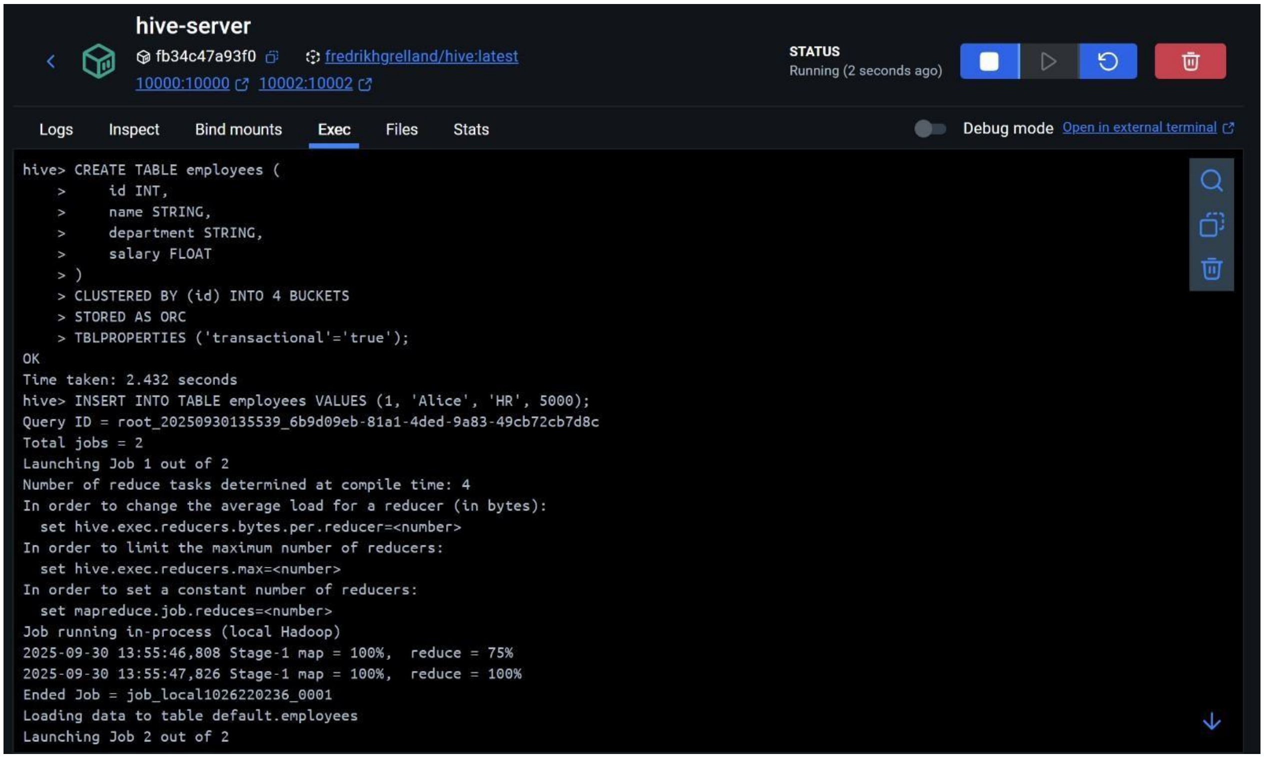
TABLE OF CONTENTS

1. DATA MANIPULATION USING HIVEQL

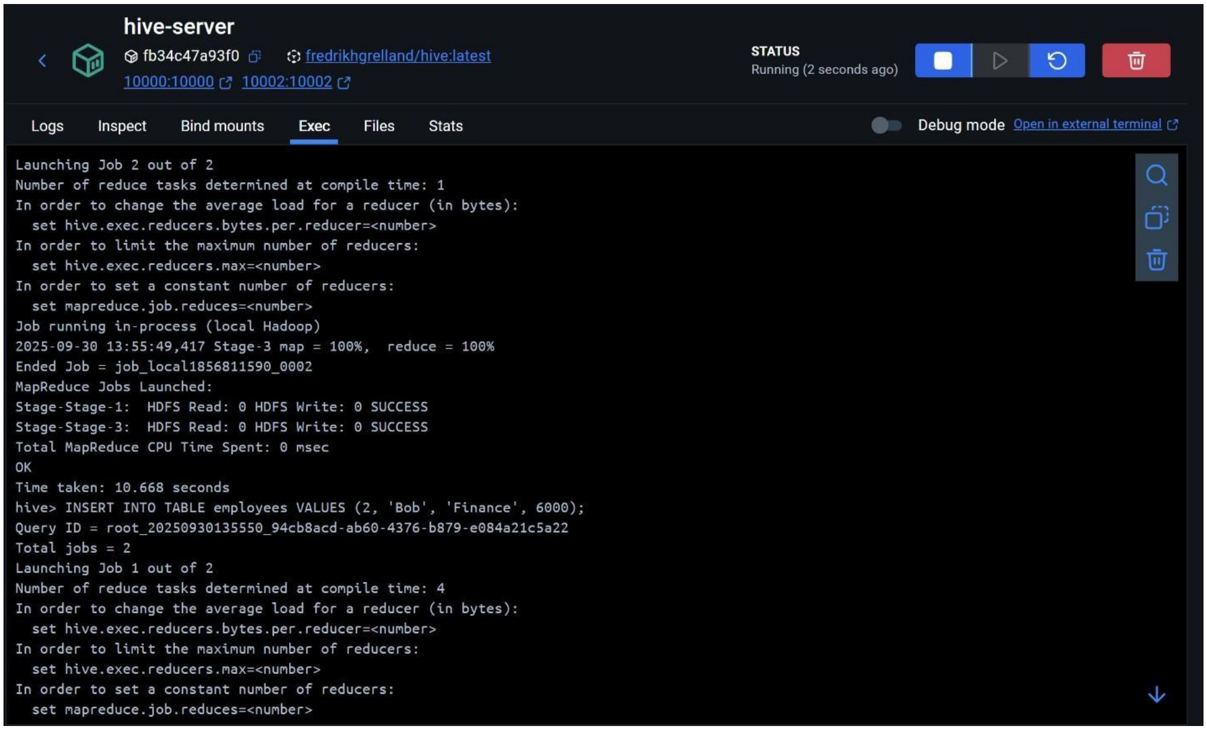
PROGRAM:

```
CREATE TABLE employees (
    id INT,
    name STRING,
    department STRING,
    salary FLOAT
)
CLUSTERED BY (id) INTO 4 BUCKETS
STORED AS ORC
TBLPROPERTIES ('transactional'='true');
INSERT INTO TABLE employees VALUES (1, 'Alice', 'HR', 5000);
INSERT INTO TABLE employees VALUES (2, 'Bob', 'Finance', 6000);
INSERT INTO TABLE employees VALUES (3, 'Charlie', 'IT', 7000);
INSERT INTO TABLE employees VALUES (4, 'David', 'Marketing', 5500);
UPDATE employees SET salary = 7500 WHERE id = 3;
DELETE FROM employees WHERE id = 1;
ALTER TABLE employees ADD COLUMNS (email STRING);
ALTER TABLE employees CHANGE COLUMN salary salary DOUBLE;
ALTER TABLE employees RENAME TO employees_new;
TRUNCATE TABLE employees_new;
```

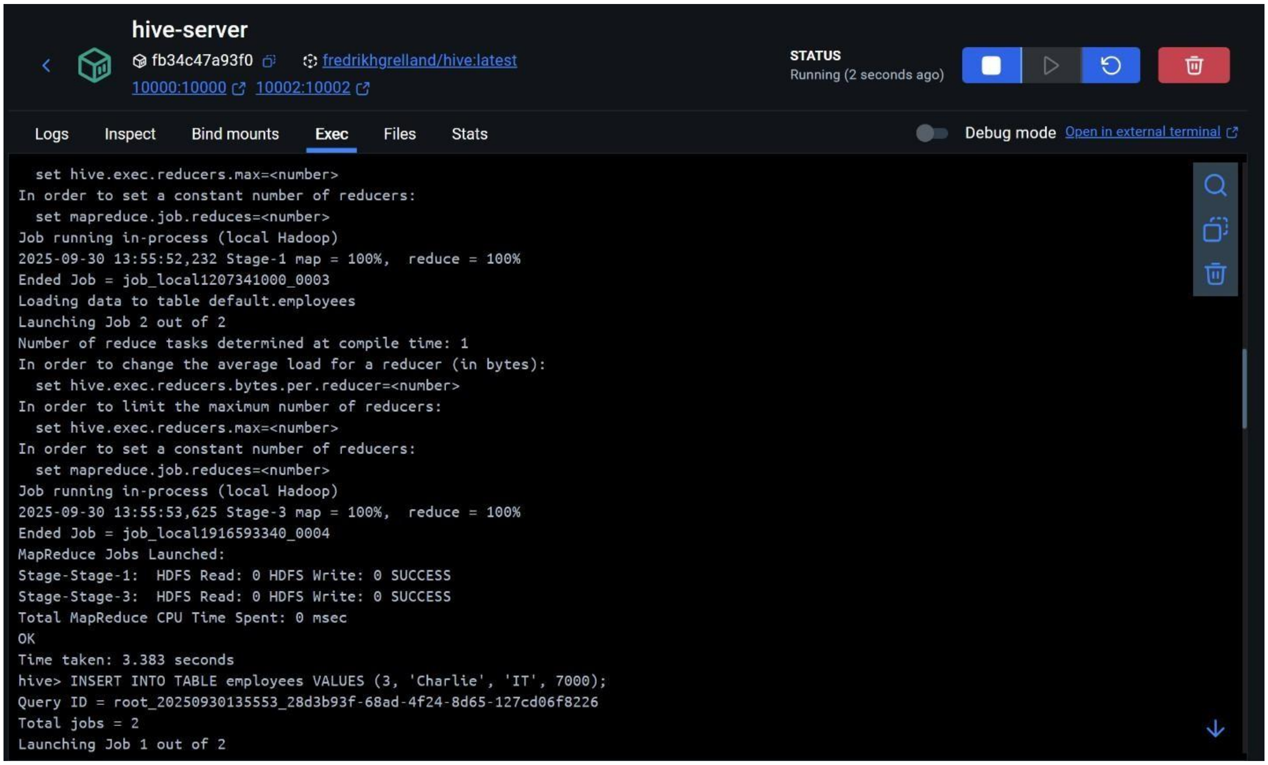
OUTPUT:



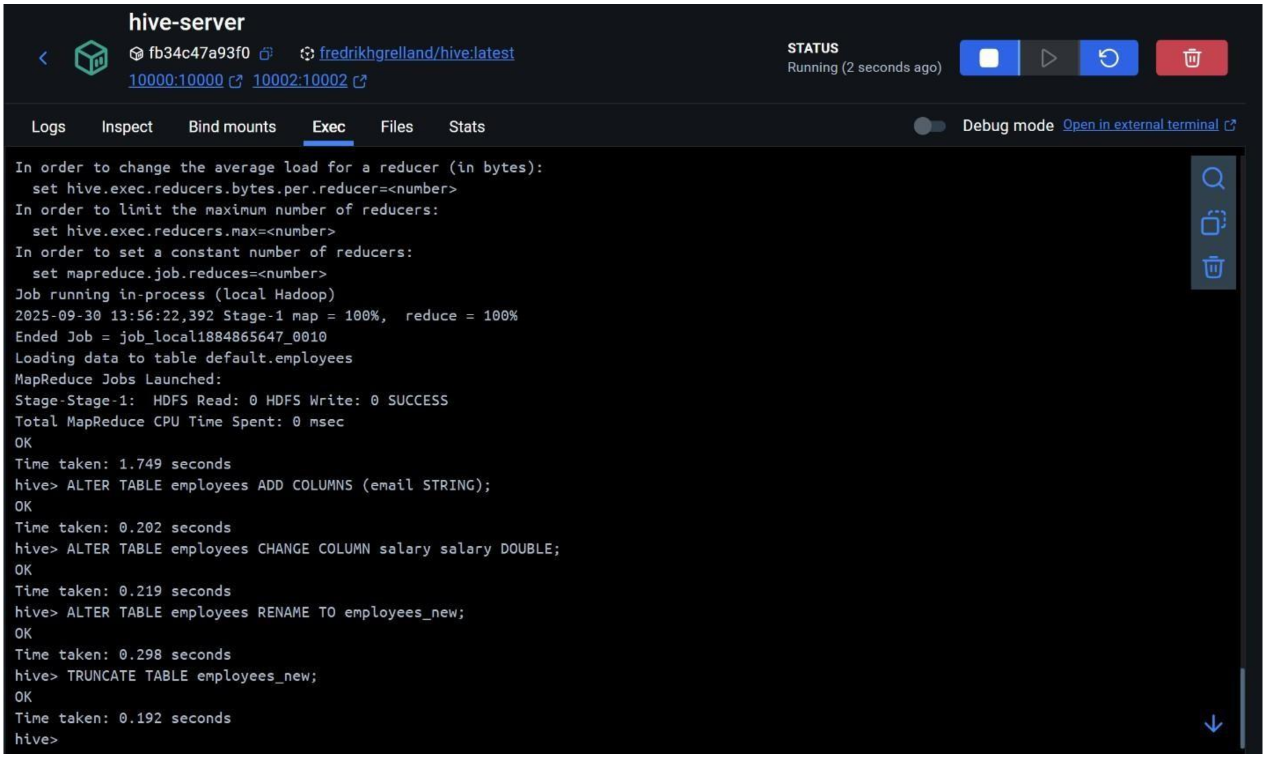
```
hive> CREATE TABLE employees (
    >     id INT,
    >     name STRING,
    >     department STRING,
    >     salary FLOAT
    > )
    > CLUSTERED BY (id) INTO 4 BUCKETS
    > STORED AS ORC
    > TBLPROPERTIES ('transactional'='true');
OK
Time taken: 2.432 seconds
hive> INSERT INTO TABLE employees VALUES (1, 'Alice', 'HR', 5000);
Query ID = root_20250930135539_6b9d09eb-81a1-4ded-9a83-49cb72cb7d8c
Total jobs = 2
Launching Job 1 out of 2
Number of reduce tasks determined at compile time: 4
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Job running in-process (local Hadoop)
2025-09-30 13:55:46,808 Stage-1 map = 100%,  reduce = 75%
2025-09-30 13:55:47,826 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_local1026220236_0001
Loading data to table default.employees
Launching Job 2 out of 2
```



```
Launching Job 2 out of 2
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Job running in-process (local Hadoop)
2025-09-30 13:55:49,417 Stage-3 map = 100%,  reduce = 100%
Ended Job = job_local1856811590_0002
MapReduce Jobs Launched:
Stage-Stage-1:  HDFS Read: 0 HDFS Write: 0 SUCCESS
Stage-Stage-3:  HDFS Read: 0 HDFS Write: 0 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
Time taken: 10.668 seconds
hive> INSERT INTO TABLE employees VALUES (2, 'Bob', 'Finance', 6000);
Query ID = root_20250930135550_94cb8acd-ab60-4376-b879-e084a21c5a22
Total jobs = 2
Launching Job 1 out of 2
Number of reduce tasks determined at compile time: 4
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
```



```
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Job running in-process (local Hadoop)
2025-09-30 13:55:52,232 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_local1207341000_0003
Loading data to table default.employees
Launching Job 2 out of 2
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Job running in-process (local Hadoop)
2025-09-30 13:55:53,625 Stage-3 map = 100%,  reduce = 100%
Ended Job = job_local1916593340_0004
MapReduce Jobs Launched:
Stage-Stage-1:  HDFS Read: 0 HDFS Write: 0 SUCCESS
Stage-Stage-3:  HDFS Read: 0 HDFS Write: 0 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
Time taken: 3.383 seconds
hive> INSERT INTO TABLE employees VALUES (3, 'Charlie', 'IT', 7000);
Query ID = root_20250930135553_28d3b93f-68ad-4f24-8d65-127cd06f8226
Total jobs = 2
Launching Job 1 out of 2
```



```
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Job running in-process (local Hadoop)
2025-09-30 13:56:22,392 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_local1884865647_0010
Loading data to table default.employees
MapReduce Jobs Launched:
Stage-Stage-1:  HDFS Read: 0 HDFS Write: 0 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
Time taken: 1.749 seconds
hive> ALTER TABLE employees ADD COLUMNS (email STRING);
OK
Time taken: 0.202 seconds
hive> ALTER TABLE employees CHANGE COLUMN salary salary DOUBLE;
OK
Time taken: 0.219 seconds
hive> ALTER TABLE employees RENAME TO employees_new;
OK
Time taken: 0.298 seconds
hive> TRUNCATE TABLE employees_new;
OK
Time taken: 0.192 seconds
hive>
```

RESULT:

2. HIVE JOIN OPERATIONS

PROGRAM:

1. Create employees table

```
CREATE TABLE employees (
    emp_id INT,
    emp_name STRING,
    dept_id INT
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE;
```

2. Create departments table

```
CREATE TABLE departments (
    dept_id INT,
    dept_name STRING
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE;
```

3. Insert data into employees

```
INSERT INTO TABLE employees VALUES
(1, 'Alice', 10),
(2, 'Bob', 20),
(3, 'Charlie', 30),
(4, 'David', 40);
```

4. **Insert data into departments**

```
INSERT INTO TABLE departments VALUES
(10, 'HR'),
(20, 'Finance'),
(30, 'IT'),
(50, 'Marketing');
```

5. **Inner Join**

```
SELECT e.emp_id, e.emp_name, d.dept_name
FROM employees e
INNER JOIN departments d
ON e.dept_id = d.dept_id;
```

6. **Left Outer Join**

```
SELECT e.emp_id, e.emp_name, d.dept_name
FROM employees e
LEFT OUTER JOIN departments d
ON e.dept_id = d.dept_id;
```

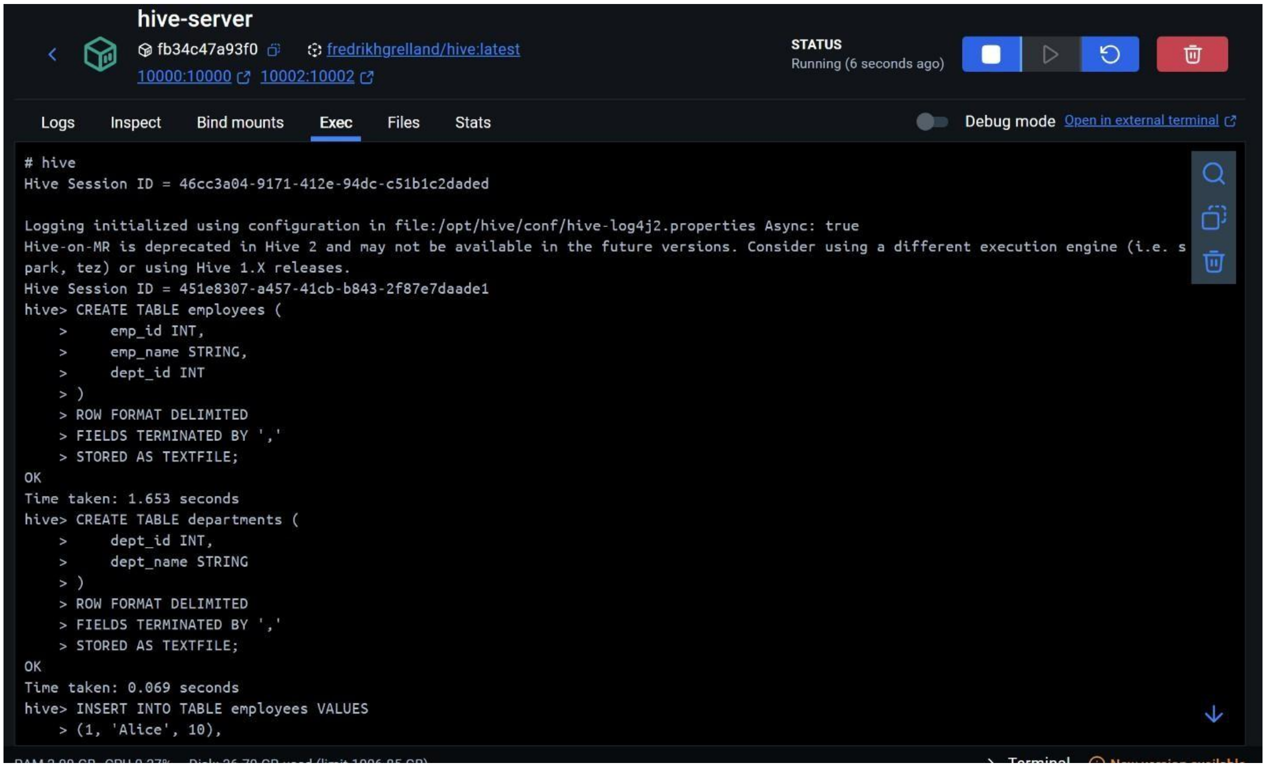
7. **Right Outer Join**

```
SELECT e.emp_id, e.emp_name, d.dept_name
FROM employees e
RIGHT OUTER JOIN departments d
ON e.dept_id = d.dept_id;
```

