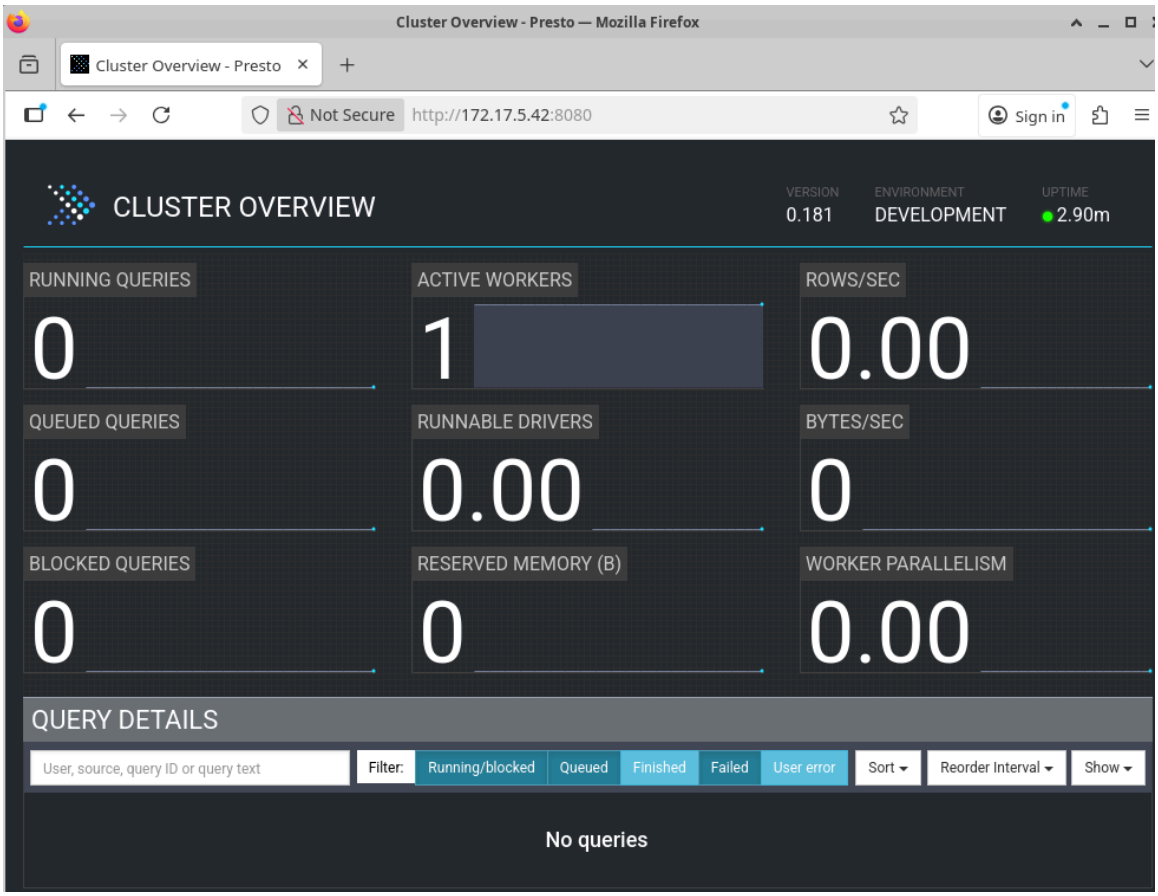
HandOut. 8: Delving into Hive (2%)

Installation

Access your docker working directory
Get the git for docker hive : git clone https://github.com/big -data -europe/docker -hive.git
Execute: docker -compose up –d
Check all containers : docker ps
Check your IP : ipconfig /all
[ToDo: Snapshot of Presto at 8080] (Presto is part of Hive)



Bash into hive : docker -compose exec hive -server bash
Execute Beeline : /opt/hive/bin/beeline -u jdbc:hive2://localhost:10000

Basics
Check Beeline help for all available commands: !help
List all current connections: !list
Set variables in Beeline for session -specific configurations: !set [ToDo: Try setting 2 variables and check their output]

```
0: jdbc:hive2://localhost:10000> !set maxwidth 100
0: jdbc:hive2://localhost:10000> !set verbose true
0: jdbc:hive2://localhost:10000> !set
properties: {beeline.maxhistoryrows=500, beeline.headerinterv
eascommand=false, beeline.authtype=, beeline.delimiterfordsv=
e.timeout=-1, beeline.showelapsedtime=true, beeline.verbose=t
etable=false, beeline.isolation=TRANSACTION_REPEATABLE_READ,
e.scriptfile=, beeline.maxwidth=100, beeline.color=false, bee
oot/.beeline/history}
authtype
autocommit          true
autosave            false
color               false
delimiterfordsv     |
entirelineascommand false
fastconnect         true
force               false
headerinterval      100
historyfile         /root/.beeline/history
hiveconfvariables   {}
hivevariables       {}
incremental         true
incrementalbufferrows1000
initfiles
isolation           TRANSACTION_REPEATABLE_READ
lastconnectedurl    jdbc:hive2://localhost:10000
maxcolumnwidth      50
maxheight           45
maxhistoryrows      500
maxwidth            100
nullemptystring     false
numberformat        default
outputformat        table
scriptfile
showdbinprompt      false
showelapsedtime     true
```

```
scriptfile
showdbinprompt      false
showelapsedtime     true
showheader          true
shownestederrs      false
showwarnings        false
timeout             -1
trimscripts         true
truncatetable       false
verbose             true
```

Modify how query results are displayed: !set outputformat=<format> (replace format by "table", "vertical", "csv")
Quit the session : !quit
Login again into Beehive

Create Database:
CREATE DATABASE [IF NOT EXISTS] userdb;
CREATE SCHEMA userdb;
SHOW DATABASES;

```
0: jdbc:hive2://localhost:10000> SHOW DATABASES;
+----------------+
| database_name  |
+----------------+
| default        |
| userdb         |
+----------------+
2 rows selected (0.057 seconds)
0: jdbc:hive2://localhost:10000>
```

Drop Database:
DROP DATABASE IF EXISTS userdb;
DROP DATABASE IF EXISTS userdb CASCADE; //drop tables then database
DROP SCHEMA userdb;
SHOW DATABASES;

Create Table:
CREATE TABLE IF NOT EXISTS employee (eid int, name String, salary String, destination String)
COMMENT 'Employee details'
ROW FORMAT DELIMITED

FIELDS TERMINATED BY ‘ \t’
LINES TERMINATED BY ‘ \n’
STORED AS TEXTFILE;

```
0: jdbc:hive2://localhost:10000> CREATE TABLE IF NOT EXISTS employee (
. . . . . .>   eid INT,
. . . . . .>   name STRING,
. . . . . .>   salary STRING,
. . . . . .>   destination STRING
. . . . . .> )
. . . . . .> COMMENT 'Employee details'
. . . . . .> ROW FORMAT DELIMITED
. . . . . .> FIELDS TERMINATED BY '\t'
. . . . . .> LINES TERMINATED BY '\n'
. . . . . .> STORED AS TEXTFILE;
No rows affected (0.56 seconds)
0: jdbc:hive2://localhost:10000> SHOW TABLES;
+-----------+
| tab_name  |
+-----------+
| employee  |
+-----------+
1 row selected (0.055 seconds)
0: jdbc:hive2://localhost:10000>
```

Alter Table:
ALTER TABLE name RENAME TO new_name
ALTER TABLE name ADD COLUMNS (col_spec[, col_spec ...])

```
1 row selected (0.039 seconds)
0: jdbc:hive2://localhost:10000> ALTER TABLE emp ADD COLUMNS (
. . . . . .>   department STRING,
. . . . . .>   joining_date STRING
. . . . . .> );
No rows affected (0.109 seconds)
0: jdbc:hive2://localhost:10000> DESCRIBE emp;
+--------------+------------+----------+
|   col_name   | data_type  | comment  |
+--------------+------------+----------+
| eid          | int        |          |
| name         | string     |          |
| salary       | string     |          |
| destination  | string     |          |
| department   | string     |          |
| joining_date | string     |          |
+--------------+------------+----------+
6 rows selected (0.078 seconds)
0: jdbc:hive2://localhost:10000>
```

ALTER TABLE name DROP [COLUMN] column_name
ALTER TABLE name CHANGE column_name new_name new_type ALTER TABLE name REPLACE COLUMNS (col_spec[, col_spec ...])
ALTER TABLE employee RENAME TO emp;

```
0: jdbc:hive2://localhost:10000> DESCRIBE emp;
+--------------+------------+----------+
|   col_name   | data_type  | comment  |
+--------------+------------+----------+
| eid          | int        |          |
| name         | string     |          |
| salary       | int        |          |
| destination  | string     |          |
+--------------+------------+----------+
4 rows selected (0.047 seconds)
0: jdbc:hive2://localhost:10000>
```

Drop Table
DROP TABLE IF EXISTS employee;

Exercise s with S tudent DB :
DDL:

• CREATE DATABASE IF NOT EXISTS student_db;
• USE student_db;
• SHOW DATABASES;
• DROP DATABASE IF EXISTS student_db CASCADE;

• CREATE TABLE students (
  student_id INT,
   name STRING,
   age INT,
   major STRING
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE;
• SHOW TABLES;
• DESCRIBE students;

```
0: jdbc:hive2://localhost:10000> DESCRIBE students;
+-------------+------------+----------+
|  col_name   | data_type  | comment  |
+-------------+------------+----------+
| student_id  | int        |          |
| name        | string     |          |
| age         | int        |          |
| major       | string     |          |
+-------------+------------+----------+
4 rows selected (0.063 seconds)
0: jdbc:hive2://localhost:10000>
```

• ALTER TABLE students ADD COLUMNS (gpa FLOAT);
• ALTER TABLE students RENAME TO student_info;
• DROP TABLE IF EXISTS student_info;
DDL:
• [ToDo : Create a CSV file with some fictitious student data – should have column names at top and data below , e.g., name, CGPA, degree program etc.]

```
madeel@bdacourse:~/data$ cat students_data.csv
student_id,name,age,major,gpa
1,John Doe,20,Computer Science,3.5
2,Jane Smith,21,Mathematics,3.8
3,Ali Khan,22,Physics,3.2
4,Sara Lee,19,Statistics,3.7
```

• Load student CSV [ToDo: paste output] : LOAD DATA LOCAL INPATH '/path/to/students_data.csv' INTO TABLE students;

```
0: jdbc:hive2://localhost:10000> SELECT * FROM student_info;
+--------------------------+--------------------+--------------------+----------------------+--------------------+
| student_info.student_id  | student_info.name  | student_info.age   | student_info.major   | student_info.gpa   |
+--------------------------+--------------------+--------------------+----------------------+--------------------+
| NULL                     | name               | NULL               | major                | NULL               |
| 1                        | John Doe           | 20                 | Computer Science     | 3.5                |
| 2                        | Jane Smith         | 21                 | Mathematics          | 3.8                |
| 3                        | Ali Khan           | 22                 | Physics              | 3.2                |
| 4                        | Sara Lee           | 19                 | Statistics           | 3.7                |
+--------------------------+--------------------+--------------------+----------------------+--------------------+
5 rows selected (1.825 seconds)
0: jdbc:hive2://localhost:10000>
```

• [ToDo: Modify the data in the CSV file]

• Replace existing data in the table [ToDo: paste output] : LOAD DATA LOCAL INPATH '/path/to/students_data.csv' OVERWRITE INTO TABLE students;

```
0: jdbc:hive2://localhost:10000> SELECT * FROM student_info;LOAD DATA LOCAL INPATH '/tmp/students_data.csv' OVERWRITE INTO TABLE student_info;
+--------------------------+--------------------+--------------------+----------------------+--------------------+
| student_info.student_id  | student_info.name  | student_info.age   | student_info.major   | student_info.gpa   |
+--------------------------+--------------------+--------------------+----------------------+--------------------+
| NULL                     | name               | NULL               | major                | NULL               |
| 1                        | John Doe           | 20                 | Computer Science     | 3.5                |
| 2                        | Jane Smith         | 21                 | Mathematics          | 3.8                |
| 3                        | Ali Khan           | 22                 | Physics              | 3.2                |
| 4                        | Sara Lee           | 19                 | Statistics           | 3.7                |
+--------------------------+--------------------+--------------------+----------------------+--------------------+
5 rows selected (0.157 seconds)
No rows affected (0.388 seconds)
0: jdbc:hive2://localhost:10000> SELECT * FROM student_info;
+--------------------------+--------------------+--------------------+----------------------+--------------------+
| student_info.student_id  | student_info.name  | student_info.age   | student_info.major   | student_info.gpa   |
+--------------------------+--------------------+--------------------+----------------------+--------------------+
| NULL                     | name               | NULL               | major                | NULL               |
| 1                        | John Doe           | 20                 | Computer Science     | 3.5                |
| 2                        | Jane Smith         | 21                 | Mathematics          | 3.8                |
| 3                        | Ali Khan           | 22                 | Physics              | 3.2                |
| 4                        | Zuha Ali           | 19                 | Statistics           | 3.7                |
+--------------------------+--------------------+--------------------+----------------------+--------------------+
5 rows selected (0.207 seconds)
0: jdbc:hive2://localhost:10000>
```

• Inserting data – this should add data after your CSV data [ToDo: paste output] : INSERT INTO TABLE students VALUES (1, 'John Doe', 20, 'Computer Science', 3.5);

```
0: jdbc:hive2://localhost:10000> INSERT INTO TABLE student_info (student_id, name, age, major, gpa)
. . . . . .> VALUES (5, 'Mahnoor Adeel', 20, 'Computer Science', 3.5);
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a di
No rows affected (1.893 seconds)
0: jdbc:hive2://localhost:10000> SELECT * FROM student_info;
+--------------------------+--------------------+--------------------+----------------------+--------------------+
| student_info.student_id  | student_info.name  | student_info.age   | student_info.major   | student_info.gpa   |
+--------------------------+--------------------+--------------------+----------------------+--------------------+
| 5                        | Mahnoor Adeel      | 20                 | Computer Science     | 3.5                |
| NULL                     | name               | NULL               | major                | NULL               |
| 1                        | John Doe           | 20                 | Computer Science     | 3.5                |
| 2                        | Jane Smith         | 21                 | Mathematics          | 3.8                |
| 3                        | Ali Khan           | 22                 | Physics              | 3.2                |
| 4                        | Zuha Ali           | 19                 | Statistics           | 3.7                |
+--------------------------+--------------------+--------------------+----------------------+--------------------+
6 rows selected (0.195 seconds)
0: jdbc:hive2://localhost:10000>
```

• Insert data from another table – you need to create another table for this – try for your own knowledge [ToDo: paste output] : INSERT INTO TABLE students SELECT * FROM student_backup;

```
+---------------------------+----------------------+---------------------+-----------------------+---------------------+
| student_backup.student_id | student_backup.name  | student_backup.age  | student_backup.major  | student_backup.gpa  |
+---------------------------+----------------------+---------------------+-----------------------+---------------------+
| 6                         | Adeel Khan           | 23                  | Data Science          | 3.9                 |
| 7                         | Sara Malik           | 21                  | Mathematics           | 3.7                 |
+---------------------------+----------------------+---------------------+-----------------------+---------------------+
2 rows selected (0.149 seconds)
0: jdbc:hive2://localhost:10000>
```

```
0: jdbc:hive2://localhost:10000> INSERT INTO TABLE student_info SELECT * FROM student_backup;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a diff
No rows affected (1.629 seconds)
0: jdbc:hive2://localhost:10000> SELECT * FROM student_info;
+--------------------------+--------------------+--------------------+---------------------+--------------------+
| student_info.student_id  | student_info.name  | student_info.age   | student_info.major  | student_info.gpa   |
+--------------------------+--------------------+--------------------+---------------------+--------------------+
| 5                        | Mahnoor Adeel      | 20                 | Computer Science    | 3.5                |
| 6                        | Adeel Khan         | 23                 | Data Science        | 3.9                |
| 7                        | Sara Malik         | 21                 | Mathematics         | 3.7                |
| NULL                     | name               | NULL               | major               | NULL               |
| 1                        | John Doe           | 20                 | Computer Science    | 3.5                |
| 2                        | Jane Smith         | 21                 | Mathematics         | 3.8                |
| 3                        | Ali Khan           | 22                 | Physics             | 3.2                |
| 4                        | Zuha Ali           | 19                 | Statistics          | 3.7                |
+--------------------------+--------------------+--------------------+---------------------+--------------------+
8 rows selected (0.165 seconds)
0: jdbc:hive2://localhost:10000>
```

• Update GPA: UPDATE students SET gpa = 3.8 WHERE student_id = 1;
• DELETE FROM students WHERE age < 18;

Queries:
• SELECT * FROM students;
• SELECT * FROM students WHERE major = 'Computer Science';

```
0: jdbc:hive2://localhost:10000> SELECT * FROM student_info WHERE major = 'Computer Science';
+--------------------------+--------------------+--------------------+---------------------+--------------------+
| student_info.student_id  | student_info.name  | student_info.age   | student_info.major  | student_info.gpa   |
+--------------------------+--------------------+--------------------+---------------------+--------------------+
| 1                        | John Doe           | 20                 | Computer Science    | 3.5                |
+--------------------------+--------------------+--------------------+---------------------+--------------------+
1 row selected (0.432 seconds)
0: jdbc:hive2://localhost:10000>
```

• SELECT * FROM students ORDER BY gpa DESC;

```
0: jdbc:hive2://localhost:10000>  SELECT * FROM student_info ORDER BY gpa DESC;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a differe
+--------------------------+--------------------+--------------------+---------------------+--------------------+
| student_info.student_id  | student_info.name  | student_info.age   | student_info.major  | student_info.gpa   |
+--------------------------+--------------------+--------------------+---------------------+--------------------+
| 2                        | Jane Smith         | 21                 | Mathematics         | 3.8                |
| 4                        | Zuha Ali           | 19                 | Statistics          | 3.7                |
| 1                        | John Doe           | 20                 | Computer Science    | 3.5                |
| 3                        | Ali Khan           | 22                 | Physics             | 3.2                |
| NULL                     | name               | NULL               | major               | NULL               |
+--------------------------+--------------------+--------------------+---------------------+--------------------+
5 rows selected (1.831 seconds)
0: jdbc:hive2://localhost:10000>
```

• SELECT major, COUNT(*) AS student_count FROM students GROUP BY major;

```
0: jdbc:hive2://localhost:10000> SELECT major, COUNT(*) AS student_count
. . . . . .> FROM student_info
. . . . . .> GROUP BY major;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in t
+--------------------+----------------+
|       major        | student_count  |
+--------------------+----------------+
| Computer Science   | 2              |
| Data Science       | 1              |
| Mathematics        | 2              |
| Physics            | 1              |
| Statistics         | 1              |
| major              | 1              |
+--------------------+----------------+
6 rows selected (1.409 seconds)
0: jdbc:hive2://localhost:10000>
```

• [ToDo: paste output – you need to create a departments table first and then fill it up with some dummy data] SELECT s.student_id, s.name, d.department_name
FROM students s
JOIN departments d ON s.major = d.major;

```
0: jdbc:hive2://localhost:10000> SELECT * FROM departments;
+----------------------+---------------------------------+--------------------+
| departments.dept_id  |   departments.department_name   | departments.major  |
+----------------------+---------------------------------+--------------------+
| 1                    | Department of Computer Science  | Computer Science   |
| 2                    | Department of Data Science      | Data Science       |
| 3                    | Department of Mathematics       | Mathematics        |
| 4                    | Department of Physics           | Physics            |
+----------------------+---------------------------------+--------------------+
```

```
0: jdbc:hive2://localhost:10000> SELECT
. . . . . .>   s.student_id,
. . . . . .>   s.name,
. . . . . .>   d.department_name
. . . . . .> FROM student_info s
. . . . . .> JOIN departments d
. . . . . .> ON s.major = d.major;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hive/lib/log4j-slf4j-impl-2.6.2.j
SLF4J: Found binding in [jar:file:/opt/hadoop-2.7.4/share/hadoop/common/
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an expl
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFa
Execution log at: /tmp/root/root_20251105065357_101d4117-810d-4f62-9400-
2025-11-05 06:54:01     Starting to launch local task to process map joi
2025-11-05 06:54:02     Dump the side-table for tag: 1 with group count:
local-10004/HashTable-Stage-3/MapJoin-mapfile01--.hashtable
2025-11-05 06:54:02     Uploaded 1 File to: file:/tmp/root/69a13f22-1344
ile01--.hashtable (488 bytes)
2025-11-05 06:54:02     End of local task; Time Taken: 1.047 sec.
+---------------+----------------+---------------------------------+
| s.student_id  |     s.name     |        d.department_name        |
+---------------+----------------+---------------------------------+
| 5             | Mahnoor Adeel  | Department of Computer Science  |
| 6             | Adeel Khan     | Department of Data Science      |
| 7             | Sara Malik     | Department of Mathematics       |
| 1             | John Doe       | Department of Computer Science  |
| 2             | Jane Smith     | Department of Mathematics       |
| 3             | Ali Khan       | Department of Physics           |
+---------------+----------------+---------------------------------+
```

Partition and Bucket:

Suppose that a table named Tab1 contains employee data such as id, name, dept, and yoj (i.e., year of joining). Suppose you need to retrieve the details of all employees who joined in 2012. A query searches the whole table for the required information. However, if you partition the employee data with the year and store it in a separate file, it reduces the query processing time. The following example shows how to partition a file and its data:

The following file contains employeedata table.

