

Comparative Analysis of Query Performance in Hadoop and Oracle DB



Overview

- Dataset
- Queries (Oracle SQL and Hive)
- Oracle Query Plans
- Hive Query
- Running time for Oracle and Hive
- Challenges
- Conclusion



Project Idea:

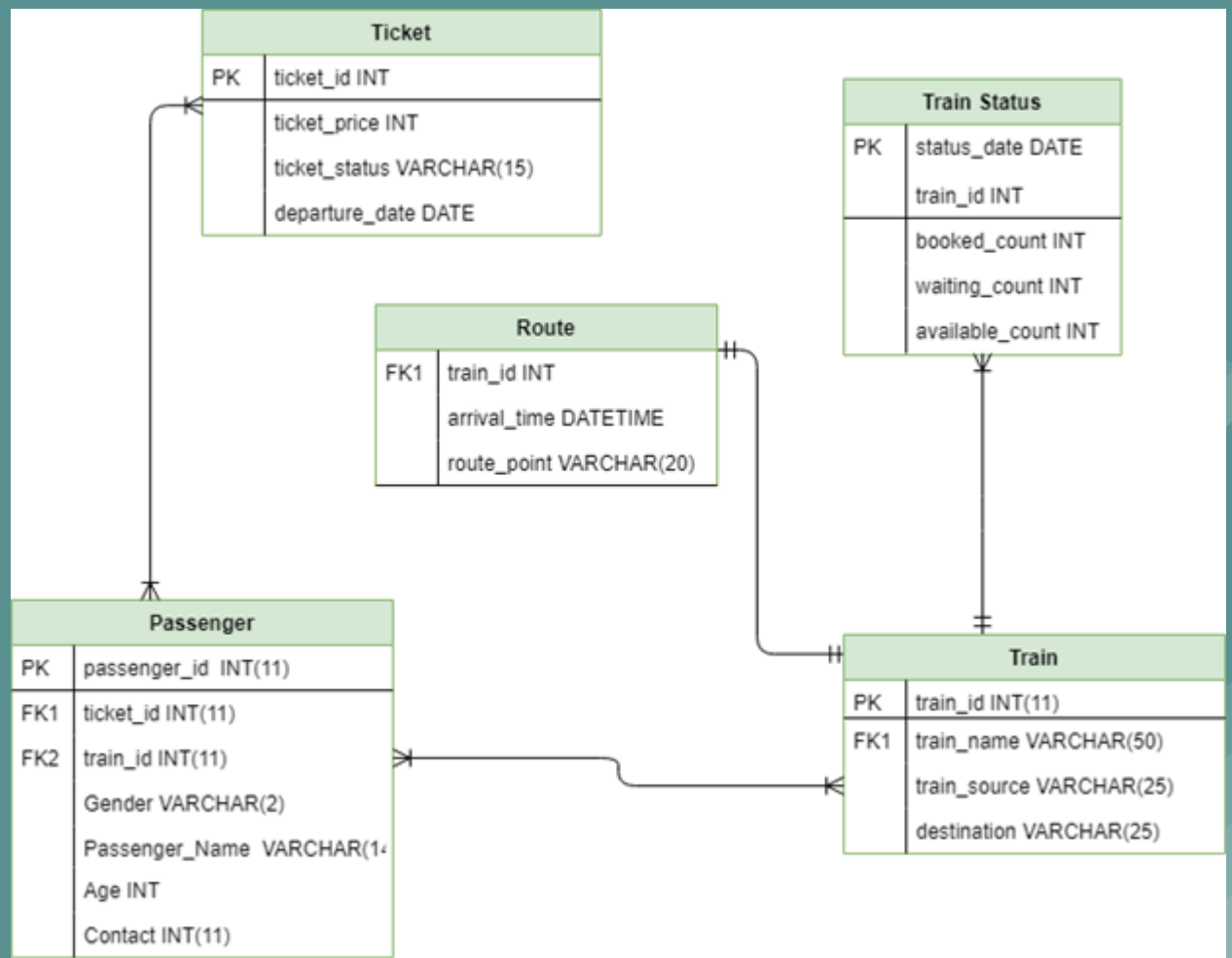
The project focuses on Train Management System which will allow the track of all train schedules, their route and passenger details. This will also allow the Train Administrator to manage, edit and add routes. Also, we can add new passenger and edit ticket details. This System simplifies the train management by maintaining all the records.

Implementation Details:

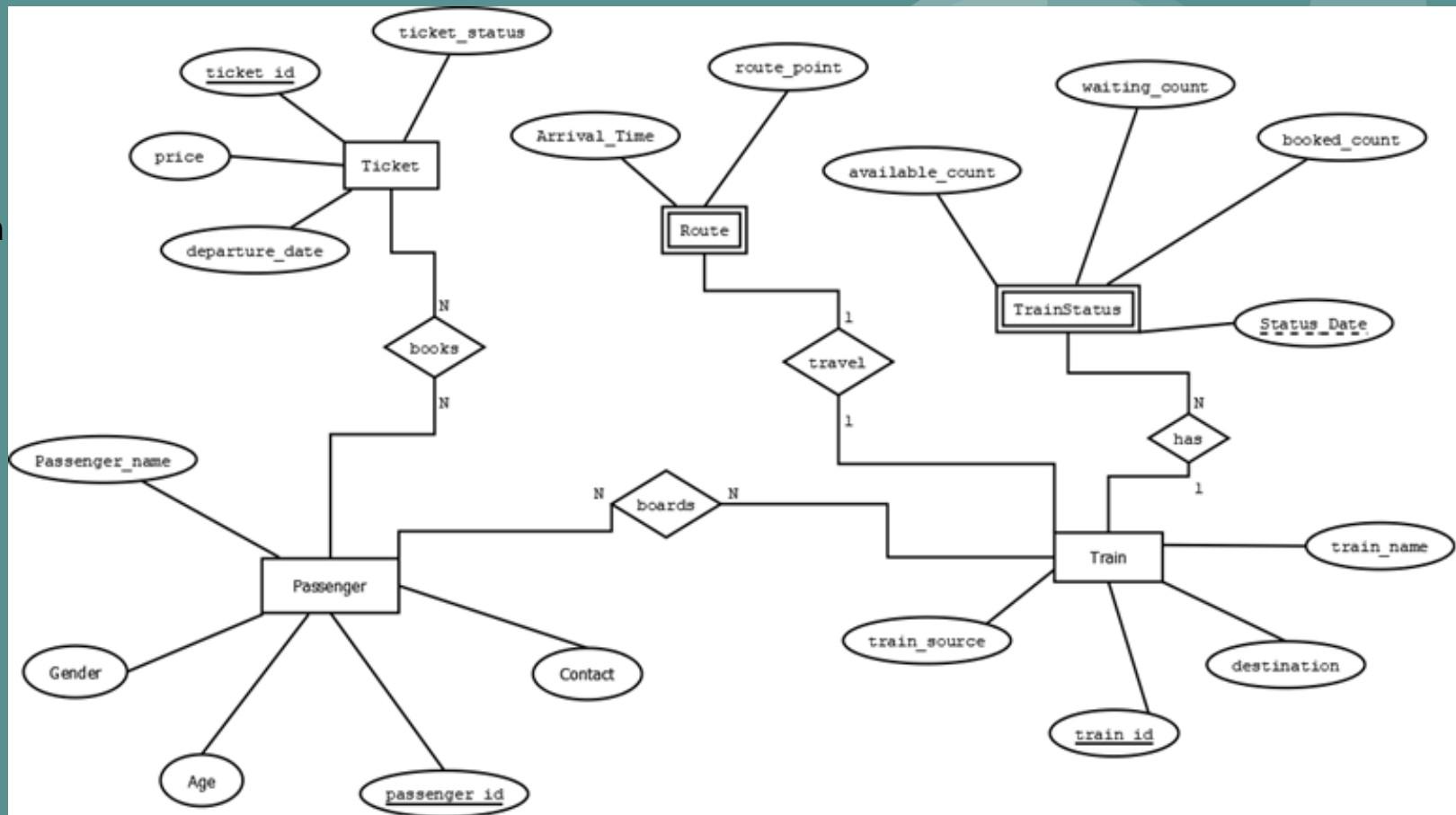
Hive/Hadoop section: We ran the queries on the hadoop cluster where data is stored using Hive QL on the design center account provided. Using this we noted the time taken for the queries to execute on hive.

Oracle DB & Explain plan: We installed Oracle 12c version on the local desktop and executed all the queries to see the performance difference. We also executed the explain plan which is a tool that you can use to have Oracle explain to you how it plans on executing your query. This is useful in tuning queries to the database to get them to perform better. Once you know how Oracle plans on executing your query, you can change your environment to run the query faster.

Schema



ER Diagram



Oracle Explain Plan

The EXPLAIN PLAN statement displays execution plans chosen by the Oracle optimizer for SELECT, UPDATE, INSERT, and DELETE statements.

The EXPLAIN PLAN helps you to understand the optimizer decisions, such as why the optimizer chose a nested loops join instead of a hash join, and lets you understand the performance of a query.

The optimizer performs the following steps:

1. The optimizer generates a set of potential plans for the SQL statement based on available access paths and hints.
2. The optimizer estimates the cost of each plan based on statistics in the data dictionary. Statistics include information on the data distribution and storage characteristics of the tables, indexes, and partitions accessed by the statement.
3. The optimizer compares the plans and chooses the plan with the lowest cost.