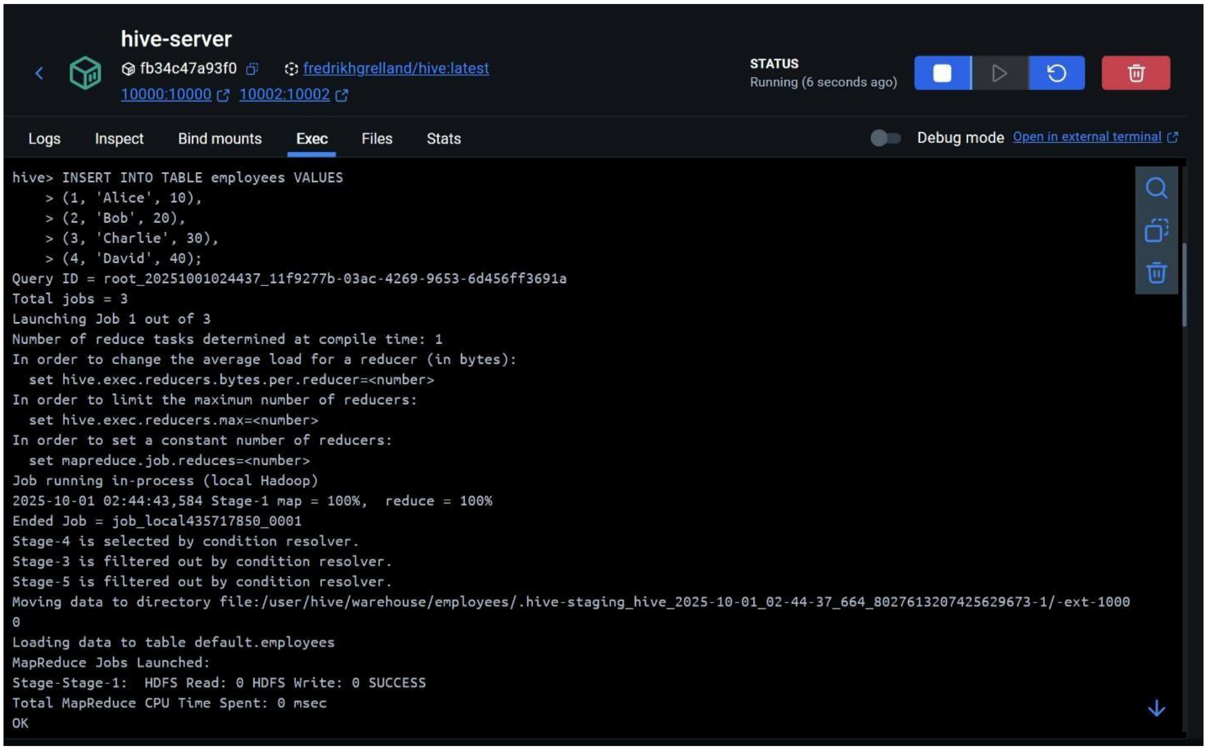
8. **Full Outer Join**

```
SELECT e.emp_id, e.emp_name, d.dept_name

FROM employees e

FULL OUTER JOIN departments d

ON e.dept_id = d.dept_id;
```

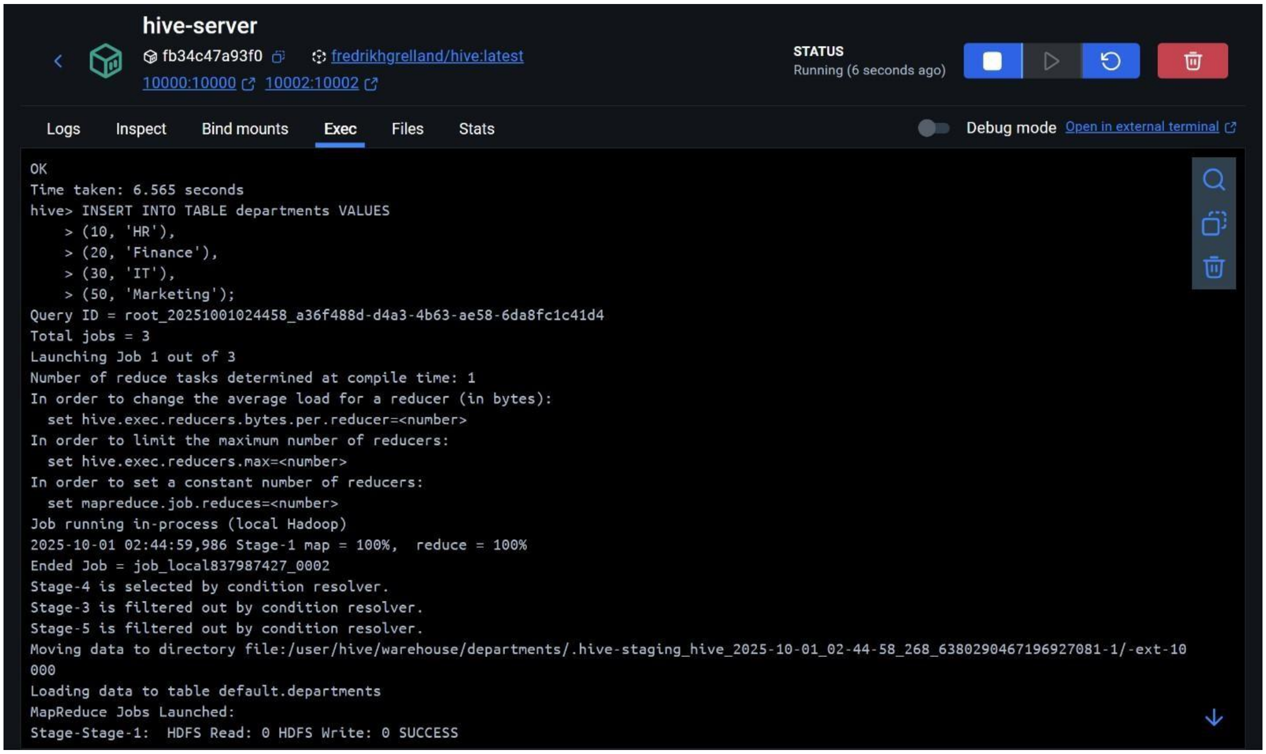
OUTPUT:



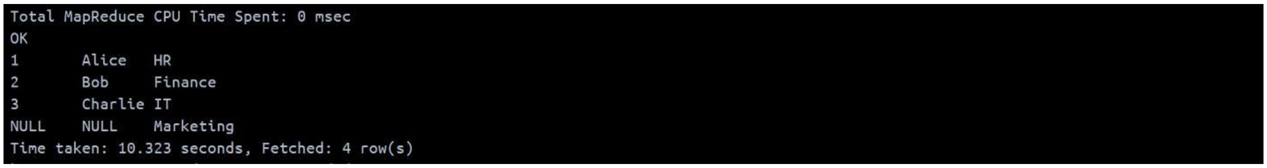
```
# hive
Hive Session ID = 46cc3a04-9171-412e-94dc-c51b1c2daded

Logging initialized using configuration in file:/opt/hive/conf/hive-log4j2.properties Async: true
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. s
park, tez) or using Hive 1.X releases.
Hive Session ID = 451e8307-a457-41cb-b843-2f87e7daade1
hive> CREATE TABLE employees (
    >     emp_id INT,
    >     emp_name STRING,
    >     dept_id INT
    > )
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY ','
    > STORED AS TEXTFILE;
OK
Time taken: 1.653 seconds
hive> CREATE TABLE departments (
    >     dept_id INT,
    >     dept_name STRING
    > )
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY ','
    > STORED AS TEXTFILE;
OK
Time taken: 0.069 seconds
hive> INSERT INTO TABLE employees VALUES
    > (1, 'Alice', 10),
```



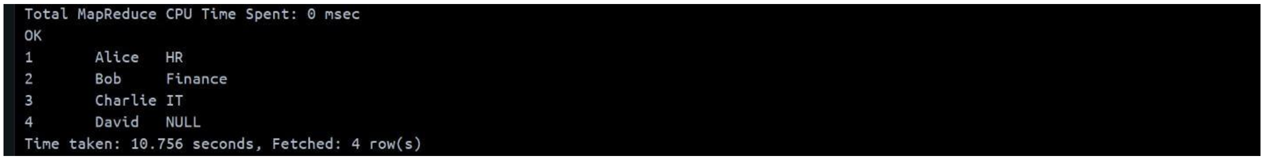
```
hive> INSERT INTO TABLE employees VALUES
    > (1, 'Alice', 10),
    > (2, 'Bob', 20),
    > (3, 'Charlie', 30),
    > (4, 'David', 40);
Query ID = root_20251001024437_11f9277b-03ac-4269-9653-6d456ff3691a
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Job running in-process (local Hadoop)
2025-10-01 02:44:43,584 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_local435717850_0001
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to directory file:/user/hive/warehouse/employees/.hive-staging_hive_2025-10-01_02-44-37_664_8027613207425629673-1/-ext-1000
0
Loading data to table default.employees
MapReduce Jobs Launched:
Stage-Stage-1:  HDFS Read: 0 HDFS Write: 0 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
```



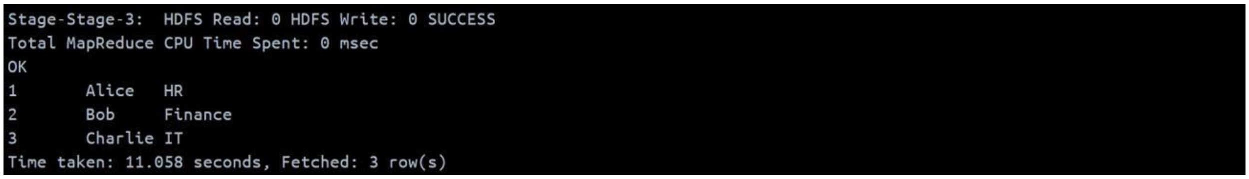
```
OK
Time taken: 6.565 seconds
hive> INSERT INTO TABLE departments VALUES
    > (10, 'HR'),
    > (20, 'Finance'),
    > (30, 'IT'),
    > (50, 'Marketing');
Query ID = root_20251001024458_a36f488d-d4a3-4b63-ae58-6da8fc1c41d4
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Job running in-process (local Hadoop)
2025-10-01 02:44:59,986 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_local837987427_0002
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to directory file:/user/hive/warehouse/departments/.hive-staging_hive_2025-10-01_02-44-58_268_6380290467196927081-1/-ext-10
000
Loading data to table default.departments
MapReduce Jobs Launched:
Stage-Stage-1:  HDFS Read: 0 HDFS Write: 0 SUCCESS
```



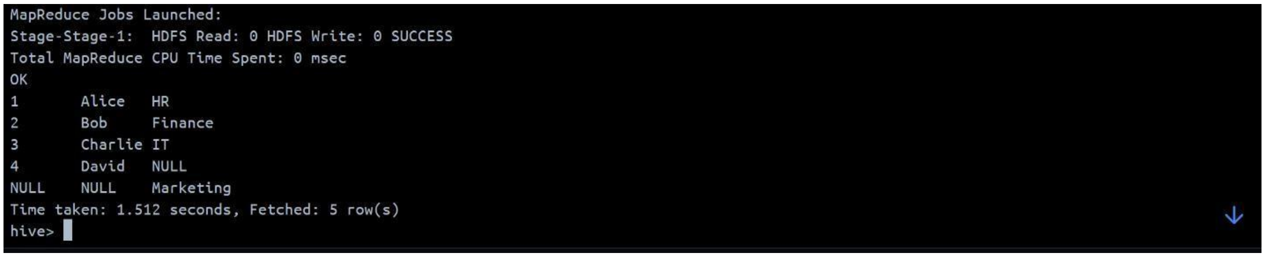
```
Total MapReduce CPU Time Spent: 0 msec
OK
1       Alice   HR
2       Bob     Finance
3       Charlie IT
NULL    NULL    Marketing
Time taken: 10.323 seconds, Fetched: 4 row(s)
```



```
Total MapReduce CPU Time Spent: 0 msec
OK
1       Alice   HR
2       Bob     Finance
3       Charlie IT
4       David   NULL
Time taken: 10.756 seconds, Fetched: 4 row(s)
```



```
Stage-Stage-3:  HDFS Read: 0 HDFS Write: 0 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
1       Alice   HR
2       Bob     Finance
3       Charlie IT
Time taken: 11.058 seconds, Fetched: 3 row(s)
```



```
MapReduce Jobs Launched:
Stage-Stage-1:  HDFS Read: 0 HDFS Write: 0 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
1       Alice   HR
2       Bob     Finance
3       Charlie IT
4       David   NULL
NULL    NULL    Marketing
Time taken: 1.512 seconds, Fetched: 5 row(s)
hive>
```

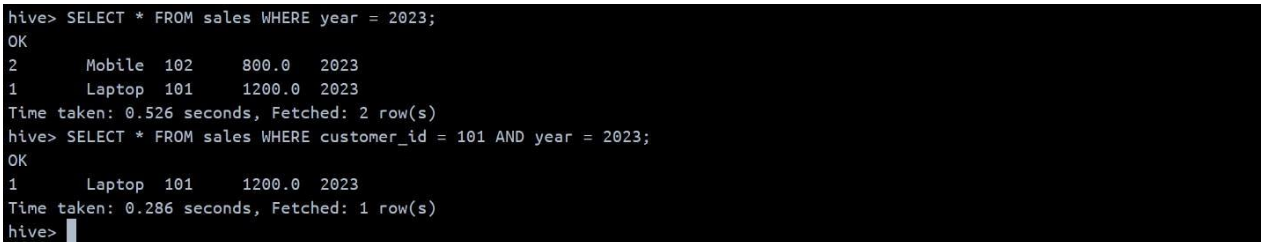
RESULT:

3. PARTITIONING AND BUCKETING IN HIVE

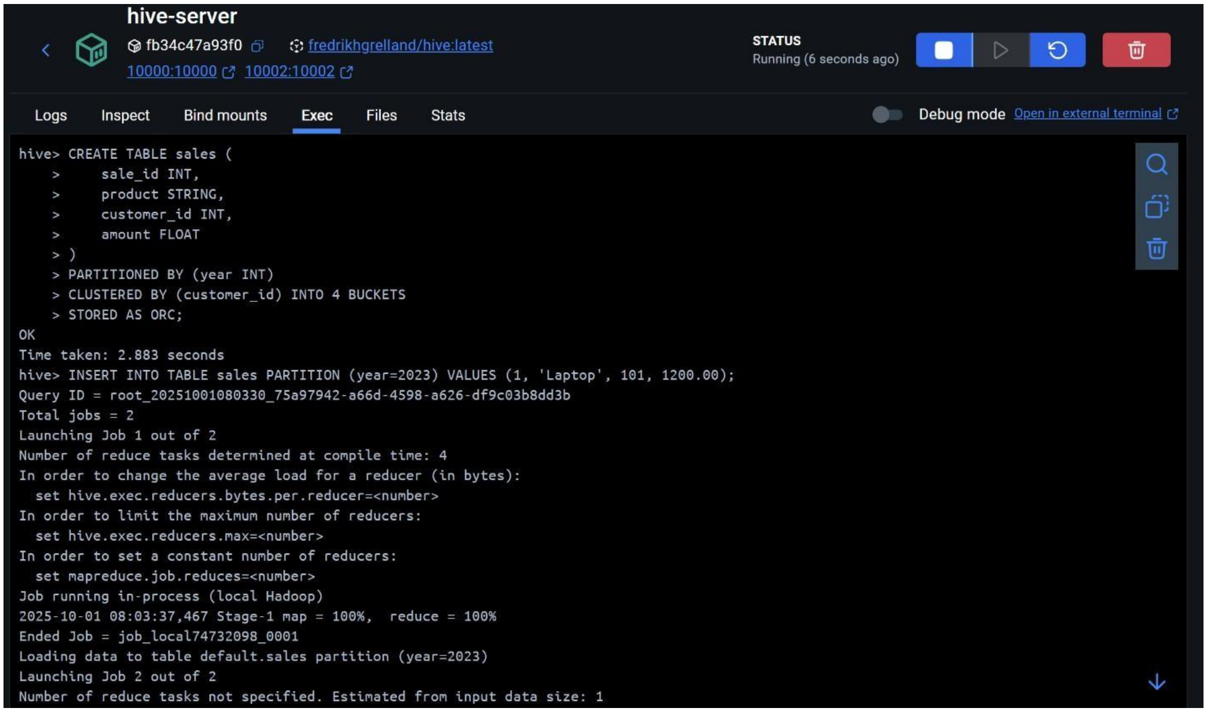
PROGRAM:

```
CREATE TABLE sales (
    sale_id INT,
    product STRING,
    customer_id INT,
    amount FLOAT
)
PARTITIONED BY (year INT)
CLUSTERED BY (customer_id) INTO 4 BUCKETS
STORED AS ORC;
INSERT INTO TABLE sales PARTITION (year=2023) VALUES (1, 'Laptop', 101, 1200.00);
INSERT INTO TABLE sales PARTITION (year=2023) VALUES (2, 'Mobile', 102, 800.00);
INSERT INTO TABLE sales PARTITION (year=2024) VALUES (3, 'Tablet', 101, 500.00);
SELECT * FROM sales WHERE year = 2023;
SELECT * FROM sales WHERE customer_id = 101 AND year = 2023;
```