Path: /tab1/employeedata/file1

id, name, dept, yoj

1, gopal, TP, 2012

2, kiran, HR, 2012

3, kaleel,SC, 2013

4, Prasanth, SC, 2013

The above data is partitioned into two files using year.

Path: /tab1/employeedata/2012/file2

1, gopal, TP, 2012
2, kiran, HR, 2012

Path: /tab1/employeedata/2013/file3
3, kaleel,SC, 2013
4, Prasanth, SC, 2013
[ToDo: paste output – you can create a partition of your own choice] Partition by year of enrollment:

```
CREATE TABLE students_by_year (
   student_id INT,
   name STRING,
   age INT,
   major STRING
)
PARTITIONED BY (year INT)
STORED AS TEXTFILE;
```

```
root@46ee3faa75fd:/# hdfs dfs -ls /user/hive/warehouse/student_db.db/students_by_year
Found 2 items
drwxrwxr-x   - root supergroup          0 2025-11-05 06:58 /user/hive/warehouse/student_db.db/students_by_year/year=2023
drwxrwxr-x   - root supergroup          0 2025-11-05 06:58 /user/hive/warehouse/student_db.db/students_by_year/year=2024
root@46ee3faa75fd:/#
```

```
0: jdbc:hive2://localhost:10000> SHOW PARTITIONS students_by_year;
+------------+
| partition  |
+------------+
| year=2023  |
| year=2024  |
+------------+
2 rows selected (0.164 seconds)
0: jdbc:hive2://localhost:10000> ls
. . . . . .> ls;
Error: Error while compiling statement: FAILED: ParseException line 1:0 cannot recognize input near 'ls' 'ls' '<EOF>' (state=42000,code=40000)
0: jdbc:hive2://localhost:10000> SELECT * FROM students_by_year;
+-----------------------------+-----------------------+----------------------+------------------------+-----------------------+
| students_by_year.student_id | students_by_year.name | students_by_year.age | students_by_year.major | students_by_year.year |
+-----------------------------+-----------------------+----------------------+------------------------+-----------------------+
| 10                          | Ahmed Ali             | 21                   | Computer Science       | 2023                  |
| 11                          | Fatima Noor           | 19                   | Mathematics            | 2024                  |
| 12                          | Bilal Khan            | 22                   | Physics                | 2024                  |
+-----------------------------+-----------------------+----------------------+------------------------+-----------------------+
3 rows selected (0.197 seconds)
0: jdbc:hive2://localhost:10000>
```

[ToDo: paste output – you can create clusters/buckets of your own choice] Clustered by student_id into 4 buckets:

```
CREATE TABLE students_bucketed (
   student_id INT,
   name STRING,
   age INT,
   major STRING
)
CLUSTERED BY (student_id) INTO 4 BUCKETS
STORED AS TEXTFILE;
```

```
root@46ee3faa75fd:/# hdfs dfs -ls /user/hive/warehouse/student_db.db/students_bucketed

Found 4 items
-rwxrwxr-x   3 root supergroup         42 2025-11-05 07:07 /user/hive/warehouse/student_db.db/students_bucketed/000000_0
-rwxrwxr-x   3 root supergroup         67 2025-11-05 07:07 /user/hive/warehouse/student_db.db/students_bucketed/000001_0
-rwxrwxr-x   3 root supergroup         57 2025-11-05 07:07 /user/hive/warehouse/student_db.db/students_bucketed/000002_0
-rwxrwxr-x   3 root supergroup         50 2025-11-05 07:07 /user/hive/warehouse/student_db.db/students_bucketed/000003_0
root@46ee3faa75fd:/#
```

```
0: jdbc:hive2://localhost:10000> SELECT * FROM students_bucketed;
+-------------------------------+-------------------------+------------------------+-------------------------+
| students_bucketed.student_id  | students_bucketed.name  | students_bucketed.age  | students_bucketed.major |
+-------------------------------+-------------------------+------------------------+-------------------------+
| 4                             | Zuha Ali                | 19                     | Statistics              |
| NULL                          | name                    | NULL                   | major                   |
| 1                             | John Doe                | 20                     | Computer Science        |
| 5                             | Mahnoor Adeel           | 20                     | Computer Science        |
| 2                             | Jane Smith              | 21                     | Mathematics             |
| 6                             | Adeel Khan              | 23                     | Data Science            |
| 3                             | Ali Khan                | 22                     | Physics                 |
| 7                             | Sara Malik              | 21                     | Mathematics             |
+-------------------------------+-------------------------+------------------------+-------------------------+
8 rows selected (0.188 seconds)
0: jdbc:hive2://localhost:10000>
```

[ToDo : paste output – you can explain plan of any query of your own choice – describe the output] Explain the query plan:

EXPLAIN SELECT * FROM students WHERE age > 20;

```
0: jdbc:hive2://localhost:10000> EXPLAIN SELECT * FROM student_info WHERE age>20;
+----------------------------------------------------+
|                      Explain                       |
+----------------------------------------------------+
| STAGE DEPENDENCIES:                                |
|   Stage-0 is a root stage                          |
|                                                    |
| STAGE PLANS:                                       |
|   Stage: Stage-0                                   |
|     Fetch Operator                                 |
|       limit: -1                                    |
|       Processor Tree:                              |
|         TableScan                                  |
|           alias: student_info                      |
|           Statistics: Num rows: 1 Data size: 257 Basic stats: COMPLETE Column stats: NONE |
|           Filter Operator                          |
|             predicate: (age > 20) (type: boolean)  |
|             Statistics: Num rows: 1 Data size: 257 Basic stats: COMPLETE Column stats: NONE |
|             Select Operator                        |
|               expressions: student_id (type: int), name (type: string), age (type: int), major (type: string), gpa (type: float) |
|               outputColumnNames: _col0, _col1, _col2, _col3, _col4 |
|               Statistics: Num rows: 1 Data size: 257 Basic stats: COMPLETE Column stats: NONE |
|               ListSink                             |
|                                                    |
+----------------------------------------------------+
20 rows selected (0.096 seconds)
0: jdbc:hive2://localhost:10000>
```

Built -in Functions:

round, floor, ceil, rand, concat, substr, upper, ucase , lower, lcase, trim, rtrim, regex_replace, size, cast, to_date, year, month, day, get_json_object

[ToDo: try 2 functions and paste output]

```
0: jdbc:hive2://localhost:10000> SELECT
. . . . . .>   name,
. . . . . .>   gpa,
. . . . . .>   ROUND(gpa) AS rounded_gpa
. . . . . .> FROM student_info;
+----------------+-------+--------------+
|      name      |  gpa  | rounded_gpa  |
+----------------+-------+--------------+
| Mahnoor Adeel  | 3.5   | 4.0          |
| Adeel Khan     | 3.9   | 4.0          |
| Sara Malik     | 3.7   | 4.0          |
| name           | NULL  | NULL         |
| John Doe       | 3.5   | 4.0          |
| Jane Smith     | 3.8   | 4.0          |
| Ali Khan       | 3.2   | 3.0          |
| Zuha Ali       | 3.7   | 4.0          |
+----------------+-------+--------------+
8 rows selected (0.122 seconds)
0: jdbc:hive2://localhost:10000> SELECT UPPER('hello hive') AS result;
+-------------+
|   result    |
+-------------+
| HELLO HIVE  |
+-------------+
1 row selected (0.055 seconds)
0: jdbc:hive2://localhost:10000>
```

Aggregate Functions:
count(*), count(expr), sum, avg, min, max

```
0: jdbc:hive2://localhost:10000> SELECT
. . . . . .>   COUNT(*) AS total_students,
. . . . . .>   COUNT(gpa) AS gpa_available,
. . . . . .>   SUM(gpa) AS total_gpa,
. . . . . .>   AVG(gpa) AS average_gpa,
. . . . . .>   MIN(gpa) AS lowest_gpa,
. . . . . .>   MAX(gpa) AS highest_gpa
. . . . . .> FROM student_info;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a di
+----------------+----------------+--------------------+--------------------+-------------+-------------+
| total_students | gpa_available  |     total_gpa      |    average_gpa     | lowest_gpa  | highest_gpa |
+----------------+----------------+--------------------+--------------------+-------------+-------------+
| 8              | 7              | 25.300000190734863 | 3.614285741533552  | 3.2         | 3.9         |
+----------------+----------------+--------------------+--------------------+-------------+-------------+
1 row selected (1.357 seconds)
0: jdbc:hive2://localhost:10000>
0: jdbc:hive2://localhost:10000>
```

Views:
The usage of view in Hive is same as that of the view in SQL. It is a standard RDBMS concept. We can execute all DML operations on a view .
Example: CREATE VIEW emp_30000 AS SELECT * FROM employee WHERE salary>30000;
[ToDo: create a view on the student's database and show output]

```
0: jdbc:hive2://localhost:10000> CREATE VIEW high_gpa_students AS
. . . . . .> SELECT student_id, name, major, gpa
. . . . . .> FROM student_info
. . . . . .> WHERE gpa > 3.5;
No rows affected (0.116 seconds)
0: jdbc:hive2://localhost:10000> SHOW VIEWS;
+--------------------+
|      tab_name      |
+--------------------+
| high_gpa_students  |
+--------------------+
1 row selected (0.044 seconds)
0: jdbc:hive2://localhost:10000> SELECT * FROM high_gpa_students;
+--------------------------------+-------------------------+--------------------------+------------------------+
| high_gpa_students.student_id  | high_gpa_students.name  | high_gpa_students.major  | high_gpa_students.gpa  |
+--------------------------------+-------------------------+--------------------------+------------------------+
| 6                              | Adeel Khan              | Data Science             | 3.9                    |
| 7                              | Sara Malik              | Mathematics              | 3.7                    |
| 2                              | Jane Smith              | Mathematics              | 3.8                    |
| 4                              | Zuha Ali                | Statistics               | 3.7                    |
+--------------------------------+-------------------------+--------------------------+------------------------+
4 rows selected (0.13 seconds)
0: jdbc:hive2://localhost:10000>
```

Indexes

CREATE INDEX inedx_salary ON TABLE employee(salary) AS 'org.apache.hadoop.hive.ql.index.compact.CompactIndexHandler';

[ToDo: create an index on the student's database and execute queries to show difference in performance after indexing]

```
0: jdbc:hive2://localhost:10000> SELECT * FROM student_info WHERE gpa > 3.5;
+--------------------------+--------------------+-------------------+----------------------+--------------------+
| student_info.student_id  | student_info.name  | student_info.age  | student_info.major   | student_info.gpa   |
+--------------------------+--------------------+-------------------+----------------------+--------------------+
| 6                        | Adeel Khan         | 23                | Data Science         | 3.9                |
| 7                        | Sara Malik         | 21                | Mathematics          | 3.7                |
| 2                        | Jane Smith         | 21                | Mathematics          | 3.8                |
| 4                        | Zuha Ali           | 19                | Statistics           | 3.7                |
+--------------------------+--------------------+-------------------+----------------------+--------------------+
4 rows selected (0.105 seconds)
```

```
0: jdbc:hive2://localhost:10000> SHOW INDEX ON student_info;
+----------------------+----------------------+----------------------+-----------------------------------+----------------------+----------+
|      idx_name        |       tab_name       |      col_names       |           idx_tab_name            |       idx_type       | comment  |
+----------------------+----------------------+----------------------+-----------------------------------+----------------------+----------+
| idx_gpa              | student_info         | gpa                  | student_db__student_info_idx_gpa__ | compact              |          |
+----------------------+----------------------+----------------------+-----------------------------------+----------------------+----------+
1 row selected (0.096 seconds)
```

Joins:

Some basic examples are given below.

Inner Join

SELECT c.ID, c.NAME, c.AGE, o.AMOUNT FROM CUSTOMERS c JOIN ORDERS o ON (c.ID = o.CUSTOMER_ID);

```
0: jdbc:hive2://localhost:10000> SELECT g.driverid, g.city, t.model
. . . . . .> FROM geos g
. . . . . .> INNER JOIN trucks t
. . . . . .> ON g.driverid = t.driverid
. . . . . .> LIMIT 10;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hive/lib/log4j-slf4j-impl-2.6.
SLF4J: Found binding in [jar:file:/opt/hadoop-2.7.4/share/hadoop/comm
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an e
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLogge
Execution log at: /tmp/root/root_20251105073910_0e98621b-9799-4c4c-94
2025-11-05 07:39:13     Starting to launch local task to process map
2025-11-05 07:39:14     Dump the side-table for tag: 1 with group cou
/-local-10004/HashTable-Stage-3/MapJoin-mapfile21--.hashtable
2025-11-05 07:39:14     Uploaded 1 File to: file:/tmp/root/69a13f22-1
ile21--.hashtable (3360 bytes)
2025-11-05 07:39:14     End of local task; Time Taken: 0.916 sec.
+-------------+-------------+---------------+
| g.driverid  |   g.city    |    t.model    |
+-------------+-------------+---------------+
| driverid    | city        | model         |
| A54         | Santa Rosa  | Western Star  |
| A20         | Aptos       | Western Star  |
| A40         | Stockton    | Ford          |
| A31         | Willits     | Kenworth      |
| A71         | Irvine      | Peterbilt     |
| A50         | Occidental  | Oshkosh       |
| A51         | Modesto     | Crane         |
| A19         | San Pablo   | Caterpillar   |
| A77         | San Pablo   | Ford          |
+-------------+-------------+---------------+
10 rows selected (5.368 seconds)
```

Left Outer Join

SELECT c.ID, c.NAME, o.AMOUNT, o.DATE FROM CUSTOMERS c LEFT OUTER JOIN ORDERS o ON (c.ID = o.CUSTOMER_ID);

```
0: jdbc:hive2://localhost:10000> SELECT g.driverid, g.city, t.model
. . . . . .> FROM geos g
. . . . . .> LEFT OUTER JOIN trucks t
. . . . . .> ON g.driverid = t.driverid
. . . . . .> LIMIT 10;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hive/lib/log4j-slf4j-impl-2.6
SLF4J: Found binding in [jar:file:/opt/hadoop-2.7.4/share/hadoop/com
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLogg
Execution log at: /tmp/root/root_20251105074017_a5144b55-886c-4c3d-b
2025-11-05 07:40:20     Starting to launch local task to process map
2025-11-05 07:40:20     Dump the side-table for tag: 1 with group co
/-local-10004/HashTable-Stage-3/MapJoin-mapfile31--.hashtable
2025-11-05 07:40:20     Uploaded 1 File to: file:/tmp/root/69a13f22-
ile31--.hashtable (3360 bytes)
2025-11-05 07:40:20     End of local task; Time Taken: 0.769 sec.
+-------------+-------------+---------------+
| g.driverid  |   g.city    |    t.model    |
+-------------+-------------+---------------+
| driverid    | city        | model         |
| A54         | Santa Rosa  | Western Star  |
| A20         | Aptos       | Western Star  |
| A40         | Stockton    | Ford          |
| A31         | Willits     | Kenworth      |
| A71         | Irvine      | Peterbilt     |
| A50         | Occidental  | Oshkosh       |
| A51         | Modesto     | Crane         |
| A19         | San Pablo   | Caterpillar   |
| A77         | San Pablo   | Ford          |
+-------------+-------------+---------------+
10 rows selected (5.221 seconds)
0: jdbc:hive2://localhost:10000>
```

Right Outer Join

SELECT c.ID, c.NAME, o.AMOUNT, o.DATE FROM CUSTOMERS c RIGHT OUTER JOIN ORDERS o ON (c.ID = o.CUSTOMER_ID);

```
0: jdbc:hive2://localhost:10000> SELECT g.driverid, g.city, t.model
. . . . . .> FROM geos g
. . . . . .> RIGHT OUTER JOIN trucks t
. . . . . .> ON g.driverid = t.driverid
. . . . . .> LIMIT 10;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the f
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hive/lib/log4j-slf4j-impl-2.6.2.jar!/c
SLF4J: Found binding in [jar:file:/opt/hadoop-2.7.4/share/hadoop/common/lib/s
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanati
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory
Execution log at: /tmp/root/root_20251105074113_23883b3c-38a0-4e06-9ab2-dafaf
2025-11-05 07:41:16     Starting to launch local task to process map join;
2025-11-05 07:41:16     Dump the side-table for tag: 0 with group count: 101
/-local-10004/HashTable-Stage-3/MapJoin-mapfile40--.hashtable
2025-11-05 07:41:16     Uploaded 1 File to: file:/tmp/root/69a13f22-1344-4735
ile40--.hashtable (120369 bytes)
2025-11-05 07:41:16     End of local task; Time Taken: 0.769 sec.
+-------------+-----------+---------------+
| g.driverid  |  g.city   |    t.model    |
+-------------+-----------+---------------+
| driverid    | city      | model         |
| A1          | Aptos     | Freightliner  |
| A1          | Oceano    | Freightliner  |
| A1          | Gilroy    | Freightliner  |
| A1          | Modesto   | Freightliner  |
| A1          | Palmdale  | Freightliner  |
| A1          | Palmdale  | Freightliner  |
| A1          | Oceano    | Freightliner  |
| A1          | Palmdale  | Freightliner  |
| A1          | Gilroy    | Freightliner  |
+-------------+-----------+---------------+
10 rows selected (5.182 seconds)
```