Queries

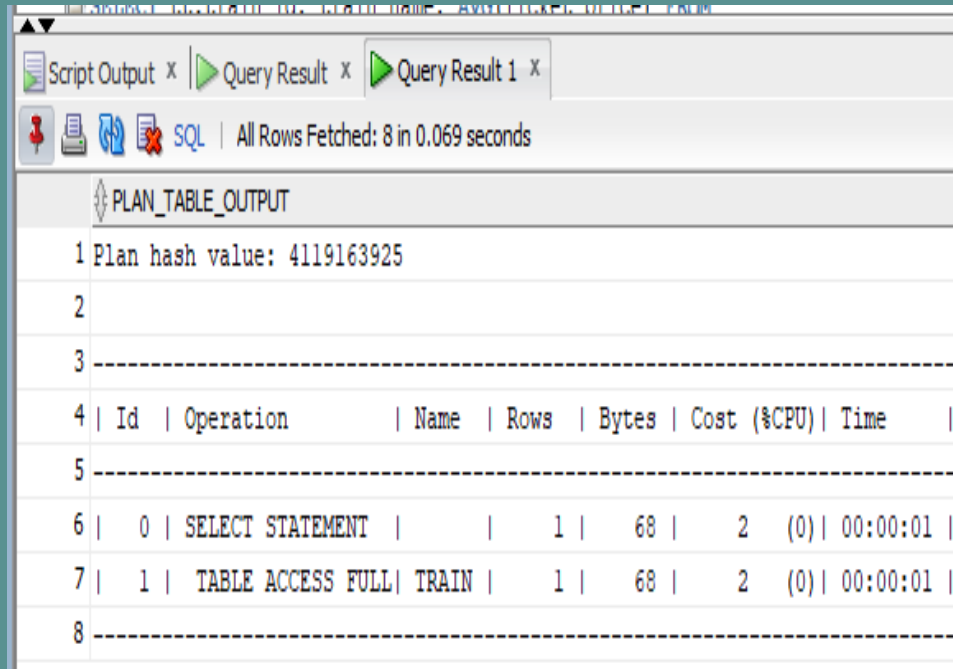


1. Display Details of Train Table

```
Select * from train;
```



Oracle Explain Plan



Script Output x Query Result x Query Result 1 x

SQL | All Rows Fetched: 8 in 0.069 seconds

PLAN_TABLE_OUTPUT

1 Plan hash value: 4119163925

2

3 -----

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	68	2 (0)	00:00:01
1	TABLE ACCESS FULL	TRAIN	1	68	2 (0)	00:00:01

8 -----

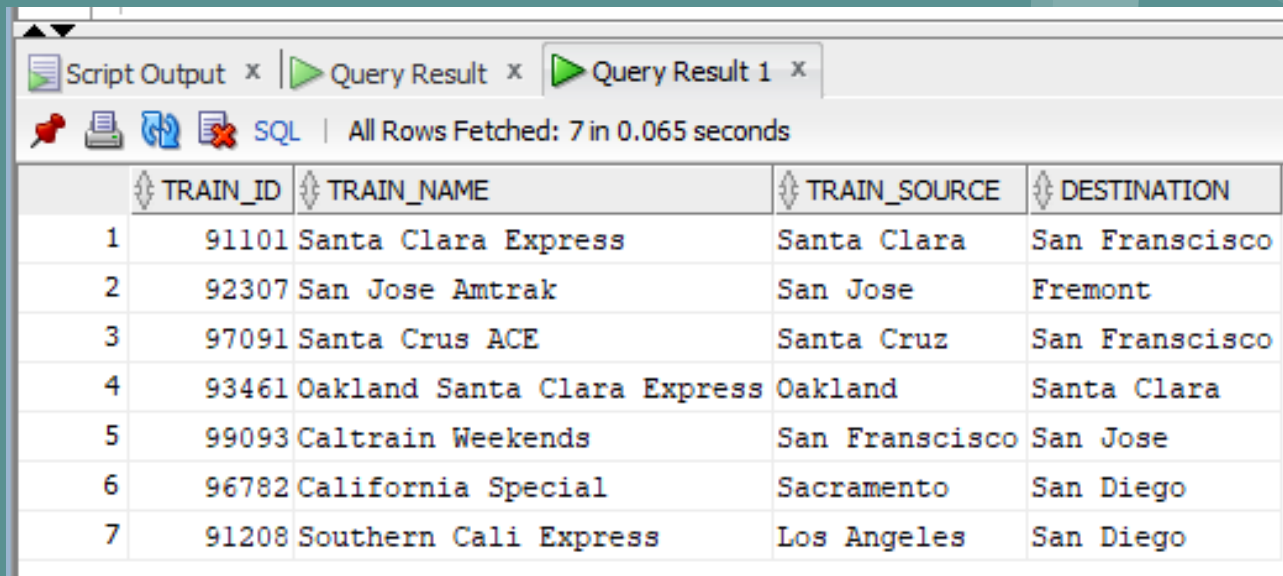
Proposed Query Plan

$\Pi_{\text{train_id,train_name,train_source,destination}}$

train



Oracle Output



The screenshot shows an Oracle SQL Developer window with a tab labeled 'Query Result 1'. The status bar indicates 'All Rows Fetched: 7 in 0.065 seconds'. The query results are displayed in a table with 5 columns: an index, TRAIN_ID, TRAIN_NAME, TRAIN_SOURCE, and DESTINATION. The data is as follows:

	TRAIN_ID	TRAIN_NAME	TRAIN_SOURCE	DESTINATION
1	91101	Santa Clara Express	Santa Clara	San Francisco
2	92307	San Jose Amtrak	San Jose	Fremont
3	97091	Santa Cruz ACE	Santa Cruz	San Francisco
4	93461	Oakland Santa Clara Express	Oakland	Santa Clara
5	99093	Caltrain Weekends	San Francisco	San Jose
6	96782	California Special	Sacramento	San Diego
7	91208	Southern Cali Express	Los Angeles	San Diego

Hive Output

```
> select * from train;
OK
91101  Santa Clara Express      Santa Clara  San Francisco
92307  San Jose Amtrak San Jose    Fremont
97091  Santa Crus ACE   Santa Cruz   San Francisco
93461  Oakland Santa Clara Express  Oakland Santa Clara
99093  Caltrain W2weekends  San Francisco  San Jose
96782  California Special  Sacramento    San Diego
91208  Southern Cali Express  Los Angeles   San Diego
Time taken: 0.172 seconds, Fetched: 7 row(s)
hive>
```

Oracle time: 0.065 secs
Hive time: 0.172 secs

2. Display all passenger details(Sorted by passenger name)

```
Select DISTINCT(passenger_id), p.passenger_name, age, t.train_id,  
t.train_name
```

```
From passenger p, train t
```

```
where t.train_id=p.train_id
```

```
order by passenger_name;
```

Oracle Explain Plan

Script Output x

Query Result x

All Rows Fetched: 18 in 0.07 seconds

PLAN_TABLE_OUTPUT

1

Plan hash value: 1804103205

2

3

4

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	88	3 (34)	00:00:01
1	SORT ORDER BY		1	88	3 (34)	00:00:01
2	NESTED LOOPS					
3	NESTED LOOPS		1	88	2 (0)	00:00:01
4	TABLE ACCESS FULL	PASSENGER	1	48	2 (0)	00:00:01
* 5	INDEX UNIQUE SCAN	SYS_C007016	1		0 (0)	00:00:01
6	TABLE ACCESS BY INDEX ROWID	TRAIN	1	40	0 (0)	00:00:01

5

6

0 | SELECT STATEMENT | | 1 | 88 | 3 (34) | 00:00:01 |

7

1 | SORT ORDER BY | | 1 | 88 | 3 (34) | 00:00:01 |

8

2 | NESTED LOOPS | | | | | |

9

3 | NESTED LOOPS | | 1 | 88 | 2 (0) | 00:00:01 |

10

4 | TABLE ACCESS FULL | PASSENGER | 1 | 48 | 2 (0) | 00:00:01 |

11

* 5 | INDEX UNIQUE SCAN | SYS_C007016 | 1 | | 0 (0) | 00:00:01 |

12

6 | TABLE ACCESS BY INDEX ROWID | TRAIN | 1 | 40 | 0 (0) | 00:00:01 |

13

14

15

Predicate Information (identified by operation id):

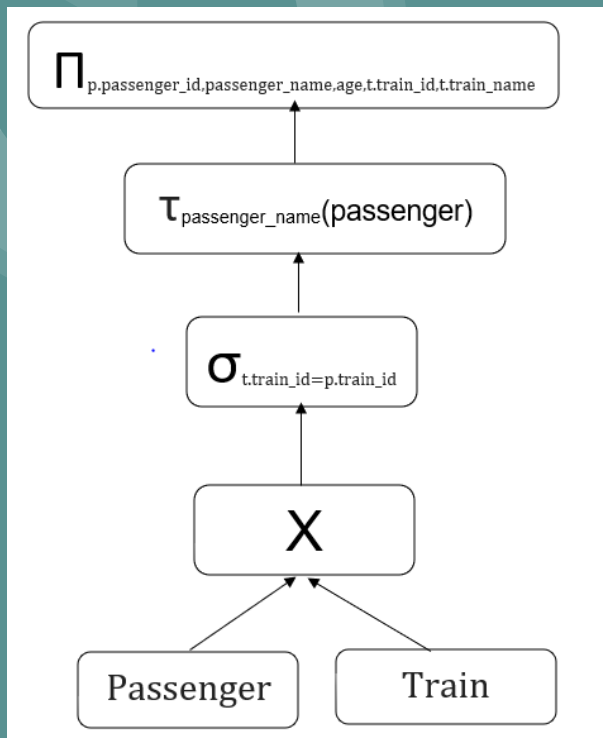
16

17

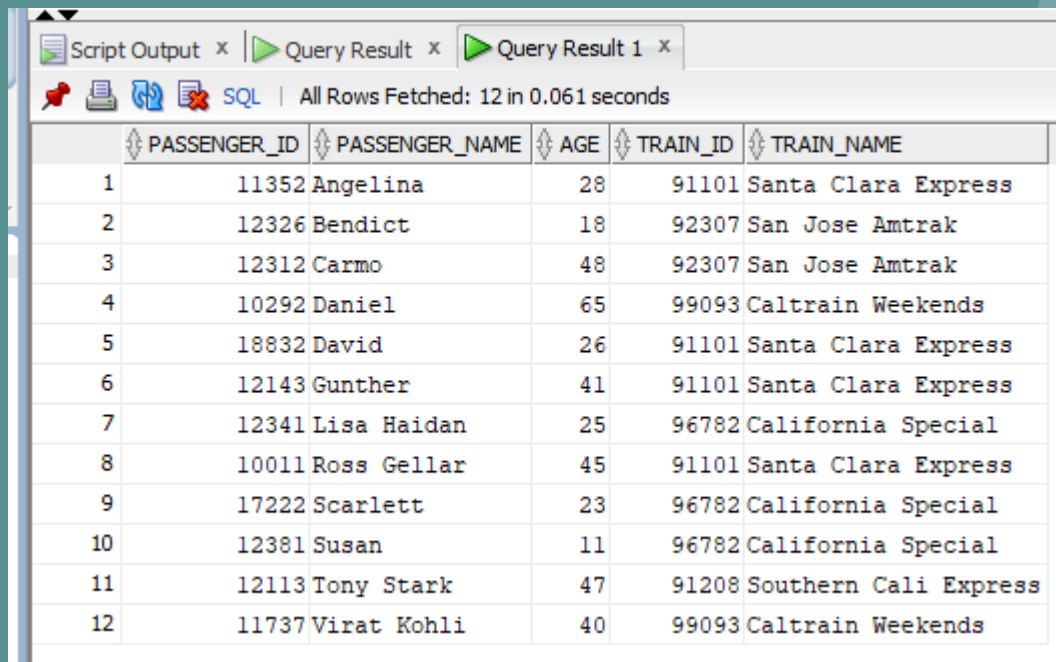
18

5 - access("T"."TRAIN_ID"="P"."TRAIN_ID")

Proposed Query Plan



Oracle Output



The screenshot shows an Oracle SQL Developer window with the following components:

- Tab Bar:** Contains three tabs: "Script Output x", "Query Result x", and "Query Result 1 x". The "Query Result 1 x" tab is active.
- Toolbar:** Includes icons for a pin, a printer, a refresh, a save, and a SQL icon.
- Status Bar:** Displays the text "All Rows Fetched: 12 in 0.061 seconds".
- Table:** A table with 5 columns: "PASSENGER_ID", "PASSENGER_NAME", "AGE", "TRAIN_ID", and "TRAIN_NAME". It contains 12 rows of data.