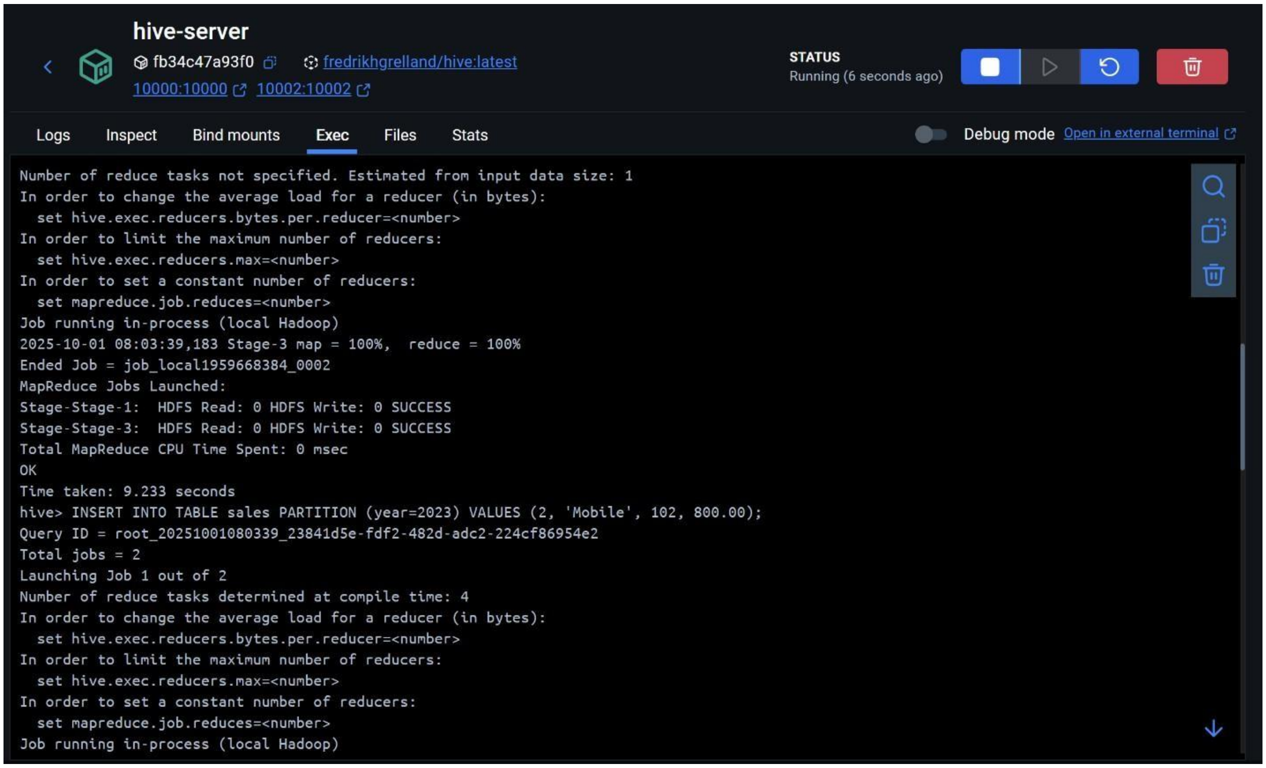
OUTPUT:



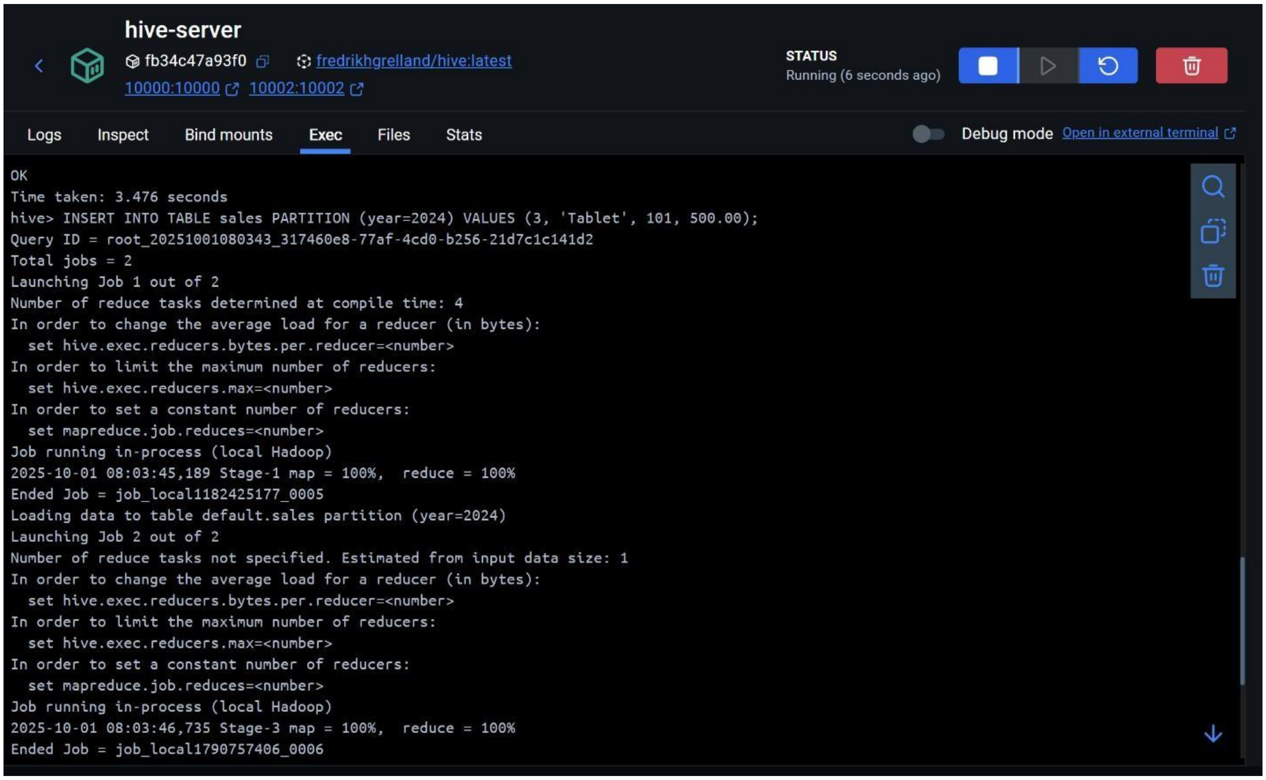
```
hive> SELECT * FROM sales WHERE year = 2023;
OK
2       Mobile  102     800.0   2023
1       Laptop  101     1200.0  2023
Time taken: 0.526 seconds, Fetched: 2 row(s)
hive> SELECT * FROM sales WHERE customer_id = 101 AND year = 2023;
OK
1       Laptop  101     1200.0  2023
Time taken: 0.286 seconds, Fetched: 1 row(s)
hive> 
```



```
hive> CREATE TABLE sales (
    >     sale_id INT,
    >     product STRING,
    >     customer_id INT,
    >     amount FLOAT
    > )
    > PARTITIONED BY (year INT)
    > CLUSTERED BY (customer_id) INTO 4 BUCKETS
    > STORED AS ORC;
OK
Time taken: 2.883 seconds
hive> INSERT INTO TABLE sales PARTITION (year=2023) VALUES (1, 'Laptop', 101, 1200.00);
Query ID = root_20251001080330_75a97942-a66d-4598-a626-df9c03b8dd3b
Total jobs = 2
Launching Job 1 out of 2
Number of reduce tasks determined at compile time: 4
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Job running in-process (local Hadoop)
2025-10-01 08:03:37,467 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_local74732098_0001
Loading data to table default.sales partition (year=2023)
Launching Job 2 out of 2
Number of reduce tasks not specified. Estimated from input data size: 1
```



```
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Job running in-process (local Hadoop)
2025-10-01 08:03:39,183 Stage-3 map = 100%,  reduce = 100%
Ended Job = job_local1959668384_0002
MapReduce Jobs Launched:
Stage-Stage-1:  HDFS Read: 0 HDFS Write: 0 SUCCESS
Stage-Stage-3:  HDFS Read: 0 HDFS Write: 0 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
Time taken: 9.233 seconds
hive> INSERT INTO TABLE sales PARTITION (year=2023) VALUES (2, 'Mobile', 102, 800.00);
Query ID = root_20251001080339_23841d5e-fdf2-482d-adc2-224cf86954e2
Total jobs = 2
Launching Job 1 out of 2
Number of reduce tasks determined at compile time: 4
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Job running in-process (local Hadoop)
```



```
OK
Time taken: 3.476 seconds
hive> INSERT INTO TABLE sales PARTITION (year=2024) VALUES (3, 'Tablet', 101, 500.00);
Query ID = root_20251001080343_317460e8-77af-4cd0-b256-21d7c1c141d2
Total jobs = 2
Launching Job 1 out of 2
Number of reduce tasks determined at compile time: 4
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Job running in-process (local Hadoop)
2025-10-01 08:03:45,189 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_local1182425177_0005
Loading data to table default.sales partition (year=2024)
Launching Job 2 out of 2
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Job running in-process (local Hadoop)
2025-10-01 08:03:46,735 Stage-3 map = 100%,  reduce = 100%
Ended Job = job_local1790757406_0006
```

RESULT:

4. LOADING AND FILTERING DATA IN APACHE PIG

PROGRAM:

1. Load employee data (id, name, dept_id)

```
employees = LOAD 'emp.txt' USING PigStorage(',') AS (id:int, name:chararray, dept_id:int);
```

2. Load department data (dept_id, dept_name)

```
departments = LOAD 'dept.txt' USING PigStorage(',') AS (dept_id:int, dept_name:chararray);
```

3. Display employees

```
DUMP employees;
```

4. Display departments

```
DUMP departments;
```

5. Join employees with departments on dept_id

```
emp_dept = JOIN employees BY dept_id, departments BY dept_id;
```

6. Display joined data

```
DUMP emp_dept;
```

7. Filter employees who belong to department 10 (HR)

```
emp_hr = FILTER employees BY dept_id == 10;
```

8. Display filtered employees

```
DUMP emp_hr;
```

9. Group employees by department id

```
grouped_emps = GROUP employees BY dept_id;
```

10. Count employees per department

```
emp_count = FOREACH grouped_emps GENERATE group AS dept_id, COUNT(employees)
AS emp_num;
```

11. Display employee counts

```
DUMP emp_count;
```

EMP.txt

1,John,10

2,Alice,20

3,Bob,10

4,Eve,30

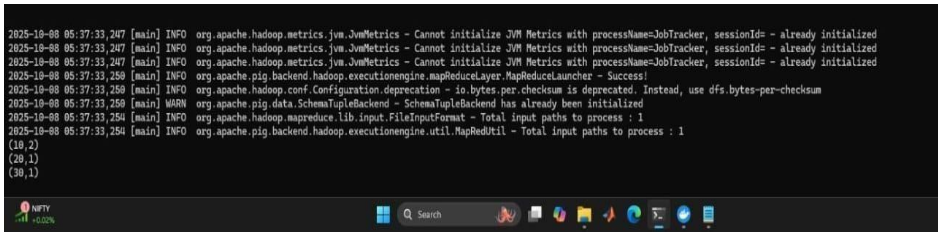
Dept.txt

10,HR

20,Engineering

30,Marketing

OUTPUT:



```
2025-10-08 05:37:33,247 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2025-10-08 05:37:33,247 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2025-10-08 05:37:33,247 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2025-10-08 05:37:33,250 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
2025-10-08 05:37:33,250 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
2025-10-08 05:37:33,250 [main] WARN  org.apache.pig.data.SchemaTupleBackend - SchemaTupleBackend has already been initialized
2025-10-08 05:37:33,254 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
2025-10-08 05:37:33,254 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(10,2)
(20,1)
(30,1)
```

RESULT:

5. AGGREGATION AND SORTING USING APACHE PIG

PROGRAM:

1. Load the Data

```
emp = LOAD 'emp.txt' USING PigStorage(',')
    AS (id:int, name:chararray, dept:chararray, salary:int);
```

2. Display Raw Data

```
DUMP emp;
```

3. Extract Required Fields with normalized case and trimmed whitespace

```
emp_subset = FOREACH emp GENERATE
   TRIM(LOWER(name)) AS name,
   TRIM(UPPER(dept)) AS dept,
   salary;
```

4. Remove Duplicates

```
distinct_emp = DISTINCT emp_subset;
DUMP distinct_emp;
```

5. Sort by salary (ascending)

```
emp_sorted = ORDER distinct_emp BY salary ASC;
DUMP emp_sorted;
```

6. Group by Department

```
emp_grouped = GROUP distinct_emp BY dept;
DUMP emp_grouped;
```

7. Average Salary per Department

```
emp_avg = FOREACH emp_grouped GENERATE
   group AS department,
   AVG(distinct_emp.salary) AS avg_salary;
DUMP emp_avg;
```

8. Count total employees per Department

```
emp_count = FOREACH emp_grouped GENERATE
   group AS department,
   COUNT(distinct_emp) AS total_employees;
DUMP emp_count;
```

9. Max and Min salary per Department

```
emp_minmax = FOREACH emp_grouped GENERATE
   group AS department,
   MAX(distinct_emp.salary) AS max_salary,
   MIN(distinct_emp.salary) AS min_salary;
DUMP emp_minmax;
```

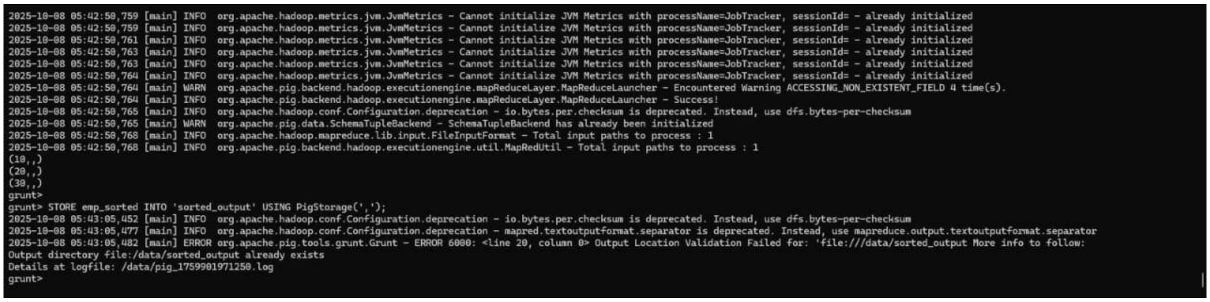
10. Store the sorted employee data

```
STORE emp_sorted INTO 'sorted_output' USING PigStorage(',');
```

Emp.txt:

```
1,John,10
2,Alice,20
3,Bob,10
4,Eve,30
```

OUTPUT:



```
2025-10-08 05:42:50,759 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2025-10-08 05:42:50,759 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2025-10-08 05:42:50,761 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2025-10-08 05:42:50,763 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2025-10-08 05:42:50,763 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2025-10-08 05:42:50,764 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2025-10-08 05:42:50,764 [main] WARN  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Encountered Warning ACCESSING_NON_EXISTENT_FIELD 4 time(s).
2025-10-08 05:42:50,764 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
2025-10-08 05:42:50,765 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
2025-10-08 05:42:50,765 [main] WARN  org.apache.pig.data.SchemaTupleBackend - SchemaTupleBackend has already been initialized
2025-10-08 05:42:50,768 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
2025-10-08 05:42:50,768 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(10,,)
(20,,)
(30,,)
grunt>
grunt> STORE emp_sorted INTO 'sorted_output' USING PigStorage(',');
2025-10-08 05:43:05,452 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
2025-10-08 05:43:05,477 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.textoutputformat.separator is deprecated. Instead, use mapreduce.output.textoutputformat.separator
2025-10-08 05:43:05,482 [main] ERROR org.apache.pig.tools.grunt.Grunt - ERROR 6000: <line 20, column 0> Output Location Validation Failed for: 'file:///data/sorted_output More info to follow:
Output directory file:/data/sorted_output already exists
Details at logfile: /data/pig_1759901971250.log
grunt>
```

RESULT:

6. CRUD OPERATIONS IN HBASE USING HBASE SHELL

PROGRAM:

1. Create Tables

```
create 'students01', 'info'
create 'courses01', 'details'
create 'faculty01', 'profile'
```

2. Insert Data (Create)

Students

```
put 'students01', 'row1', 'info:name', 'Keerthana'
put 'students01', 'row1', 'info:dept', 'CSE'
put 'students01', 'row1', 'info:year', '3'

put 'students01', 'row2', 'info:name', 'Arun'
put 'students01', 'row2', 'info:dept', 'ECE'
put 'students01', 'row2', 'info:year', '2'
```

Courses

```
put 'courses01', 'c101', 'details:name', 'Data Structures'
put 'courses01', 'c101', 'details:credits', '4'

put 'courses01', 'c102', 'details:name', 'Database Systems'
put 'courses01', 'c102', 'details:credits', '3'
```

Faculty

```
put 'faculty01', 'F001', 'profile:name', 'Dr. Meena'
put 'faculty01', 'F001', 'profile:dept', 'CSE'

put 'faculty01', 'F002', 'profile:name', 'Dr. Ravi'
put 'faculty01', 'F002', 'profile:dept', 'ECE'
```

3. Read Data (Retrieve)

Single row from each table

```
get 'students01', 'row1'

get 'courses01', 'c101'

get 'faculty01', 'F001'
```

4. Scan all rows

```
scan 'students01'

scan  'courses01'

scan 'faculty01'
```

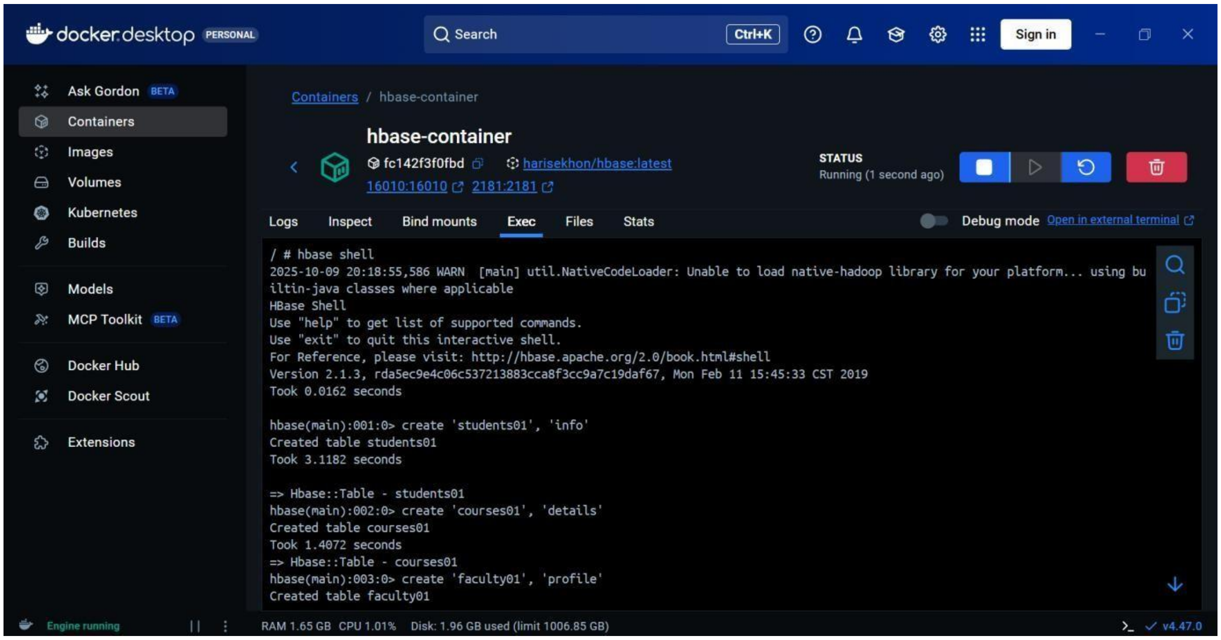
5. Update Data

```
put 'students01', 'row1', 'info:dept', 'IT'

put 'courses01', 'c101', 'details:credits', '5'

put 'faculty01', 'F002', 'profile:dept', 'EEE'
```