Full Outer Join:
SELECT c.ID, c.NAME, o.AMOUNT, o.DATE FROM CUSTOMERS c FULL OUTER JOIN ORDERS o ON (c.ID = o.CUSTOMER_ID);

```
0: jdbc:hive2://localhost:10000> SELECT g.driverid, g.city, t.model
. . . . . .> FROM geos g
. . . . . .> FULL OUTER JOIN trucks t
. . . . . .> ON g.driverid = t.driverid
. . . . . .> LIMIT 10;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available
+-------------+---------------+---------------+
| g.driverid  |    g.city     |    t.model    |
+-------------+---------------+---------------+
| A1          | Lodi          | Freightliner  |
| A1          | Klamath       | Freightliner  |
| A1          | Markleeville  | Freightliner  |
| A1          | San Diego     | Freightliner  |
| A1          | Palmdale      | Freightliner  |
| A1          | Palmdale      | Freightliner  |
| A1          | Arbuckle      | Freightliner  |
| A1          | Oceano        | Freightliner  |
| A1          | Oceano        | Freightliner  |
| A1          | Gilroy        | Freightliner  |
+-------------+---------------+---------------+
10 rows selected (1.391 seconds) _
```

Experiment with geolocation file :
Make directories for data files in container (geolocation data, available as geolocation.csv)
docker -compose exec hive -server bash
cd ..
whereis hive
cd home
mkdir geolocationmkdir trucks
exit

```
madeel@bdacourse:~/lab-6-hive/docker-hive$ docker exec -it docker-hive-hive-server
-1 bash
root@a0c1ac0c18b3:/opt# cd /home
root@a0c1ac0c18b3:/home# mkdir geolocation trucks
root@a0c1ac0c18b3:/home# ls -la
total 16
drwxr-xr-x 1 root root 4096 Nov  7 05:41 .
drwxr-xr-x 1 root root 4096 Nov  5 07:26 ..
drwxr-xr-x 2 root root 4096 Nov  7 05:41 geolocation
drwxr-xr-x 2 root root 4096 Nov  7 05:41 trucks
root@a0c1ac0c18b3:/home# exit
exit
madeel@bdacourse:~/lab-6-hive/docker-hive$
```

Copy from local to container
Access hive folder (where yml file is located)
docker cp geolocations.csv docker -hive -master_hive -server_1:/home/geolocation

docker -compose exec hive -server bash
check the copied file

```
madeel@bdacourse:~/lab-6-hive/docker-hive$ docker exec -it docker-hive-hive-server
-1 bash
root@a0c1ac0c18b3:/opt# mv -f /tmp/geolocation.csv /home/geolocation/
root@a0c1ac0c18b3:/opt# mv -f /tmp/trucks.csv      /home/trucks/
root@a0c1ac0c18b3:/opt# ls -la /home/geolocation
total 524
drwxr-xr-x 2 root root   4096 Nov  7 05:56 .
drwxr-xr-x 1 root root   4096 Nov  7 05:41 ..
-rw------- 1 1001 1001 526677 Nov  5 07:24 geolocation.csv
root@a0c1ac0c18b3:/opt# ls -la /home/trucks
total 68
drwxr-xr-x 2 root root  4096 Nov  7 05:56 .
drwxr-xr-x 1 root root  4096 Nov  7 05:41 ..
-rw------- 1 1001 1001 61378 Nov  5 07:24 trucks.csv
root@a0c1ac0c18b3:/opt#
```

Copy from container to hdfs
hadoop fs -mkdir /user/data/
hadoop fs -put -f /home/geolocation /user/data/
hadoop fs -ls /user/data/geolocation

```
root@a0c1ac0c18b3:/opt# hadoop fs -mkdir -p /user/data

root@a0c1ac0c18b3:/opt# hadoop fs -put -f /home/geolocation/geolocation.csv /user/
data/
root@a0c1ac0c18b3:/opt# hadoop fs -put -f /home/trucks/trucks.csv          /user/d
ata/
root@a0c1ac0c18b3:/opt# hadoop fs -ls /user/data
Found 2 items
-rw-r--r--   3 root supergroup     526677 2025-11-07 05:57 /user/data/geolocation.
csv
-rw-r--r--   3 root supergroup      61378 2025-11-07 05:57 /user/data/trucks.csv
root@a0c1ac0c18b3:/opt#
root@a0c1ac0c18b3:/opt#
```

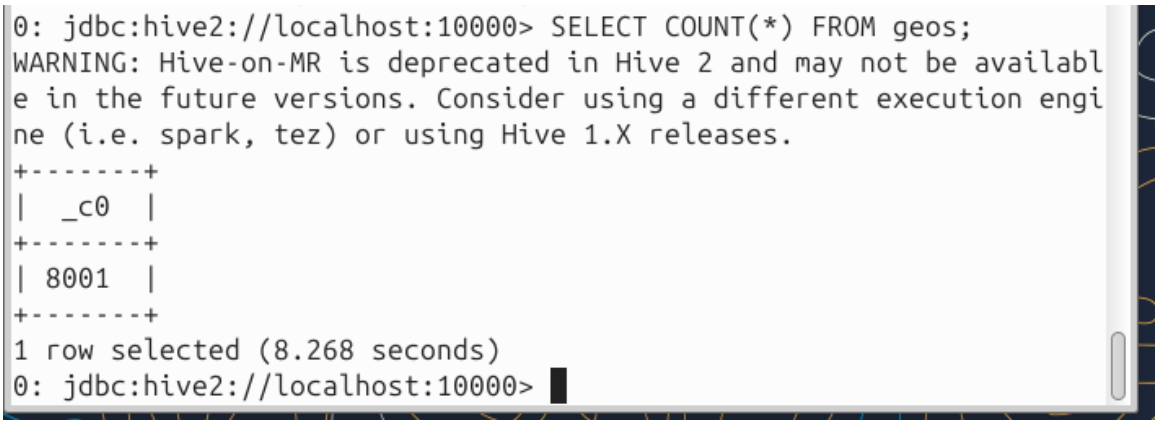
Access hive command prompt
/opt/hive/bin/beeline -u jdbc:hive2://localhost:10000

First, we need to create the table according to the schema of geolocation.csv

```
CREATE TABLE IF NOT EXISTS geos
(
truckid string,
driverid string,
event string,
latitude decimal(5,0),
longitude decimal(5,0),
city string,
state string,
velocity int,
```

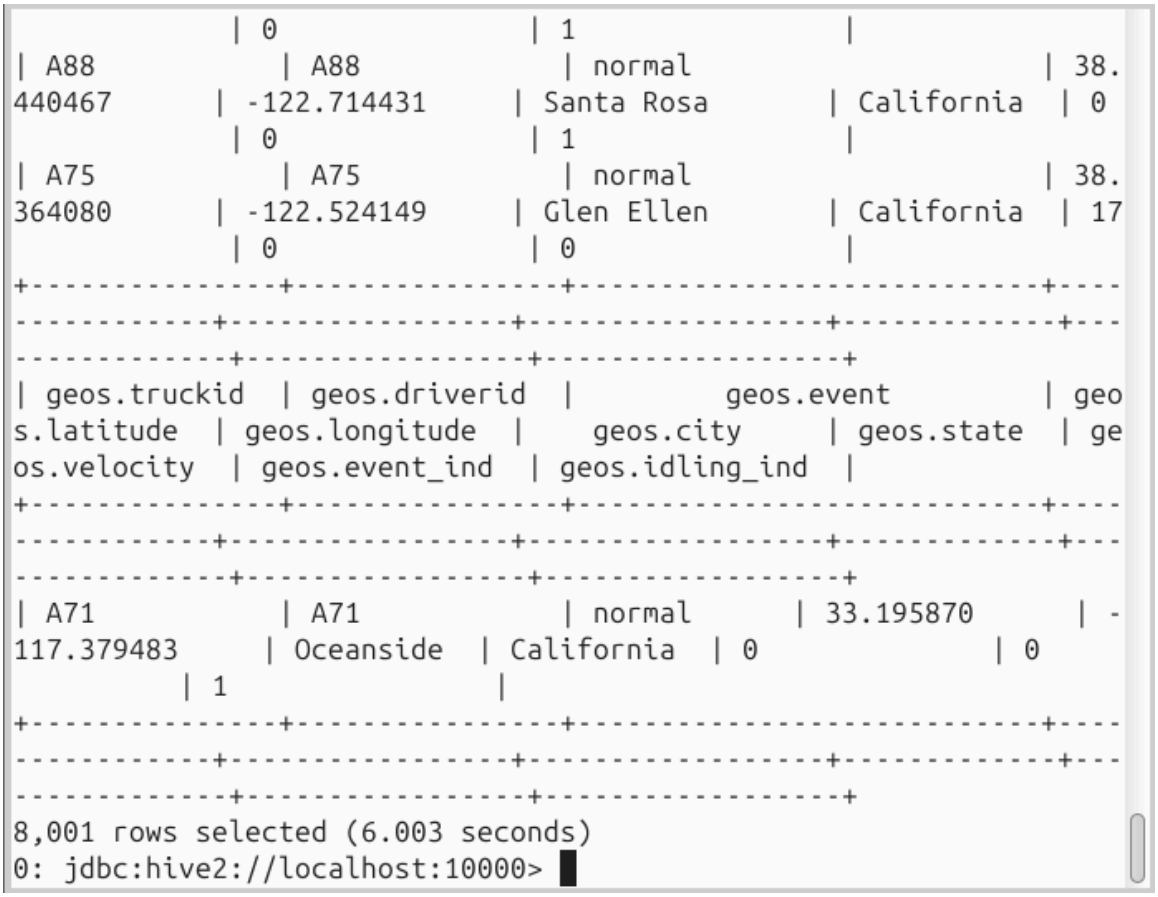
event_ind int,
idling_ind int
)
COMMENT 'Geo Table' ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';

LOAD DATA INPATH '/user/data/geolocation/geolocation.csv' INTO TABLE geos;

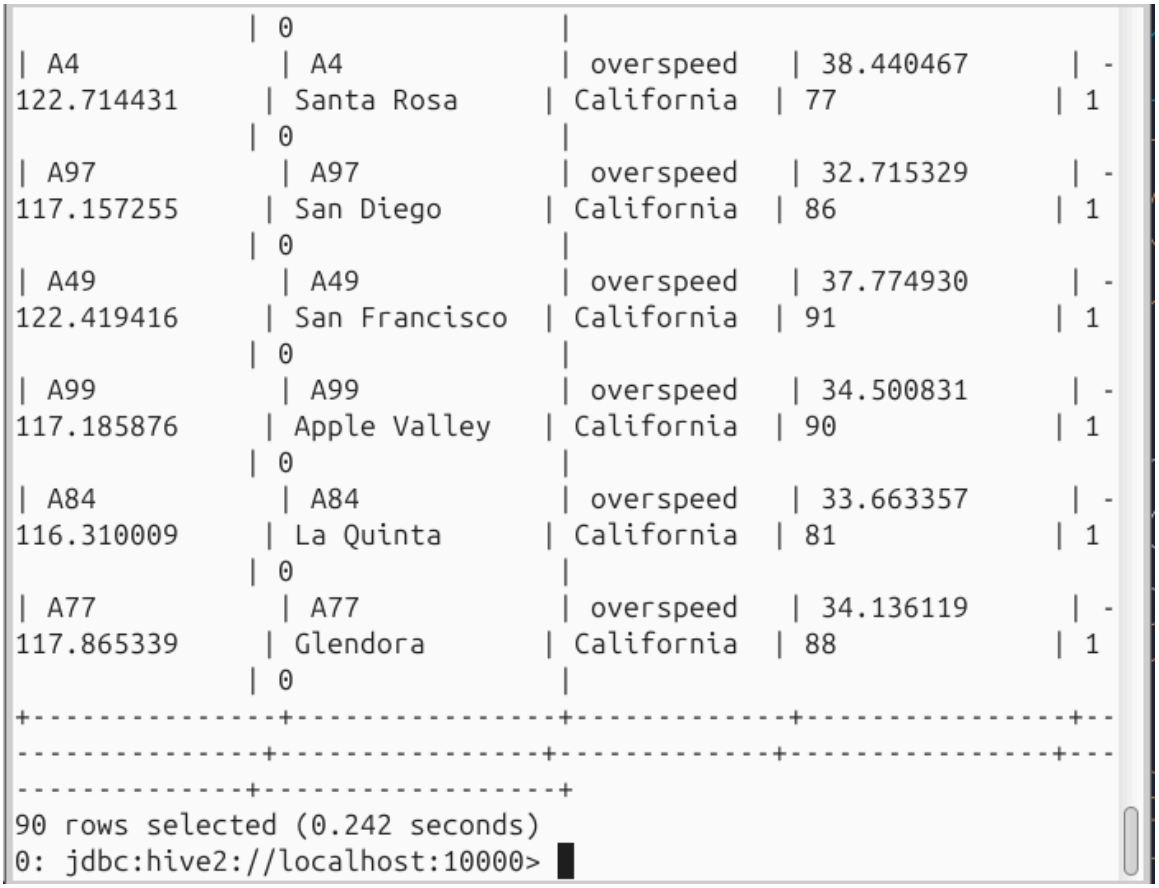


Now, run the following queries:

• select * from geos;

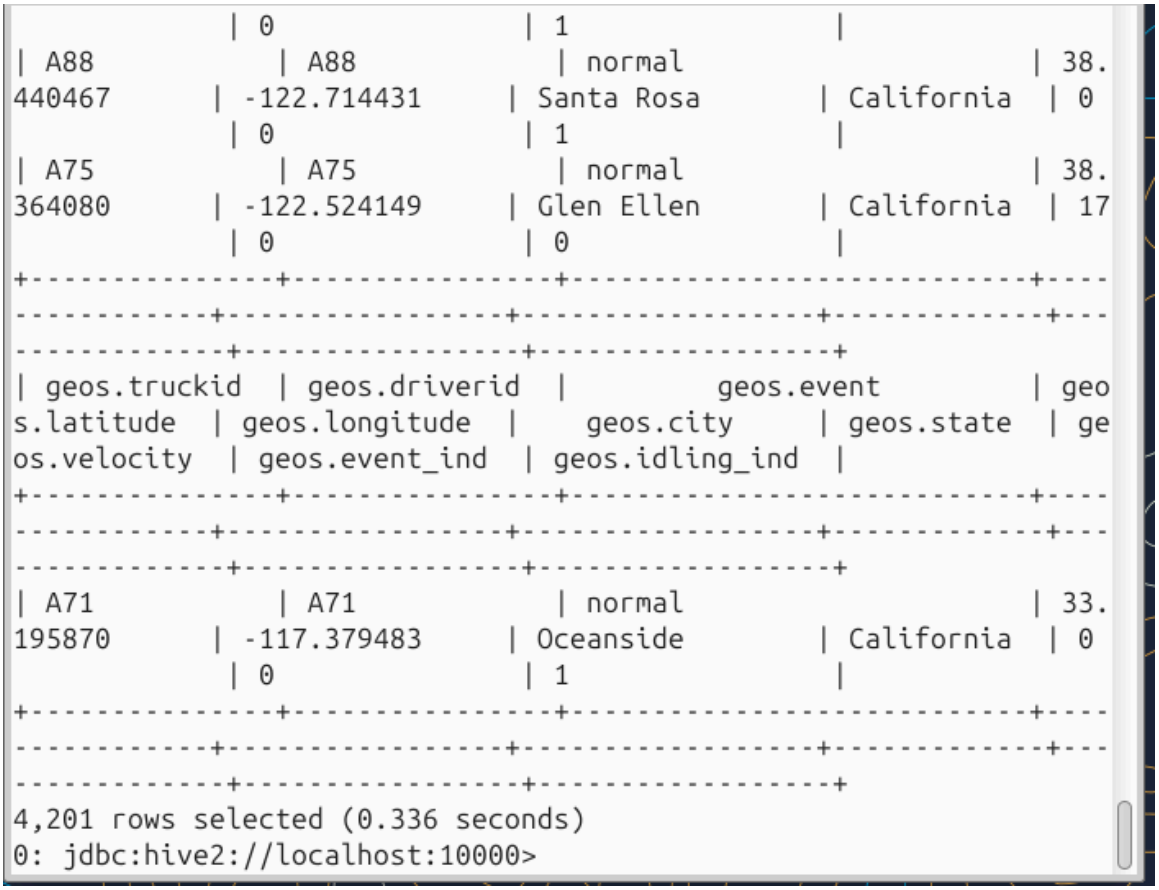


• select * from geos where event = "overspeed";