	PASSENGER_ID	PASSENGER_NAME	AGE	TRAIN_ID	TRAIN_NAME
1	11352	Angelina	28	91101	Santa Clara Express
2	12326	Bendict	18	92307	San Jose Amtrak
3	12312	Carmo	48	92307	San Jose Amtrak
4	10292	Daniel	65	99093	Caltrain Weekends
5	18832	David	26	91101	Santa Clara Express
6	12143	Gunther	41	91101	Santa Clara Express
7	12341	Lisa Haidan	25	96782	California Special
8	10011	Ross Gellar	45	91101	Santa Clara Express
9	17222	Scarlett	23	96782	California Special
10	12381	Susan	11	96782	California Special
11	12113	Tony Stark	47	91208	Southern Cali Express
12	11737	Virat Kohli	40	99093	Caltrain Weekends

Hive Output

```
Hadoop job information for Stage-3: number of mappers: 1; number of reducers: 1
2020-03-07 13:49:13,650 Stage-3 map = 0%, reduce = 0%
2020-03-07 13:49:17,912 Stage-3 map = 100%, reduce = 0%, Cumulative CPU 0.7 sec
2020-03-07 13:49:23,219 Stage-3 map = 100%, reduce = 100%, Cumulative CPU 1.67 sec
MapReduce Total cumulative CPU time: 1 seconds 670 msec
Ended Job = job_1581338931449_0639
MapReduce Jobs Launched:
Stage-Stage-2: Map: 1 Reduce: 1 Cumulative CPU: 2.16 sec HDFS Read: 13827 HDFS Write: 731 SUCCESS
Stage-Stage-3: Map: 1 Reduce: 1 Cumulative CPU: 1.67 sec HDFS Read: 6769 HDFS Write: 515 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 830 msec
OK
11352 Angelina 28 91101 Santa Clara Express
12326 Bendict 18 92307 San Jose Amtrak
12312 Carmo 48 92307 San Jose Amtrak
10292 Daniel 65 99093 Caltrain W2weekends
18832 David 26 91101 Santa Clara Express
12143 Gunther 41 91101 Santa Clara Express
12341 Lisa Haidan 25 96782 California Special
10011 Ross Gellar 45 91101 Santa Clara Express
17222 Scarlett 23 96782 California Special
12381 Susan 11 96782 California Special
12113 Tony Stark 47 91208 Southern Cali Express
11737 Virat Kohli 40 99093 Caltrain W2weekends
Time taken: 78.845 seconds, Fetched: 12 row(s)
hive>
```

Oracle time: 0.061 secs
Hive time: 78.845 secs

3. Display Trains passing through San Francisco(Nested Queries)

```
SELECT train_name FROM train
WHERE train_id IN
(SELECT route.train_id FROM route
WHERE route.route_point="San Francisco");
```


Oracle Explain Plan

Script Output x Query Result x Query Result 1 x Query Result 2 x

SQL | All Rows Fetched: 19 in 0.089 seconds

PLAN_TABLE_OUTPUT

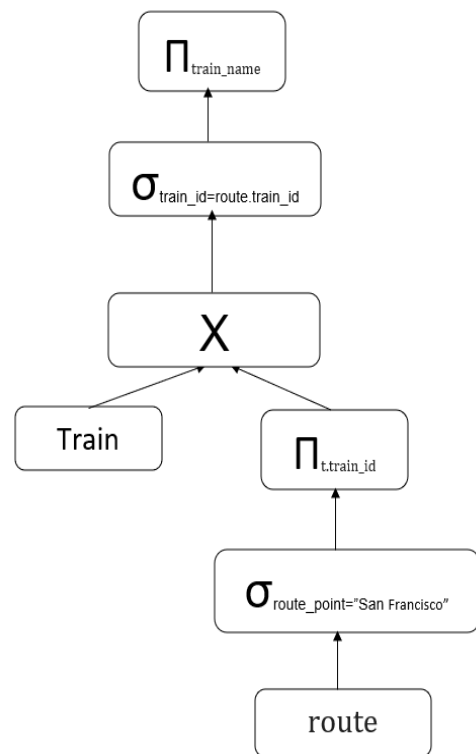
1 Plan hash value: 3402099364

2

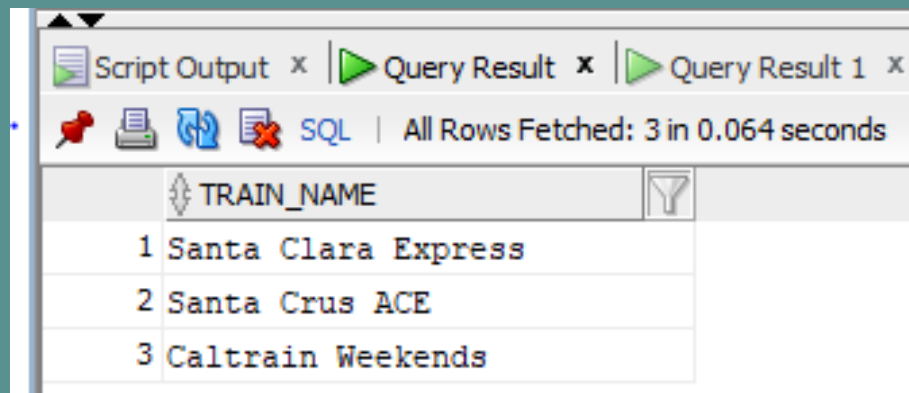
3 -----

4	Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
5							
6	0	SELECT STATEMENT		1	65	3 (34)	00:00:01
7	1	MERGE JOIN SEMI		1	65	3 (34)	00:00:01
8	2	TABLE ACCESS BY INDEX ROWID	TRAIN	1	40	0 (0)	00:00:01
9	3	INDEX FULL SCAN	SYS_C007016	1		0 (0)	00:00:01
10	* 4	SORT UNIQUE		1	25	3 (34)	00:00:01
11	* 5	TABLE ACCESS FULL	ROUTE	1	25	2 (0)	00:00:01
12							
13							
14	Predicate Information (identified by operation id):						
15							
16							
17	4	access("TRAIN_ID"="ROUTE"."TRAIN_ID")					
18		filter("TRAIN_ID"="ROUTE"."TRAIN_ID")					
19	5	filter("ROUTE"."ROUTE_POINT"='San Francisco')					

Proposed Query Plan



Oracle Output



The screenshot shows a window titled 'Script Output x | Query Result x | Query Result 1 x'. Below the title bar is a toolbar with icons for a pin, printer, undo, redo, and a document with a red X. To the right of the toolbar, it says 'SQL | All Rows Fetched: 3 in 0.064 seconds'. The main area displays a table with one column, 'TRAIN_NAME', which is highlighted. The table contains three rows of data.

	TRAIN_NAME
1	Santa Clara Express
2	Santa Crus ACE
3	Caltrain Weekends

Hive Output

```
hive> SELECT train_name FROM train
> WHERE train_id IN
> (SELECT route.train_id FROM route
> WHERE route.route_point = 'San Francisco');
FAILED: SemanticException [Error 10249]: Line 2:6 Unsupported SubQuery Expression 'train_id': Correlating expression can
not contain unqualified column references.
hive>
```

```
Execution completed successfully
MapredLocal task succeeded
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1581338931449_0791, Tracking URL = http://name1.h
Kill Command = /DCNFS/applications/cdh/5.12/app/hadoop-2.6.0-cdh5.12
Hadoop job information for Stage-3: number of mappers: 1; number of
2020-03-09 13:33:57,640 Stage-3 map = 0%, reduce = 0%
2020-03-09 13:34:02,954 Stage-3 map = 100%, reduce = 0%, Cumulative
MapReduce Total cumulative CPU time: 1 seconds 190 msec
Ended Job = job_1581338931449_0791
MapReduce Jobs Launched:
Stage-Stage-3: Map: 1 Cumulative CPU: 1.19 sec HDFS Read: 6906 H
Total MapReduce CPU Time Spent: 1 seconds 190 msec
OK
Santa Clara Express
Santa Crus ACE
Caltrain Wweekends
Time taken: 45.236 seconds, Fetched: 3 row(s)
hive>
```

Explicitly mention the column which we are referring to.