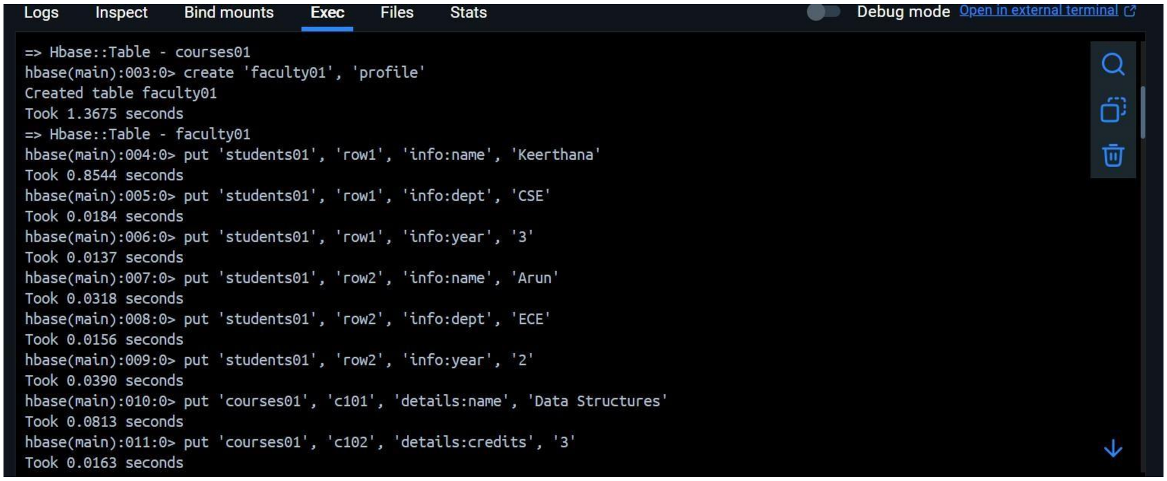
6. Delete Data

```
delete 'students01', 'row2', 'info:year'

deleteall 'courses01', 'c102'

deleteall 'faculty01', 'F001'
```

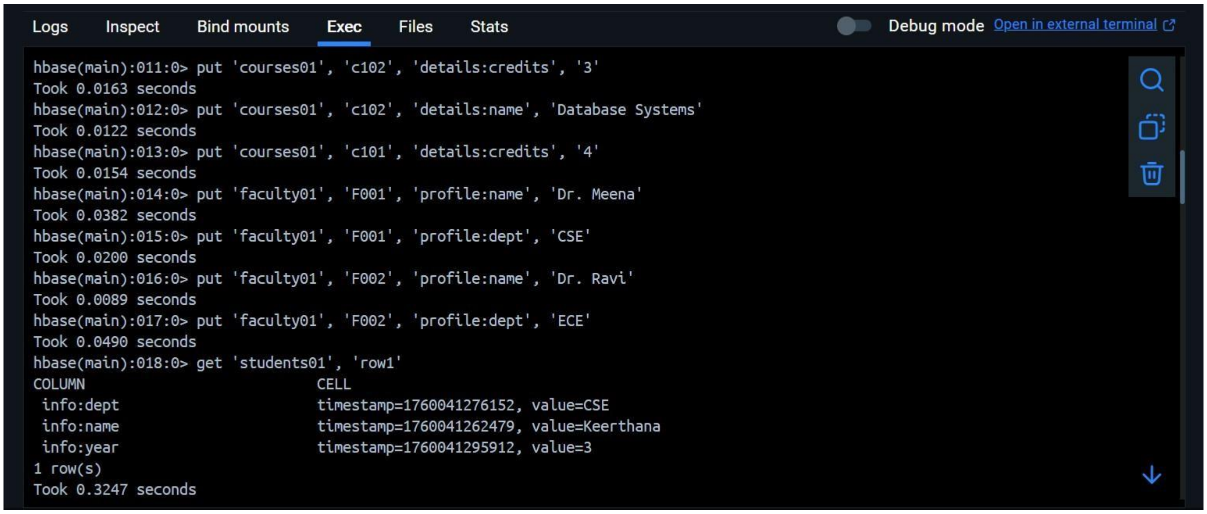
OUTPUT:



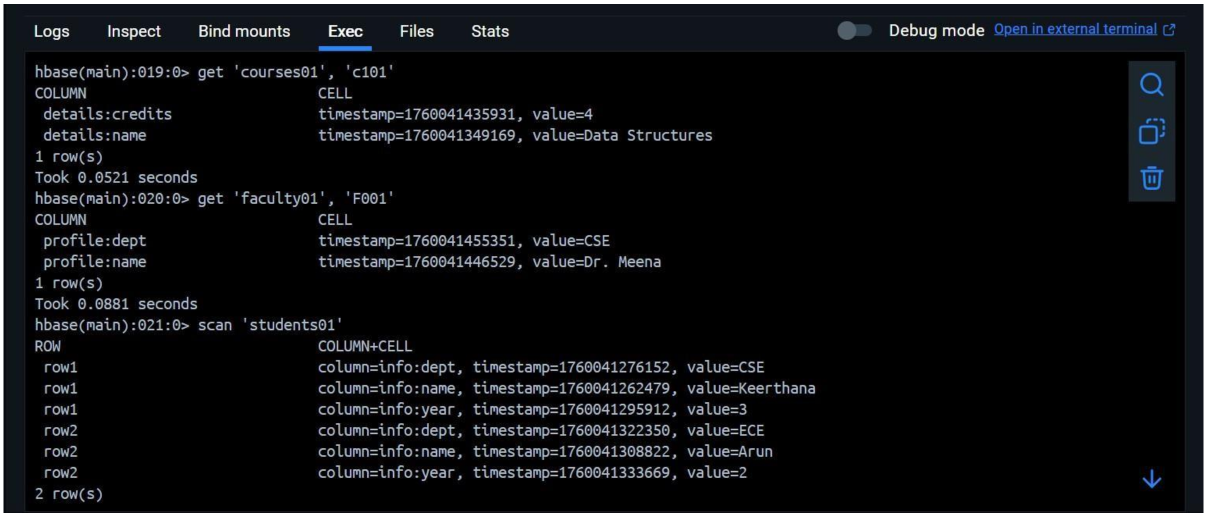
```
/ # hbase shell
2025-10-09 20:18:55,586 WARN  [main] util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using bu
iltin-java classes where applicable
HBase Shell
Use "help" to get list of supported commands.
Use "exit" to quit this interactive shell.
For Reference, please visit: http://hbase.apache.org/2.0/book.html#shell
Version 2.1.3, rda5ec9e4c06c537213883cca8f3cc9a7c19daf67, Mon Feb 11 15:45:33 CST 2019
Took 0.0162 seconds

hbase(main):001:0> create 'students01', 'info'
Created table students01
Took 3.1182 seconds

=> Hbase::Table - students01
hbase(main):002:0> create 'courses01', 'details'
Created table courses01
Took 1.4072 seconds
=> Hbase::Table - courses01
hbase(main):003:0> create 'faculty01', 'profile'
Created table faculty01
```



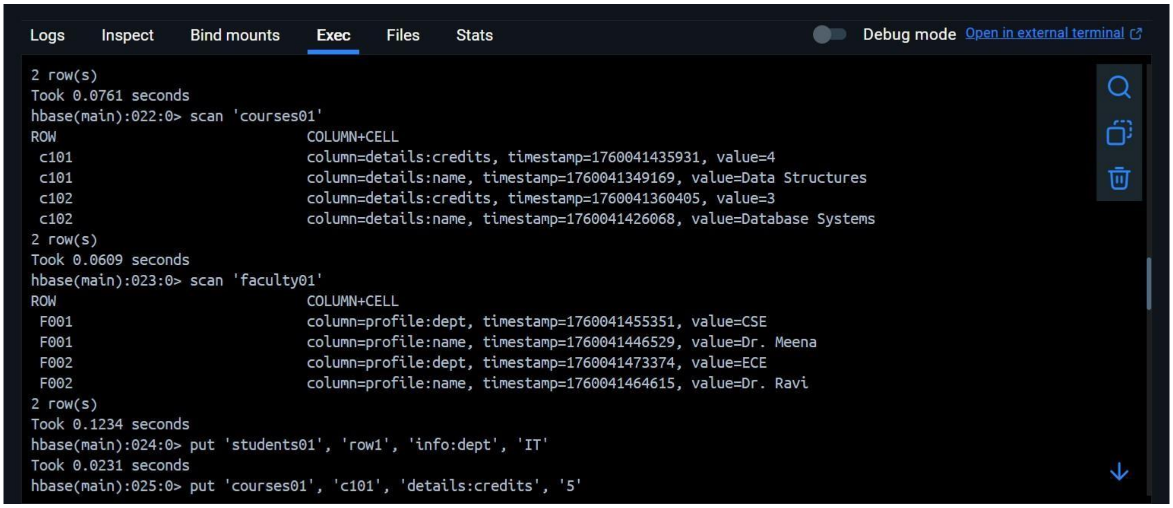
```
=> Hbase::Table - courses01
hbase(main):003:0> create 'faculty01', 'profile'
Created table faculty01
Took 1.3675 seconds
=> Hbase::Table - faculty01
hbase(main):004:0> put 'students01', 'row1', 'info:name', 'Keerthana'
Took 0.8544 seconds
hbase(main):005:0> put 'students01', 'row1', 'info:dept', 'CSE'
Took 0.0184 seconds
hbase(main):006:0> put 'students01', 'row1', 'info:year', '3'
Took 0.0137 seconds
hbase(main):007:0> put 'students01', 'row2', 'info:name', 'Arun'
Took 0.0318 seconds
hbase(main):008:0> put 'students01', 'row2', 'info:dept', 'ECE'
Took 0.0156 seconds
hbase(main):009:0> put 'students01', 'row2', 'info:year', '2'
Took 0.0390 seconds
hbase(main):010:0> put 'courses01', 'c101', 'details:name', 'Data Structures'
Took 0.0813 seconds
hbase(main):011:0> put 'courses01', 'c102', 'details:credits', '3'
Took 0.0163 seconds
```



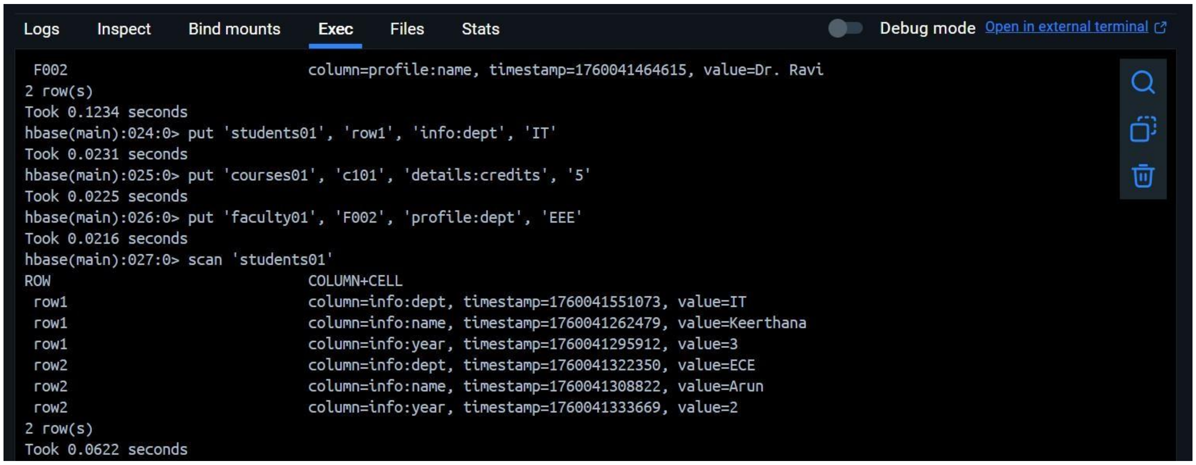
```
hbase(main):011:0> put 'courses01', 'c102', 'details:credits', '3'
Took 0.0163 seconds
hbase(main):012:0> put 'courses01', 'c102', 'details:name', 'Database Systems'
Took 0.0122 seconds
hbase(main):013:0> put 'courses01', 'c101', 'details:credits', '4'
Took 0.0154 seconds
hbase(main):014:0> put 'faculty01', 'F001', 'profile:name', 'Dr. Meena'
Took 0.0382 seconds
hbase(main):015:0> put 'faculty01', 'F001', 'profile:dept', 'CSE'
Took 0.0200 seconds
hbase(main):016:0> put 'faculty01', 'F002', 'profile:name', 'Dr. Ravi'
Took 0.0089 seconds
hbase(main):017:0> put 'faculty01', 'F002', 'profile:dept', 'ECE'
Took 0.0490 seconds
hbase(main):018:0> get 'students01', 'row1'
COLUMN                          CELL
 info:dept                      timestamp=1760041276152, value=CSE
 info:name                      timestamp=1760041262479, value=Keerthana
 info:year                      timestamp=1760041295912, value=3
1 row(s)
Took 0.3247 seconds
```



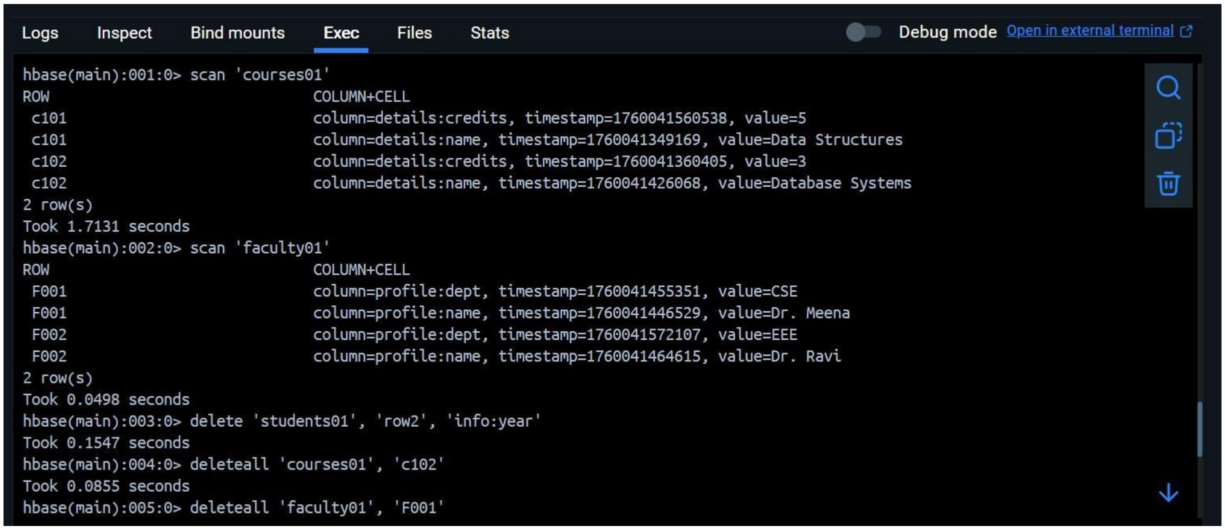
```
hbase(main):019:0> get 'courses01', 'c101'
COLUMN                          CELL
 details:credits                timestamp=1760041435931, value=4
 details:name                   timestamp=1760041349169, value=Data Structures
1 row(s)
Took 0.0521 seconds
hbase(main):020:0> get 'faculty01', 'F001'
COLUMN                          CELL
 profile:dept                   timestamp=1760041455351, value=CSE
 profile:name                   timestamp=1760041446529, value=Dr. Meena
1 row(s)
Took 0.0881 seconds
hbase(main):021:0> scan 'students01'
ROW                             COLUMN+CELL
 row1                           column=info:dept, timestamp=1760041276152, value=CSE
 row1                           column=info:name, timestamp=1760041262479, value=Keerthana
 row1                           column=info:year, timestamp=1760041295912, value=3
 row2                           column=info:dept, timestamp=1760041322350, value=ECE
 row2                           column=info:name, timestamp=1760041308822, value=Arun
 row2                           column=info:year, timestamp=1760041333669, value=2
2 row(s)
```



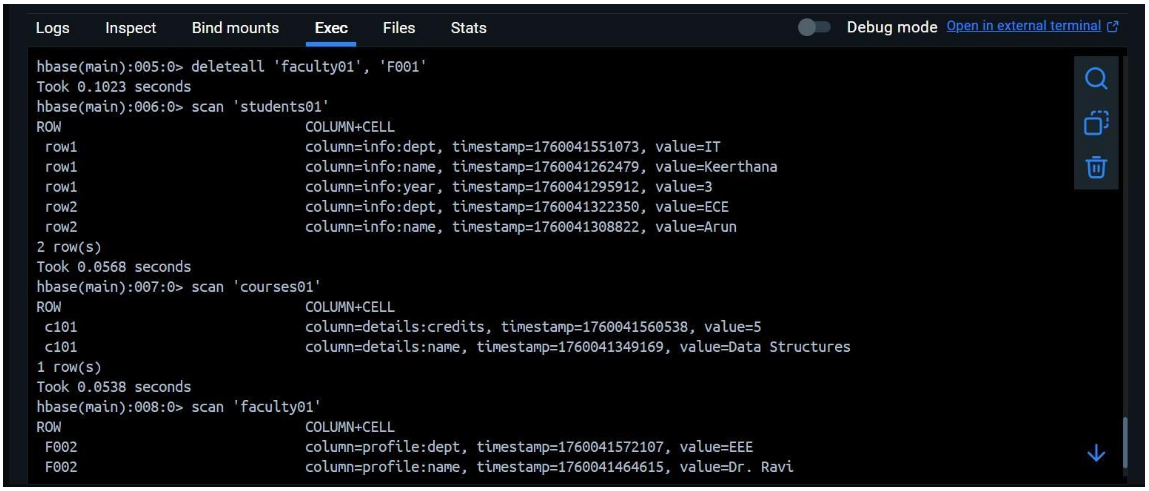
```
2 row(s)
Took 0.0761 seconds
hbase(main):022:0> scan 'courses01'
ROW                             COLUMN+CELL
 c101                           column=details:credits, timestamp=1760041435931, value=4
 c101                           column=details:name, timestamp=1760041349169, value=Data Structures
 c102                           column=details:credits, timestamp=1760041360405, value=3
 c102                           column=details:name, timestamp=1760041426068, value=Database Systems
2 row(s)
Took 0.0609 seconds
hbase(main):023:0> scan 'faculty01'
ROW                             COLUMN+CELL
 F001                           column=profile:dept, timestamp=1760041455351, value=CSE
 F001                           column=profile:name, timestamp=1760041446529, value=Dr. Meena
 F002                           column=profile:dept, timestamp=1760041473374, value=ECE
 F002                           column=profile:name, timestamp=1760041464615, value=Dr. Ravi
2 row(s)
Took 0.1234 seconds
hbase(main):024:0> put 'students01', 'row1', 'info:dept', 'IT'
Took 0.0231 seconds
hbase(main):025:0> put 'courses01', 'c101', 'details:credits', '5'
```