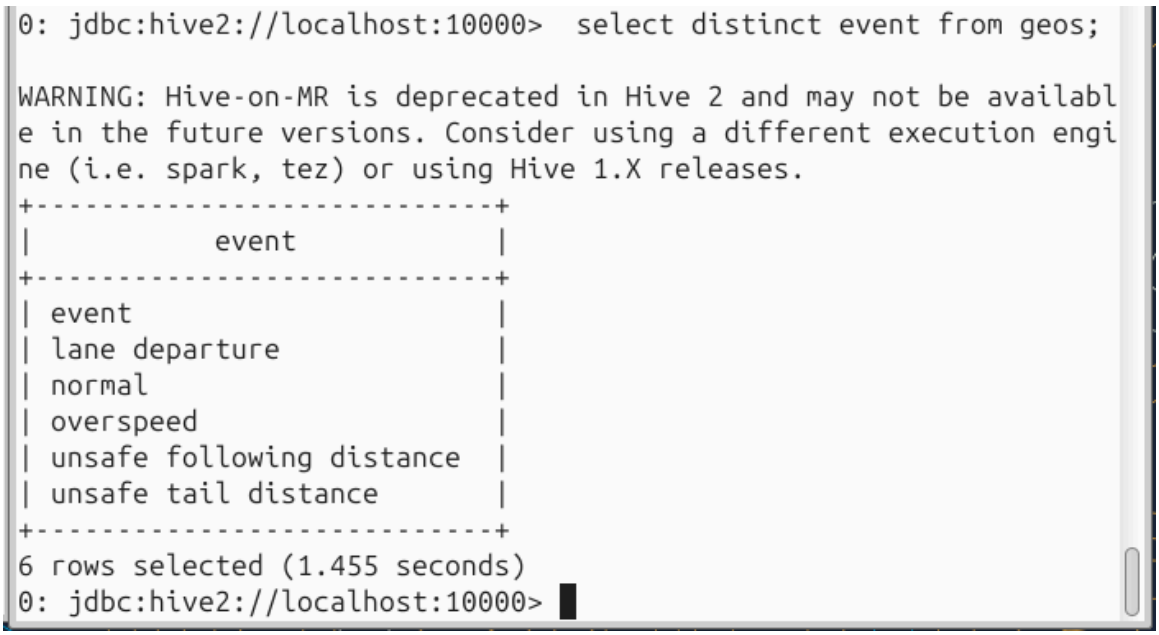
```
               | 0                 |
| A4            | A4              | overspeed   | 38.440467      | -
122.714431     | Santa Rosa      | California  | 77             | 1
               | 0                 |
| A97           | A97             | overspeed   | 32.715329      | -
117.157255     | San Diego       | California  | 86             | 1
               | 0                 |
| A49           | A49             | overspeed   | 37.774930      | -
122.419416     | San Francisco   | California  | 91             | 1
               | 0                 |
| A99           | A99             | overspeed   | 34.500831      | -
117.185876     | Apple Valley    | California  | 90             | 1
               | 0                 |
| A84           | A84             | overspeed   | 33.663357      | -
116.310009     | La Quinta       | California  | 81             | 1
               | 0                 |
| A77           | A77             | overspeed   | 34.136119      | -
117.865339     | Glendora        | California  | 88             | 1
               | 0                 |
+---------------+-----------------+-------------+----------------+--
---------------+-----------------+-------------+----------------+---
--------------+------------------+
90 rows selected (0.242 seconds)
0: jdbc:hive2://localhost:10000>
```

• select * from geos where velocity < 40;



```
             | 0               | 1                 |
| A88           | A88           | normal                      | 38.
440467      | -122.714431     | Santa Rosa       | California  | 0
             | 0               | 1                 |
| A75           | A75           | normal                      | 38.
364080      | -122.524149     | Glen Ellen       | California  | 17
             | 0               | 0                 |
+--------------+---------------+---------------------------+----
-----------+----------------+-----------------+------------+---
------------+----------------+-----------------+
| geos.truckid  | geos.driverid  |         geos.event         | geo
s.latitude  | geos.longitude  |    geos.city     | geos.state  | ge
os.velocity  | geos.event_ind  | geos.idling_ind  |
+--------------+---------------+---------------------------+----
-----------+----------------+-----------------+------------+---
------------+----------------+-----------------+
| A71           | A71           | normal                      | 33.
195870      | -117.379483     | Oceanside        | California  | 0
             | 0               | 1                 |
+--------------+---------------+---------------------------+----
-----------+----------------+-----------------+------------+---
------------+----------------+-----------------+
4,201 rows selected (0.336 seconds)
0: jdbc:hive2://localhost:10000>
```

• select distinct event from geos;



```
0: jdbc:hive2://localhost:10000>  select distinct event from geos;

WARNING: Hive-on-MR is deprecated in Hive 2 and may not be availabl
e in the future versions. Consider using a different execution engi
ne (i.e. spark, tez) or using Hive 1.X releases.
+----------------------------+
|           event            |
+----------------------------+
| event                      |
| lane departure             |
| normal                     |
| overspeed                  |
| unsafe following distance  |
| unsafe tail distance       |
+----------------------------+
6 rows selected (1.455 seconds)
0: jdbc:hive2://localhost:10000>
```

• select avg(velocity) from geos where event ="lane departure" group by event;

```
0: jdbc:hive2://localhost:10000> select avg(velocity) from geos whe
re event ="lane departure" group by event;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be availabl
e in the future versions. Consider using a different execution engi
ne (i.e. spark, tez) or using Hive 1.X releases.
+---------------------+
|         _c0         |
+---------------------+
| 42.328947368421055  |
+---------------------+
1 row selected (1.403 seconds)
0: jdbc:hive2://localhost:10000>
```

- select * from geos order by velocity desc limit 5;

```
0: jdbc:hive2://localhost:10000> select * from geos order by velocity desc limit 5;
WARNING: 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. spark, tez) or using Hive 1.X releases.
+---------------+----------------+-------------+----------------+---------------+----------------+-----------------+
| geos.truckid  | geos.driverid  | geos.event  | geos.latitude  | geos.longitude  |   geos.city    | geos.state  | geos.velocity  | geos.event_ind  | geos.idling_ind  |
+---------------+----------------+-------------+----------------+---------------+----------------+-----------------+
| A71           | A71            | overspeed   | 34.106676      | -117.806726     | San Dimas      | California  | 100            | 1               | 0                |
| A38           | A38            | overspeed   | 38.440467      | -122.714431     | Santa Rosa     | California  | 94             | 1               | 0                |
| A27           | A27            | overspeed   | 37.484938      | -119.966284     | Mariposa       | California  | 94             | 1               | 0                |
| A26           | A26            | overspeed   | 37.774930      | -122.419416     | San Francisco  | California  | 94             | 1               | 0                |
| A49           | A49            | overspeed   | 37.774930      | -122.419416     | San Francisco  | California  | 91             | 1               | 0                |
+---------------+----------------+-------------+----------------+---------------+----------------+-----------------+
5 rows selected (1.321 seconds)
0: jdbc:hive2://localhost:10000>
```

[ToDo: Show queries and their output]

Exercises

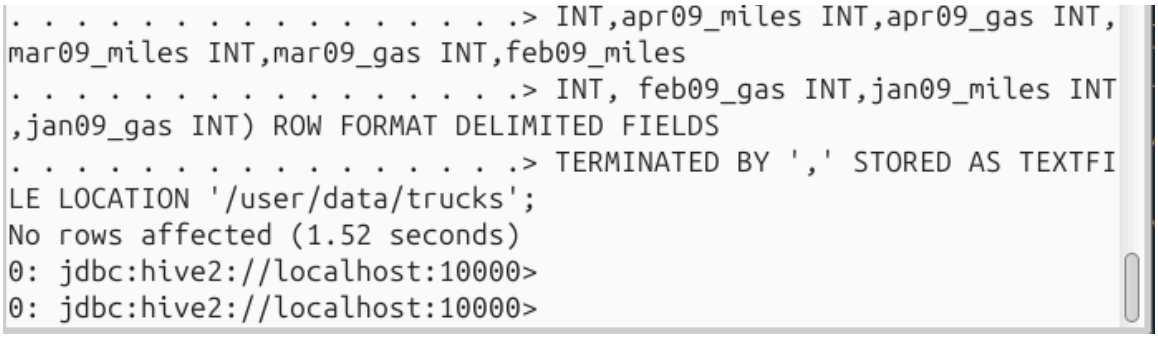
- Get all the cities for driver id A54
- Get driver who have visited the least amount of cities
- Get driver who on average drives the slowest
- Using the like statement get all cities with name starting from A

Importing trucks file

We need to import the trucks data file now.

Make a “trucks” directory, copy from local to container, copy from container to hdfs, and access hive command prompt (as for the above ge olocation file)

CREATE EXTERNAL TABLE IF NOT EXISTS trucks (driverid STRING,truckid STRING,model STRING,jun13_miles INT,jun13_gas INT,may13_miles INT,may13_gas INT,apr13_miles INT,apr13_gas INT,mar13_miles INT,mar13_gas INT,feb13_miles INT,feb13_gas INT,jan13_miles INT,jan13_gas INT,dec12_miles INT,dec12_gas INT,nov12_miles INT,nov12_gas INT,oct12_miles INT,oct12_gas INT,sep12_miles INT,sep12_gas INT,aug12_miles INT,aug12_gas INT,jul12_miles INT,jul12_gas INT,jun12_miles INT,jun12_gas INT,may12_miles INT,may12_gas INT,apr12_miles INT,apr12_gas INT,mar12_miles INT,mar12_gas INT,feb12_miles INT,feb12_gas INT,jan12_miles INT,jan12_gas INT,dec11_miles INT,dec11_gas INT,nov11_miles INT,nov11_gas INT,oct11_miles INT,oct11_gas INT,sep11_miles INT,sep11_gas INT,aug11_miles INT,aug11_gas INT,jul11_miles INT,jul11_gas INT,jun11_miles INT,jun11_gas INT,may11_miles INT,may11_gas INT,apr11_miles INT,apr11_gas INT,mar11_miles INT,mar11_gas INT,feb11_miles INT,feb11_gas INT,jan11_miles INT,jan11_gas INT,dec10_miles INT,dec10_gas INT,nov10_miles INT,nov10_gas INT,oct10_miles INT,oct10_gas INT,sep10_miles INT,sep10_gas INT,aug10_miles INT,aug10_gas INT,jul10_miles INT,jul10_gas INT,jun10_miles INT,jun10_gas INT,may10_miles INT,may10_gas INT,apr10_miles INT,apr10_gas INT,mar10_miles INT,mar10_gas INT,feb10_miles INT,feb10_gas INT,jan10_miles INT,jan10_gas INT,dec09_miles INT,dec09_gas INT,nov09_miles INT,nov09_gas INT, oct09_miles INT,oct09_gas INT,sep09_miles INT,sep09_gas INT,aug09_miles INT,aug09_gas INT,jul09_miles INT,jul09_gas INT,jun09_miles INT,jun09_gas INT,may09_miles INT,may09_gas INT,apr09_miles INT,apr09_gas INT,mar09_miles INT,mar09_gas INT,feb09_miles INT, feb09_gas INT,jan09_miles INT,jan09_gas INT) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS TEXTFILE LOCATION '/user/data/trucks.csv';



[ToDo: Show output of above – what advantage do we have for an external table]

When we drop an EXTERNAL table, Hive deletes only the table metadata but keeps the actual data files in HDFS intact.

This means the data is not lost and the table can be recreated later using the same LOCATION.

Managed Table (default)	**External Table**
Dropping the table deletes the data from HDFS	Dropping the table keeps the data in HDFS
Hive owns the data	Hive only references the data

Good for temporary/derived data	Good for shared data / data ingestion

```
0: jdbc:hive2://localhost:10000> DESCRIBE FORMATTED trucks;
+-------------------------------+----------------------------------------------------+-----------------------+
|           col_name            |                     data_type                      |        comment        |
+-------------------------------+----------------------------------------------------+-----------------------+
| # col_name                    | data_type                                          | comment               |
|                               | NULL                                               | NULL                  |
| driverid                      | string                                             |                       |
| truckid                       | string                                             |                       |
| model                         | string                                             |                       |
| jun13_miles                   | int                                                |                       |
| jun13_gas                     | int                                                |                       |
| may13_miles                   | int                                                |                       |
| may13_gas                     | int                                                |                       |
| apr13_miles                   | int                                                |                       |
| apr13_gas                     | int                                                |                       |
| mar13_miles                   | int                                                |                       |
| mar13_gas                     | int                                                |                       |
| feb13_miles                   | int                                                |                       |
| feb13_gas                     | int                                                |                       |
| jan13_miles                   | int                                                |                       |
| jan13_gas                     | int                                                |                       |
| dec12_miles                   | int                                                |                       |
| dec12_gas                     | int                                                |                       |
| nov12_miles                   | int                                                |                       |
| nov12_gas                     | int                                                |                       |
| oct12_miles                   | int                                                |                       |
| oct12_gas                     | int                                                |                       |
| sep12_miles                   | int                                                |                       |
| sep12_gas                     | int                                                |                       |
| aug12_miles                   | int                                                |                       |
| aug12_gas                     | int                                                |                       |
| jul12_miles                   | int                                                |                       |
```

```
+-------------------------------+----------------------------------------------------+-----------------------+
|           col_name            |                     data_type                      |        comment        |
+-------------------------------+----------------------------------------------------+-----------------------+
| jul09_gas                     | int                                                |                       |
| jun09_miles                   | int                                                |                       |
| jun09_gas                     | int                                                |                       |
| may09_miles                   | int                                                |                       |
| may09_gas                     | int                                                |                       |
| apr09_miles                   | int                                                |                       |
| apr09_gas                     | int                                                |                       |
| mar09_miles                   | int                                                |                       |
| mar09_gas                     | int                                                |                       |
| feb09_miles                   | int                                                |                       |
| feb09_gas                     | int                                                |                       |
| jan09_miles                   | int                                                |                       |
| jan09_gas                     | int                                                |                       |
|                               | NULL                                               | NULL                  |
| # Detailed Table Information  | NULL                                               | NULL                  |
| Database:                     | student_db                                         | NULL                  |
| Owner:                        | root                                               | NULL                  |
| CreateTime:                   | Wed Nov 05 07:29:18 UTC 2025                       | NULL                  |
| LastAccessTime:               | UNKNOWN                                            | NULL                  |
| Retention:                    | 0                                                  | NULL                  |
| Location:                     | hdfs://namenode:8020/user/hive/warehouse/student_db.db/trucks | NULL          |
| Table Type:                   | EXTERNAL_TABLE                                     | NULL                  |
| Table Parameters:             | NULL                                               | NULL                  |
|                               | EXTERNAL                                           | TRUE                  |
|                               | numFiles                                           | 1                     |
|                               | numRows                                            | 0                     |
|                               | rawDataSize                                        | 0                     |
|                               | totalSize                                          | 61378                 |
|                               | transient_lastDdlTime                              | 1762328178            |
|                               | NULL                                               | NULL                  |
```

```
|                               | totalSize                                          | 61378                | |
|                               | transient_lastDdlTime                              | 1762328178           |
|                               | NULL                                               | NULL                 |
| # Storage Information         | NULL                                               | NULL                 |
| SerDe Library:                | org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe | NULL                 |
| InputFormat:                  | org.apache.hadoop.mapred.TextInputFormat           | NULL                 |
| OutputFormat:                 | org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat | NULL          |
| Compressed:                   | No                                                 | NULL                 |
| Num Buckets:                  | -1                                                 | NULL                 |
| Bucket Columns:               | []                                                 | NULL                 |
| Sort Columns:                 | []                                                 | NULL                 |
| Storage Desc Params:          | NULL                                               | NULL                 |
|                               | field.delim                                        | ,                    |
|                               | serialization.format                               | ,                    |
+-------------------------------+----------------------------------------------------+----------------------+
141 rows selected (1.321 seconds)
0: jdbc:hive2://localhost:10000>
```

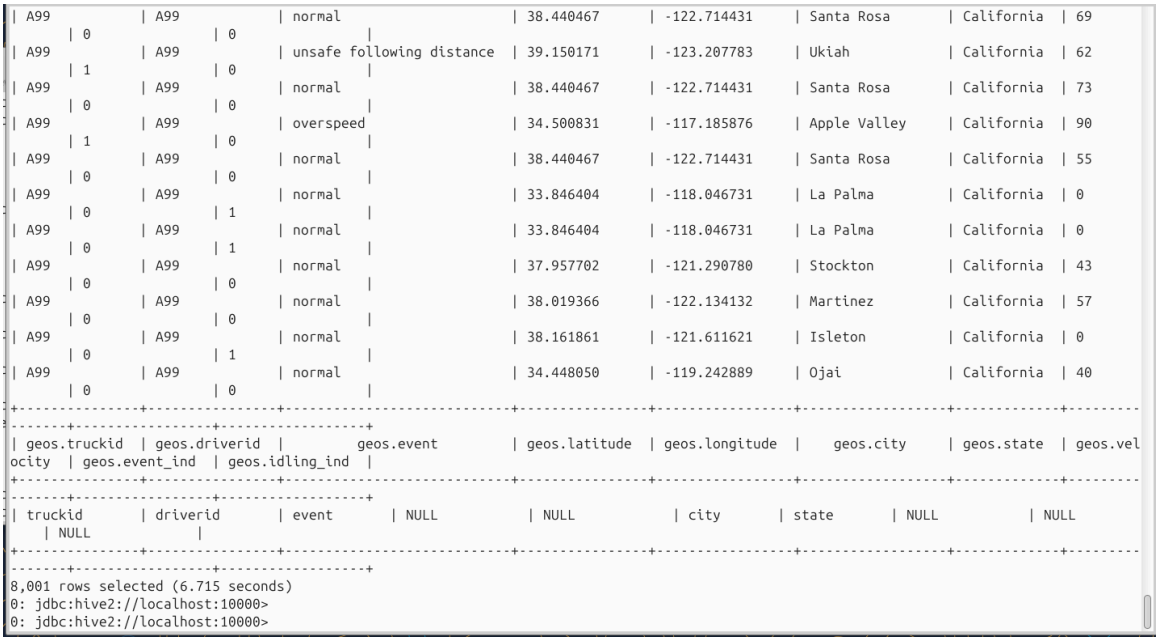
[ToDo: What is each query doing – explain in a single sentence only in your own words:]

• select count(model) as modelCount, model from trucks group by model order by modelCount desc;

```
0: jdbc:hive2://localhost:10000> SELECT COUNT(model) AS modelCo
unt, model
. . . . . .> FROM trucks
. . . . . .> GROUP BY model
. . . . . .> ORDER BY modelCount DESC;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be avai
lable in the future versions. Consider using a different execut
ion engine (i.e. spark, tez) or using Hive 1.X releases.
+-------------+---------------+
| modelcount  |     model     |
+-------------+---------------+
| 20          | Ford          |
| 19          | Caterpillar   |
| 16          | Peterbilt     |
| 9           | Volvo         |
| 9           | Navistar      |
| 7           | Hino          |
| 6           | Kenworth      |
| 5           | Freightliner  |
| 4           | Oshkosh       |
| 3           | Western Star  |
| 2           | Crane         |
| 1           | model         |
+-------------+---------------+
12 rows selected (3.88 seconds)
0: jdbc:hive2://localhost:10000>
```

Counts how many trucks there are of each model and shows the model with the highest count first.