Oracle time: 0.064 secs

Hive time: 45.236 secs

4. Top 4 trains having maximum booked reservations

ORACLE:

```
SELECT Train_name, booked_count FROM Train t
INNER JOIN TrainStatus ts ON
t.train_id = ts.train_id
ORDER BY (Booked_count DESC) tr
WHERE ROWNUM < 5 ;
```

HIVE:

```
SELECT Train_name, booked_count FROM Train t
INNER JOIN TrainStatus ts ON
t.train_id = ts.train_id
ORDER BY Booked_count DESC limit 4;
```

Oracle Explain Plan

Script Output x Query Result x Query Result 1 x Query Result 2 x Query Result 3 x

SQL | All Rows Fetched: 22 in 0.263 seconds

PLAN_TABLE_OUTPUT

1 Plan hash value: 3714950130

2

3

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	40	3 (34)	00:00:01
1	COUNT STOPKEY					
2	VIEW		1	40	3 (34)	00:00:01
3	SORT ORDER BY STOPKEY		1	66	3 (34)	00:00:01
4	NESTED LOOPS					
5	NESTED LOOPS		1	66	2 (0)	00:00:01
6	TABLE ACCESS FULL	TRAIN	1	40	2 (0)	00:00:01
7	INDEX RANGE SCAN	SYS_C007019	1		0 (0)	00:00:01
8	TABLE ACCESS BY INDEX ROWID	TRAINSTATUS	1	26	0 (0)	00:00:01

17 Predicate Information (identified by operation id):

18

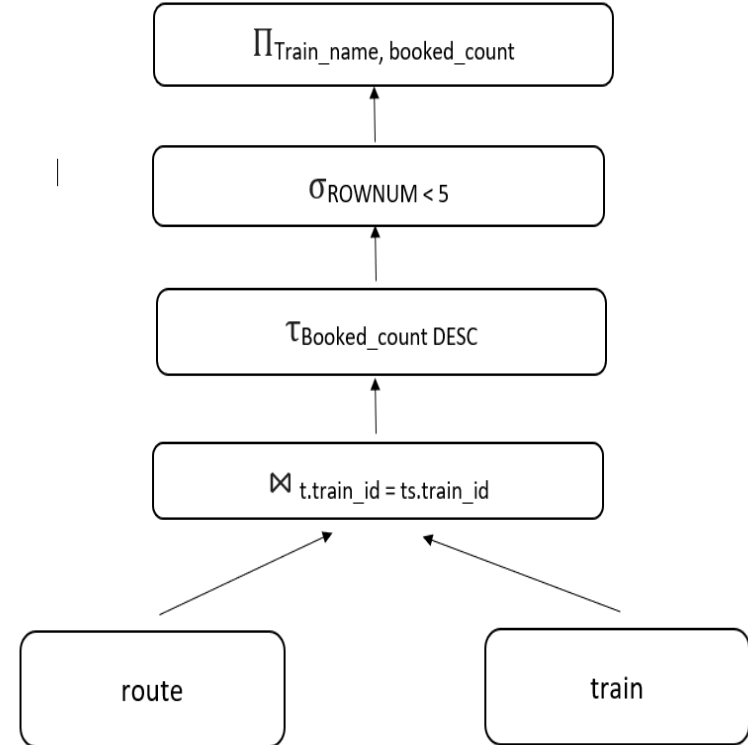
19

20 1 - filter(ROWNUM<5)

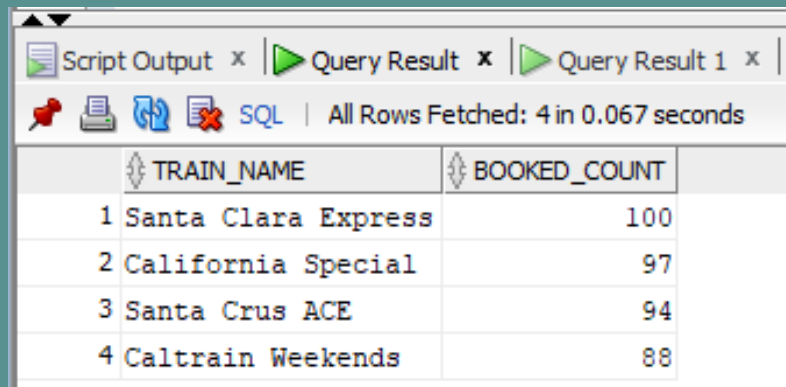
21 3 - filter(ROWNUM<5)

22 7 - access("T"."TRAIN_ID"="TS"."TRAIN_ID")

Proposed Query Plan



Oracle Output



Script Output x | Query Result x | Query Result 1 x |

SQL | All Rows Fetched: 4 in 0.067 seconds

	TRAIN_NAME	BOOKED_COUNT
1	Santa Clara Express	100
2	California Special	97
3	Santa Crus ACE	94
4	Caltrain Weekends	88

Hive Output

```
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2020-03-07 14:30:31,645 Stage-2 map = 0%, reduce = 0%
2020-03-07 14:30:35,901 Stage-2 map = 100%, reduce = 0%
2020-03-07 14:30:42,355 Stage-2 map = 100%, reduce = 100%, Cumulative CPU 2.16 sec
MapReduce Total cumulative CPU time: 2 seconds 160 msec
Ended Job = job_1581338931449_0644
MapReduce Jobs Launched:
Stage-Stage-2: Map: 1 Reduce: 1 Cumulative CPU: 2.16 sec HDFS Read: 11918 HDFS Write: 86 SUCCESS
Total MapReduce CPU Time Spent: 2 seconds 160 msec
OK
Santa Clara Express      100
California Special       97
Santa Crus ACE           94
Caltrain Wweekends       88
Time taken: 57.875 seconds, Fetched: 4 row(s)
hive>
```

Oracle time: 0.067 secs
Hive time: 57.875 secs

5. Count number of Train Stops for each train

```
SELECT train_id train_name, count(*) as Stops  
FROM train T  
JOIN  
route r  
ON r.train_id = t.train_id  
GROUP BY t.train_id, t.train_name, Date(arrival_time);
```



Oracle Explain Plan

Script Output x Query Result x

All Rows Fetched: 18 in 0.071 seconds

PLAN_TABLE_OUTPUT

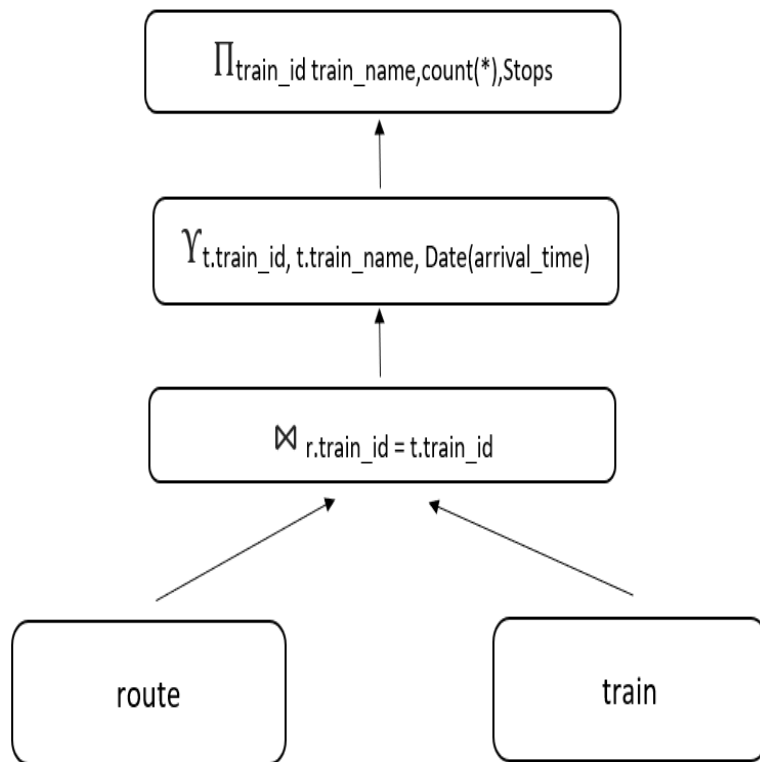
1 Plan hash value: 3631851623

2

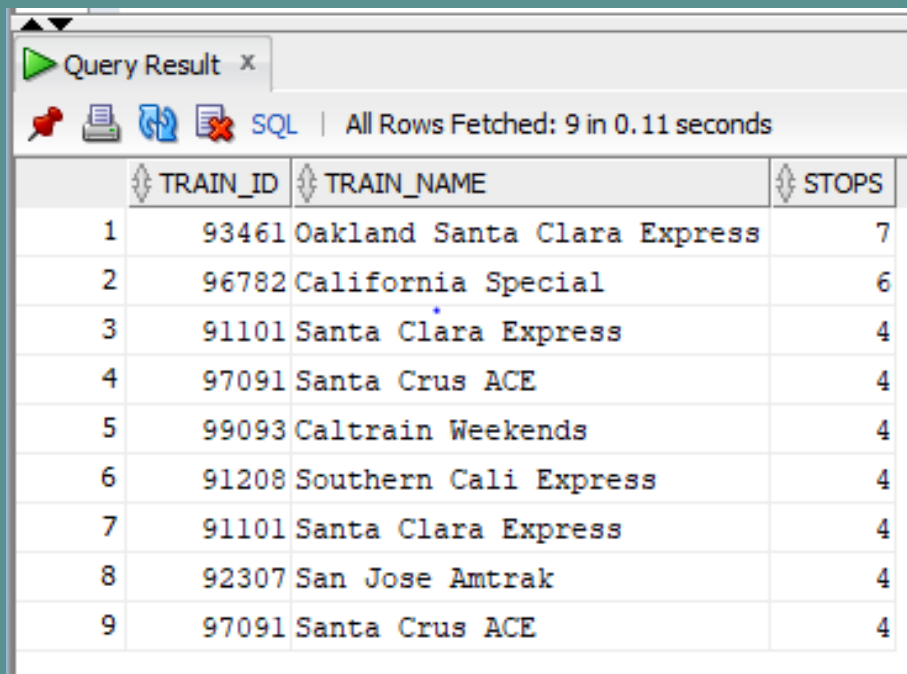
3 -----

4	Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
5							
6	0	SELECT STATEMENT		1	62	3 (34)	00:00:01
7	1	HASH GROUP BY		1	62	3 (34)	00:00:01
8	2	NESTED LOOPS					
9	3	NESTED LOOPS		1	62	2 (0)	00:00:01
10	4	TABLE ACCESS FULL	ROUTE	1	22	2 (0)	00:00:01
11	* 5	INDEX UNIQUE SCAN	SYS_C007016	1		0 (0)	00:00:01
12	6	TABLE ACCESS BY INDEX ROWID	TRAIN	1	40	0 (0)	00:00:01
13							
14							
15	Predicate Information (identified by operation id):						
16	-----						
17							
18	5	access("T"."TRAIN_ID"="R"."TRAIN_ID")					

Proposed Query Plan



Oracle Output



Query Result x

SQL | All Rows Fetched: 9 in 0.11 seconds

	TRAIN_ID	TRAIN_NAME	STOPS
1	93461	Oakland Santa Clara Express	7
2	96782	California Special	6
3	91101	Santa Clara Express	4
4	97091	Santa Crus ACE	4
5	99093	Caltrain Weekends	4
6	91208	Southern Cali Express	4
7	91101	Santa Clara Express	4
8	92307	San Jose Amtrak	4
9	97091	Santa Crus ACE	4