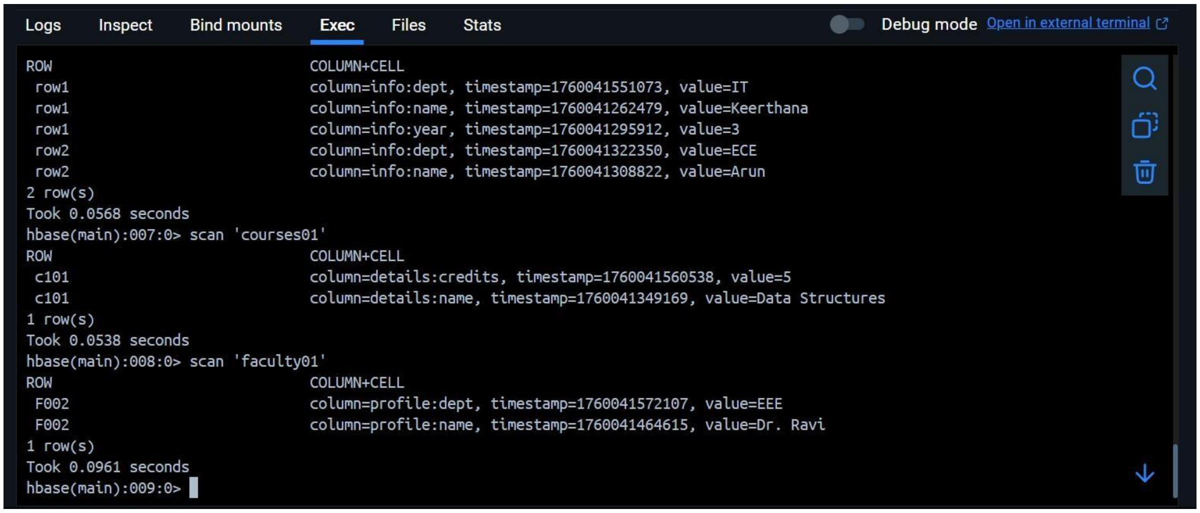
```
 F002                           column=profile:name, timestamp=1760041464615, value=Dr. Ravi
2 row(s)
Took 0.1234 seconds
hbase(main):024:0> put 'students01', 'row1', 'info:dept', 'IT'
Took 0.0231 seconds
hbase(main):025:0> put 'courses01', 'c101', 'details:credits', '5'
Took 0.0225 seconds
hbase(main):026:0> put 'faculty01', 'F002', 'profile:dept', 'EEE'
Took 0.0216 seconds
hbase(main):027:0> scan 'students01'
ROW                             COLUMN+CELL
 row1                           column=info:dept, timestamp=1760041551073, value=IT
 row1                           column=info:name, timestamp=1760041262479, value=Keerthana
 row1                           column=info:year, timestamp=1760041295912, value=3
 row2                           column=info:dept, timestamp=1760041322350, value=ECE
 row2                           column=info:name, timestamp=1760041308822, value=Arun
 row2                           column=info:year, timestamp=1760041333669, value=2
2 row(s)
Took 0.0622 seconds
```



```
hbase(main):001:0> scan 'courses01'
ROW                             COLUMN+CELL
 c101                           column=details:credits, timestamp=1760041560538, value=5
 c101                           column=details:name, timestamp=1760041349169, value=Data Structures
 c102                           column=details:credits, timestamp=1760041360405, value=3
 c102                           column=details:name, timestamp=1760041426068, value=Database Systems
2 row(s)
Took 1.7131 seconds
hbase(main):002:0> scan 'faculty01'
ROW                             COLUMN+CELL
 F001                           column=profile:dept, timestamp=1760041455351, value=CSE
 F001                           column=profile:name, timestamp=1760041446529, value=Dr. Meena
 F002                           column=profile:dept, timestamp=1760041572107, value=EEE
 F002                           column=profile:name, timestamp=1760041464615, value=Dr. Ravi
2 row(s)
Took 0.0498 seconds
hbase(main):003:0> delete 'students01', 'row2', 'info:year'
Took 0.1547 seconds
hbase(main):004:0> deleteall 'courses01', 'c102'
Took 0.0855 seconds
hbase(main):005:0> deleteall 'faculty01', 'F001'
```



```
hbase(main):005:0> deleteall 'faculty01', 'F001'
Took 0.1023 seconds
hbase(main):006:0> scan 'students01'
ROW                             COLUMN+CELL
 row1                           column=info:dept, timestamp=1760041551073, value=IT
 row1                           column=info:name, timestamp=1760041262479, value=Keerthana
 row1                           column=info:year, timestamp=1760041295912, value=3
 row2                           column=info:dept, timestamp=1760041322350, value=ECE
 row2                           column=info:name, timestamp=1760041308822, value=Arun
2 row(s)
Took 0.0568 seconds
hbase(main):007:0> scan 'courses01'
ROW                             COLUMN+CELL
 c101                           column=details:credits, timestamp=1760041560538, value=5
 c101                           column=details:name, timestamp=1760041349169, value=Data Structures
1 row(s)
Took 0.0538 seconds
hbase(main):008:0> scan 'faculty01'
ROW                             COLUMN+CELL
 F002                           column=profile:dept, timestamp=1760041572107, value=EEE
 F002                           column=profile:name, timestamp=1760041464615, value=Dr. Ravi
```



```
ROW                                COLUMN+CELL
 row1                              column=info:dept, timestamp=1760041551073, value=IT
 row1                              column=info:name, timestamp=1760041262479, value=Keerthana
 row1                              column=info:year, timestamp=1760041295912, value=3
 row2                              column=info:dept, timestamp=1760041322350, value=ECE
 row2                              column=info:name, timestamp=1760041308822, value=Arun
2 row(s)
Took 0.0568 seconds
hbase(main):007:0> scan 'courses01'
ROW                                COLUMN+CELL
 c101                              column=details:credits, timestamp=1760041560538, value=5
 c101                              column=details:name, timestamp=1760041349169, value=Data Structures
1 row(s)
Took 0.0538 seconds
hbase(main):008:0> scan 'faculty01'
ROW                                COLUMN+CELL
 F002                              column=profile:dept, timestamp=1760041572107, value=EEE
 F002                              column=profile:name, timestamp=1760041464615, value=Dr. Ravi
1 row(s)
Took 0.0961 seconds
hbase(main):009:0> 
```

RESULT:

7. ADMINISTRATIVE OPERATIONS FOR MANAGING HBASE TABLES

PROGRAM:

1. Create Tables with Column Families

```
create 'students02', 'info', 'contact'

create 'courses02', 'details'

create 'faculty02', 'profile', 'contact'
```

2. Insert Table Attributes (Data)

Students

```
put 'students02', 'row1', 'info:name', 'Kiruthika'

put 'students02', 'row1', 'info:dept', 'CSE'

put 'students02', 'row1', 'info:year', '3'

put 'students02', 'row1', 'contact:email', 'kiru@college.edu'


put 'students02', 'row2', 'info:name', 'Arun'

put 'students02', 'row2', 'info:dept', 'ECE'

put 'students02', 'row2', 'info:year', '2'

put 'students02', 'row2', 'contact:email', 'arun@college.edu'
```

Courses

```
put 'courses02', 'c101', 'details:name', 'Data Structures'

put 'courses02', 'c101', 'details:credits', '4'

put 'courses02', 'c101', 'details:faculty', 'F001'


put 'courses02', 'c102', 'details:name', 'Database Systems'

put 'courses02', 'c102', 'details:credits', '3'

put 'courses02', 'c102', 'details:faculty', 'F002'
```

Faculty

```
put 'faculty02', 'F001', 'profile:name', 'Dr. Meena'

put 'faculty02', 'F001', 'profile:dept', 'CSE'

put 'faculty02', 'F001', 'contact:email', 'meena@college.edu'


put 'faculty02', 'F002', 'profile:name', 'Dr. Ravi'

put 'faculty02', 'F002', 'profile:dept', 'ECE'

put 'faculty02', 'F002', 'contact:email', 'ravi@college.edu'
```

3. Scan all rows

```
scan 'students02'

scan  'courses02'

scan 'faculty02'
```

4. Update Data

```
put 'students02', 'row1', 'info:dept', 'IT'

put 'courses02', 'c101', 'details:credits', '5'

put 'faculty02', 'F002', 'profile:dept', 'EEE'
```

5. Delete Data

```
delete 'students02', 'row2', 'info:year'

deleteall 'courses02', 'c102'

deleteall 'faculty02', 'F001'
```

6. Check table existence

```
exists 'students02'

exists 'faculty02'
```

7. Describe table schema

```
describe 'students02'
```

8. Alter table schema

alter 'students02', 'add' => 'extra'

alter 'students02', 'delete' => 'extra'

9. Disable / Enable table

disable 'students02'

enable 'students02'

10. Truncate table (delete all rows, keep schema)

truncate 'students02'

11.1Drop tables

disable 'students02'

drop 'students02'

disable 'courses02'

drop 'courses02'

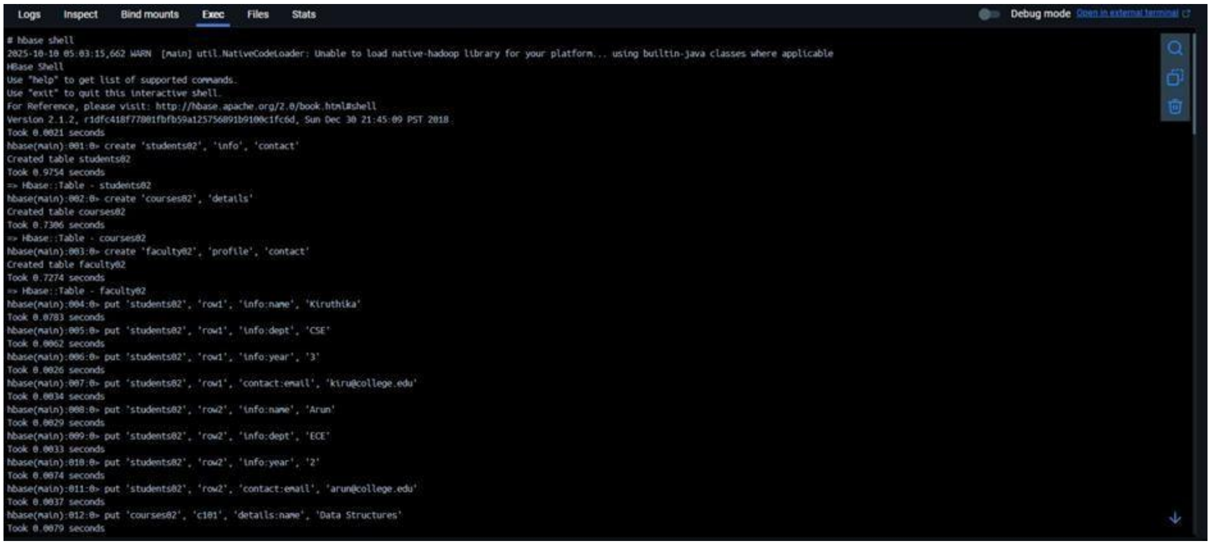
disable 'faculty02'

drop 'faculty02'

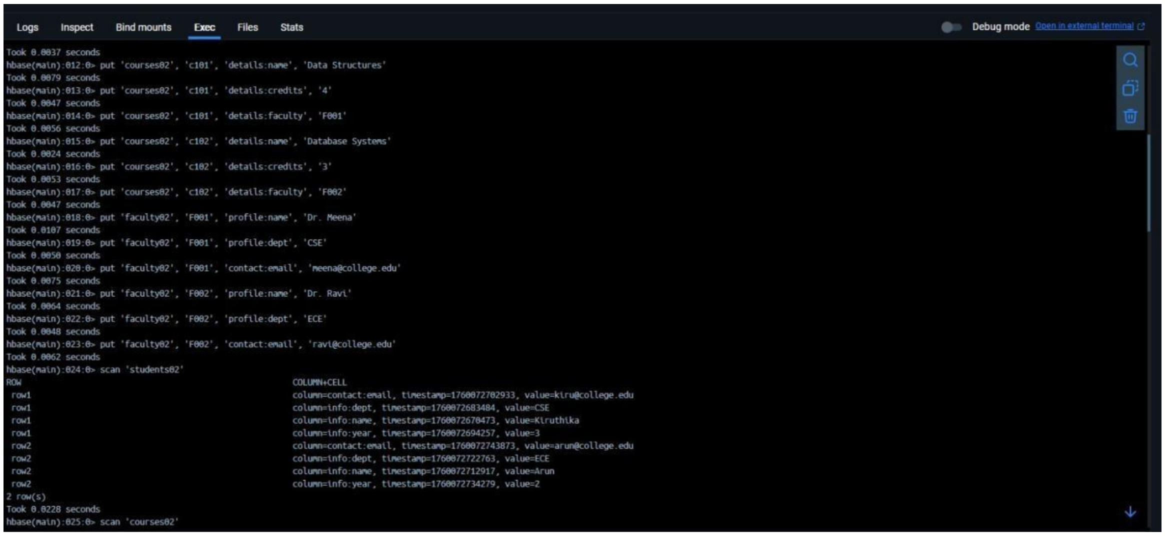
12.Verify tables removed

List

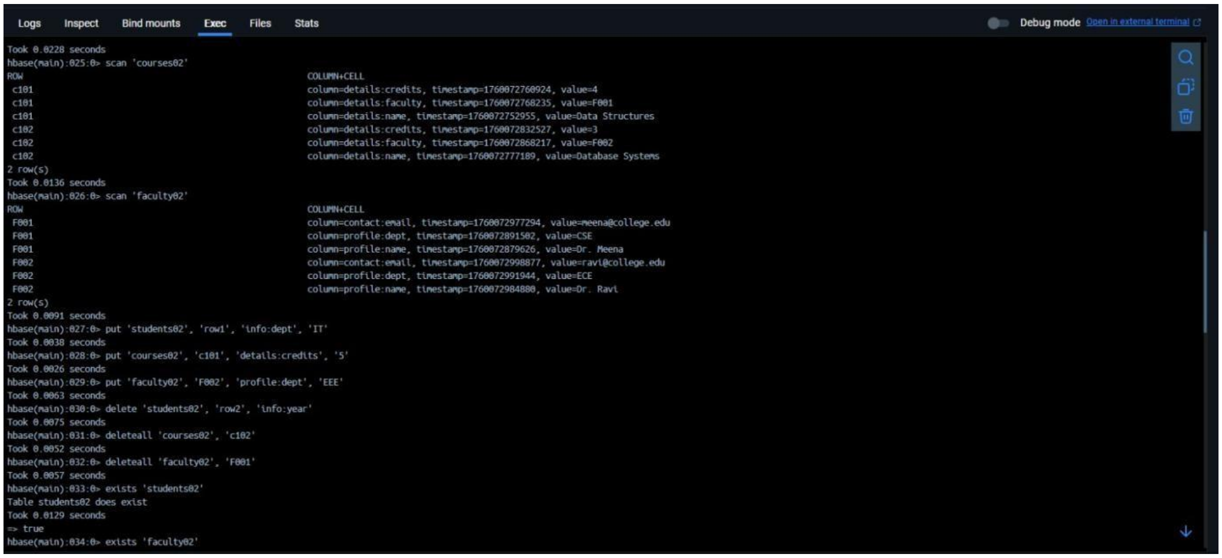
OUTPUT:



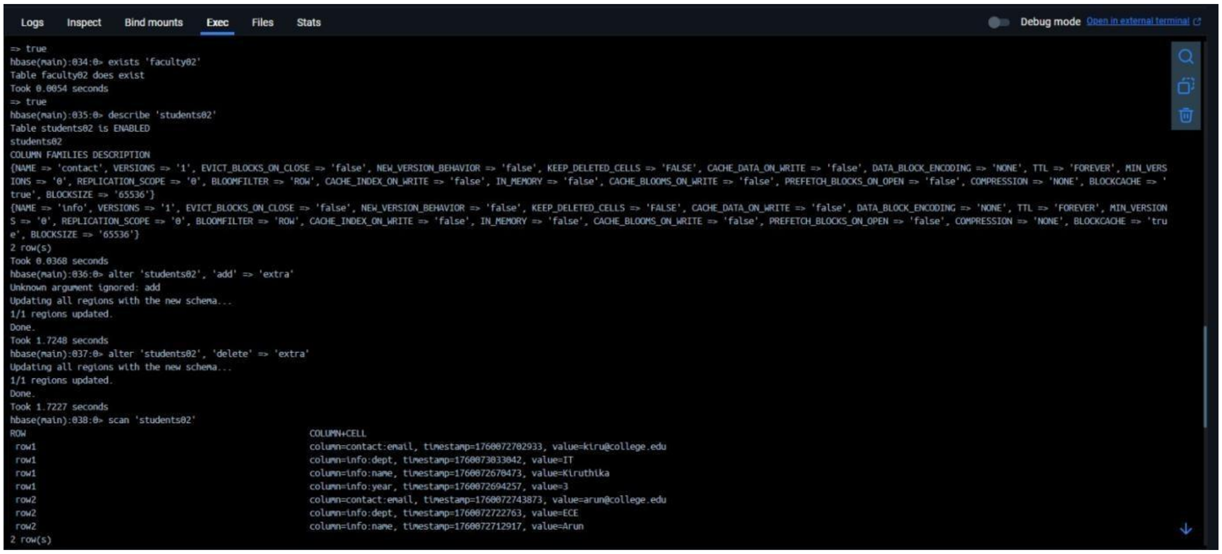
```
# hbase shell
2025-10-10 05:03:15,662 WARN  [main] util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
HBase Shell
Use "help" to get list of supported commands.
Use "exit" to quit this interactive shell.
For Reference, please visit: http://hbase.apache.org/2.0/book.html#shell
Version 2.1.2, r1dfc418f77801fbfb59a125756891b9100c1fc6d, Sun Dec 30 21:45:09 PST 2018
Took 0.0021 seconds
hbase(main):001:0> create 'students02', 'info', 'contact'
Created table students02
Took 0.9754 seconds
=> Hbase::Table - students02
hbase(main):002:0> create 'courses02', 'details'
Created table courses02
Took 0.7306 seconds
=> Hbase::Table - courses02
hbase(main):003:0> create 'faculty02', 'profile', 'contact'
Created table faculty02
Took 0.7274 seconds
=> Hbase::Table - faculty02
hbase(main):004:0> put 'students02', 'row1', 'info:name', 'Kiruthika'
Took 0.0783 seconds
hbase(main):005:0> put 'students02', 'row1', 'info:dept', 'CSE'
Took 0.0062 seconds
hbase(main):006:0> put 'students02', 'row1', 'info:year', '3'
Took 0.0026 seconds
hbase(main):007:0> put 'students02', 'row1', 'contact:email', 'kiru@college.edu'
Took 0.0034 seconds
hbase(main):008:0> put 'students02', 'row2', 'info:name', 'Arun'
Took 0.0029 seconds
hbase(main):009:0> put 'students02', 'row2', 'info:dept', 'ECE'
Took 0.0033 seconds
hbase(main):010:0> put 'students02', 'row2', 'info:year', '2'
Took 0.0074 seconds
hbase(main):011:0> put 'students02', 'row2', 'contact:email', 'arun@college.edu'
Took 0.0037 seconds
hbase(main):012:0> put 'courses02', 'c101', 'details:name', 'Data Structures'
Took 0.0079 seconds
```