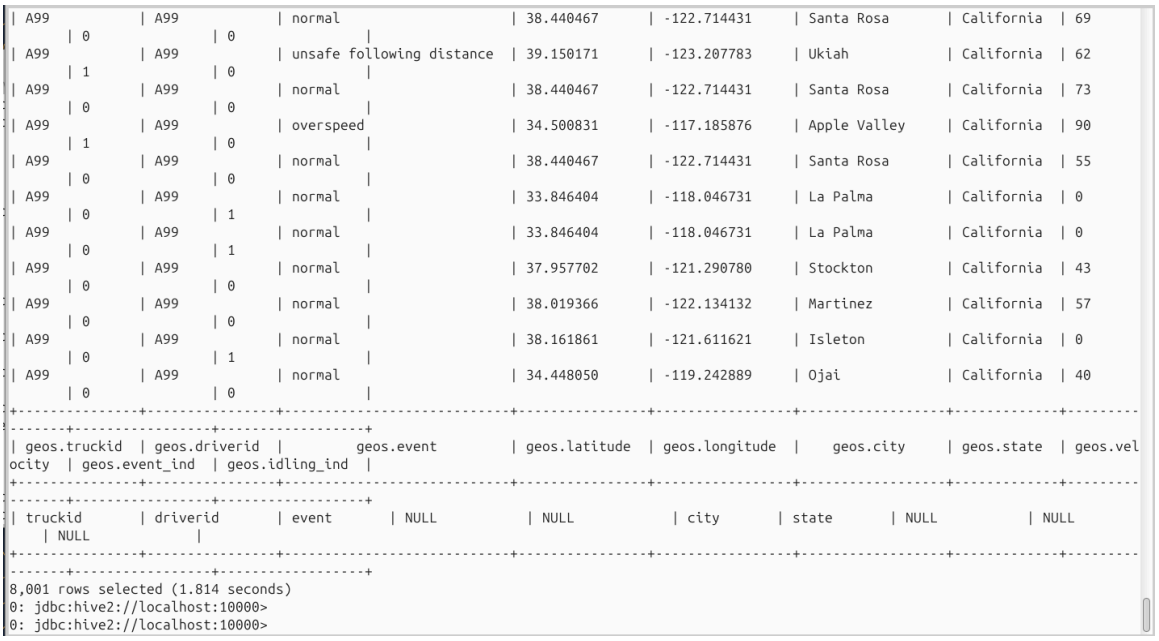
• select * from geos ORDER BY truckid;



```
| A99           | A99           | normal                    | 38.440467     | -122.714431    | Santa Rosa       | California  | 69
       | 0             | 0             |
| A99           | A99           | unsafe following distance | 39.150171     | -123.207783    | Ukiah            | California  | 62
       | 1             | 0             |
| A99           | A99           | normal                    | 38.440467     | -122.714431    | Santa Rosa       | California  | 73
       | 0             | 0             |
| A99           | A99           | overspeed                 | 34.500831     | -117.185876    | Apple Valley     | California  | 90
       | 1             | 0             |
| A99           | A99           | normal                    | 38.440467     | -122.714431    | Santa Rosa       | California  | 55
       | 0             | 0             |
| A99           | A99           | normal                    | 33.846404     | -118.046731    | La Palma         | California  | 0
       | 0             | 1             |
| A99           | A99           | normal                    | 33.846404     | -118.046731    | La Palma         | California  | 0
       | 0             | 1             |
| A99           | A99           | normal                    | 37.957702     | -121.290780    | Stockton         | California  | 43
       | 0             | 0             |
| A99           | A99           | normal                    | 38.019366     | -122.134132    | Martinez         | California  | 57
       | 0             | 0             |
| A99           | A99           | normal                    | 38.161861     | -121.611621    | Isleton          | California  | 0
       | 0             | 1             |
| A99           | A99           | normal                    | 34.448050     | -119.242889    | Ojai             | California  | 40
       | 0             | 0             |
+---------------+---------------+---------------------------+---------------+----------------+------------------+-------------+---------
-------+---------------+-----------------+
| geos.truckid  | geos.driverid |         geos.event        | geos.latitude | geos.longitude |    geos.city     | geos.state  | geos.vel
ocity  | geos.event_ind  | geos.idling_ind  |
+---------------+---------------+---------------------------+---------------+----------------+------------------+-------------+---------
-------+---------------+-----------------+
| truckid       | driverid      | event       | NULL          | NULL           | city       | state       | NULL          | NULL
    | NULL            |
+---------------+---------------+---------------------------+---------------+----------------+------------------+-------------+---------
-------+---------------+-----------------+
8,001 rows selected (6.715 seconds)
0: jdbc:hive2://localhost:10000>
0: jdbc:hive2://localhost:10000>
```

Globally sorts the whole geos table by truckid using a single reducer (total order).

• select * from geos SORT BY driverid ASC;



```
| A99           | A99           | normal                    | 38.440467     | -122.714431    | Santa Rosa       | California  | 69
       | 0             | 0             |
| A99           | A99           | unsafe following distance | 39.150171     | -123.207783    | Ukiah            | California  | 62
       | 1             | 0             |
| A99           | A99           | normal                    | 38.440467     | -122.714431    | Santa Rosa       | California  | 73
       | 0             | 0             |
| A99           | A99           | overspeed                 | 34.500831     | -117.185876    | Apple Valley     | California  | 90
       | 1             | 0             |
| A99           | A99           | normal                    | 38.440467     | -122.714431    | Santa Rosa       | California  | 55
       | 0             | 0             |
| A99           | A99           | normal                    | 33.846404     | -118.046731    | La Palma         | California  | 0
       | 0             | 1             |
| A99           | A99           | normal                    | 33.846404     | -118.046731    | La Palma         | California  | 0
       | 0             | 1             |
| A99           | A99           | normal                    | 37.957702     | -121.290780    | Stockton         | California  | 43
       | 0             | 0             |
| A99           | A99           | normal                    | 38.019366     | -122.134132    | Martinez         | California  | 57
       | 0             | 0             |
| A99           | A99           | normal                    | 38.161861     | -121.611621    | Isleton          | California  | 0
       | 0             | 1             |
| A99           | A99           | normal                    | 34.448050     | -119.242889    | Ojai             | California  | 40
       | 0             | 0             |
+---------------+---------------+---------------------------+---------------+----------------+------------------+-------------+---------
-------+---------------+-----------------+
| geos.truckid  | geos.driverid |         geos.event        | geos.latitude | geos.longitude |    geos.city     | geos.state  | geos.vel
ocity  | geos.event_ind  | geos.idling_ind  |
+---------------+---------------+---------------------------+---------------+----------------+------------------+-------------+---------
-------+---------------+-----------------+
| truckid       | driverid      | event       | NULL          | NULL           | city       | state       | NULL          | NULL
    | NULL            |
+---------------+---------------+---------------------------+---------------+----------------+------------------+-------------+---------
-------+---------------+-----------------+
8,001 rows selected (1.814 seconds)
0: jdbc:hive2://localhost:10000>
0: jdbc:hive2://localhost:10000>
```

Sorts rows by driverid within each reducer (not a global total order).

• select driverid, city from geos DISTRIBUTE BY driverid; (what is distribute by and its advantage?)

```
| A10        | San Dimas         |
| A72        | Stockton          |
| A49        | Ukiah             |
| A68        | Santa Rosa        |
| A94        | San Quentin       |
| A62        | San Dimas         |
| A65        | Mariposa          |
| A43        | Knightsen         |
| A12        | Knightsen         |
| A10        | Kneeland          |
| A98        | Oceanside         |
| A2         | Jacumba           |
| A48        | Roseville         |
| A54        | Bakersfield       |
| A5         | Klamath           |
| A86        | San Diego         |
| A89        | Arbuckle          |
| A92        | Mariposa          |
| A77        | San Pablo         |
| A19        | San Pablo         |
| A51        | Modesto           |
| A50        | Occidental        |
| A71        | Irvine            |
| A31        | Willits           |
| A40        | Stockton          |
| A20        | Aptos             |
| A54        | Santa Rosa        |
+------------+-------------------+
| driverid   |       city        |
+------------+-------------------+
| driverid   | city   |
+------------+-------------------+
8,001 rows selected (1.403 seconds)
0: jdbc:hive2://localhost:10000>
0: jdbc:hive2://localhost:10000>
```

Sends rows with the same driverid to the same reducer (good for parallel grouping/joins); advantage: reduces data shuffling for same-key work.

• select driverid, city from geos CLUSTER BY driverid; (what is cluster by and its advantage?)

```
| A99       | Redding          |
| A99       | San Fernando     |
| A99       | Homeland         |
| A99       | Santa Rosa       |
| A99       | Santa Rosa       |
| A99       | Isleton          |
| A99       | Santa Rosa       |
| A99       | San Francisco    |
| A99       | Santa Rosa       |
| A99       | Ukiah            |
| A99       | Santa Rosa       |
| A99       | Homeland         |
| A99       | San Francisco    |
| A99       | Ojai             |
| A99       | Ojai             |
| A99       | Isleton          |
| A99       | Santa Rosa       |
| A99       | Ukiah            |
| A99       | Santa Rosa       |
| A99       | Apple Valley     |
| A99       | Santa Rosa       |
| A99       | La Palma         |
| A99       | La Palma         |
| A99       | Stockton         |
| A99       | Martinez         |
| A99       | Isleton          |
| A99       | Ojai             |
+-----------+------------------+
| driverid  |       city       |
+-----------+------------------+
| driverid  | city  |
+-----------+------------------+
8,001 rows selected (1.346 seconds)
0: jdbc:hive2://localhost:10000>
0: jdbc:hive2://localhost:10000>
```

Combination of DISTRIBUTE BY driverid and SORT BY driverid in one step; advantage: same keys land together and are sorted, which speeds downstream processing.

• UPDATE geolocation SET driverid = null WHERE event = “normal”;

docker Hive uses **TEXTFILE non-transactional** tables, so UPDATE/MERGE won’t execute. So thats why we will use INSERT OVERWRITE

```
0: jdbc:hive2://localhost:10000> -- Overwrite geos with the transformed result
0: jdbc:hive2://localhost:10000> INSERT OVERWRITE TABLE geos
. . . . . . . . . . . . . . . .> SELECT
. . . . . . . . . . . . . . . .>   truckid,
. . . . . . . . . . . . . . . .>   CASE WHEN event = 'normal' THEN NULL ELSE driverid END AS driverid,
. . . . . . . . . . . . . . . .>   event,
. . . . . . . . . . . . . . . .>   latitude,
. . . . . . . . . . . . . . . .>   longitude,
. . . . . . . . . . . . . . . .>   city,
. . . . . . . . . . . . . . . .>   state,
. . . . . . . . . . . . . . . .>   velocity,
. . . . . . . . . . . . . . . .>   event_ind,
. . . . . . . . . . . . . . . .>   idling_ind
. . . . . . . . . . . . . . . .> FROM geos;
WARNING: 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. spark, tez) or using Hive 1.X releases.
No rows affected (2.802 seconds)
0: jdbc:hive2://localhost:10000>
0: jdbc:hive2://localhost:10000>
```

Rewrites the geos table so any row with event='normal' gets driverid set to NULL.

• MERGE INTO geos USING
(SELECT * FROM trucks) sub ON sub.driverid=geos.driverid
WHEN MATCHED THEN UPDATE SET truckid = sub.truckid, driverid = sub.driverid
WHEN NOT MATCHED THEN INSERT VALUES (sub.truckid, sub.driverid);

Our Hive (2.3, TEXTFILE) is non-ACID, so we used the portable INSERT OVERWRITE approach which is the standard workaround.

```
0: jdbc:hive2://localhost:10000> -- 1) Update: overwrite geos with values from trucks where driver ids match
0: jdbc:hive2://localhost:10000> INSERT OVERWRITE TABLE geos
. . . . . .> SELECT
. . . . . .>   COALESCE(t.truckid, g.truckid)    AS truckid,
. . . . . .>   COALESCE(t.driverid, g.driverid)  AS driverid,
. . . . . .>   g.event, g.latitude, g.longitude, g.city, g.state,
. . . . . .>   g.velocity, g.event_ind, g.idling_ind
. . . . . .> FROM geos g
. . . . . .> LEFT JOIN trucks t
. . . . . .>   ON t.driverid = g.driverid;
WARNING: 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. spark, tez) or using Hive 1.X releases.
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hive/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hadoop-2.7.4/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.clas
s]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Execution log at: /tmp/root/root_20251108154443_0a40afbd-459e-4ad2-9122-be1fd86b6f65.log
2025-11-08 15:44:48     Starting to launch local task to process map join;      maximum memory = 477626368
2025-11-08 15:44:50     Dump the side-table for tag: 1 with group count: 101 into file: file:/tmp/root/69a13f22-1344-4735-9e95-44cc29041616/
hive_2025-11-08_15-44-43_483_8760890177756052994-4/-local-10002/HashTable-Stage-4/MapJoin-mapfile51--.hashtable
2025-11-08 15:44:50     Uploaded 1 File to: file:/tmp/root/69a13f22-1344-4735-9e95-44cc29041616/hive_2025-11-08_15-44-43_483_876089017775605
2994-4/-local-10002/HashTable-Stage-4/MapJoin-mapfile51--.hashtable (2892 bytes)
2025-11-08 15:44:50     End of local task; Time Taken: 1.423 sec.
No rows affected (8.651 seconds)
0: jdbc:hive2://localhost:10000>
0: jdbc:hive2://localhost:10000> -- 2) Insert not-matched: add rows present in trucks but missing in geos
0: jdbc:hive2://localhost:10000> INSERT INTO TABLE geos
. . . . . .> SELECT
. . . . . .>   t.truckid,
. . . . . .>   t.driverid,
. . . . . .>   'from_trucks_only' AS event,
```

```
0: jdbc:hive2://localhost:10000> -- 2) Insert not-matched: add rows present in trucks but missing in geos
0: jdbc:hive2://localhost:10000> INSERT INTO TABLE geos
. . . . . .> SELECT
. . . . . .>   t.truckid,
. . . . . .>   t.driverid,
. . . . . .>   'from_trucks_only' AS event,
. . . . . .>   CAST(NULL AS DECIMAL(10,6)) AS latitude,
. . . . . .>   CAST(NULL AS DECIMAL(10,6)) AS longitude,
. . . . . .>   NULL AS city,
. . . . . .>   NULL AS state,
. . . . . .>   CAST(NULL AS INT) AS velocity,
. . . . . .>   CAST(NULL AS INT) AS event_ind,
. . . . . .>   CAST(NULL AS INT) AS idling_ind
. . . . . .> FROM trucks t
. . . . . .> LEFT JOIN geos g
. . . . . .>   ON g.driverid = t.driverid
. . . . . .> WHERE g.driverid IS NULL;
WARNING: 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. spark, tez) or using Hive 1.X releases.
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hive/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hadoop-2.7.4/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.clas
s]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Execution log at: /tmp/root/root_20251108154452_d3be516e-4cba-46e5-89d7-9618a5a3ccc6.log
2025-11-08 15:44:56     Starting to launch local task to process map join;      maximum memory = 477626368
2025-11-08 15:44:57     Dump the side-table for tag: 1 with group count: 101 into file: file:/tmp/root/69a13f22-1344-4735-9e95-44cc29041616/
hive_2025-11-08_15-44-52_189_1606934293283349949-4/-local-10002/HashTable-Stage-4/MapJoin-mapfile61--.hashtable
2025-11-08 15:44:57     Uploaded 1 File to: file:/tmp/root/69a13f22-1344-4735-9e95-44cc29041616/hive_2025-11-08_15-44-52_189_160693429328334
9949-4/-local-10002/HashTable-Stage-4/MapJoin-mapfile61--.hashtable (34137 bytes)
2025-11-08 15:44:57     End of local task; Time Taken: 1.146 sec.
No rows affected (6.949 seconds)
0: jdbc:hive2://localhost:10000>
```

First part updates matching rows from trucks; second part inserts new rows for drivers that exist only in trucks.

• Joins:

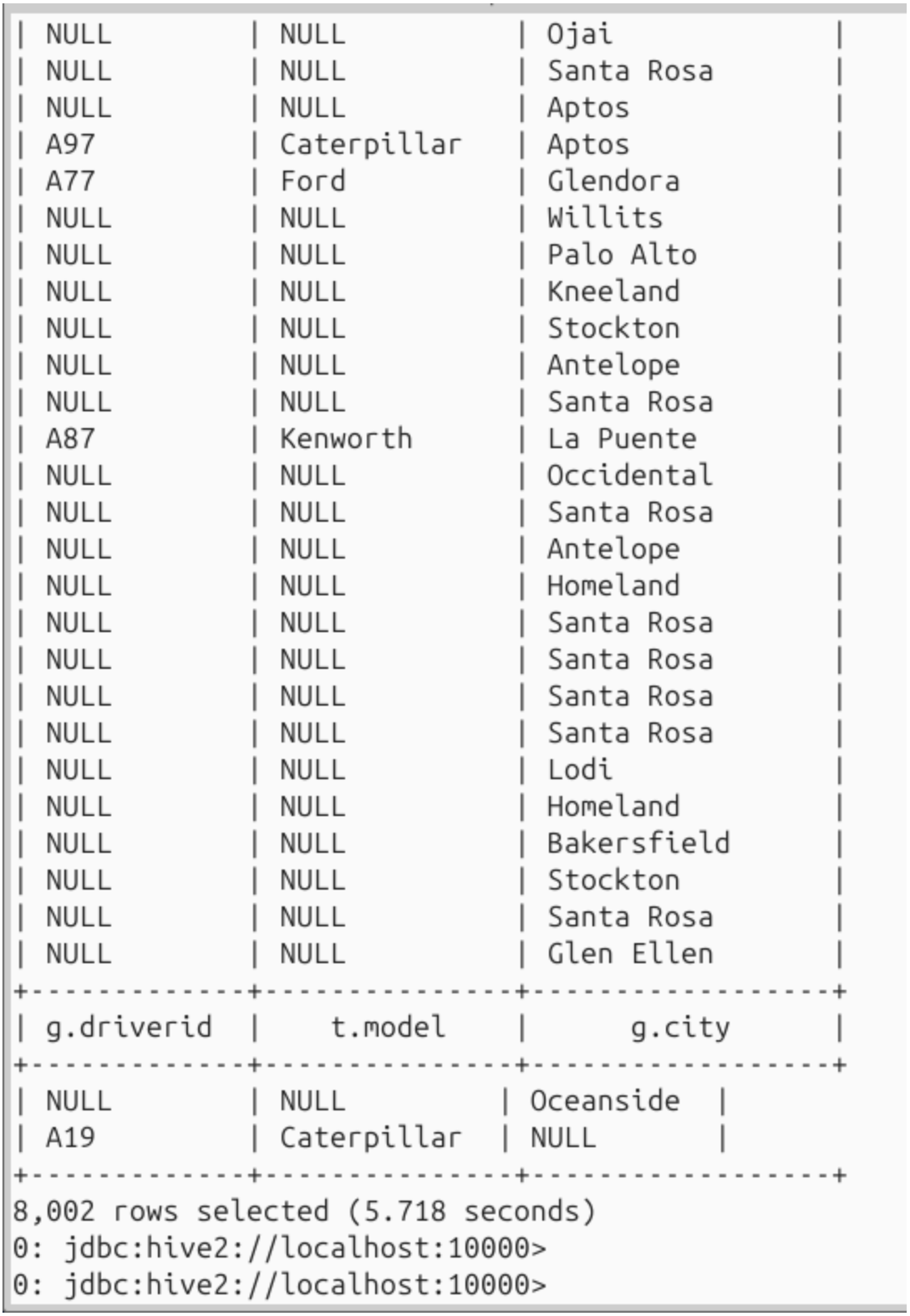
o select geos.driverid,trucks.model,geos.city from geos join trucks where geos.driverid=trucks.driverid;

A51	Crane	Modesto
A15	Ford	Apple Valley
A10	Peterbilt	Palo Alto
A27	Peterbilt	San Pablo
A71	Peterbilt	Roseville
A77	Ford	San Francisco
A84	Freightliner	Apple Valley
A5	Hino	Willits
A25	Volvo	San Fernando
A97	Caterpillar	Lodi
A4	Kenworth	Santa Rosa
A24	Navistar	Arbuckle
A97	Caterpillar	San Diego
A33	Caterpillar	Goleta
A38	Hino	Aptos
A12	Caterpillar	Antelope
A100	Peterbilt	Apple Valley
A57	Freightliner	Redding
A86	Ford	Arbuckle
A77	Ford	Glendora
A11	Peterbilt	La Quinta
A49	Caterpillar	San Francisco
A99	Peterbilt	Apple Valley
A38	Hino	Willits
A27	Peterbilt	Willits
A90	Ford	Arbuckle
A84	Freightliner	La Quinta
A97	Caterpillar	Aptos
A77	Ford	Glendora
A87	Kenworth	La Puente
A19	Caterpillar	NULL

459 rows selected (5.492 seconds)
0: jdbc:hive2://localhost:10000>
0: jdbc:hive2://localhost:10000>

Returns only drivers present in both geos and trucks (inner join).

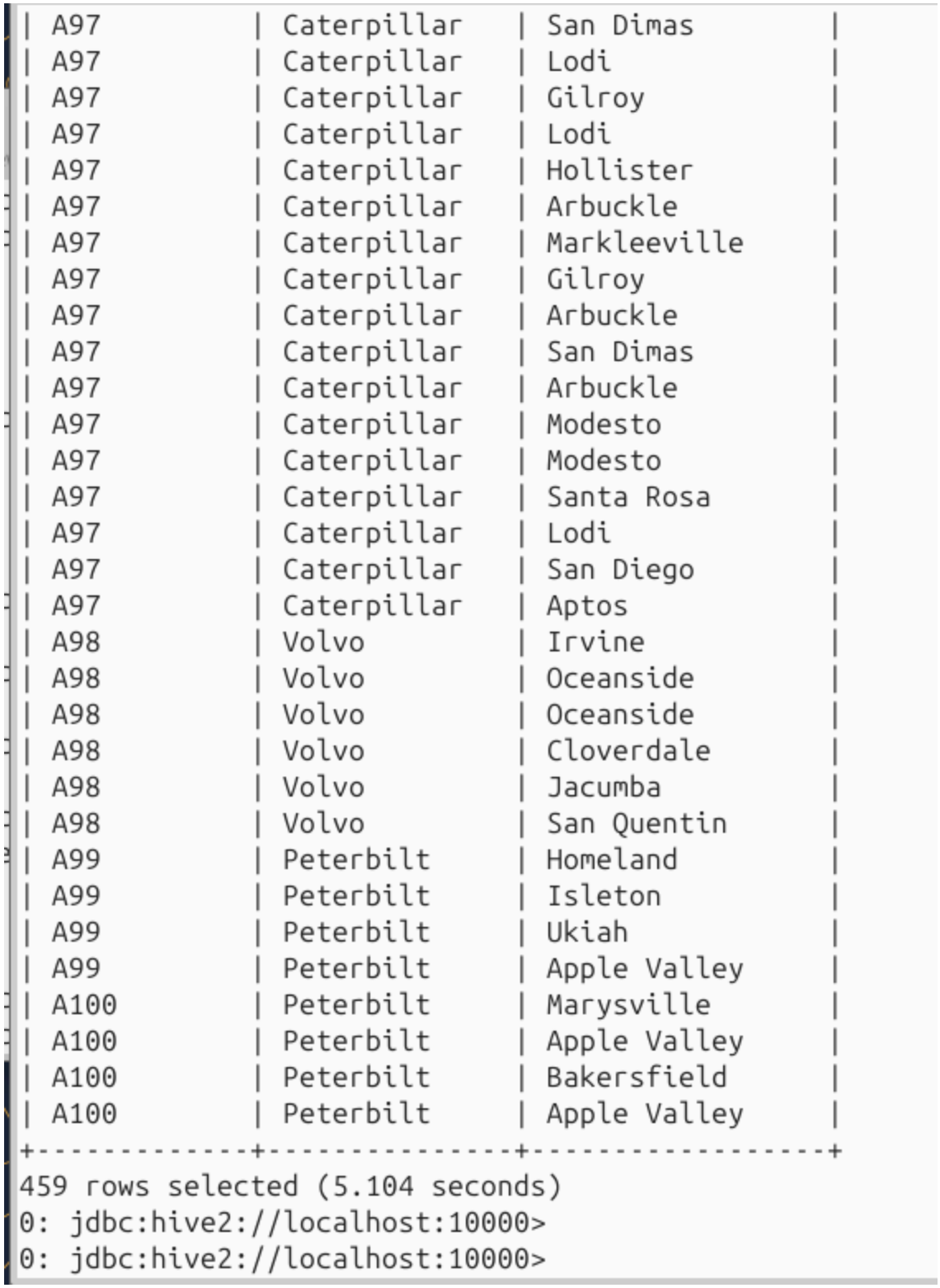
o select geos.driverid,trucks.model,geos.city from geos left outer join trucks where geos.driverid=trucks.driverid;



```
| NULL         | NULL           | Ojai              |
| NULL         | NULL           | Santa Rosa        |
| NULL         | NULL           | Aptos             |
| A97          | Caterpillar    | Aptos             |
| A77          | Ford           | Glendora          |
| NULL         | NULL           | Willits           |
| NULL         | NULL           | Palo Alto         |
| NULL         | NULL           | Kneeland          |
| NULL         | NULL           | Stockton          |
| NULL         | NULL           | Antelope          |
| NULL         | NULL           | Santa Rosa        |
| A87          | Kenworth       | La Puente         |
| NULL         | NULL           | Occidental        |
| NULL         | NULL           | Santa Rosa        |
| NULL         | NULL           | Antelope          |
| NULL         | NULL           | Homeland          |
| NULL         | NULL           | Santa Rosa        |
| NULL         | NULL           | Santa Rosa        |
| NULL         | NULL           | Santa Rosa        |
| NULL         | NULL           | Santa Rosa        |
| NULL         | NULL           | Lodi              |
| NULL         | NULL           | Homeland          |
| NULL         | NULL           | Bakersfield       |
| NULL         | NULL           | Stockton          |
| NULL         | NULL           | Santa Rosa        |
| NULL         | NULL           | Glen Ellen        |
+--------------+----------------+-------------------+
| g.driverid   |    t.model     |      g.city       |
+--------------+----------------+-------------------+
| NULL         | NULL           | Oceanside  |
| A19          | Caterpillar    | NULL       |
+--------------+----------------+-------------------+
8,002 rows selected (5.718 seconds)
0: jdbc:hive2://localhost:10000>
0: jdbc:hive2://localhost:10000>
```

Returns all geos rows and matches model when a trucks row exists (left keeps all left).

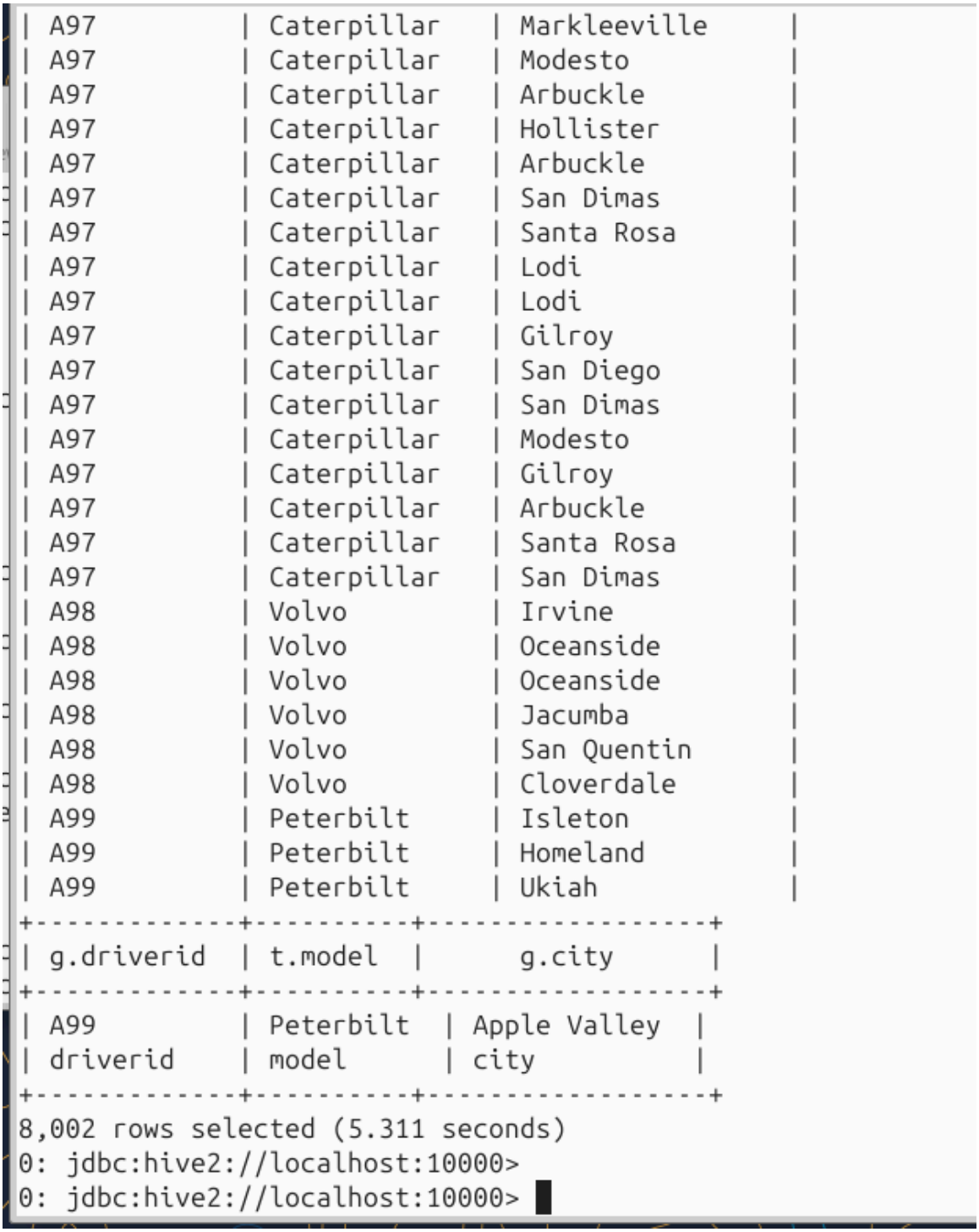
o select geos.driverid,trucks.model, geos.city from geos right outer join trucks where geos.driverid=trucks.driverid;



```
| A97          | Caterpillar    | San Dimas          |
| A97          | Caterpillar    | Lodi               |
| A97          | Caterpillar    | Gilroy             |
| A97          | Caterpillar    | Lodi               |
| A97          | Caterpillar    | Hollister          |
| A97          | Caterpillar    | Arbuckle           |
| A97          | Caterpillar    | Markleeville       |
| A97          | Caterpillar    | Gilroy             |
| A97          | Caterpillar    | Arbuckle           |
| A97          | Caterpillar    | San Dimas          |
| A97          | Caterpillar    | Arbuckle           |
| A97          | Caterpillar    | Modesto            |
| A97          | Caterpillar    | Modesto            |
| A97          | Caterpillar    | Santa Rosa         |
| A97          | Caterpillar    | Lodi               |
| A97          | Caterpillar    | San Diego          |
| A97          | Caterpillar    | Aptos              |
| A98          | Volvo          | Irvine             |
| A98          | Volvo          | Oceanside          |
| A98          | Volvo          | Oceanside          |
| A98          | Volvo          | Cloverdale         |
| A98          | Volvo          | Jacumba            |
| A98          | Volvo          | San Quentin        |
| A99          | Peterbilt      | Homeland           |
| A99          | Peterbilt      | Isleton            |
| A99          | Peterbilt      | Ukiah              |
| A99          | Peterbilt      | Apple Valley       |
| A100         | Peterbilt      | Marysville         |
| A100         | Peterbilt      | Apple Valley       |
| A100         | Peterbilt      | Bakersfield        |
| A100         | Peterbilt      | Apple Valley       |
+--------------+----------------+--------------------+
459 rows selected (5.104 seconds)
0: jdbc:hive2://localhost:10000>
0: jdbc:hive2://localhost:10000>
```

Returns all trucks rows and merges geos info when present (right keeps all right).

o select geos.driverid,trucks.model, geos.city from geos full outer join trucks where geos.driverid=trucks.driverid;



```
| A97          | Caterpillar   | Markleeville     |
| A97          | Caterpillar   | Modesto          |
| A97          | Caterpillar   | Arbuckle         |
| A97          | Caterpillar   | Hollister        |
| A97          | Caterpillar   | Arbuckle         |
| A97          | Caterpillar   | San Dimas        |
| A97          | Caterpillar   | Santa Rosa       |
| A97          | Caterpillar   | Lodi             |
| A97          | Caterpillar   | Lodi             |
| A97          | Caterpillar   | Gilroy           |
| A97          | Caterpillar   | San Diego        |
| A97          | Caterpillar   | San Dimas        |
| A97          | Caterpillar   | Modesto          |
| A97          | Caterpillar   | Gilroy           |
| A97          | Caterpillar   | Arbuckle         |
| A97          | Caterpillar   | Santa Rosa       |
| A97          | Caterpillar   | San Dimas        |
| A98          | Volvo         | Irvine           |
| A98          | Volvo         | Oceanside        |
| A98          | Volvo         | Oceanside        |
| A98          | Volvo         | Jacumba          |
| A98          | Volvo         | San Quentin      |
| A98          | Volvo         | Cloverdale       |
| A99          | Peterbilt     | Isleton          |
| A99          | Peterbilt     | Homeland         |
| A99          | Peterbilt     | Ukiah            |
+--------------+---------+------------------+
| g.driverid   | t.model  |      g.city      |
+--------------+---------+------------------+
| A99          | Peterbilt  | Apple Valley  |
| driverid     | model      | city          |
+--------------+---------+------------------+
8,002 rows selected (5.311 seconds)
0: jdbc:hive2://localhost:10000>
0: jdbc:hive2://localhost:10000>
```

Returns rows from either table even when the other side is missing (complete union of both).

• [ToDo: Write queries for the following] :

o Display truckid, driverid and model for every abnormal event

-- Display truckid, driverid and model for every abnormal event

SELECT g.truckid, g.driverid, t.model

```
FROM geos g
JOIN trucks t
  ON t.driverid = g.driverid
WHERE g.event_ind = 1;   -- abnormal = flagged rows (works even if event text varies)
```

```
| A38         | A38         | Hino           |
| A51         | A51         | Crane          |
| A15         | A15         | Ford           |
| A10         | A10         | Peterbilt      |
| A27         | A27         | Peterbilt      |
| A71         | A71         | Peterbilt      |
| A77         | A77         | Ford           |
| A84         | A84         | Freightliner   |
| A5          | A5          | Hino           |
| A25         | A25         | Volvo          |
| A97         | A97         | Caterpillar    |
| A4          | A4          | Kenworth       |
| A24         | A24         | Navistar       |
| A97         | A97         | Caterpillar    |
| A33         | A33         | Caterpillar    |
| A38         | A38         | Hino           |
| A12         | A12         | Caterpillar    |
| A100        | A100        | Peterbilt      |
| A57         | A57         | Freightliner   |
| A86         | A86         | Ford           |
| A77         | A77         | Ford           |
| A11         | A11         | Peterbilt      |
| A49         | A49         | Caterpillar    |
| A99         | A99         | Peterbilt      |
| A38         | A38         | Hino           |
| A27         | A27         | Peterbilt      |
| A90         | A90         | Ford           |
| A84         | A84         | Freightliner   |
| A97         | A97         | Caterpillar    |
| A77         | A77         | Ford           |
| A87         | A87         | Kenworth       |
+-------------+-------------+----------------+
451 rows selected (5.692 seconds)
0: jdbc:hive2://localhost:10000>
0: jdbc:hive2://localhost:10000>
```

Shows truck/model only for rows that are flagged abnormal via event_ind=1.

o Display truckid, driverid, model and city for velocity > 25.

```
-- Display truckid, driverid, model and city for velocity > 25
SELECT g.truckid, g.driverid, t.model, g.city
```

```
FROM geos g
JOIN trucks t
  ON t.driverid = g.driverid
WHERE g.velocity > 25;
```

A35	A35	Ford	Arbuckle
A44	A44	Peterbilt	Antelope
A36	A36	Ford	Palmdale
A34	A34	Freightliner	Redding
A14	A14	Crane	Kneeland
A34	A34	Freightliner	Palo Cedro
A8	A8	Navistar	San Pablo
A5	A5	Hino	Gilroy
A51	A51	Crane	Modesto
A15	A15	Ford	Apple Valley
A27	A27	Peterbilt	San Pablo
A71	A71	Peterbilt	Roseville
A77	A77	Ford	San Francisco
A5	A5	Hino	Willits
A97	A97	Caterpillar	Lodi
A4	A4	Kenworth	Santa Rosa
A97	A97	Caterpillar	San Diego
A33	A33	Caterpillar	Goleta
A38	A38	Hino	Aptos
A100	A100	Peterbilt	Apple Valley
A86	A86	Ford	Arbuckle
A77	A77	Ford	Glendora
A11	A11	Peterbilt	La Quinta
A49	A49	Caterpillar	San Francisco
A99	A99	Peterbilt	Apple Valley
A38	A38	Hino	Willits
A27	A27	Peterbilt	Willits
A90	A90	Ford	Arbuckle
A84	A84	Freightliner	La Quinta
A97	A97	Caterpillar	Aptos
A77	A77	Ford	Glendora

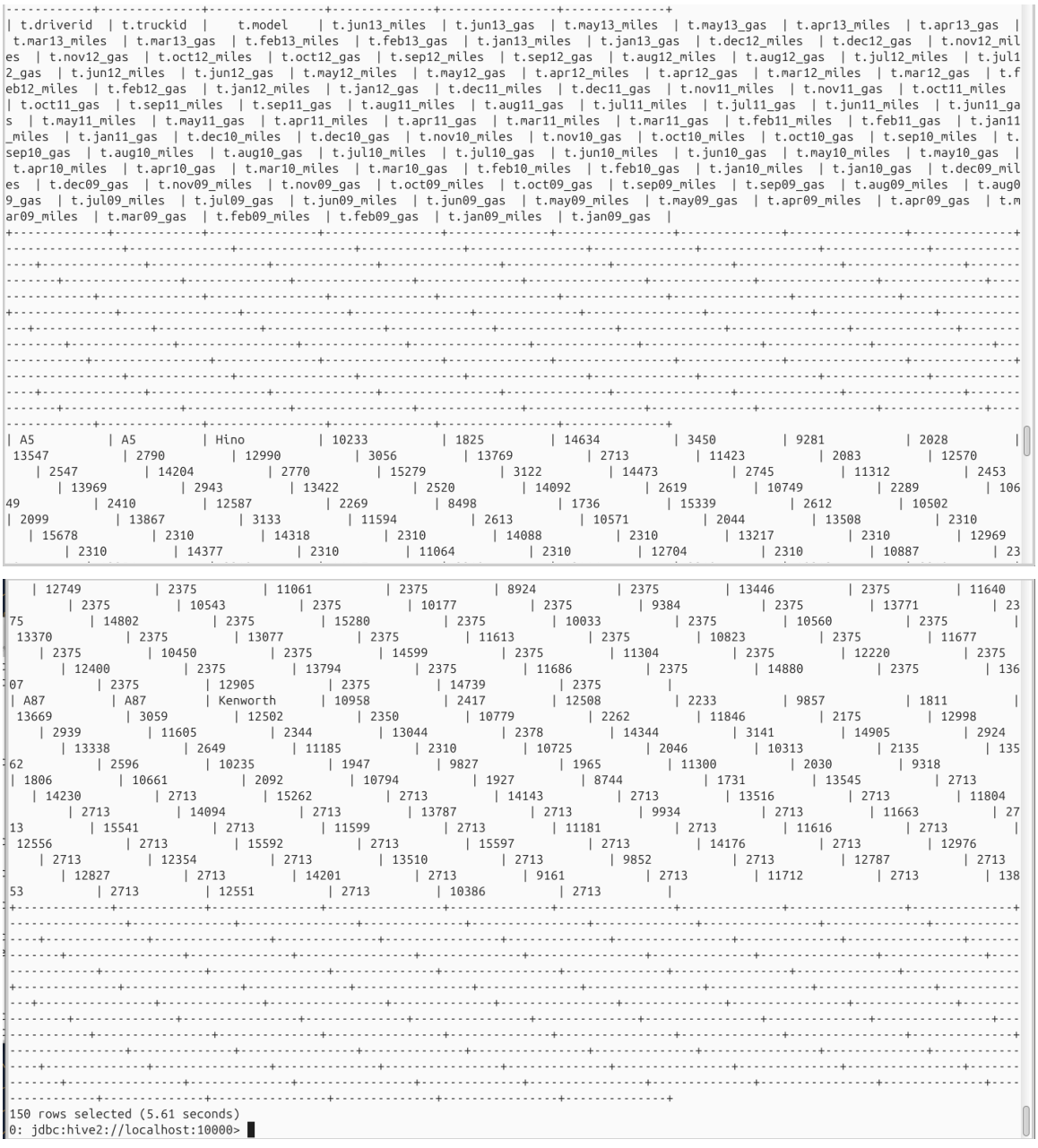
```
334 rows selected (5.485 seconds)
0: jdbc:hive2://localhost:10000>
0: jdbc:hive2://localhost:10000>
```

Lists faster-moving trucks with their model and the city where that reading occurred.

o Display complete record of the trucks for events where they were on unsafe following distance.

```
-- Display complete record of the trucks for "unsafe following distance" events
-- (handle both exact text and variants)
```

```
SELECT t.*
FROM geos g
JOIN trucks t
  ON t.driverid = g.driverid
WHERE LOWER(g.event) = 'unsafe_follow_distance'
  OR LOWER(g.event) LIKE '%unsafe%follow%distance%';
```



Pulls the full trucks row for any geos record whose event indicates unsafe following distance (handles naming variations).

Hive Transaction Manager:

One of the important properties that you need to know is hive.txn.manager which is used to set Hive Transaction manager, by default hive uses DummyTxnManager, to enable ACID, we need to set it to DbTxnManager.

SET hive.support.concurrency=true;

```
0: jdbc:hive2://localhost:10000> SET hive.txn.manager;
+-----------------------------------------------------+
|                         set                         |
+-----------------------------------------------------+
| hive.txn.manager=org.apache.hadoop.hive.ql.lockmgr.DummyTxnManager |
+-----------------------------------------------------+
1 row selected (0.008 seconds)
0: jdbc:hive2://localhost:10000>
```

SET hive.txn.manager=org.apache.hadoop.hive.ql.lockmgr.DbTxnManager;

The following are not required if you are using Hive 2.0
SET hive.enforce.bucketing=true;
SET hive.exec.dynamic.partition.mode=nostrict;

The following parameters are required for standalone hive metastore:
SET hive.compactor.initiator.on=true;
SET hive.compactor.worker.threads=1

[ToDo: execute several hive commands from truck data and show the difference in performance between ACID and non -ACID transaction managers]
Below are some of the limitations of using Hive ACID transactions:

- To support ACID, Hive tables should be created with TRANSACTIONAL table property. Currently, Hive supports ACID transactions on tables that store ORC file format.
- Enable ACID support by setting transaction manager to DbTxnManager
- Transaction tables cannot be accessed from the non -ACID Transaction Manager (DummyTxnManager) session.
- External tables cannot be created to support ACID since the changes on external tables are beyond Hive control.
- LOAD is not supported on ACID transactional Tables. hence use INSERT INTO.
- On Transactional session, all operations are auto commit as BEGIN, COMMIT and ROLLBACK are not yet supported.

NEW HIVE, create database as it doesnt exist

```
0: jdbc:hive2://localhost:10000> SHOW DATABASES;
+----------------+
| database_name  |
+----------------+
| default        |
+----------------+
1 row selected (0.908 seconds)
0: jdbc:hive2://localhost:10000> CREATE DATABASE IF NOT EXISTS student_db;
No rows affected (0.14 seconds)
0: jdbc:hive2://localhost:10000> USE student_db;
No rows affected (0.029 seconds)
```

Created table as it doesnt exist

```
0: jdbc:hive2://localhost:10000> SHOW TABLES;
+-----------+
| tab_name  |
+-----------+
+-----------+
No rows selected (0.172 seconds)
0: jdbc:hive2://localhost:10000> DROP TABLE IF EXISTS demo_txn;
No rows affected (0.069 seconds)
0: jdbc:hive2://localhost:10000> CREATE TABLE demo_txn (
. . . . . .>   id INT,
. . . . . .>   name STRING
. . . . . .> )
. . . . . .> CLUSTERED BY (id) INTO 2 BUCKETS
. . . . . .> STORED AS ORC
. . . . . .> TBLPROPERTIES ('transactional'='true');
No rows affected (0.506 seconds)
0: jdbc:hive2://localhost:10000>
```

Inserted data

```
0: jdbc:hive2://localhost:10000> INSERT INTO demo_txn VALUES (1, 'Ali'), (2, 'Sara');
WARNING: 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. spark, tez) or using Hive 1.X re
leases.
No rows affected (5.199 seconds)
0: jdbc:hive2://localhost:10000>
```

Updated data (it worked!)

```
0: jdbc:hive2://localhost:10000> UPDATE demo_txn SET name = 'Zahrah' WHERE id = 2;
WARNING: 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. spark, tez) or using Hive 1.X re
leases.
No rows affected (2.833 seconds)
0: jdbc:hive2://localhost:10000> SELECT * FROM demo_txn;
+--------------+----------------+
| demo_txn.id  | demo_txn.name  |
+--------------+----------------+
| 2            | Zahrah         |
| 1            | Ali            |
+--------------+----------------+
2 rows selected (0.237 seconds)
0: jdbc:hive2://localhost:10000>
```

Deleted data (it worked!)

```
0: jdbc:hive2://localhost:10000> DELETE FROM demo_txn WHERE id = 1;
WARNING: 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. spark, tez) or using Hive 1.X re
leases.
No rows affected (1.585 seconds)
0: jdbc:hive2://localhost:10000>
0: jdbc:hive2://localhost:10000> SELECT * FROM demo_txn;
+--------------+----------------+
| demo_txn.id  | demo_txn.name  |
+--------------+----------------+
| 2            | Zahrah         |
+--------------+----------------+
1 row selected (0.218 seconds)
0: jdbc:hive2://localhost:10000>
```

Loaded geos data

```
0: jdbc:hive2://localhost:10000> CREATE TABLE IF NOT EXISTS geos
. . . . . .> (
. . . . . .> truckid string,
. . . . . .> driverid string,
. . . . . .> event string,
. . . . . .> latitude decimal(5,0),
. . . . . .> longitude decimal(5,0),
. . . . . .> city string,
. . . . . .> state string,
. . . . . .> velocity int,
. . . . . .> event_ind int,
. . . . . .> idling_ind int
. . . . . .> )
. . . . . .> COMMENT 'Geo Table' ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';
No rows affected (0.068 seconds)
0: jdbc:hive2://localhost:10000> LOAD DATA INPATH '/user/data/geolocation/geolocation.csv' INTO TABLE geos;
Error: Error while compiling statement: FAILED: SemanticException Line 1:17 Invalid path ''/user/data/geolocation/geolocation.csv'': No files matching path hdfs://namenode:802
0/user/data/geolocation/geolocation.csv (state=42000,code=40000)
0: jdbc:hive2://localhost:10000> LOAD DATA INPATH '/user/data/geolocation.csv' INTO TABLE geos;
No rows affected (0.329 seconds)
```

```
0: jdbc:hive2://localhost:10000> ALTER TABLE geos CHANGE COLUMN latitude  latitude  DECIMAL(10,6);
No rows affected (0.126 seconds)
0: jdbc:hive2://localhost:10000> ALTER TABLE geos CHANGE COLUMN longitude longitude DECIMAL(10,6);
No rows affected (0.08 seconds)
0: jdbc:hive2://localhost:10000> ALTER TABLE geos SET TBLPROPERTIES ('skip.header.line.count'='1');
No rows affected (0.077 seconds)
```

```
0: jdbc:hive2://localhost:10000> SELECT * FROM geos LIMIT 5;
+--------------+---------------+-------------+---------------+-----------------+-------------+-------------+---------------+-----------------+-----------------+
| geos.truckid | geos.driverid | geos.event  | geos.latitude | geos.longitude  |  geos.city  | geos.state  | geos.velocity | geos.event_ind  | geos.idling_ind |
+--------------+---------------+-------------+---------------+-----------------+-------------+-------------+---------------+-----------------+-----------------+
| A54          | NULL          | normal      | 38.440467     | -122.714431     | Santa Rosa  | California  | 17            | 0               | 0               |
| A20          | NULL          | normal      | 36.977173     | -121.899402     | Aptos       | California  | 27            | 0               | 0               |
| A40          | A40           | overspeed   | 37.957702     | -121.290780     | Stockton    | California  | 77            | 1               | 0               |
| A31          | NULL          | normal      | 39.409608     | -123.355566     | Willits     | California  | 22            | 0               | 0               |
| A71          | NULL          | normal      | 33.683947     | -117.794694     | Irvine      | California  | 43            | 0               | 0               |
+--------------+---------------+-------------+---------------+-----------------+-------------+-------------+---------------+-----------------+-----------------+
5 rows selected (0.188 seconds)
0: jdbc:hive2://localhost:10000>
```

Loaded trucks data

```
root@63fd52321451:/opt# hadoop fs -mv /user/data/trucks.csv /user/data/trucks/truc
ks.csv
root@63fd52321451:/opt# hadoop fs -ls /user/data
Found 1 items
drwxr-xr-x   - root supergroup          0 2025-11-08 20:19 /user/data/trucks
root@63fd52321451:/opt# hadoop fs -put -f /user/data/trucks.csv /user/data/trucks/
trucks.csv
put: `/user/data/trucks.csv': No such file or directory

root@63fd52321451:/opt#
root@63fd52321451:/opt# hadoop fs -ls /user/data/trucks

Found 1 items
-rw-r--r--   3 root supergroup      61378 2025-11-07 05:57 /user/data/trucks/truck
s.csv
root@63fd52321451:/opt#
root@63fd52321451:/opt#
```

```
. . . . . .>   apr09_miles INT, apr09_gas INT,
. . . . . .>   mar09_miles INT, mar09_gas INT,
. . . . . .>   feb09_miles INT, feb09_gas INT,
. . . . . .>   jan09_miles INT, jan09_gas INT
. . . . . .> )
. . . . . .> ROW FORMAT DELIMITED
. . . . . .> FIELDS TERMINATED BY ','
. . . . . .> STORED AS TEXTFILE
. . . . . .> LOCATION '/user/data/trucks'
. . . . . .> TBLPROPERTIES ('skip.header.line.count'='1');
No rows affected (0.091 seconds)
0: jdbc:hive2://localhost:10000>
```

Make geos table transactional

```
0: jdbc:hive2://localhost:10000> SET hive.support.concurrency=true;
No rows affected (0.022 seconds)
0: jdbc:hive2://localhost:10000> SET hive.txn.manager=org.apache.hadoop.hive.ql.lockmgr.DbTxnManager;
No rows affected (0.011 seconds)
0: jdbc:hive2://localhost:10000> SET hive.compactor.initiator.on=true;
No rows affected (0.005 seconds)
0: jdbc:hive2://localhost:10000> SET hive.compactor.worker.threads=1;
No rows affected (0.004 seconds)
0: jdbc:hive2://localhost:10000> SET hive.support.concurrency=true;
No rows affected (0.005 seconds)
0: jdbc:hive2://localhost:10000> SET hive.txn.manager=org.apache.hadoop.hive.ql.lockmgr.DbTxnManager;
No rows affected (0.004 seconds)
0: jdbc:hive2://localhost:10000> SET hive.compactor.initiator.on=true;
No rows affected (0.005 seconds)
0: jdbc:hive2://localhost:10000> SET hive.compactor.worker.threads=1;
No rows affected (0.009 seconds)
0: jdbc:hive2://localhost:10000> SET hive.txn.manager;
+-----------------------------------------------------+
|                         set                         |
+-----------------------------------------------------+
| hive.txn.manager=org.apache.hadoop.hive.ql.lockmgr.DbTxnManager |
+-----------------------------------------------------+
1 row selected (0.007 seconds)
0: jdbc:hive2://localhost:10000>
```

```
0: jdbc:hive2://localhost:10000> DROP TABLE IF EXISTS geos_txn;
No rows affected (0.025 seconds)
0: jdbc:hive2://localhost:10000> CREATE TABLE geos_txn (
. . . . . . . . . . . . . . . .>   truckid     STRING,
. . . . . . . . . . . . . . . .>   driverid    STRING,
. . . . . . . . . . . . . . . .>   event       STRING,
. . . . . . . . . . . . . . . .>   latitude    DECIMAL(10,6),
. . . . . . . . . . . . . . . .>   longitude   DECIMAL(10,6),
. . . . . . . . . . . . . . . .>   city        STRING,
. . . . . . . . . . . . . . . .>   state       STRING,
. . . . . . . . . . . . . . . .>   velocity    INT,
. . . . . . . . . . . . . . . .>   event_ind   INT,
. . . . . . . . . . . . . . . .>   idling_ind  INT
. . . . . . . . . . . . . . . .> )
. . . . . . . . . . . . . . . .> CLUSTERED BY (driverid) INTO 4 BUCKETS
. . . . . . . . . . . . . . . .> STORED AS ORC
. . . . . . . . . . . . . . . .> TBLPROPERTIES ('transactional'='true');
No rows affected (0.134 seconds)
0: jdbc:hive2://localhost:10000> INSERT INTO geos_txn
. . . . . . . . . . . . . . . .> SELECT truckid, driverid, event, latitude, longitude, city, state, velocity, event_ind, idling_ind
. . . . . . . . . . . . . . . .> FROM geos;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution eng
leases.
No rows affected (2.763 seconds)
0: jdbc:hive2://localhost:10000>
```

```
0: jdbc:hive2://localhost:10000> UPDATE geos_txn
. . . . . . . . . . . . . . . .>    SET driverid = NULL
. . . . . . . . . . . . . . . .>  WHERE event = 'normal';
WARNING: 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. spark, tez) or using Hive 1.X re
leases.
No rows affected (3.612 seconds)
0: jdbc:hive2://localhost:10000>
```

Regular HIVE SELECT TIME

```
0: jdbc:hive2://localhost:10000> SELECT COUNT(*) AS geos_rows FROM geos;
+------------+
| geos_rows  |
+------------+
| 8002       |
+------------+
1 row selected (0.111 seconds)
0: jdbc:hive2://localhost:10000>
```

UPDATED HIVE SELECT TIME

```
0: jdbc:hive2://localhost:10000> SELECT COUNT(*) FROM geos_txn;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available
leases.
+--------+
|  _c0   |
+--------+
| 16000  |
+--------+
1 row selected (1.477 seconds)
```

REGULAR HIVE EXPLAIN TIME

```
0: jdbc:hive2://localhost:10000> EXPLAIN SELECT driverid, city FROM geos WHERE velocity > 25;
+----------------------------------------------------+
|                      Explain                       |
+----------------------------------------------------+
| STAGE DEPENDENCIES:                                |
|   Stage-0 is a root stage                          |
|                                                    |
| STAGE PLANS:                                       |
|   Stage: Stage-0                                   |
|     Fetch Operator                                 |
|       limit: -1                                    |
|       Processor Tree:                              |
|         TableScan                                  |
|           alias: geos                              |
|           Statistics: Num rows: 8002 Data size: 504769 Basic stats: COMPLETE Column stats: NONE |
|           Filter Operator                          |
|             predicate: (velocity > 25) (type: boolean) |
|             Statistics: Num rows: 2667 Data size: 168235 Basic stats: COMPLETE Column stats: NONE |
|             Select Operator                        |
|               expressions: driverid (type: string), city (type: string) |
|               outputColumnNames: _col0, _col1      |
|               Statistics: Num rows: 2667 Data size: 168235 Basic stats: COMPLETE Column stats: NONE |
|               ListSink                             |
|                                                    |
+----------------------------------------------------+
20 rows selected (0.075 seconds)
0: jdbc:hive2://localhost:10000>
```

UPDATED HIVE EXPLAIN TIME

```
0: jdbc:hive2://localhost:10000> EXPLAIN SELECT driverid, city FROM geos WHERE velocity > 25;
+----------------------------------------------------+
|                      Explain                       |
+----------------------------------------------------+
| STAGE DEPENDENCIES:                                |
|   Stage-0 is a root stage                          |
|                                                    |
| STAGE PLANS:                                       |
|   Stage: Stage-0                                   |
|     Fetch Operator                                 |
|       limit: -1                                    |
|       Processor Tree:                              |
|         TableScan                                  |
|           alias: geos                              |
|           Statistics: Num rows: 5095 Data size: 1039448 Basic stats: COMPLETE Column stats: NONE |
|           Filter Operator                          |
|             predicate: (velocity > 25) (type: boolean) |
|             Statistics: Num rows: 1698 Data size: 346414 Basic stats: COMPLETE Column stats: NONE |
|             Select Operator                        |
|               expressions: driverid (type: string), city (type: string) |
|               outputColumnNames: _col0, _col1      |
|               Statistics: Num rows: 1698 Data size: 346414 Basic stats: COMPLETE Column stats: NONE |
|               ListSink                             |
|                                                    |
+----------------------------------------------------+
20 rows selected (0.207 seconds)
0: jdbc:hive2://localhost:10000>
```

REGULAR HIVE UPDATE FAILED

```
0: jdbc:hive2://localhost:10000> UPDATE geos SET driverid = NULL WHERE event = 'normal';
Error: Error while compiling statement: FAILED: SemanticException [Error 10294]: Attempt to do update or delete using transaction manager th
at does not support these operations. (state=42000,code=10294)
0: jdbc:hive2://localhost:10000> s
```

UPDATED HIVE UPDATE WORKED

```
0: jdbc:hive2://localhost:10000> UPDATE geos_txn
. . . . . .>    SET driverid = NULL
. . . . . .>  WHERE event = 'normal';
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available i
leases.
No rows affected (3.612 seconds)
```

Bonus Question (1%):

Experiment with Tez, Spark and LLAP execution engines and benchmark wrt MR.

TEZ

Setup tez

```
madeel@bdacourse:~/lab-6-hive/docker-hive$ docker exec -it docker-hive-hive-server
-1 bash
root@944c1a3483a8:/opt# cd /opt
root@944c1a3483a8:/opt# curl -LO https://archive.apache.org/dist/tez/0.9.1/apache-
tez-0.9.1-bin.tar.gz
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 58.2M  100 58.2M    0     0  2685k      0  0:00:22  0:00:22 --:--:-- 2415k
root@944c1a3483a8:/opt# tar -xzf apache-tez-0.9.1-bin.tar.gz
root@944c1a3483a8:/opt# mv apache-tez-0.9.1-bin tez
root@944c1a3483a8:/opt# hdfs dfs -mkdir -p /apps/tez
root@944c1a3483a8:/opt# hdfs dfs -put -f /opt/tez/* /apps/tez/
root@944c1a3483a8:/opt# hdfs dfs -chmod -R 755 /apps/tez
root@944c1a3483a8:/opt# exit
exit
madeel@bdacourse:~/lab-6-hive/docker-hive$
```

```
madeel@bdacourse:~/lab-6-hive/docker-hive$ docker compose exec hive-server bash
WARN[0000] /home/madeel/lab-6-hive/docker-hive/docker-compose.yml: the attribute `version` is obsolete, it will be ig
root@944c1a3483a8:/opt# /opt/hive/bin/beeline -u jdbc:hive2://localhost:10000
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hive/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hadoop-2.7.4/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Connecting to jdbc:hive2://localhost:10000
Connected to: Apache Hive (version 2.3.2)
Driver: Hive JDBC (version 2.3.2)
Transaction isolation: TRANSACTION_REPEATABLE_READ
Beeline version 2.3.2 by Apache Hive
0: jdbc:hive2://localhost:10000> SET hive.execution.engine;
+----------------------------+
|            set             |
+----------------------------+
| hive.execution.engine=tez  |
+----------------------------+
1 row selected (0.092 seconds)
0: jdbc:hive2://localhost:10000>
```

Created dataset

```
madeel@bdacourse:~/lab-6-hive/docker-hive$ docker exec -it docker-hive-hive-server
-1 bash
root@944c1a3483a8:/opt# L='T999999,A999999,normal,33.123456,-117.987654,Karachi,Si
ndh,45,1,0'
root@944c1a3483a8:/opt# yes "$L" | head -n 10000000 > /tmp/geos_500m.csv   # ~10M
lines
root@944c1a3483a8:/opt# ls -lh /tmp/geos_500m.csv
-rw-r--r-- 1 root root 630M Nov  9 15:53 /tmp/geos_500m.csv
root@944c1a3483a8:/opt# hdfs dfs -mkdir -p /user/data/bench
root@944c1a3483a8:/opt# hdfs dfs -put -f /tmp/geos_500m.csv /user/data/bench/geos_
500m.csv
root@944c1a3483a8:/opt# hdfs dfs -ls -h /user/data/bench
Found 1 items
-rw-r--r--   3 root supergroup    629.4 M 2025-11-09 15:53 /user/data/bench/geos_5
00m.csv
root@944c1a3483a8:/opt#
```

Created extrernal database on hive that loads this dataset

```
0: jdbc:hive2://localhost:10000> DROP TABLE IF EXISTS geos_bench_ext;
No rows affected (1.568 seconds)
0: jdbc:hive2://localhost:10000> CREATE EXTERNAL TABLE geos_bench_ext (
. . . . . .>   truckid     STRING,
. . . . . .>   driverid    STRING,
. . . . . .>   event       STRING,
. . . . . .>   latitude    DECIMAL(10,6),
. . . . . .>   longitude   DECIMAL(10,6),
. . . . . .>   city        STRING,
. . . . . .>   state       STRING,
. . . . . .>   velocity    INT,
. . . . . .>   event_ind   INT,
. . . . . .>   idling_ind  INT
. . . . . .> )
. . . . . .> ROW FORMAT DELIMITED
. . . . . .> FIELDS TERMINATED BY ','
. . . . . .> STORED AS TEXTFILE
. . . . . .> LOCATION '/user/data/bench';
No rows affected (0.098 seconds)
0: jdbc:hive2://localhost:10000>
```

Created transactional database to load

```
0: jdbc:hive2://localhost:10000> DROP TABLE IF EXISTS geos_bench_txn;
No rows affected (0.028 seconds)
0: jdbc:hive2://localhost:10000> CREATE TABLE geos_bench_txn (
. . . . . .>   truckid     STRING,
. . . . . .>   driverid    STRING,
. . . . . .>   event       STRING,
. . . . . .>   latitude    DECIMAL(10,6),
. . . . . .>   longitude   DECIMAL(10,6),
. . . . . .>   city        STRING,
. . . . . .>   state       STRING,
. . . . . .>   velocity    INT,
. . . . . .>   event_ind   INT,
. . . . . .>   idling_ind  INT
. . . . . .> )
. . . . . .> CLUSTERED BY (truckid) INTO 8 BUCKETS
 bucket on truckid so we can UPDATE driverid later
. . . . . .> STORED AS ORC
. . . . . .> TBLPROPERTIES ('transactional'='true');
No rows affected (0.205 seconds)
0: jdbc:hive2://localhost:10000>
```

Then set Tez

```
0: jdbc:hive2://localhost:10000> SET hive.execution.engine=Tez;
No rows affected (0.003 seconds)
0: jdbc:hive2://localhost:10000> SET hive.execution.engine;
+----------------------------+
|            set             |
+----------------------------+
| hive.execution.engine=Tez  |
+----------------------------+
1 row selected (0.011 seconds)
0: jdbc:hive2://localhost:10000> INSERT INTO geos_bench_txn
. . . . . . . . . . . . . . . .> SELECT * FROM geos_bench_ext;
```

Now loading the data into transactional database

```
0: jdbc:hive2://localhost:10000> INSERT INTO geos_bench_txn
. . . . . . . . . . . . . . . .> SELECT * FROM geos_bench_ext;
No rows affected (65.361 seconds)
0: jdbc:hive2://localhost:10000>
```

Completed.

TEZ SELECT TIME

```
0: jdbc:hive2://localhost:10000> SELECT COUNT(*) FROM geos_bench_txn;
+-----------+
|    _c0    |
+-----------+
| 10000000  |
+-----------+
1 row selected (5.411 seconds)
0: jdbc:hive2://localhost:10000>
```

TEZ EXPLAIN TIME

```
0: jdbc:hive2://localhost:10000> EXPLAIN
. . . . . .> SELECT driverid, city
. . . . . .> FROM geos_bench_txn
. . . . . .> WHERE velocity > 40;
+----------------------------------------------------+
|                      Explain                       |
+----------------------------------------------------+
| Plan optimized by CBO.                             |
|                                                    |
| Stage-0                                            |
|   Fetch Operator                                   |
|     limit:-1                                       |
|     Select Operator [SEL_2]                        |
|       Output:["_col0","_col1"]                     |
|       Filter Operator [FIL_4]                      |
|         predicate:(velocity > 40)                  |
|         TableScan [TS_0]                           |
|           Output:["driverid","city","velocity"]    |
|                                                    |
+----------------------------------------------------+
12 rows selected (0.169 seconds)
0: jdbc:hive2://localhost:10000>
```

TEZ UPDATE TIME

```
0: jdbc:hive2://localhost:10000> UPDATE geos_bench_txn
. . . . . .>    SET driverid = NULL
. . . . . .>  WHERE event = 'normal';
No rows affected (73.775 seconds)
0: jdbc:hive2://localhost:10000>
```

TEZ DELETE TIME

```
0: jdbc:hive2://localhost:10000> DELETE FROM geos_bench_txn
. . . . . .>  WHERE velocity = 45;
No rows affected (41.729 seconds)
0: jdbc:hive2://localhost:10000>
```

SPARK

Downloading spark from github repo

```
madeel@bdacourse:~/lab-6-hive$ git clone https://github.com/Marcel-Jan/docker-ha
doop-spark
Cloning into 'docker-hadoop-spark'...
remote: Enumerating objects: 644, done.
remote: Counting objects: 100% (241/241), done.
remote: Compressing objects: 100% (35/35), done.
remote: Total 644 (delta 215), reused 206 (delta 206), pack-reused 403 (from 2)
Receiving objects: 100% (644/644), 155.91 KiB | 4.45 MiB/s, done.
Resolving deltas: 100% (302/302), done.
madeel@bdacourse:~/lab-6-hive$ ls
docker-hadoop-spark  docker-hive
madeel@bdacourse:~/lab-6-hive$ cd docker-hadoop-spark/
madeel@bdacourse:~/lab-6-hive/docker-hadoop-spark$ docker compose up -d
WARN[0000] /home/madeel/lab-6-hive/docker-hadoop-spark/docker-compose.yml: the a
ttribute `version` is obsolete, it will be ignored, please remove it to avoid po
tential confusion
[+] Running 21/24
 ⠧ spark-worker-1 [⣿⣿⣿⣿⣿⣿⣿⣿⣿⣿⣿⣿] 282.5MB / 313.5MB Pulling                    120.6s
 ✔ historyserver Pulled                                                          8.9s
 ⠧ spark-master [⣿⣿] Pulling                                                    120.6s
```

```
[+] Running 15/15
 ✔ Network docker-hadoop-spark_default                Created                    0.1s
 ✔ Volume docker-hadoop-spark_hadoop_historyserver    Created                    0.0s
 ✔ Volume docker-hadoop-spark_hadoop_namenode         Created                    0.0s
 ✔ Volume docker-hadoop-spark_hadoop_datanode         Created                    0.0s
 ✔ Container nodemanager                              Started                    2.3s
 ✔ Container namenode                                 Started                    2.4s
 ✔ Container datanode                                 Started                    2.5s
 ✔ Container historyserver                            Started                    2.6s
 ✔ Container hive-metastore                           Started                    2.8s
 ✔ Container presto-coordinator                       Started                    2.6s
 ✔ Container hive-metastore-postgresql                Started                    2.9s
 ✔ Container resourcemanager                          Started                    2.8s
 ✔ Container spark-master                             Started                    1.5s
 ✔ Container hive-server                              Started                    1.3s
 ✔ Container spark-worker-1                           Started                    2.0s
madeel@bdacourse:~/lab-6-hive/docker-hadoop-spark$
```

Created data, created databases and then inserted data in the databases.

```
0: jdbc:hive2://localhost:10000> SET hive.txn.manager;
+----------------------------------------------------+
|                        set                         |
+----------------------------------------------------+
| hive.txn.manager=org.apache.hadoop.hive.ql.lockmgr.DbTxnManager |
+----------------------------------------------------+
1 row selected (0.013 seconds)
0: jdbc:hive2://localhost:10000> USE student_db;
No rows affected (0.199 seconds)
0: jdbc:hive2://localhost:10000> show tables;
+-----------------+
|    tab_name     |
+-----------------+
| geos_bench_ext  |
| geos_bench_txn  |
+-----------------+
2 rows selected (0.198 seconds)
0: jdbc:hive2://localhost:10000> INSERT INTO geos_bench_txn
. . . . . .> SELECT * FROM geos_bench_ext;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the futu
re versions. Consider using a different execution engine (i.e. spark, tez) or us
ing Hive 1.X releases.
No rows affected (71.936 seconds)
0: jdbc:hive2://localhost:10000>
```

SELECT TIME

```
0: jdbc:hive2://localhost:10000> SELECT COUNT(*) AS total_rows FROM geos_bench_t
xn;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the futu
re versions. Consider using a different execution engine (i.e. spark, tez) or us
ing Hive 1.X releases.
+-------------+
| total_rows  |
+-------------+
| 10000000    |
+-------------+
1 row selected (6.75 seconds)
0: jdbc:hive2://localhost:10000>
```

EXPLAIN TIME

```
|                                                     |
| STAGE PLANS:                                        |
|   Stage: Stage-0                                    |
|     Fetch Operator                                  |
|       limit: -1                                     |
|       Processor Tree:                               |
|         TableScan                                   |
|           alias: geos_bench_txn                     |
|           Statistics: Num rows: 3138 Data size: 640305 Basic stats: COMPLETE C
olumn stats: NONE |
|           Filter Operator                           |
|             predicate: (velocity > 40) (type: boolean) |
|             Statistics: Num rows: 1046 Data size: 213435 Basic stats: COMPLETE
 Column stats: NONE |
|             Select Operator                         |
|               expressions: driverid (type: string), city (type: string) |
|               outputColumnNames: _col0, _col1       |
|               Statistics: Num rows: 1046 Data size: 213435 Basic stats: COMPLE
TE Column stats: NONE |
|               ListSink                              |
|                                                     |
+-----------------------------------------------------+
20 rows selected (0.213 seconds)
0: jdbc:hive2://localhost:10000>
```

UPDATE

```
0: jdbc:hive2://localhost:10000> UPDATE geos_bench_txn
. . . . . .>    SET driverid = NULL
. . . . . .>  WHERE event = 'normal';
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the futu
re versions. Consider using a different execution engine (i.e. spark, tez) or us
ing Hive 1.X releases.
No rows affected (79.901 seconds)
0: jdbc:hive2://localhost:10000>
```

DELETE

```
0: jdbc:hive2://localhost:10000> DELETE FROM geos_bench_txn
. . . . . .>  WHERE velocity = 45;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the futu
re versions. Consider using a different execution engine (i.e. spark, tez) or us
ing Hive 1.X releases.
No rows affected (47.819 seconds)
0: jdbc:hive2://localhost:10000>
```

MR

Created dataset and tables. Now doing SELECT

```
0: jdbc:hive2://localhost:10000> SELECT COUNT(*) AS total_rows FROM geos_bench_t
xn;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the futu
re versions. Consider using a different execution engine (i.e. spark, tez) or us
ing Hive 1.X releases.
+-------------+
| total_rows  |
+-------------+
| 10000000    |
+-------------+
1 row selected (6.402 seconds)
0: jdbc:hive2://localhost:10000>
```

EXPLAIN

```
|                                                     |
| STAGE PLANS:                                        |
|   Stage: Stage-0                                    |
|     Fetch Operator                                  |
|       limit: -1                                     |
|       Processor Tree:                               |
|         TableScan                                   |
|           alias: geos_bench_txn                     |
|           Statistics: Num rows: 6501 Data size: 1326242 Basic stats: COMPLETE
Column stats: NONE |
|           Filter Operator                           |
|             predicate: (velocity > 40) (type: boolean) |
|             Statistics: Num rows: 2167 Data size: 442080 Basic stats: COMPLETE
 Column stats: NONE |
|             Select Operator                         |
|               expressions: driverid (type: string), city (type: string) |
|               outputColumnNames: _col0, _col1       |
|               Statistics: Num rows: 2167 Data size: 442080 Basic stats: COMPLE
TE Column stats: NONE |
|               ListSink                              |
|                                                     |
+-----------------------------------------------------+
20 rows selected (0.224 seconds)
0: jdbc:hive2://localhost:10000>
```

UPDATE

```
0: jdbc:hive2://localhost:10000> UPDATE geos_bench_txn
. . . . . .>    SET driverid = NULL
. . . . . .>  WHERE event = 'normal';
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the futu
re versions. Consider using a different execution engine (i.e. spark, tez) or us
ing Hive 1.X releases.
No rows affected (81.859 seconds)
0: jdbc:hive2://localhost:10000>
0: jdbc:hive2://localhost:10000>
```

DELETE

```
0: jdbc:hive2://localhost:10000> DELETE FROM geos_bench_txn
. . . . . .> WHERE velocity = 45;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the futu
re versions. Consider using a different execution engine (i.e. spark, tez) or us
ing Hive 1.X releases.
No rows affected (10.956 seconds)
0: jdbc:hive2://localhost:10000>
```

LLAP

Created dataset, database and was setting the values

```
0: jdbc:hive2://localhost:10000> SET hive.execution.engine=llap;
Error: Error while processing statement: 'SET hive.execution.engine=llap' FAILED
 in validation : Invalid value.. expects one of [mr, tez, spark]. (state=42000,c
ode=1)
0: jdbc:hive2://localhost:10000> SET hive.llap.enabled=true;
Error: Error while processing statement: hive configuration hive.llap.enabled do
es not exists. (state=42000,code=1)
0: jdbc:hive2://localhost:10000> SET hive.llap.daemon.num.executors=4;
No rows affected (0.004 seconds)
0: jdbc:hive2://localhost:10000> SET hive.llap.io.enabled=true;
No rows affected (0.005 seconds)
0: jdbc:hive2://localhost:10000> SET hive.llap.io.memory.mode=cache;
No rows affected (0.004 seconds)
```

It is not working as it requires hive version 3. We have hive version 2. Could not find any setup/compose file for hive version3 to support llap.

COMPARISION TABLE

seconds	MR	TEZ	SPARK
SELECT `SELECT COUNT(*) AS total_rows FROM geos_bench_txn;`	6.402	5.411	6.75
EXPLAIN `EXPLAIN SELECT driverid, city FROM geos_bench_txn WHERE velocity > 40;`	0.224	0.169	0.213
UPDATE `UPDATE geos_bench_txn SET driverid = NULL WHERE event = 'normal';`	81.859	73.775	79.901
DELETE `DELETE FROM geos_bench_txn WHERE velocity = 45;`	10.956	41.729	47.819

DATASET CREATION

```
# 2) Generate ~550–600 MB CSV quickly (10M identical rows; safe
>500MB)
L='T999999,A999999,normal,33.123456,-117.987654,Karachi,Sindh,45,
1,0'
yes "$L" | head -n 10000000 > /tmp/geos_500m.csv   # ~10M lines

# 3) Verify size (should be ~500–650MB depending on newline)
ls -lh /tmp/geos_500m.csv

# 4) Put into HDFS under a dedicated folder
hdfs dfs -mkdir -p /user/data/bench
hdfs dfs -put -f /tmp/geos_500m.csv
/user/data/bench/geos_500m.csv
hdfs dfs -ls -h /user/data/bench
```