Oracle Time: 0.11 secs
Hive Time: 61.207 secs

Hive Output

```
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2020-03-07 15:06:25,187 Stage-2 map = 0%, reduce = 0%
2020-03-07 15:06:30,514 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 1.07 sec
2020-03-07 15:06:35,894 Stage-2 map = 100%, reduce = 100%, Cumulative CPU 2.13 sec
MapReduce Total cumulative CPU time: 2 seconds 130 msec
Ended Job = job_1581338931449_0650
MapReduce Jobs Launched:
Stage-Stage-2: Map: 1 Reduce: 1 Cumulative CPU: 2.13 sec HDFS Read: 14552 HDFS Write: 246 SUCCESS
Total MapReduce CPU Time Spent: 2 seconds 130 msec
OK
91101 Santa Clara Express 4
91101 Santa Clara Express 4
91208 Southern Cali Express 4
92307 San Jose Amtrak 4
93461 Oakland Santa Clara Express 7
96782 California Special 6
97091 Santa Crus ACE 4
97091 Santa Crus ACE 4
99093 Caltrain W2eekends 4
Time taken: 61.207 seconds, Fetched: 9 row(s)
hive>
```

6. Calculate male to female ratio

In Oracle:

```
SELECT gender, Count(*) / (select count(*) FROM Passenger) AS Sex_Ratio FROM passenger  
GROUP BY gender;
```

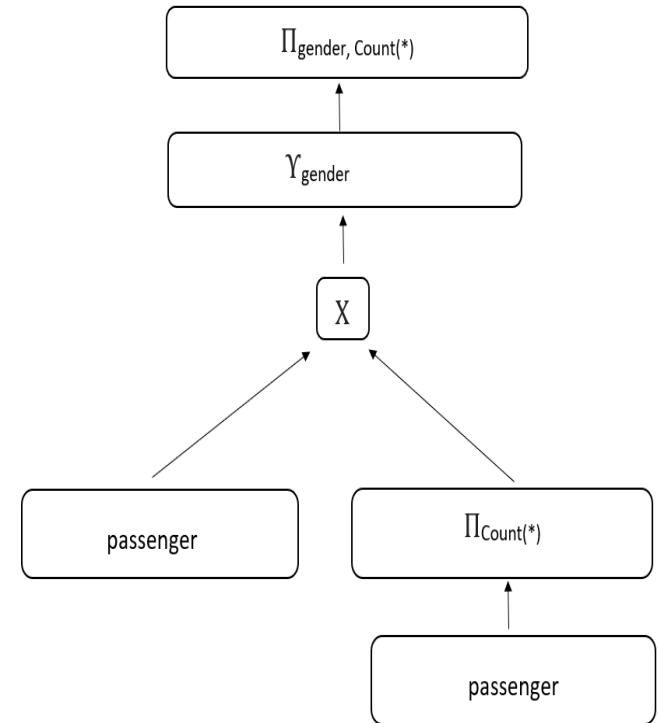
In Hive:

```
with q1 as ( SELECT COUNT(*) AS total_count FROM Passenger),  
q2 as (SELECT gender,COUNT(*) as gender_count FROM Passenger GROUP BY  
gender)  
select gender,gender_count/total_count as Sex_Ratio from q1,q2;
```

Oracle Explain Plan

Script Output x Query Result x Query Result 1 x							
SQL All Rows Fetched: 11 in 0.074 seconds							
PLAN_TABLE_OUTPUT							
1	Plan hash value: 3057175957						
2							
3	-----						
4	Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
5	-----						
6	0	SELECT STATEMENT		1	3	3 (34)	00:00:01
7	1	SORT AGGREGATE		1			
8	2	INDEX FULL SCAN	SYS_C007028	1		0 (0)	00:00:01
9	3	HASH GROUP BY		1	3	3 (34)	00:00:01
10	4	TABLE ACCESS FULL	PASSENGER	1	3	2 (0)	00:00:01
11	-----						

Proposed Query Plan



Oracle Output

Oracle Time: 0.149 secs

Hive Time: 95.357 secs

[illegible]

Hive Output

```
Hadoop job information for Stage-3: number of mappers: 1; number of reducers: 1
2020-03-07 12:49:44,816 Stage-3 map = 0%, reduce = 0%
2020-03-07 12:49:49,084 Stage-3 map = 100%, reduce = 0%, Cumulative CPU 0.77 sec
2020-03-07 12:49:53,353 Stage-3 map = 100%, reduce = 100%, Cumulative CPU 1.76 sec
MapReduce Total cumulative CPU time: 1 seconds 760 msec
Ended Job = job_1581338931449_0632
Stage-7 is filtered out by condition resolver.
Stage-8 is selected by condition resolver.
Stage-2 is filtered out by condition resolver.
Execution log at: /tmp/coen38305/coen38305_20200307124949_bc474d0e-0a22-462c-a5f9-91db160c7539.log
2020-03-07 12:50:37 Starting to launch local task to process map join; maximum memory = 1908932608
2020-03-07 12:50:38 Dump the side-table for tag: 0 with group count: 1 into file: file:/tmp/coen38305/b06c093e-2511-4531-be8c-51baa97f7901/hive_2020-03-07_12-49-16_698_4334107522297512247-1/-local-10007/HashTable-Stage-5/MapJoin-mapfile10--.hashtable
2020-03-07 12:50:38 Uploaded 1 File to: file:/tmp/coen38305/b06c093e-2511-4531-be8c-51baa97f7901/hive_2020-03-07_12-49-16_698_4334107522297512247-1/-local-10007/HashTable-Stage-5/MapJoin-mapfile10--.hashtable (278 bytes)
2020-03-07 12:50:38 End of local task; Time Taken: 0.694 sec.
Execution completed successfully
MapredLocal task succeeded
Launching Job 4 out of 5
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1581338931449_0633, Tracking URL = http://name1.hadoop.dc.engr.scu.edu:8088/proxy/application_1581338931449_0633/Kill Command = /DCNFS/applications/cdh/5.12/app/hadoop-2.6.0-cdh5.12.1/bin/hadoop job -kill job_1581338931449_0633
Hadoop job information for Stage-5: number of mappers: 1; number of reducers: 0
2020-03-07 12:50:45,605 Stage-5 map = 0%, reduce = 0%
2020-03-07 12:50:50,913 Stage-5 map = 100%, reduce = 0%, Cumulative CPU 1.2 sec
MapReduce Total cumulative CPU time: 1 seconds 200 msec
Ended Job = job_1581338931449_0633
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 1.78 sec HDFS Read: 7440 HDFS Write: 114 SUCCESS
Stage-Stage-3: Map: 1 Reduce: 1 Cumulative CPU: 1.76 sec HDFS Read: 7747 HDFS Write: 136 SUCCESS
Stage-Stage-5: Map: 1 Cumulative CPU: 1.2 sec HDFS Read: 5895 HDFS Write: 42 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 740 msec
OK
F 0.4166666666666667
M 0.5833333333333334
Time taken: 95.357 seconds, Fetched: 2 row(s)
```

7. Display Trains scheduled between 12AM to 12PM

In ORACLE:

```
SELECT DISTINCT train_name, r.* FROM route r  
LEFT JOIN Train t  
ON t.train_id = r.train_id  
Where TO_Char(Arrival_time,'hh24')  
between 0 AND 11;
```

In HIVE:

```
Select t.train_id, train_name, arrival_time, route_point from route r  
join train t on t.train_id = r.train_id  
where HOUR(arrival_time) between 0 AND 11;
```



Oracle Explain Plan

Script Output x Query Result x Query Result 1 x Query Result 2 x

SQL | All Rows Fetched: 19 in 0.08 seconds

PLAN_TABLE_OUTPUT

1 Plan hash value: 839668916

2

3

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	49	4 (25)	00:00:01
1	HASH UNIQUE		1	49	4 (25)	00:00:01
2	NESTED LOOPS OUTER		1	49	3 (0)	00:00:01
* 3	TABLE ACCESS FULL	ROUTE	1	24	2 (0)	00:00:01
4	TABLE ACCESS BY INDEX ROWID	TRAIN	1	25	1 (0)	00:00:01
* 5	INDEX UNIQUE SCAN	SYS_C007016	1		0 (0)	00:00:01