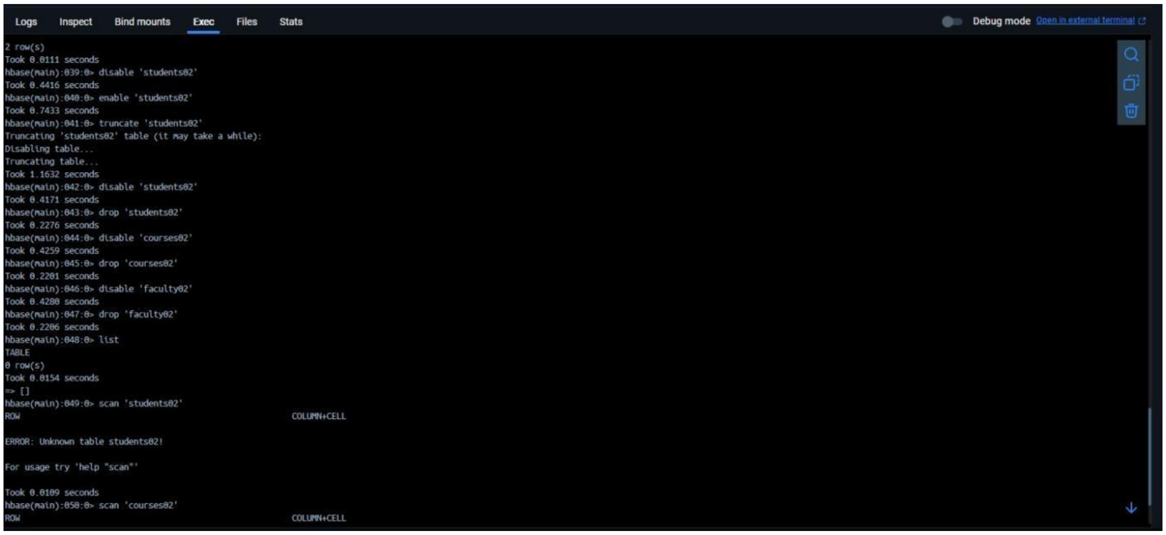
```
Took 0.0037 seconds
hbase(main):012:0> put 'courses02', 'c101', 'details:name', 'Data Structures'
Took 0.0079 seconds
hbase(main):013:0> put 'courses02', 'c101', 'details:credits', '4'
Took 0.0047 seconds
hbase(main):014:0> put 'courses02', 'c101', 'details:faculty', 'F001'
Took 0.0056 seconds
hbase(main):015:0> put 'courses02', 'c102', 'details:name', 'Database Systems'
Took 0.0024 seconds
hbase(main):016:0> put 'courses02', 'c102', 'details:credits', '3'
Took 0.0053 seconds
hbase(main):017:0> put 'courses02', 'c102', 'details:faculty', 'F002'
Took 0.0047 seconds
hbase(main):018:0> put 'faculty02', 'F001', 'profile:name', 'Dr. Meena'
Took 0.0107 seconds
hbase(main):019:0> put 'faculty02', 'F001', 'profile:dept', 'CSE'
Took 0.0050 seconds
hbase(main):020:0> put 'faculty02', 'F001', 'contact:email', 'meena@college.edu'
Took 0.0075 seconds
hbase(main):021:0> put 'faculty02', 'F002', 'profile:name', 'Dr. Ravi'
Took 0.0064 seconds
hbase(main):022:0> put 'faculty02', 'F002', 'profile:dept', 'ECE'
Took 0.0048 seconds
hbase(main):023:0> put 'faculty02', 'F002', 'contact:email', 'ravi@college.edu'
Took 0.0062 seconds
hbase(main):024:0> scan 'students02'
ROW                                                        COLUMN+CELL
 row1                                                      column=contact:email, timestamp=1760072702933, value=kiru@college.edu
 row1                                                      column=info:dept, timestamp=1760072683484, value=CSE
 row1                                                      column=info:name, timestamp=1760072670473, value=Kiruthika
 row1                                                      column=info:year, timestamp=1760072694257, value=3
 row2                                                      column=contact:email, timestamp=1760072743873, value=arun@college.edu
 row2                                                      column=info:dept, timestamp=1760072722763, value=ECE
 row2                                                      column=info:name, timestamp=1760072712917, value=Arun
 row2                                                      column=info:year, timestamp=1760072734279, value=2
2 row(s)
Took 0.0228 seconds
hbase(main):025:0> scan 'courses02'
```



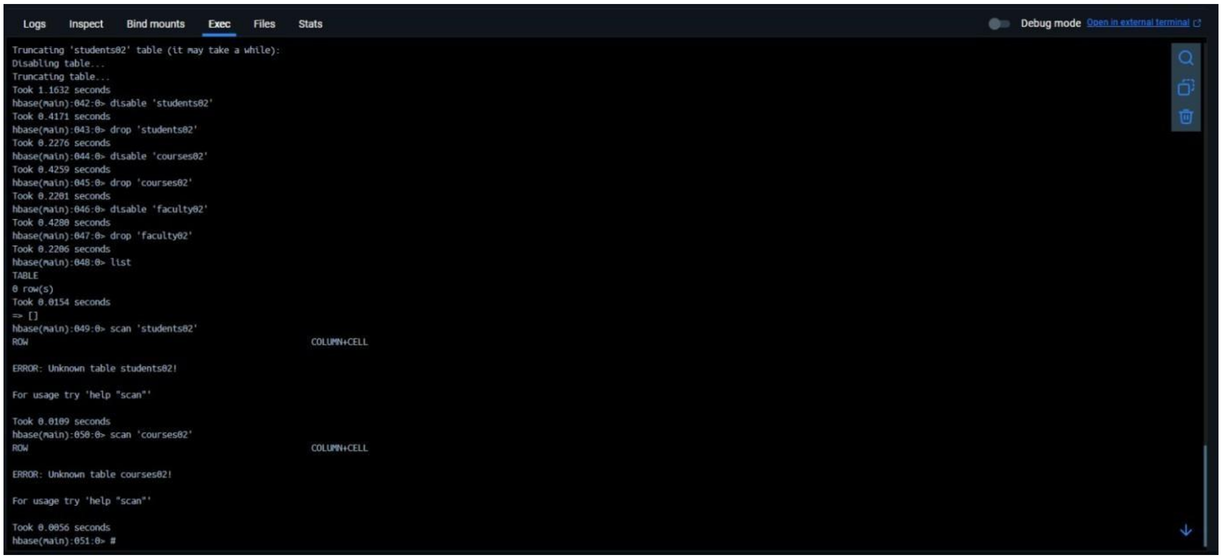
```
Took 0.0228 seconds
hbase(main):025:0> scan 'courses02'
ROW                                                       COLUMN+CELL
 c101                                                     column=details:credits, timestamp=1760072760924, value=4
 c101                                                     column=details:faculty, timestamp=1760072768235, value=F001
 c101                                                     column=details:name, timestamp=1760072752955, value=Data Structures
 c102                                                     column=details:credits, timestamp=1760072832527, value=3
 c102                                                     column=details:faculty, timestamp=1760072868217, value=F002
 c102                                                     column=details:name, timestamp=1760072777189, value=Database Systems
2 row(s)
Took 0.0136 seconds
hbase(main):026:0> scan 'faculty02'
ROW                                                       COLUMN+CELL
 F001                                                     column=contact:email, timestamp=1760072977294, value=meena@college.edu
 F001                                                     column=profile:dept, timestamp=1760072891502, value=CSE
 F001                                                     column=profile:name, timestamp=1760072879626, value=Dr. Meena
 F002                                                     column=contact:email, timestamp=1760072998877, value=ravi@college.edu
 F002                                                     column=profile:dept, timestamp=1760072991944, value=ECE
 F002                                                     column=profile:name, timestamp=1760072984880, value=Dr. Ravi
2 row(s)
Took 0.0091 seconds
hbase(main):027:0> put 'students02', 'row1', 'info:dept', 'IT'
Took 0.0038 seconds
hbase(main):028:0> put 'courses02', 'c101', 'details:credits', '5'
Took 0.0026 seconds
hbase(main):029:0> put 'faculty02', 'F002', 'profile:dept', 'EEE'
Took 0.0063 seconds
hbase(main):030:0> delete 'students02', 'row2', 'info:year'
Took 0.0075 seconds
hbase(main):031:0> deleteall 'courses02', 'c102'
Took 0.0052 seconds
hbase(main):032:0> deleteall 'faculty02', 'F001'
Took 0.0057 seconds
hbase(main):033:0> exists 'students02'
Table students02 does exist
Took 0.0129 seconds
=> true
hbase(main):034:0> exists 'faculty02'
```



```
=> true
hbase(main):034:0> exists 'faculty02'
Table faculty02 does exist
Took 0.0054 seconds
=> true
hbase(main):035:0> describe 'students02'
Table students02 is ENABLED
students02
COLUMN FAMILIES DESCRIPTION
{NAME => 'contact', VERSIONS => '1', EVICT_BLOCKS_ON_CLOSE => 'false', NEW_VERSION_BEHAVIOR => 'false', KEEP_DELETED_CELLS => 'FALSE', CACHE_DATA_ON_WRITE => 'false', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', MIN_VERS
IONS => '0', REPLICATION_SCOPE => '0', BLOOMFILTER => 'ROW', CACHE_INDEX_ON_WRITE => 'false', IN_MEMORY => 'false', CACHE_BLOOMS_ON_WRITE => 'false', PREFETCH_BLOCKS_ON_OPEN => 'false', COMPRESSION => 'NONE', BLOCKCACHE => '
true', BLOCKSIZE => '65536'}
{NAME => 'info', VERSIONS => '1', EVICT_BLOCKS_ON_CLOSE => 'false', NEW_VERSION_BEHAVIOR => 'false', KEEP_DELETED_CELLS => 'FALSE', CACHE_DATA_ON_WRITE => 'false', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', MIN_VERSION
S => '0', REPLICATION_SCOPE => '0', BLOOMFILTER => 'ROW', CACHE_INDEX_ON_WRITE => 'false', IN_MEMORY => 'false', CACHE_BLOOMS_ON_WRITE => 'false', PREFETCH_BLOCKS_ON_OPEN => 'false', COMPRESSION => 'NONE', BLOCKCACHE => 'tru
e', BLOCKSIZE => '65536'}
2 row(s)
Took 0.0368 seconds
hbase(main):036:0> alter 'students02', 'add' => 'extra'
Unknown argument ignored: add
Updating all regions with the new schema...
1/1 regions updated.
Done.
Took 1.7248 seconds
hbase(main):037:0> alter 'students02', 'delete' => 'extra'
Updating all regions with the new schema...
1/1 regions updated.
Done.
Took 1.7227 seconds
hbase(main):038:0> scan 'students02'
ROW                                                       COLUMN+CELL
 row1                                                     column=contact:email, timestamp=1760072702933, value=kiru@college.edu
 row1                                                     column=info:dept, timestamp=1760073033042, value=IT
 row1                                                     column=info:name, timestamp=1760072670473, value=Kiruthika
 row1                                                     column=info:year, timestamp=1760072694257, value=3
 row2                                                     column=contact:email, timestamp=1760072743873, value=arun@college.edu
 row2                                                     column=info:dept, timestamp=1760072722763, value=ECE
 row2                                                     column=info:name, timestamp=1760072712917, value=Arun
2 row(s)
```



```
2 row(s)
Took 0.0111 seconds
hbase(main):039:0> disable 'students02'
Took 0.4416 seconds
hbase(main):040:0> enable 'students02'
Took 0.7433 seconds
hbase(main):041:0> truncate 'students02'
Truncating 'students02' table (it may take a while):
Disabling table...
Truncating table...
Took 1.1632 seconds
hbase(main):042:0> disable 'students02'
Took 0.4171 seconds
hbase(main):043:0> drop 'students02'
Took 0.2276 seconds
hbase(main):044:0> disable 'courses02'
Took 0.4259 seconds
hbase(main):045:0> drop 'courses02'
Took 0.2201 seconds
hbase(main):046:0> disable 'faculty02'
Took 0.4280 seconds
hbase(main):047:0> drop 'faculty02'
Took 0.2206 seconds
hbase(main):048:0> list
TABLE
0 row(s)
Took 0.0154 seconds
=> []
hbase(main):049:0> scan 'students02'
ROW                                                       COLUMN+CELL

ERROR: Unknown table students02!

For usage try 'help "scan"'

Took 0.0109 seconds
hbase(main):050:0> scan 'courses02'
ROW                                                       COLUMN+CELL
```



```
Truncating 'students02' table (it may take a while):
Disabling table...
Truncating table...
Took 1.1632 seconds
hbase(main):042:0> disable 'students02'
Took 0.4171 seconds
hbase(main):043:0> drop 'students02'
Took 0.2276 seconds
hbase(main):044:0> disable 'courses02'
Took 0.4259 seconds
hbase(main):045:0> drop 'courses02'
Took 0.2201 seconds
hbase(main):046:0> disable 'faculty02'
Took 0.4280 seconds
hbase(main):047:0> drop 'faculty02'
Took 0.2206 seconds
hbase(main):048:0> list
TABLE
0 row(s)
Took 0.0154 seconds
=> []
hbase(main):049:0> scan 'students02'
ROW                                                   COLUMN+CELL

ERROR: Unknown table students02!

For usage try 'help "scan"'

Took 0.0109 seconds
hbase(main):050:0> scan 'courses02'
ROW                                                   COLUMN+CELL

ERROR: Unknown table courses02!

For usage try 'help "scan"'

Took 0.0056 seconds
hbase(main):051:0> #
```

RESULT:

8. MODELING A UNIVERSITY ACADEMIC NETWORK USING NEO4J

PROGRAM:

1. Creating Node

Students

```
CREATE (:Student {name:'marimuthu', age:21, address:'BU'});

CREATE (:Student {name:'ravi', age:21, address:'BU'});

CREATE (:Student {name:'nadesan', age:22, address:'MKU'});

CREATE (:Student {name:'manoj', age:22, address:'TU'});

CREATE (:Student {name:'kabilan', age:22, address:'RVS'});
```

Professors

```
CREATE (:Professor {name:'Dr. Ram', age:45, address:'BU'});

CREATE (:Professor {name:'Dr. Siva', age:50, address:'MKU'});
```

Departments

```
CREATE (:Department {name:'Computer Science'});

CREATE (:Department {name:'Mathematics'});
```

Courses

```
CREATE (:Course {code:'CS101', title:'Programming Fundamentals'});

CREATE (:Course {code:'CS102', title:'Data Structures'});

CREATE (:Course {code:'MATH101', title:'Linear Algebra'});
```

2. Relationships

Assign Students to Departments

```
MATCH (s:Student {name:'marimuthu'}), (d:Department {name:'Computer Science'})

CREATE (s)-[:BELONGS_TO]->(d);

MATCH (s:Student {name:'ravi'}), (d:Department {name:'Computer Science'})

CREATE (s)-[:BELONGS_TO]->(d);
```

```
MATCH (s:Student {name:'nadesan'}), (d:Department {name:'Mathematics'})
CREATE (s)-[:BELONGS_TO]->(d);
```

Assign Professors to Departments

```
MATCH (p:Professor {name:'Dr. Ram'}), (d:Department {name:'Computer Science'})
CREATE (p)-[:BELONGS_TO]->(d);
MATCH (p:Professor {name:'Dr. Siva'}), (d:Department {name:'Mathematics'})
CREATE (p)-[:BELONGS_TO]->(d);
```

Professors teach Courses

```
MATCH (p:Professor {name:'Dr. Ram'}), (c:Course {code:'CS101'})
CREATE (p)-[:TEACHES]->(c);
MATCH (p:Professor {name:'Dr. Ram'}), (c:Course {code:'CS102'})
CREATE (p)-[:TEACHES]->(c);
MATCH (p:Professor {name:'Dr. Siva'}), (c:Course {code:'MATH101'})
CREATE (p)-[:TEACHES]->(c);
```

Students enrolled in Courses

```
MATCH (s:Student {name:'marimuthu'}), (c:Course {code:'CS101'})
CREATE (s)-[:ENROLLED_IN]->(c);
MATCH (s:Student {name:'ravi'}), (c:Course {code:'CS101'})
CREATE (s)-[:ENROLLED_IN]->(c);
MATCH (s:Student {name:'nadesan'}), (c:Course {code:'MATH101'})
CREATE (s)-[:ENROLLED_IN]->(c);
```

3. Using scalar functions

Display node properties using keys()

```
MATCH (n)
RETURN n.name, n.age, n.address, keys(n);
```

Use size() on name length

```
MATCH (a:Student)

WHERE size(a.name) > 6

RETURN a.name, size(a.name) AS nameLength;
```

Use timestamp()

```
RETURN timestamp() AS currentTime;
```

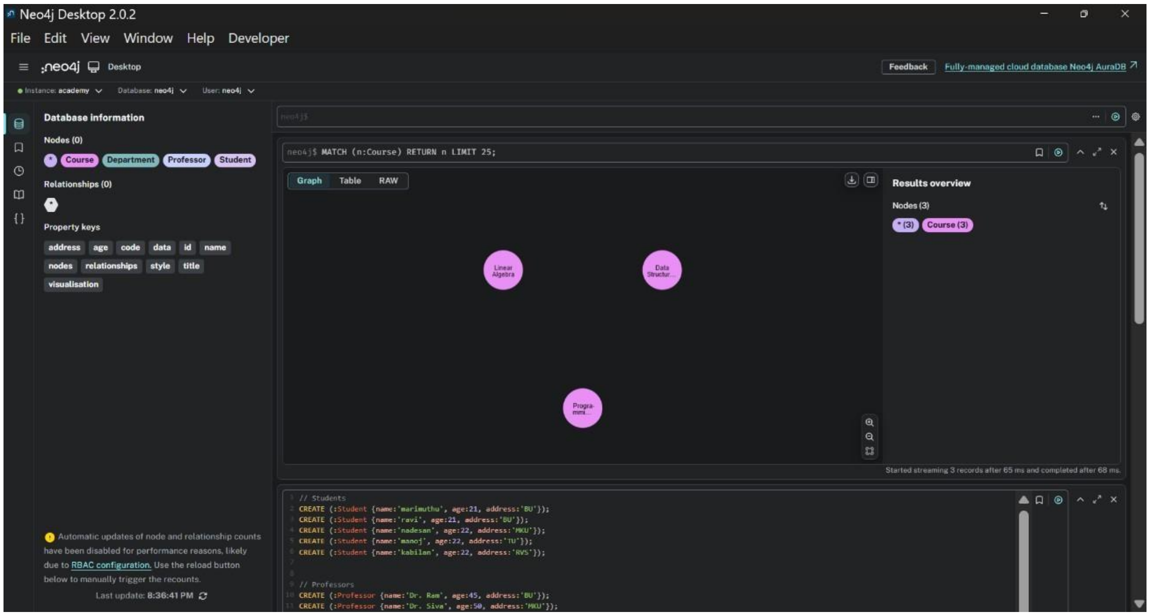
4. Aggregate Function

```
MATCH (n:Student)

RETURN collect(n.age) AS AllAges, collect(n.address) AS AllAddresses;
```

5. Display Who Teaches Which Course

```
MATCH (p:Professor)-[:TEACHES]->(c:Course)

RETURN p.name AS Professor, c.title AS Course;
```

OUTPUT:



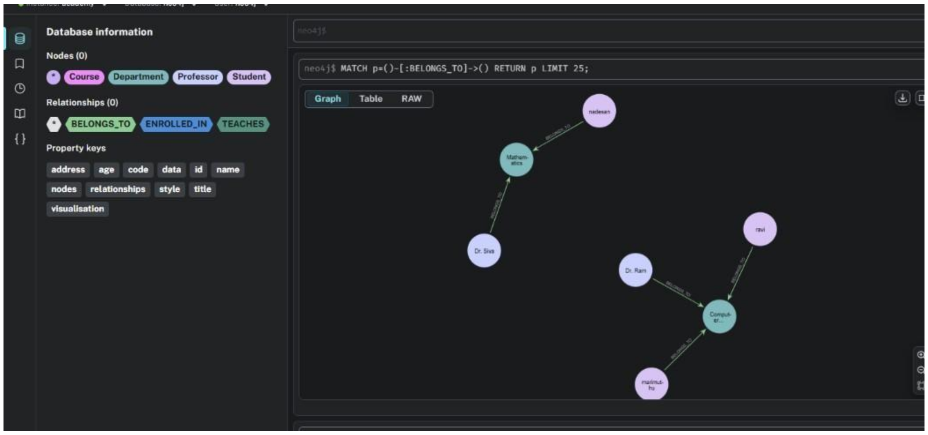
```
neo4j$ MATCH (n:Course) RETURN n LIMIT 25;
```

Started streaming 3 records after 65 ms and completed after 68 ms.

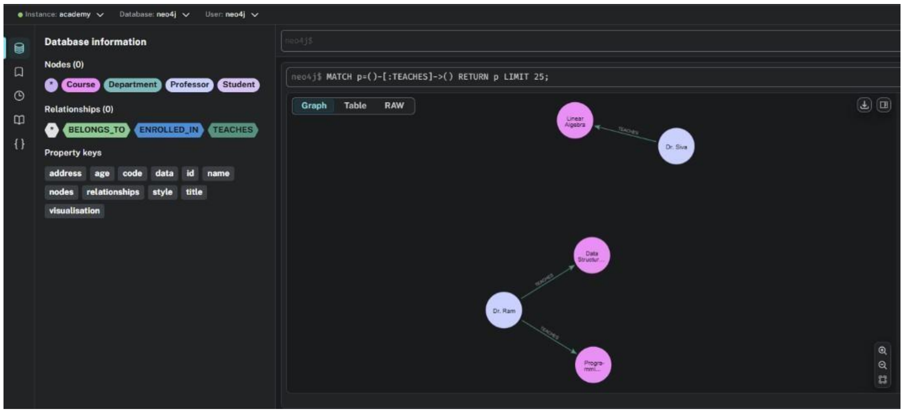
Automatic updates of node and relationship counts have been disabled for performance reasons, likely due to RBAC configuration. Use the reload button below to manually trigger the recounts.

Last update: 8:36:41 PM

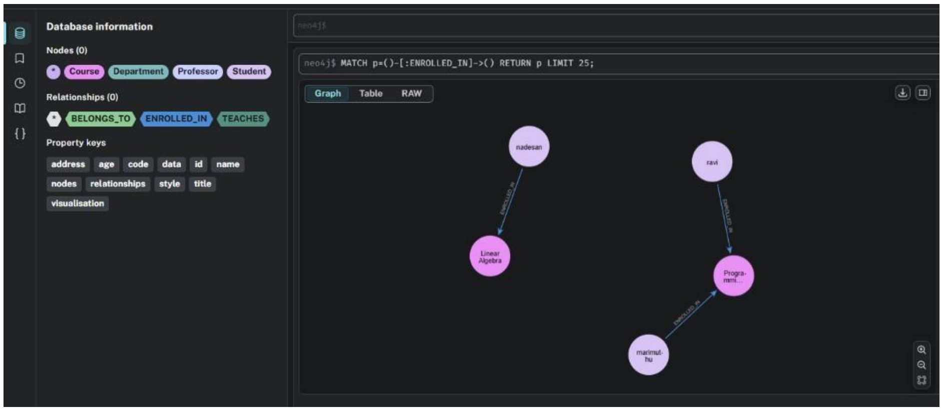
```
// Students
CREATE (:Student {name:'marimuthu', age:21, address:'BU'});
CREATE (:Student {name:'ravi', age:21, address:'BU'});
CREATE (:Student {name:'nadesan', age:22, address:'MKU'});
CREATE (:Student {name:'manoj', age:22, address:'TU'});
CREATE (:Student {name:'kabilan', age:22, address:'RVS'});


// Professors
CREATE (:Professor {name:'Dr. Ram', age:45, address:'BU'});
CREATE (:Professor {name:'Dr. Siva', age:50, address:'MKU'});
```



```
neo4j$ MATCH p=()-[:BELONGS_TO]->() RETURN p LIMIT 25;
```



```
neo4j$ MATCH p=()-[:TEACHES]->() RETURN p LIMIT 25;
```



Database information

Nodes (0)

* Course Department Professor Student

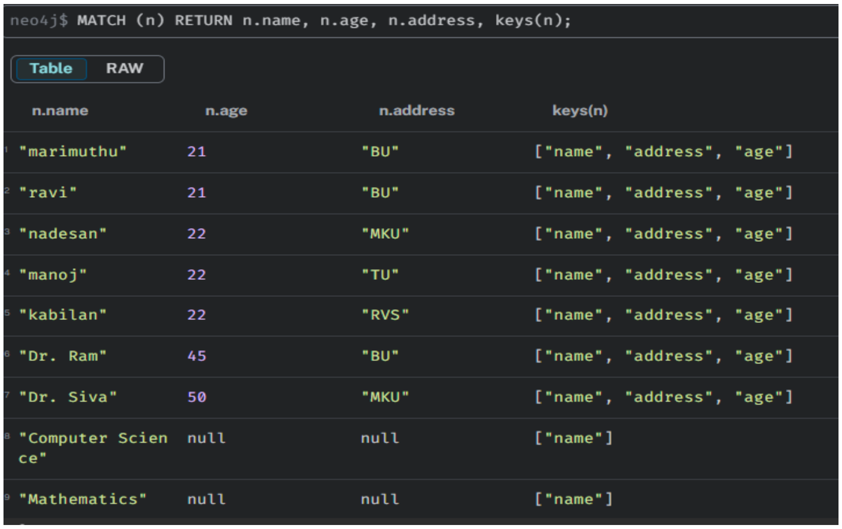
Relationships (0)

* BELONGS_TO ENROLLED_IN TEACHES

Property keys

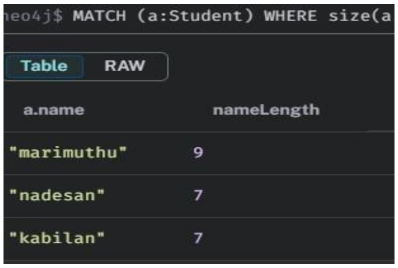
address age code data id name nodes relationships style title visualisation

```
neo4j$ MATCH p=()-[:ENROLLED_IN]->() RETURN p LIMIT 25;
```



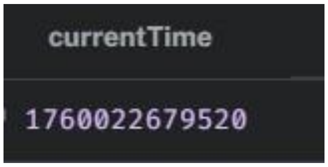
```
neo4j$ MATCH (n) RETURN n.name, n.age, n.address, keys(n);
```

n.name	n.age	n.address	keys(n)
"marimuthu"	21	"BU"	["name", "address", "age"]
"ravi"	21	"BU"	["name", "address", "age"]
"nadesan"	22	"MKU"	["name", "address", "age"]
"manoj"	22	"TU"	["name", "address", "age"]
"kabilan"	22	"RVS"	["name", "address", "age"]
"Dr. Ram"	45	"BU"	["name", "address", "age"]
"Dr. Siva"	50	"MKU"	["name", "address", "age"]
"Computer Science"	null	null	["name"]
"Mathematics"	null	null	["name"]

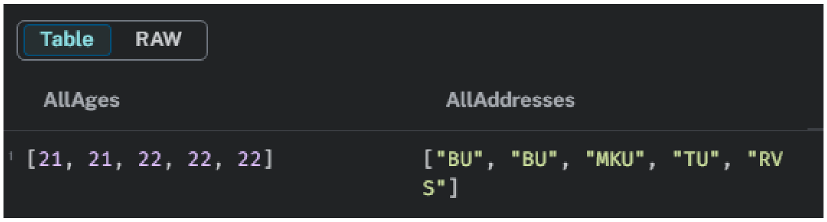


```
neo4j$ MATCH (a:Student) WHERE size(a
```

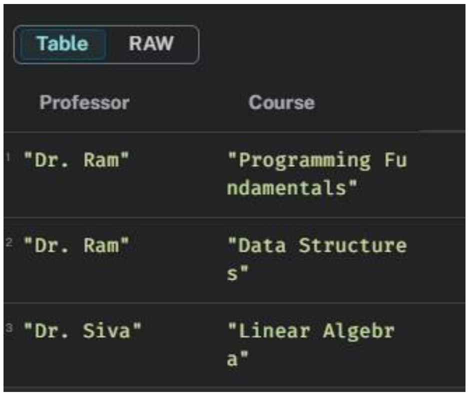
a.name	nameLength
"marimuthu"	9
"nadesan"	7
"kabilan"	7



currentTime
1760022679520



AllAges	AllAddresses
[21, 21, 22, 22, 22]	["BU", "BU", "MKU", "TU", "RVS"]



	Professor	Course
1	"Dr. Ram"	"Programming Fundamentals"
2	"Dr. Ram"	"Data Structures"
3	"Dr. Siva"	"Linear Algebra"

RESULT:

9. MAINTAINING PRODUCT INFORMATION USING CASSANDRA

PROGRAM:

1. Create Keyspace

```
CREATE KEYSPACE IF NOT EXISTS shop
WITH REPLICATION = {
  'class': 'SimpleStrategy',
  'replication_factor': 1
};
```

2. Use Keyspace

```
USE shop;
```

3. Create Table

```
CREATE TABLE IF NOT EXISTS products (
  product_id UUID PRIMARY KEY,
  name TEXT,
  category TEXT,
  price DECIMAL,
  in_stock BOOLEAN
);
```

4. insert Data

```
INSERT INTO products (product_id, name, category, price, in_stock)
VALUES (uuid(), 'Laptop', 'Electronics', 1200.00, true);
INSERT INTO products (product_id, name, category, price, in_stock)
VALUES (uuid(), 'Desk Chair', 'Furniture', 150.50, true);
INSERT INTO products (product_id, name, category, price, in_stock)
VALUES (uuid(), 'Headphones', 'Electronics', 89.99, false);
```

5. Update a product (requires known product_id)

For demonstration, assume we know a product_id = 11111111-1111-1111-1111-111111111111

Replace with a real UUID from SELECT output

```
UPDATE products

SET pric15:46 22-08-202515:46 22-08-202515:46 22-08-2025e = 1299.99, in_stock = false

WHERE product_id = 11111111-1111-1111-1111-111111111111;
```

6. Delete a product (by product_id)

```
DELETE FROM products

WHERE product_id = 22222222-2222-2222-2222-222222222222;
```

7. Select all products

```
SELECT * FROM products;
```

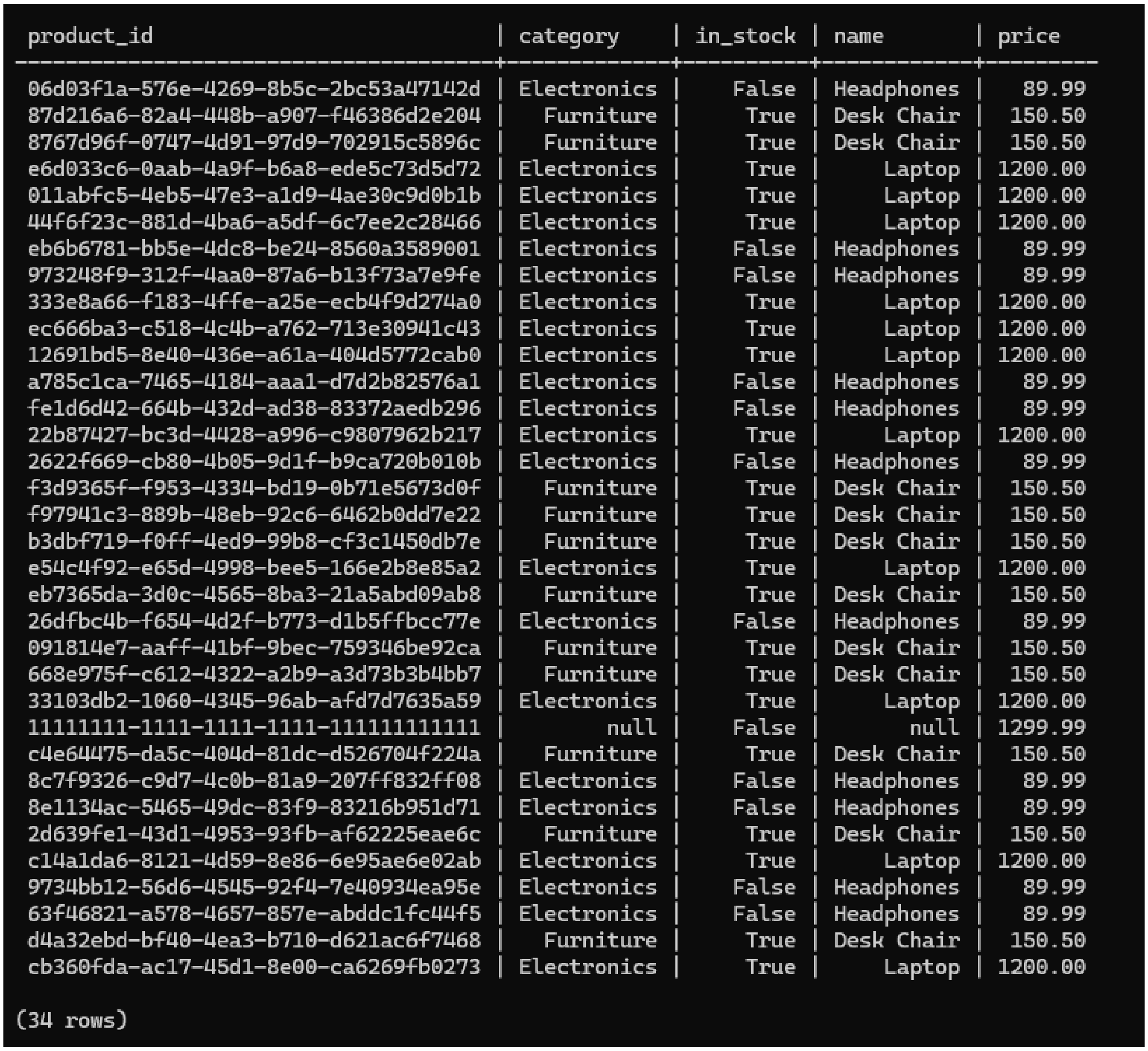
Select products WHERE in_stock = true

```
SELECT * FROM products WHERE in_stock = true ALLOW FILTERING;
```

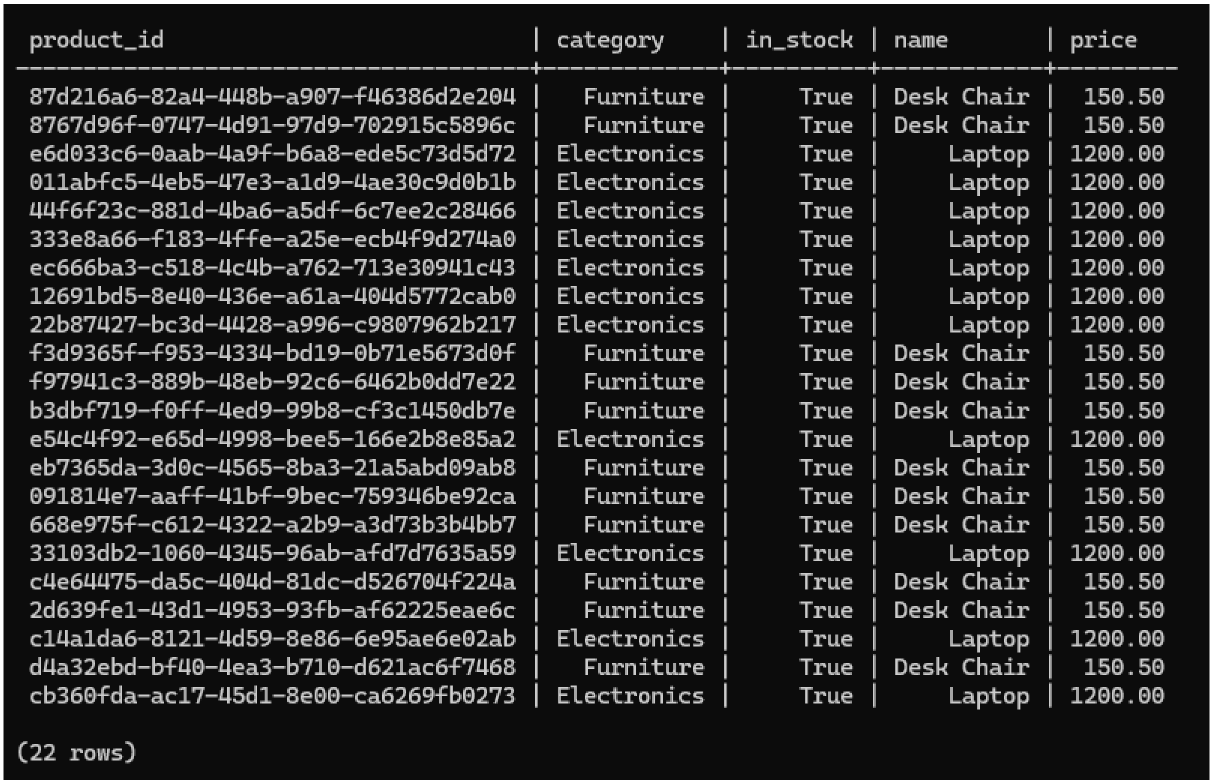
Select products WHERE category = 'Electronics'

```
SELECT * FROM products WHERE category = 'Electronics' ALLOW FILTERING;
```