14 Predicate Information (identified by operation id):

15

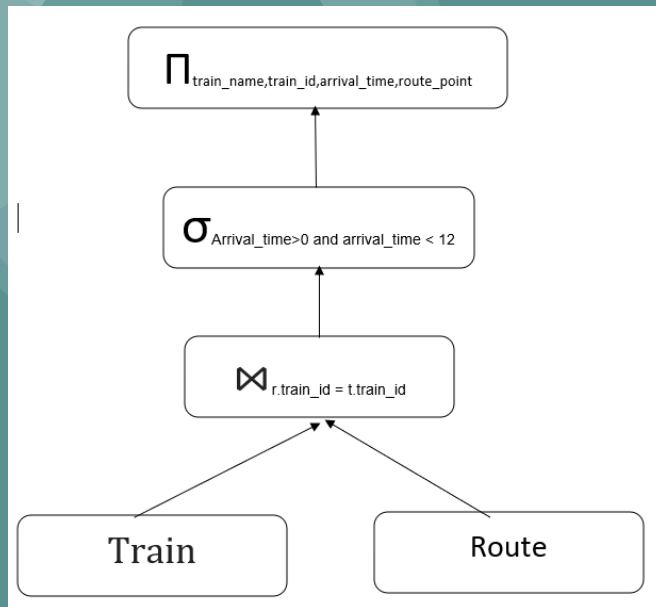
16

17 3 - filter(TO_NUMBER(TO_CHAR(INTERNAL_FUNCTION("R"."ARRIVAL_TIME"),'hh24'))>=0

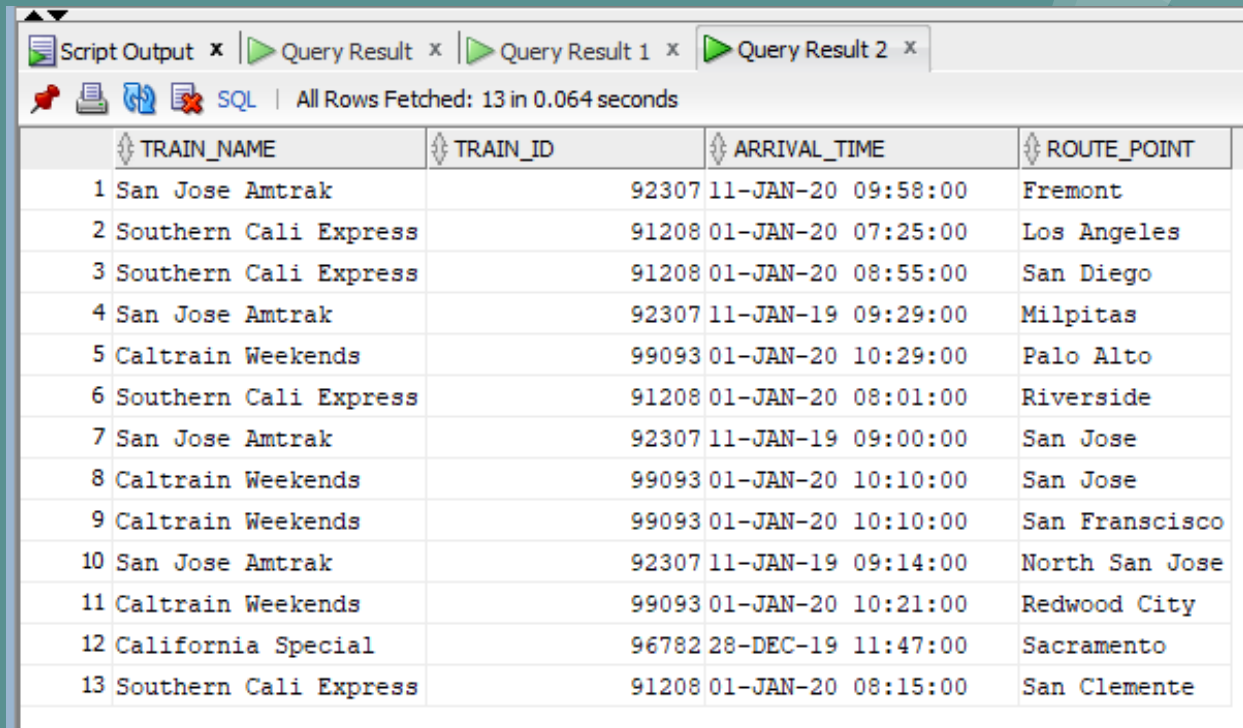
18 AND TO_NUMBER(TO_CHAR(INTERNAL_FUNCTION("R"."ARRIVAL_TIME"),'hh24'))<=11)

19 5 - access("T"."TRAIN_ID" (+)="R"."TRAIN_ID")

Proposed Query Plan



Oracle Output



The screenshot shows an Oracle SQL Developer window with a query result. The window has a title bar with tabs for 'Script Output', 'Query Result', 'Query Result 1', and 'Query Result 2'. Below the tabs is a toolbar with icons for saving, refreshing, and other functions. The main area displays a table with 4 columns: TRAIN_NAME, TRAIN_ID, ARRIVAL_TIME, and ROUTE_POINT. The table contains 13 rows of data, numbered 1 through 13. The data includes train names like 'San Jose Amtrak', 'Southern Cali Express', 'Caltrain Weekends', and 'California Special', along with their respective IDs, arrival times, and route points.

	TRAIN_NAME	TRAIN_ID	ARRIVAL_TIME	ROUTE_POINT
1	San Jose Amtrak	92307	11-JAN-20 09:58:00	Fremont
2	Southern Cali Express	91208	01-JAN-20 07:25:00	Los Angeles
3	Southern Cali Express	91208	01-JAN-20 08:55:00	San Diego
4	San Jose Amtrak	92307	11-JAN-19 09:29:00	Milpitas
5	Caltrain Weekends	99093	01-JAN-20 10:29:00	Palo Alto
6	Southern Cali Express	91208	01-JAN-20 08:01:00	Riverside
7	San Jose Amtrak	92307	11-JAN-19 09:00:00	San Jose
8	Caltrain Weekends	99093	01-JAN-20 10:10:00	San Jose
9	Caltrain Weekends	99093	01-JAN-20 10:10:00	San Francisco
10	San Jose Amtrak	92307	11-JAN-19 09:14:00	North San Jose
11	Caltrain Weekends	99093	01-JAN-20 10:21:00	Redwood City
12	California Special	96782	28-DEC-19 11:47:00	Sacramento
13	Southern Cali Express	91208	01-JAN-20 08:15:00	San Clemente

Hive Output

```
Execution completed successfully
MapredLocal task succeeded
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1581338931449_0785, Tracking URL = http://name1.hadoop.dc.engr.scu.edu:8088/pro
Kill Command = /DCNFS/applications/cdh/5.12/app/hadoop-2.6.0-cdh5.12.1/bin/hadoop job -kill job_1
Hadoop job information for Stage-3: number of mappers: 1; number of reducers: 0
2020-03-09 12:38:24,313 Stage-3 map = 0%, reduce = 0%
2020-03-09 12:38:29,639 Stage-3 map = 100%, reduce = 0%, Cumulative CPU 1.28 sec
MapReduce Total cumulative CPU time: 1 seconds 280 msec
Ended Job = job_1581338931449_0785
MapReduce Jobs Launched:
Stage-Stage-3: Map: 1 Cumulative CPU: 1.28 sec HDFS Read: 9039 HDFS Write: 728 SUCCESS
Total MapReduce CPU Time Spent: 1 seconds 280 msec
OK
92307 San Jose Amtrak 2020-01-11 09:00:00 San Jose
92307 San Jose Amtrak 2020-01-11 09:14:00 North San Jose
92307 San Jose Amtrak 2020-01-11 09:29:00 Milpitas
92307 San Jose Amtrak 2020-01-11 09:58:00 Fremont
99093 Caltrain W2eekends 2020-01-01 10:10:00 San Francisco
99093 Caltrain W2eekends 2020-01-01 10:21:00 Redwood City
99093 Caltrain W2eekends 2020-01-01 10:29:00 Palo Alto
99093 Caltrain W2eekends 2020-01-01 10:10:00 San Jose
96782 California Special 2019-12-28 11:47:00 Sacramento
91208 Southern Cali Express 2020-01-01 07:25:00 Los Angeles
91208 Southern Cali Express 2020-01-01 08:01:00 Riverside
91208 Southern Cali Express 2020-01-01 08:15:00 San Clemente
91208 Southern Cali Express 2020-01-01 08:55:00 San Diego
Time taken: 59.055 seconds, Fetched: 13 row(s)
hive>
```

Oracle Time: 0.064 secs
Hive Time: 59.055 secs

8. Display Trains who takes more than 4 stops from source to destination

ORACLE

```
SELECT train_name, count(*) FROM train T
JOIN
route r
ON r.train_id = t.train_id
GROUP BY train_name, TRUNC(arrival_time)
HAVING Count(*) > 4;
```

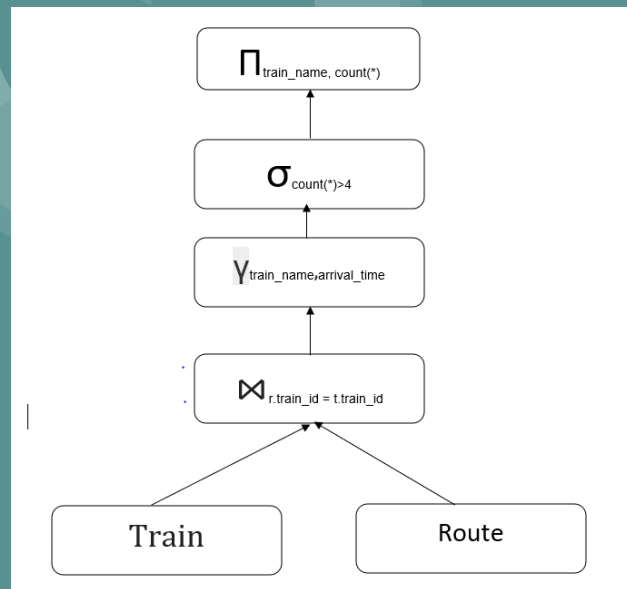
HIVE:

```
SELECT train_name, count(*) FROM train T
JOIN
route r
ON r.train_id = t.train_id
GROUP BY train_name, DATE(arrival_time)
HAVING Count(*) > 4;
```

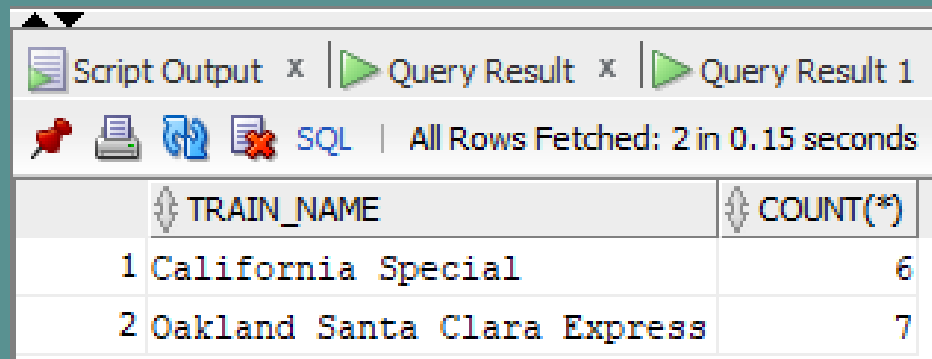
Oracle Explain Plan

Script Output x Query Result x Query Result 1 x Query Result 2 x Query Result 3 x Query R							
SQL All Rows Fetched: 18 in 0.071 seconds							
PLAN_TABLE_OUTPUT							
1 Plan hash value: 3701953459							
2							
3 -----							
4	Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
5 -----							
6	0	SELECT STATEMENT		3	114	6 (34)	00:00:01
7	* 1	FILTER					
8	2	HASH GROUP BY		3	114	6 (34)	00:00:01
9	* 3	HASH JOIN		41	1558	5 (20)	00:00:01
10	4	TABLE ACCESS FULL	TRAIN	7	175	2 (0)	00:00:01
11	5	TABLE ACCESS FULL	ROUTE	41	533	2 (0)	00:00:01
12 -----							
13							
14 Predicate Information (identified by operation id):							
15 -----							
16							
17	1 - filter(COUNT(*)>4)						
18	3 - access("R"."TRAIN_ID"="T"."TRAIN_ID")						

Proposed Query Plan



Oracle Output



TRAIN_NAME	COUNT(*)
1 California Special	6
2 Oakland Santa Clara Express	7

Hive Output

```
Execution completed successfully
MapredLocal task succeeded
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1581338931449_0880, Tracking URL = http://name1.hadoop.dc.engr.s
931449_0880/
Kill Command = /DCNFS/applications/cdh/5.12/app/hadoop-2.6.0-cdh5.12.1/bin/hadoop j
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2020-03-09 22:20:10,334 Stage-2 map = 0%, reduce = 0%
2020-03-09 22:20:14,663 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 1.08 sec
2020-03-09 22:20:21,108 Stage-2 map = 100%, reduce = 100%, Cumulative CPU 2.44 sec
MapReduce Total cumulative CPU time: 2 seconds 440 msec
Ended Job = job_1581338931449_0880
MapReduce Jobs Launched:
Stage-Stage-2: Map: 1 Reduce: 1 Cumulative CPU: 2.44 sec HDFS Read: 14272 HDFS
Total MapReduce CPU Time Spent: 2 seconds 440 msec
OK
California Special      6
Oakland Santa Clara Express  7
Time taken: 60.108 seconds, Fetched: 2 row(s)
```

Oracle Time: 0.15 secs
Hive Time: 60.108 secs

9. Display Passenger and train details combined where ticket price is equal to 35 (4 Way Join)

```
SELECT DISTINCT(Passenger_name), Train_name,  
Ticket_price, ti.Ticket_id, Ticket_status, Status_date FROM  
Passenger p  
INNER JOIN Train t ON t.train_id = p.train_id  
INNER JOIN Ticket ti ON p.ticket_id = ti.ticket_id  
INNER JOIN TrainStatus ts ON t.train_id = ts.train_id  
WHERE Ticket_price = 35
```


Oracle Explain Plan

Script Output x Query Result x Query Result 1 x Query Result 2 x Query Result 3 x

SQL | All Rows Fetched: 23 in 0.36 seconds

PLAN_TABLE_OUTPUT

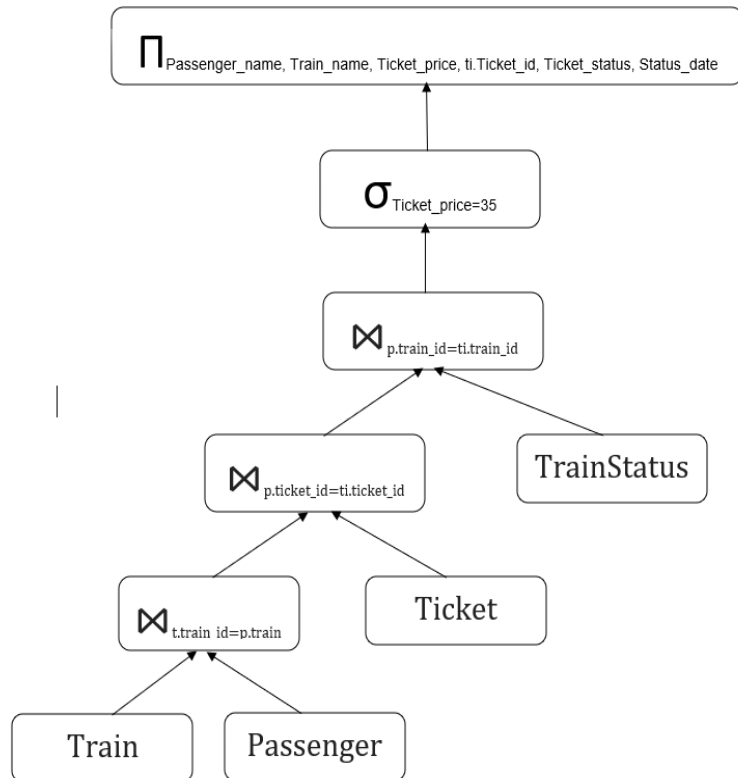
1 Plan hash value: 3952128676

2

3 -----

4	Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
5							
6	0	SELECT STATEMENT		62	4526	9 (23)	00:00:01
7	1	HASH UNIQUE		62	4526	9 (23)	00:00:01
8	* 2	HASH JOIN		62	4526	8 (13)	00:00:01
9	* 3	HASH JOIN		24	1440	7 (15)	00:00:01
10	4	MERGE JOIN CARTESIAN		14	574	4 (0)	00:00:01
11	* 5	TABLE ACCESS FULL	TICKET	2	32	2 (0)	00:00:01
12	6	BUFFER SORT		7	175	2 (0)	00:00:01
13	7	TABLE ACCESS FULL	TRAIN	7	175	1 (0)	00:00:01
14	8	TABLE ACCESS FULL	PASSENGER	12	228	2 (0)	00:00:01
15	9	INDEX FULL SCAN	SYS_C007019	18	234	1 (0)	00:00:01
16							
17							
18	Predicate Information (identified by operation id):						
19	-----						
20							
21	2	access("T"."TRAIN_ID"="TS"."TRAIN_ID")					
22	3	access("T"."TRAIN_ID"="P"."TRAIN_ID")					
23	5	filter("TI"."TICKET_PRICE"=35)					

Proposed Query Plan



Oracle Output

Script Output x Query Result x Query Result 1 x Query Result 2 x Query Result 3 x

SQL | All Rows Fetched: 70 in 0.216 seconds

	PASSENGER_NAME	TRAIN_NAME	TICKET_PRICE	TICKET_ID	TICKET_STATUS	STATUS_DATE
1	Ross Gellar	Santa Clara Express	35	30244	Waiting	13-DEC-19 00:00:00
2	Gunther	Santa Clara Express	35	30110	Waiting	01-JAN-20 00:00:00
3	David	Santa Clara Express	35	30244	Waiting	01-JAN-20 00:00:00
4	Ross Gellar	Santa Clara Express	35	30110	Waiting	01-JAN-20 00:00:00
5	Ross Gellar	Santa Clara Express	35	30244	Waiting	01-JAN-20 00:00:00
6	Gunther	Santa Clara Express	35	30110	Waiting	28-DEC-20 00:00:00
7	David	Santa Clara Express	35	30110	Waiting	28-DEC-20 00:00:00
8	Tony Stark	Southern Cali Express	35	30110	Waiting	17-JAN-20 00:00:00
9	Scarlett	California Special	35	30244	Waiting	01-JAN-20 00:00:00
10	Susan	California Special	35	30244	Waiting	09-FEB-20 00:00:00
11	Virat Kohli	Caltrain Weekends	35	30110	Waiting	01-JAN-20 00:00:00
12	David	Santa Clara Express	35	30110	Waiting	13-DEC-19 00:00:00
13	Benedict	San Jose Amtrak	35	30244	Waiting	11-JAN-20 00:00:00
14	Benedict	San Jose Amtrak	35	30110	Waiting	17-JAN-20 00:00:00
15	Carmo	San Jose Amtrak	35	30110	Waiting	17-JAN-20 00:00:00
16	Scarlett	California Special	35	30110	Waiting	01-JAN-20 00:00:00
17	Scarlett	California Special	35	30244	Waiting	09-FEB-20 00:00:00
18	Lisa Haidan	California Special	35	30110	Waiting	09-FEB-20 00:00:00
19	Susan	California Special	35	30244	Waiting	28-DEC-20 00:00:00
20	Scarlett	California Special	35	30244	Waiting	28-DEC-20 00:00:00
21	Daniel	Caltrain Weekends	35	30110	Waiting	01-JAN-20 00:00:00
22	Virat Kohli	Caltrain Weekends	35	30244	Waiting	22-JAN-20 00:00:00
23	Angelina	Santa Clara Express	35	30244	Waiting	24-DEC-20 00:00:00
24	David	Santa Clara Express	35	30244	Waiting	24-DEC-20 00:00:00
25	David	Santa Clara Express	35	30244	Waiting	28-DEC-20 00:00:00
26	Ross Gellar	Santa Clara Express	35	30244	Waiting	28-DEC-20 00:00:00

Hive Output

```
Hadoop job information for Stage-3: number of mappers: 1; number of reducers: 1
2020-03-09 12:48:14,954 Stage-3 map = 0%, reduce = 0%
2020-03-09 12:48:20,372 Stage-3 map = 100%, reduce = 0%, Cumulative CPU 1.14 sec
2020-03-09 12:48:25,740 Stage-3 map = 100%, reduce = 100%, Cumulative CPU 2.23 sec
MapReduce Total cumulative CPU time: 2 seconds 230 msec
Ended Job = job_1581338931449_0787
MapReduce Jobs Launched:
Stage-Stage-3: Map: 1 Reduce: 1 Cumulative CPU: 2.23 sec HDFS Read: 19465 HDFS Write: 3928 SUCCESS
Total MapReduce CPU Time Spent: 2 seconds 230 msec
OK
Angelina Santa Clara Express 35 30110 Waiting 2019-12-13
Angelina Santa Clara Express 35 30110 Waiting 2020-01-01
Angelina Santa Clara Express 35 30110 Waiting 2020-12-24
Angelina Santa Clara Express 35 30110 Waiting 2020-12-28
Angelina Santa Clara Express 35 30244 Waiting 2019-12-13
Angelina Santa Clara Express 35 30244 Waiting 2020-01-01
Angelina Santa Clara Express 35 30244 Waiting 2020-12-24
Angelina Santa Clara Express 35 30244 Waiting 2020-12-28
Bendict San Jose Amtrak 35 30110 Waiting 2020-01-11
Bendict San Jose Amtrak 35 30110 Waiting 2020-01-17
Bendict San Jose Amtrak 35 30244 Waiting 2020-01-11
Bendict San Jose Amtrak 35 30244 Waiting 2020-01-17
Carmo San Jose Amtrak 35 30110 Waiting 2020-01-11
Carmo San Jose Amtrak 35 30110 Waiting 2020-01-17
Carmo San Jose Amtrak 35 30244 Waiting 2020-01-11
Carmo San Jose Amtrak 35 30244 Waiting 2020-01-17
Daniel Caltrain W2weekends 35 30110 Waiting 2020-01-01
Daniel Caltrain W2weekends 35 30110 Waiting 2020-01-22
Daniel Caltrain W2weekends 35 30244 Waiting 2020-01-01
Daniel Caltrain W2weekends 35 30244 Waiting 2020-01-22
David Santa Clara Express 35 30110 Waiting 2019-12-13
David Santa Clara Express 35 30110 Waiting 2020-01-01
David Santa Clara Express 35 30110 Waiting 2020-12-24
David Santa Clara Express 35 30110 Waiting 2020-12-28
David Santa Clara Express 35 30244 Waiting 2019-12-13
David Santa Clara Express 35 30244 Waiting 2020-01-01
David Santa Clara Express 35 30244 Waiting 2020-12-24
David Santa Clara Express 35 30244 Waiting 2020-12-28
Gunther Santa Clara Express 35 30110 Waiting 2019-12-13
Gunther Santa Clara Express 35 30110 Waiting 2020-01-01
```

Oracle Time: 0.216 secs
Hive Time: 60.325 secs

Lisa Haidan	California Special	35	30110	Waiting	2020-01-01
Lisa Haidan	California Special	35	30110	Waiting	2020-02-09
Lisa Haidan	California Special	35	30110	Waiting	2020-12-28
Lisa Haidan	California Special	35	30244	Waiting	2020-01-01
Lisa Haidan	California Special	35	30244	Waiting	2020-02-09
Lisa Haidan	California Special	35	30244	Waiting	2020-12-28
Ross Gellar	Santa Clara Express	35	30110	Waiting	2019-12-13
Ross Gellar	Santa Clara Express	35	30110	Waiting	2020-01-01
Ross Gellar	Santa Clara Express	35	30110	Waiting	2020-12-24
Ross Gellar	Santa Clara Express	35	30110	Waiting	2020-12-28
Ross Gellar	Santa Clara Express	35	30244	Waiting	2019-12-13
Ross Gellar	Santa Clara Express	35	30244	Waiting	2020-01-01
Ross Gellar	Santa Clara Express	35	30244	Waiting	2020-12-24
Ross Gellar	Santa Clara Express	35	30244	Waiting	2020-12-28
Scarlett	California Special	35	30110	Waiting	2020-01-01
Scarlett	California Special	35	30110	Waiting	2020-02-09
Scarlett	California Special	35	30110	Waiting	2020-12-28
Scarlett	California Special	35	30244	Waiting	2020-01-01
Scarlett	California Special	35	30244	Waiting	2020-02-09
Scarlett	California Special	35	30244	Waiting	2020-12-28
Susan	California Special	35	30110	Waiting	2020-01-01
Susan	California Special	35	30110	Waiting	2020-02-09
Susan	California Special	35	30110	Waiting	2020-12-28
Susan	California Special	35	30244	Waiting	2020-01-01
Susan	California Special	35	30244	Waiting	2020-02-09
Susan	California Special	35	30244	Waiting	2020-12-28
Tony Stark	Southern Cali Express	35	30110	Waiting	2020-01-01
Tony Stark	Southern Cali Express	35	30110	Waiting	2020-01-17
Tony Stark	Southern Cali Express	35	30244	Waiting	2020-01-01
Tony Stark	Southern Cali Express	35	30244	Waiting	2020-01-17
Virat Kohli	Caltrain W2weekends	35	30110	Waiting	2020-01-01
Virat Kohli	Caltrain W2weekends	35	30110	Waiting	2020-01-22
Virat Kohli	Caltrain W2weekends	35	30244	Waiting	2020-01-01
Virat Kohli	Caltrain W2weekends	35	30244	Waiting	2020-01-22

Time taken: 60.325 seconds, Fetched: 70 row(s)
hive>

10. Average ticket price of Each train (Select query in From clause)

```
SELECT tt.train_id, train_name, AVG(Ticket_price) FROM  
(SELECT t.train_id, train_name, ticket_id FROM train t  
LEFT JOIN passenger p ON t.train_id = p.train_id)  
tt  
LEFT JOIN ticket ti ON  
tt.ticket_id = ti.ticket_id  
GROUP BY tt.train_id,train_name;
```

Oracle Explain Plan

SQL | All Rows Fetched: 19 in 0.366 seconds

PLAN_TABLE_OUTPUT

1 Plan hash value: 298895792

2

3 -----

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		12	516	8 (25)	00:00:01
1	HASH GROUP BY		12	516	8 (25)	00:00:01
* 2	HASH JOIN OUTER		12	516	7 (15)	00:00:01
* 3	HASH JOIN OUTER		12	420	5 (20)	00:00:01
4	TABLE ACCESS FULL	TRAIN	7	175	2 (0)	00:00:01
5	TABLE ACCESS FULL	PASSENGER	12	120	2 (0)	00:00:01
6	TABLE ACCESS FULL	TICKET	18	144	2 (0)	00:00:01

13 -----

14

15 Predicate Information (identified by operation id):

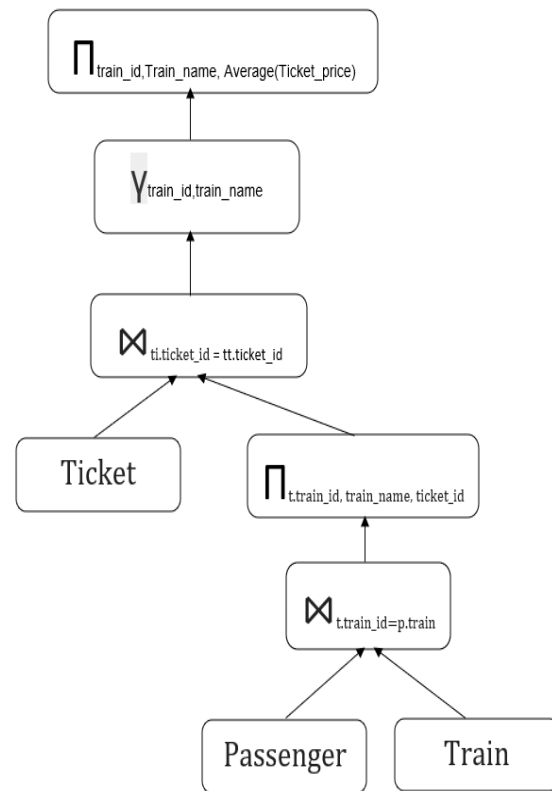
16 -----

17

18 2 - access("P"."TICKET_ID"="TI"."TICKET_ID" (+))

19 3 - access("T"."TRAIN_ID"="P"."TRAIN_ID" (+))

Proposed Query Plan



Oracle Output

[illegible]

Hive Output

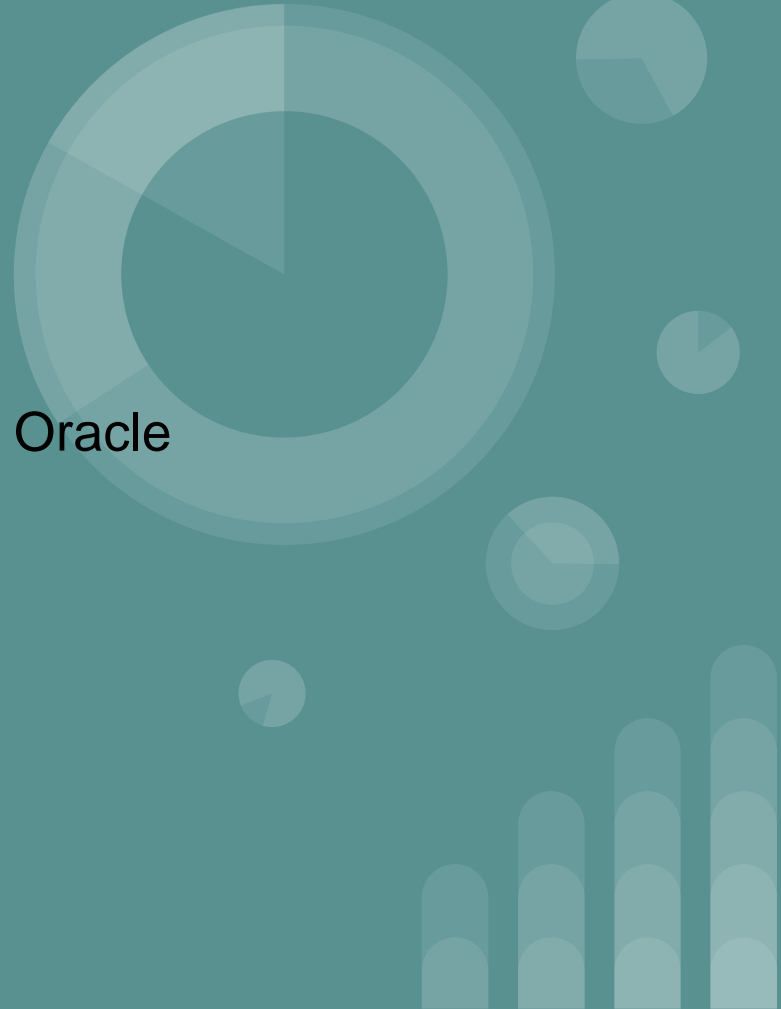
```
Execution completed successfully
MapredLocal task succeeded
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1581338931449_0790, Tracking URL = http://name1.hadoop
Kill Command = /DCNFS/applications/cdh/5.12/app/hadoop-2.6.0-cdh5.12.1/b
Hadoop job information for Stage-3: number of mappers: 1; number of redu
2020-03-09 13:11:50,495 Stage-3 map = 0%, reduce = 0%
2020-03-09 13:11:55,859 Stage-3 map = 100%, reduce = 0%, Cumulative CPU
2020-03-09 13:12:02,296 Stage-3 map = 100%, reduce = 100%, Cumulative C
MapReduce Total cumulative CPU time: 1 seconds 880 msec
Ended Job = job_1581338931449_0790
MapReduce Jobs Launched:
Stage-Stage-3: Map: 1 Reduce: 1 Cumulative CPU: 1.88 sec HDFS Read:
Total MapReduce CPU Time Spent: 1 seconds 880 msec
OK
91101 Santa Clara Express 30.0
91208 Southern Cali Express 15.0
92307 San Jose Amtrak 29.0
93461 Oakland Santa Clara Express NULL
96782 California Special 10.333333333333334
97091 Santa Crus ACE NULL
99093 Caltrain W2eekends 45.5
Time taken: 61.99 seconds, Fetched: 7 row(s)
hive>
```

Oracle Time: 0.145 secs

Hive Time: 61.99 secs

CHALLENGES FACES

- Data cleaning
- Data generation
- Comparing proposed query plan with the Oracle generated plan



CONCLUSION

- Oracle SQL is more efficient for most of our queries if compared to hive.
- Hive-Hadoop does not enforce any constraints on table/data while writing into tables but it does while reading, hence it is called schema on read.
- Oracle is highly efficient to work on small datasets and row level DML operations than Hive-Hadoop.

Q&A