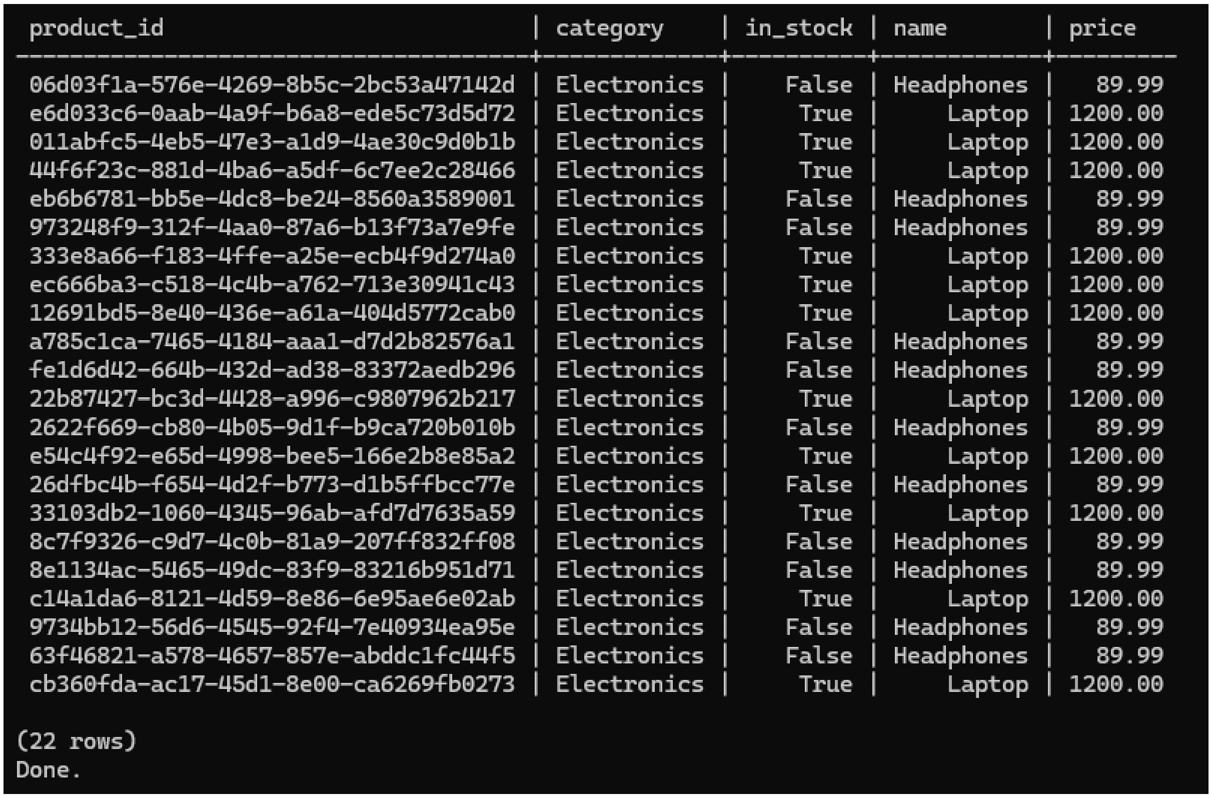
OUTPUT:



```
 product_id                           | category    | in_stock | name       | price
--------------------------------------+-------------+----------+------------+---------
 06d03f1a-576e-4269-8b5c-2bc53a47142d | Electronics |    False | Headphones |   89.99
 87d216a6-82a4-448b-a907-f46386d2e204 |   Furniture |     True | Desk Chair |  150.50
 8767d96f-0747-4d91-97d9-702915c5896c |   Furniture |     True | Desk Chair |  150.50
 e6d033c6-0aab-4a9f-b6a8-ede5c73d5d72 | Electronics |     True |     Laptop | 1200.00
 011abfc5-4eb5-47e3-a1d9-4ae30c9d0b1b | Electronics |     True |     Laptop | 1200.00
 44f6f23c-881d-4ba6-a5df-6c7ee2c28466 | Electronics |     True |     Laptop | 1200.00
 eb6b6781-bb5e-4dc8-be24-8560a3589001 | Electronics |    False | Headphones |   89.99
 973248f9-312f-4aa0-87a6-b13f73a7e9fe | Electronics |    False | Headphones |   89.99
 333e8a66-f183-4ffe-a25e-ecb4f9d274a0 | Electronics |     True |     Laptop | 1200.00
 ec666ba3-c518-4c4b-a762-713e30941c43 | Electronics |     True |     Laptop | 1200.00
 12691bd5-8e40-436e-a61a-404d5772cab0 | Electronics |     True |     Laptop | 1200.00
 a785c1ca-7465-4184-aaa1-d7d2b82576a1 | Electronics |    False | Headphones |   89.99
 fe1d6d42-664b-432d-ad38-83372aedb296 | Electronics |    False | Headphones |   89.99
 22b87427-bc3d-4428-a996-c9807962b217 | Electronics |     True |     Laptop | 1200.00
 2622f669-cb80-4b05-9d1f-b9ca720b010b | Electronics |    False | Headphones |   89.99
 f3d9365f-f953-4334-bd19-0b71e5673d0f |   Furniture |     True | Desk Chair |  150.50
 f97941c3-889b-48eb-92c6-6462b0dd7e22 |   Furniture |     True | Desk Chair |  150.50
 b3dbf719-f0ff-4ed9-99b8-cf3c1450db7e |   Furniture |     True | Desk Chair |  150.50
 e54c4f92-e65d-4998-bee5-166e2b8e85a2 | Electronics |     True |     Laptop | 1200.00
 eb7365da-3d0c-4565-8ba3-21a5abd09ab8 |   Furniture |     True | Desk Chair |  150.50
 26dfbc4b-f654-4d2f-b773-d1b5ffbcc77e | Electronics |    False | Headphones |   89.99
 091814e7-aaff-41bf-9bec-759346be92ca |   Furniture |     True | Desk Chair |  150.50
 668e975f-c612-4322-a2b9-a3d73b3b4bb7 |   Furniture |     True | Desk Chair |  150.50
 33103db2-1060-4345-96ab-afd7d7635a59 | Electronics |     True |     Laptop | 1200.00
 11111111-1111-1111-1111-111111111111 |        null |    False |       null | 1299.99
 c4e64475-da5c-404d-81dc-d526704f224a |   Furniture |     True | Desk Chair |  150.50
 8c7f9326-c9d7-4c0b-81a9-207ff832ff08 | Electronics |    False | Headphones |   89.99
 8e1134ac-5465-49dc-83f9-83216b951d71 | Electronics |    False | Headphones |   89.99
 2d639fe1-43d1-4953-93fb-af62225eae6c |   Furniture |     True | Desk Chair |  150.50
 c14a1da6-8121-4d59-8e86-6e95ae6e02ab | Electronics |     True |     Laptop | 1200.00
 9734bb12-56d6-4545-92f4-7e40934ea95e | Electronics |    False | Headphones |   89.99
 63f46821-a578-4657-857e-abddc1fc44f5 | Electronics |    False | Headphones |   89.99
 d4a32ebd-bf40-4ea3-b710-d621ac6f7468 |   Furniture |     True | Desk Chair |  150.50
 cb360fda-ac17-45d1-8e00-ca6269fb0273 | Electronics |     True |     Laptop | 1200.00

(34 rows)
```



```
 product_id                           | category    | in_stock | name       | price
--------------------------------------+-------------+----------+------------+---------
 87d216a6-82a4-448b-a907-f46386d2e204 |   Furniture |     True | Desk Chair |  150.50
 8767d96f-0747-4d91-97d9-702915c5896c |   Furniture |     True | Desk Chair |  150.50
 e6d033c6-0aab-4a9f-b6a8-ede5c73d5d72 | Electronics |     True |     Laptop | 1200.00
 011abfc5-4eb5-47e3-a1d9-4ae30c9d0b1b | Electronics |     True |     Laptop | 1200.00
 44f6f23c-881d-4ba6-a5df-6c7ee2c28466 | Electronics |     True |     Laptop | 1200.00
 333e8a66-f183-4ffe-a25e-ecb4f9d274a0 | Electronics |     True |     Laptop | 1200.00
 ec666ba3-c518-4c4b-a762-713e30941c43 | Electronics |     True |     Laptop | 1200.00
 12691bd5-8e40-436e-a61a-404d5772cab0 | Electronics |     True |     Laptop | 1200.00
 22b87427-bc3d-4428-a996-c9807962b217 | Electronics |     True |     Laptop | 1200.00
 f3d9365f-f953-4334-bd19-0b71e5673d0f |   Furniture |     True | Desk Chair |  150.50
 f97941c3-889b-48eb-92c6-6462b0dd7e22 |   Furniture |     True | Desk Chair |  150.50
 b3dbf719-f0ff-4ed9-99b8-cf3c1450db7e |   Furniture |     True | Desk Chair |  150.50
 e54c4f92-e65d-4998-bee5-166e2b8e85a2 | Electronics |     True |     Laptop | 1200.00
 eb7365da-3d0c-4565-8ba3-21a5abd09ab8 |   Furniture |     True | Desk Chair |  150.50
 091814e7-aaff-41bf-9bec-759346be92ca |   Furniture |     True | Desk Chair |  150.50
 668e975f-c612-4322-a2b9-a3d73b3b4bb7 |   Furniture |     True | Desk Chair |  150.50
 33103db2-1060-4345-96ab-afd7d7635a59 | Electronics |     True |     Laptop | 1200.00
 c4e64475-da5c-404d-81dc-d526704f224a |   Furniture |     True | Desk Chair |  150.50
 2d639fe1-43d1-4953-93fb-af62225eae6c |   Furniture |     True | Desk Chair |  150.50
 c14a1da6-8121-4d59-8e86-6e95ae6e02ab | Electronics |     True |     Laptop | 1200.00
 d4a32ebd-bf40-4ea3-b710-d621ac6f7468 |   Furniture |     True | Desk Chair |  150.50
 cb360fda-ac17-45d1-8e00-ca6269fb0273 | Electronics |     True |     Laptop | 1200.00

(22 rows)
```



```
 product_id                           | category    | in_stock | name       | price
--------------------------------------+-------------+----------+------------+---------
 06d03f1a-576e-4269-8b5c-2bc53a47142d | Electronics |    False | Headphones |   89.99
 e6d033c6-0aab-4a9f-b6a8-ede5c73d5d72 | Electronics |     True |     Laptop | 1200.00
 011abfc5-4eb5-47e3-a1d9-4ae30c9d0b1b | Electronics |     True |     Laptop | 1200.00
 44f6f23c-881d-4ba6-a5df-6c7ee2c28466 | Electronics |     True |     Laptop | 1200.00
 eb6b6781-bb5e-4dc8-be24-8560a3589001 | Electronics |    False | Headphones |   89.99
 973248f9-312f-4aa0-87a6-b13f73a7e9fe | Electronics |    False | Headphones |   89.99
 333e8a66-f183-4ffe-a25e-ecb4f9d274a0 | Electronics |     True |     Laptop | 1200.00
 ec666ba3-c518-4c4b-a762-713e30941c43 | Electronics |     True |     Laptop | 1200.00
 12691bd5-8e40-436e-a61a-404d5772cab0 | Electronics |     True |     Laptop | 1200.00
 a785c1ca-7465-4184-aaa1-d7d2b82576a1 | Electronics |    False | Headphones |   89.99
 fe1d6d42-664b-432d-ad38-83372aedb296 | Electronics |    False | Headphones |   89.99
 22b87427-bc3d-4428-a996-c9807962b217 | Electronics |     True |     Laptop | 1200.00
 2622f669-cb80-4b05-9d1f-b9ca720b010b | Electronics |    False | Headphones |   89.99
 e54c4f92-e65d-4998-bee5-166e2b8e85a2 | Electronics |     True |     Laptop | 1200.00
 26dfbc4b-f654-4d2f-b773-d1b5ffbcc77e | Electronics |    False | Headphones |   89.99
 33103db2-1060-4345-96ab-afd7d7635a59 | Electronics |     True |     Laptop | 1200.00
 8c7f9326-c9d7-4c0b-81a9-207ff832ff08 | Electronics |    False | Headphones |   89.99
 8e1134ac-5465-49dc-83f9-83216b951d71 | Electronics |    False | Headphones |   89.99
 c14a1da6-8121-4d59-8e86-6e95ae6e02ab | Electronics |     True |     Laptop | 1200.00
 9734bb12-56d6-4545-92f4-7e40934ea95e | Electronics |    False | Headphones |   89.99
 63f46821-a578-4657-857e-abddc1fc44f5 | Electronics |    False | Headphones |   89.99
 cb360fda-ac17-45d1-8e00-ca6269fb0273 | Electronics |     True |     Laptop | 1200.00

(22 rows)
Done.
```

RESULT:

10. MAPREDUCE FOR WORD COUNT

PROGRAM:

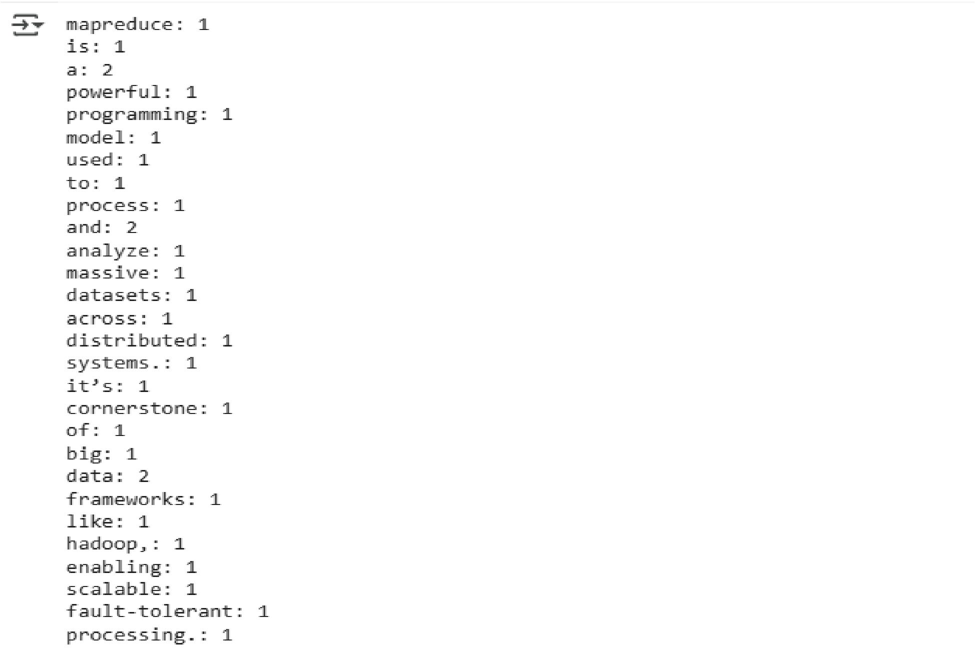
```
from google.colab import files
from collections import defaultdict
def mapper(t): return [(w.lower(), 1) for w in t.split()]
def reducer(m):
    c = defaultdict(int)
    [c._setitem_(w, c[w]+cnt) for w,cnt in m]
    return c
text = open(list(files.upload().keys())[0]).read()
for w,c in reducer(mapper(text)).items():
print(w, ":", c)
```

Example1.txt:

MapReduce is a powerful programming model used to process and analyze massive datasets across distributed systems.

It’s a cornerstone of big data frameworks like Hadoop, enabling scalable and fault-tolerant data processing.

OUTPUT:



```
mapreduce: 1
is: 1
a: 2
powerful: 1
programming: 1
model: 1
used: 1
to: 1
process: 1
and: 2
analyze: 1
massive: 1
datasets: 1
across: 1
distributed: 1
systems.: 1
it’s: 1
cornerstone: 1
of: 1
big: 1
data: 2
frameworks: 1
like: 1
hadoop,: 1
enabling: 1
scalable: 1
fault-tolerant: 1
processing.: 1
```

RESULT:

11. MAPREDUCE TO ANALYZE STUDENT MARKS LIST

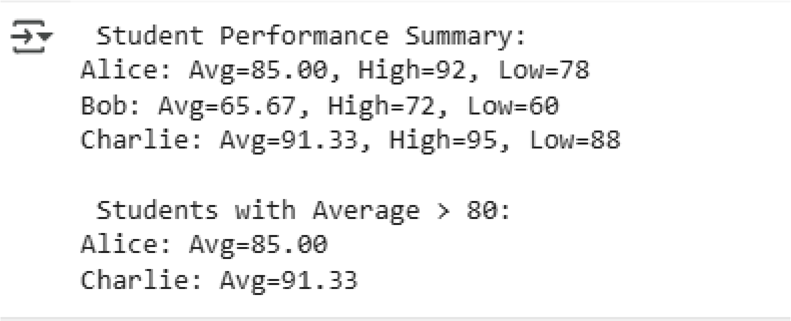
PROGRAM:

```
from google.colab import files
from collections import defaultdict
def mapper(lines):
    return [(s,int(m)) for s,_,m in (l.split(',') for l in lines)]
def reducer(mapped):
    g=defaultdict(list)
    for s,m in mapped: g[s]+=[m]
    return {s:{"Avg":sum(v)/len(v),"High":max(v),"Low":min(v)} for s,v in g.items()}
res = reducer(mapper(open(list(files.upload().keys())[0]).read().splitlines()))
print(
    "Student Performance Summary:\n" +
    '\n'.join([f"{s}: Avg={d['Avg']:.2f}, High={d['High']}, Low={d['Low']}" for s,d in
res.items()]) +
    "\n\nStudents with Average > 80:\n" +
    ', '.join([s for s,d in res.items() if d["Avg"]>80])
)
```

Example2.txt:

```
Alice,Math,85
Alice,Physics,78
Alice,Chemistry,92
Bob,Math,65
Bob,Physics,72
Bob,Chemistry,60
Charlie,Math,95
Charlie,Physics,88
Charlie,Chemistry,91
```

OUTPUT:



```
 Student Performance Summary:
Alice: Avg=85.00, High=92, Low=78
Bob: Avg=65.67, High=72, Low=60
Charlie: Avg=91.33, High=95, Low=88

 Students with Average > 80:
Alice: Avg=85.00
Charlie: Avg=91.33
```

RESULT:

12. MAPREDUCE FOR FILTERING OUT COMMON WORDS

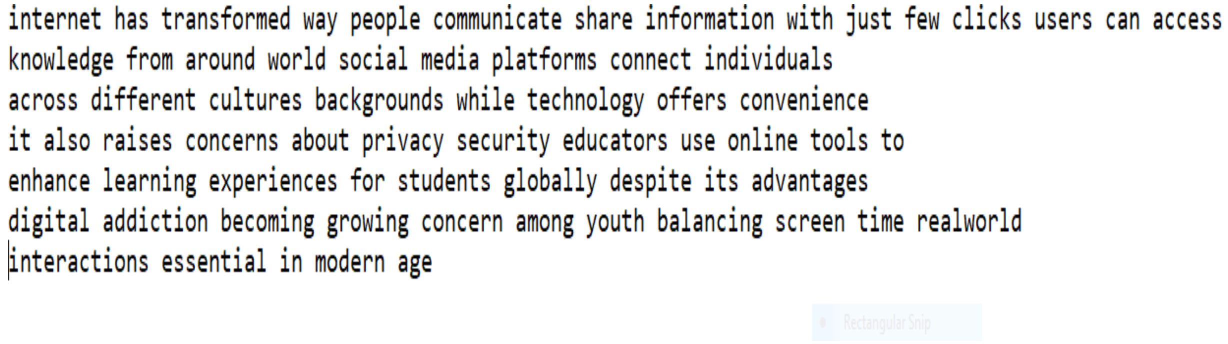
PROGRAM:

```
from google.colab import files
from collections import defaultdict
import string
def mapper(t, stop): return [(w,w) for w in t.translate(str.maketrans("","",'!"#$%&\'()*+,-
./:;<=>?@[\\]^_`{|}~')).lower().split() if w not in stop]
def reducer(m):
    g=defaultdict(list)
    [g[k].append(v) for k,v in m]
    return ' '.join(v for vals in g.values() for v in vals)
stop={"is","the","a","an","and"}
text=open(list(files.upload().keys())[0]).read()
print(reducer(mapper(text, stop)))
```

Example3.txt:

```
The internet has transformed the way people communicate and share information.
With just a few clicks, users can access knowledge from around the world.
Social media platforms connect individuals across different cultures and backgrounds.
While technology offers convenience, it also raises concerns about privacy and security.
Educators use online tools to enhance learning experiences for students globally.
Despite its advantages, digital addiction is becoming a growing concern among youth.
Balancing screen time and real-world interactions is essential in the modern age.
```

OUTPUT:



```
internet has transformed way people communicate share information with just few clicks users can access
knowledge from around world social media platforms connect individuals
across different cultures backgrounds while technology offers convenience
it also raises concerns about privacy security educators use online tools to
enhance learning experiences for students globally despite its advantages
digital addiction becoming growing concern among youth balancing screen time realworld
interactions essential in modern age
```

RESULT: