Oracle and Hive Query Project

By-
Ashvita Hadge (W1270748)
Sai Lalitha Sree Vuddamarry (W1270081)
Snehal Patil (W1159432)
Chetan Pangam (W1282207)

Description of the Dataset

- We used Yelp academic Dataset to run our queries on Oracle and Hive.
- The dataset is downloaded from https://www.yelp.com/dataset_challenge
- The size of the Dataset is 0.9 GB.
- There are 3 tables in this dataset.
  - Users (user_id, name, review_count, yelping_since, friends_count, fans, average_stars)
  - Review (review_id, date_of_review, stars, user_id, business_id, votes)
  - Business (business_id, name, city, state, stars, review_count, open, longitude, latitude)

CONFIGURATION

Oracle SQL Plus: Quad Core Processor

Oracle SQL Plus: Single Core Processor

Hadoop - Spark (v 1.5) - Hive :

Santa Clara University Design Center

40 Nodes each with 8 cores = 320 Cores

Query on Single table having predicates with special operator LIKE

1. Find every business in WI that has the word ‘Pizza‘ in its name.

```
SELECT B.BUSINESS_ID, B.NAME
   FROM BUSINESS B
WHERE B.STATE=’WI’
     AND B.NAME LIKE ‘%Pizza%’;
```

Oracle

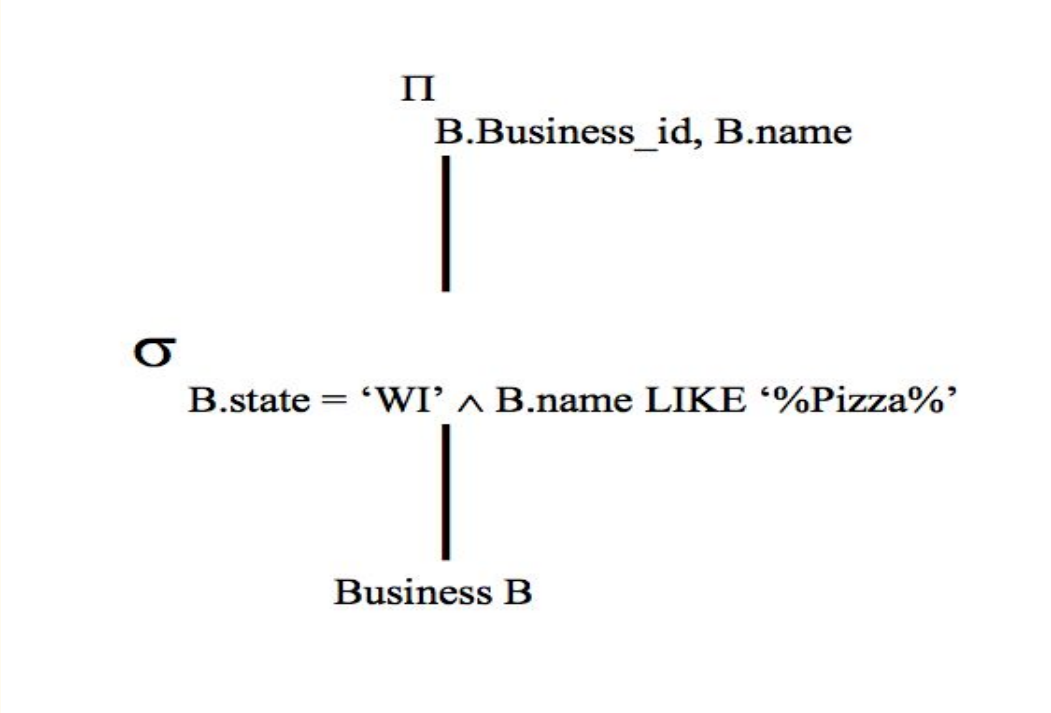
Command Prompt - sqlplus

```
SQL> select b.business_id, b.name from business b where b.state='WI' and b.name like '%Pizza%';

BUSINESS_ID                                                              NAME
------------------------------------------------------------------------ ------------------------------
------------------------------------------------------------------------------------------------------------
jqOPL979k-t_Vc049CnAjg                                                   Rosati's Authentic Chicago Pizza
xMyTUliefguc3bfetGebJg                                                   Pizza Pit
pHBQ-dQRwN3z7Rq7iSog-A                                                   Pizza Hut
brkJrGv2Eaj9NTl9g2RMgA                                                   Rocky Rococo Pan Style Pizza
QfCHvy6v39xPH7CvCASlFg                                                   Buck's Pizza
zOc8lbjViUZajbY7M0aUCQ                                                   Spartan Pizza
KTqNU4plO23583DYAMGXYg                                                   Domino's Pizza
cruBFtsFaBuhX_I72uU8pA                                                   Papa John's Pizza
OC8AUJshLVimn_-P_INvJw                                                   Pizza Pit
jo-BEjs7UYovOAupDAN_yw                                                   Rocky Rococo Pan Style Pizza
hSl4HdPdu6ohwFKJPfj93A                                                   Ian's Pizza By The Slice
fBBMlJIud_LXrTV-zfkM0g                                                   Pizza Di Roma
lWtpoG_7K7wwyvggpdIDRQ                                                   Glass Nickel Pizza
LHyB122HpkWueoWj-0bX4A                                                   Gumby's Pizza
Br9o4CNzhwUw4YUHseGA-g                                                   Buck's Pizza
qmbHSjielEKCiVilOarRmw                                                   Pizza Extreme
2WxwgDMKm8XW6grbuMsDwg                                                   Rocky Rococo Pan Style Pizza
D4KioPzFvBwb04Pf7rtCeA                                                   Pizza Pit Extreme
vK5sub1Y993EtLHVtnt9Pg                                                   Falbo Bros Pizza
OAXWfsklsATjpui2C8HilQ                                                   Topper's Pizza
1qz8Ws6tEu04X632x08pzg                                                   Rocky Rococo Pan Style Pizza
fc7QD3p7hOyewBFNOvTdOg                                                   Pizza Hut
```

Hive

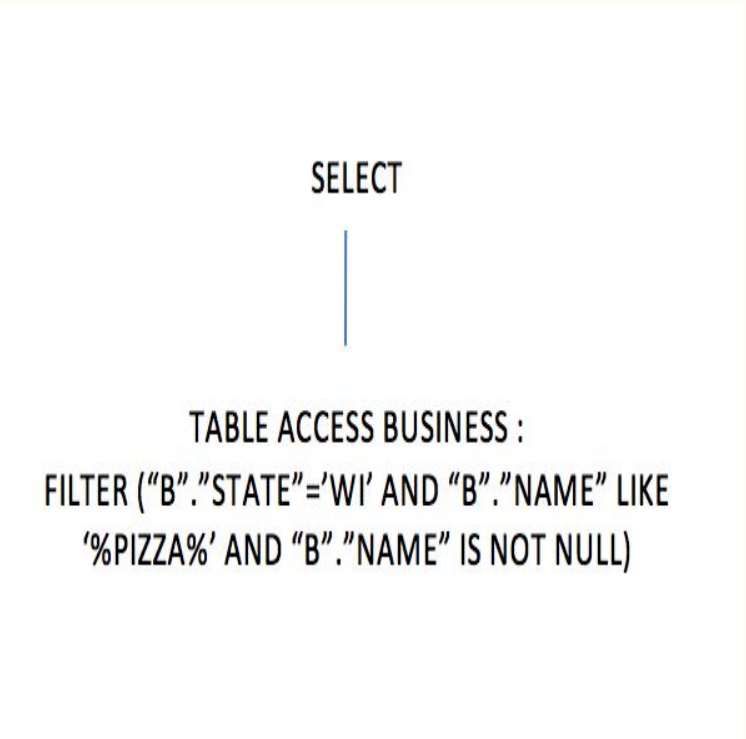
```
scala> sparkHive.sql("SELECT b.business_id, b.name FROM yelp_business b where b.state='WI' AND b.name LIKE'%Pizza%'").show()
+------------------+------------------+
|        business_id|              name|
+------------------+------------------+
|jqOPL979k-t_Vc049...|Rosati's Authenti...|
|xMyTUliefguc3bfet...|          Pizza Pit|
|pHBQ-dQRwN3z7Rq7i...|          Pizza Hut|
|brkJrGv2Eaj9NTl9g...|Rocky Rococo Pan ...|
|QfCHvy6v39xPH7CvC...|        Buck's Pizza|
|zOc8lbjViUZajbY7M...|       Spartan Pizza|
|KTqNU4plO23583DYA...|      Domino's Pizza|
|cruBFtsFaBuhX_I72...|   Papa John's Pizza|
|OC8AUJshLVimn_-P_...|          Pizza Pit|
|jo-BEjs7UYovOAupD...|Rocky Rococo Pan ...|
|hSl4HdPdu6ohwFKJP...|Ian's Pizza By Th...|
|fBBMlJIud_LXrTV-z...|       Pizza Di Roma|
|lWtpoG_7K7wwyvggp...|  Glass Nickel Pizza|
|LHyB122HpkWueoWj-...|       Gumby's Pizza|
|Br9o4CNzhwUw4YUHs...|        Buck's Pizza|
|qmbHSjielEKCiVilO...|       Pizza Extreme|
|2WxwgDMKm8XW6grbu...|Rocky Rococo Pan ...|
|D4KioPzFvBwb04Pf7...|   Pizza Pit Extreme|
|vK5sub1Y993EtLHVt...|    Falbo Bros Pizza|
|OAXWfsklsATjpui2C...|      Topper's Pizza|
+------------------+------------------+
only showing top 20 rows
```

Query Plan



Plan Table

```
SQL> explain plan for(SELECT B.BUSINESS_ID, B.NAME FROM BUSINESS B WHERE B.STATE='WI' AND B.NAME LIKE '%Pizza%');
explain plan for(SELECT B.BUSINESS_ID, B.NAME FROM BUSINESS B WHERE B.STATE='WI' AND B.NAME LIKE `%Pizza%')
                                                                                                *

ERROR at line 1:
ORA-00911: invalid character


SQL> explain plan for(SELECT B.BUSINESS_ID, B.NAME FROM BUSINESS B WHERE B.STATE='WI' AND B.NAME LIKE '%Pizza%');

Explained.

SQL> select plan_table_output from table (dbms_xplan.display());

PLAN_TABLE_OUTPUT
--------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Plan hash value: 2270958855

-------------------------------------------------------------------------------
| Id  | Operation          | Name     | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |          |    79 |  3634 |   137   (1)| 00:00:02 |
|*  1 |  TABLE ACCESS FULL| BUSINESS |    79 |  3634 |   137   (1)| 00:00:02 |
-------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

PLAN_TABLE_OUTPUT
--------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------

   1 - filter("B"."STATE"='WI' AND "B"."NAME" LIKE '%Pizza%' AND
              "B"."NAME" IS NOT NULL)

14 rows selected.
```



Execution Time of Query in a Single Core Processor: 0.98 secs

```
SQL> SELECT B.BUSINESS_ID, B.NAME FROM BUSINESS B WHERE B.STATE='WI' AND B.NAME LIKE '%Pizza%';

48 rows selected.

Elapsed: 00:00:00.98
```

Execution Time of Query in Quad Core Processor : 0.20 secs

```
SQL> SELECT B.BUSINESS_ID, B.NAME FROM BUSINESS B WHERE B.STATE='WI' AND B.NAME LIKE '%Pizza%';

48 rows selected.

Elapsed: 00:00:00.20
```

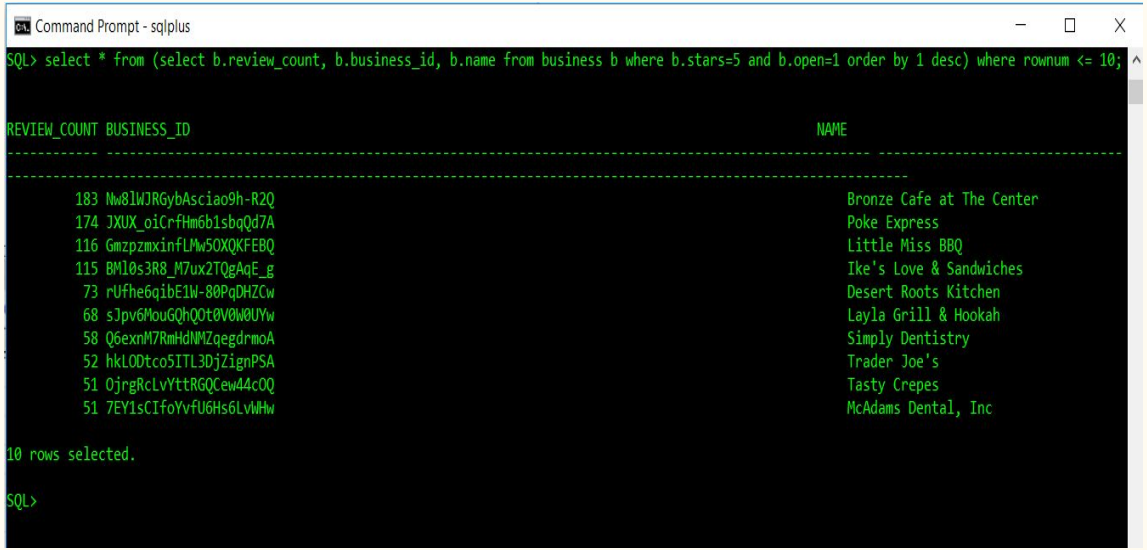
Spark Hive-Execution Time of Query : 24.40 sec

```
scala> time {
     | sparkHive.sql("SELECT B.BUSINESS_ID, B.NAME FROM YELP_BUSINESS B WHERE B.STATE='WI' AND B.NAME LIKE '%Pizza%'").show()
     | }
+--------------------+--------------------+
|         BUSINESS_ID|                NAME|
+--------------------+--------------------+
|jqOPL979k-t_Vc049...|Rosati's Authenti...|
|xMyTUliefguc3bfet...|           Pizza Pit|
|pHBQ-dQRwN3z7Rq7i...|           Pizza Hut|
|brkJrGv2Eaj9NTl9g...|Rocky Rococo Pan ...|
|QfCHvy6v39xPH7CvC...|        Buck's Pizza|
|zOc8lbjViUZajbY7M...|       Spartan Pizza|
|KTqNU4plO23583DYA...|      Domino's Pizza|
|cruBFtsFaBuhX_I72...|   Papa John's Pizza|
|OC8AUJshLVimn_-P_...|           Pizza Pit|
|jo-BEjs7UYovOAupD...|Rocky Rococo Pan ...|
|hSl4HdPdu6ohwFKJP...|Ian's Pizza By Th...|
|fBBMlJIud_LXrTV-z...|       Pizza Di Roma|
|lWtpoG_7K7wwyvggp...|  Glass Nickel Pizza|
|LHyB122HpkWueoWj-...|       Gumby's Pizza|
|Br9o4CNzhwUw4YUHs...|        Buck's Pizza|
|qmbHSjielEKCiVilO...|       Pizza Extreme|
|2WxwgDMKm8XW6grbu...|Rocky Rococo Pan ...|
|D4KioPzFvBwb04Pf7...|   Pizza Pit Extreme|
|vK5sub1Y993EtLHVt...|    Falbo Bros Pizza|
|OAXWfsklsATjpui2C...|      Topper's Pizza|
+--------------------+--------------------+
only showing top 20 rows

time: 24.407481169 sec
```

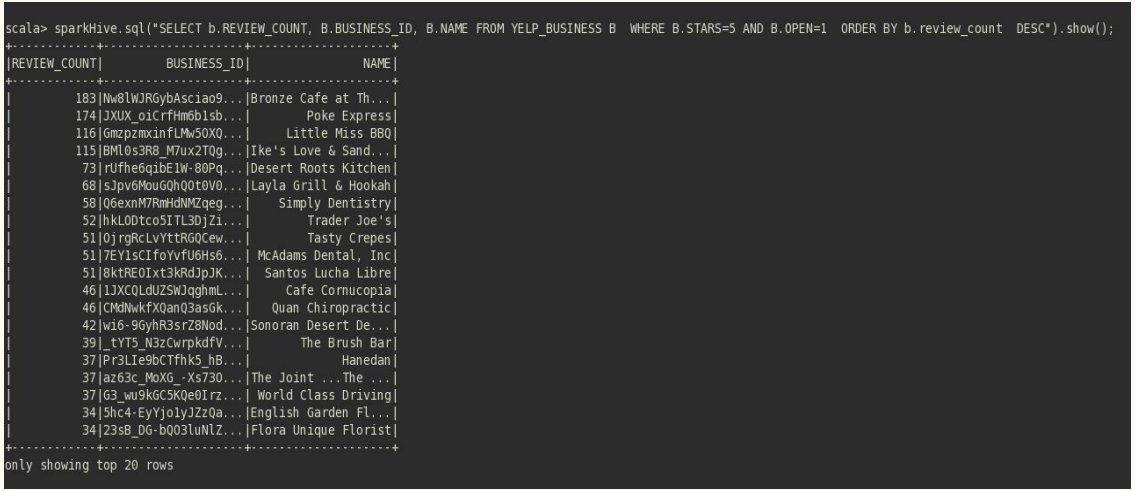
Single table query with predicates

2. Get the businesses who have a 5 star rating and are still in business. The results should be sorted by review counts in descending order. Return the top 10 businesses, the business ID, name, review count.

```
SELECT *
  FROM (SELECT B.REVIEW_COUNT, B.BUSINESS_ID, B.NAME
          FROM BUSINESS B
         WHERE B.STARS=5
           AND B.OPEN=1
         ORDER BY REVIEW_COUNT DESC)
WHERE ROWNUM <=10;
```

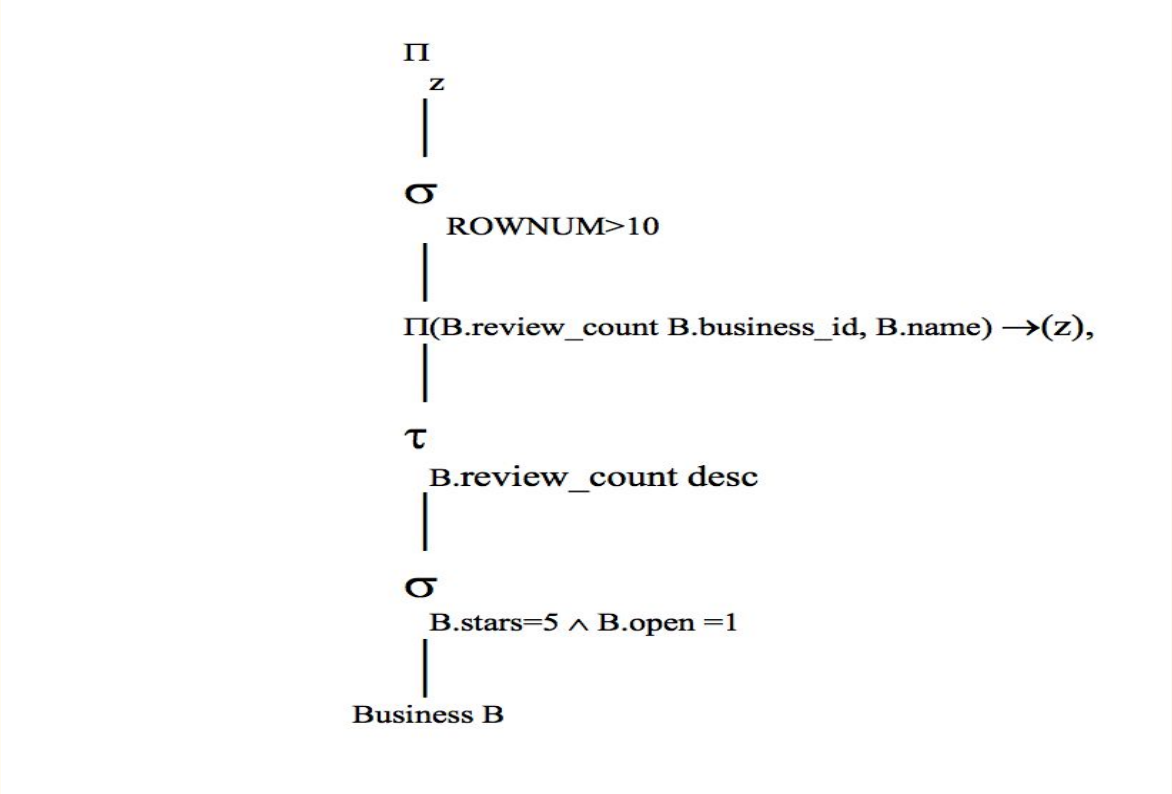

Command Prompt - sqlplus
SQL> select * from (select b.review_count, b.business_id, b.name from business b where b.stars=5 and b.open=1 order by 1 desc) where rownum <= 10;
REVIEW_COUNT BUSINESS_ID NAME
183 Nw8lWJRGybAsciao9h-R2Q Bronze Cafe at The Center
174 JXUX_oiCrfHm6b1sbqQd7A Poke Express
116 GmzpzmxinfLMw5OXQKFEBQ Little Miss BBQ
115 BMl0s3R8_M7ux2TQgAqE_g Ike's Love & Sandwiches
73 rUfhe6qibE1W-80PqDHZCw Desert Roots Kitchen
68 sJpv6MouGQhQOt0V0W0UYw Layla Grill & Hookah
58 Q6exnM7RmHdNMZqegdrmoA Simply Dentistry
52 hkLODtco5ITL3DjZignPSA Trader Joe's
51 OjrgRcLvYttRGQCew44cOQ Tasty Crepes
51 7EY1sCIfoYvfU6Hs6LvWHw McAdams Dental, Inc
10 rows selected.
SQL>

Oracle


scala> sparkHive.sql("SELECT b.REVIEW_COUNT, B.BUSINESS_ID, B.NAME FROM YELP_BUSINESS B WHERE B.STARS=5 AND B.OPEN=1 ORDER BY b.review_count DESC").show();
|REVIEW_COUNT| BUSINESS_ID| NAME|
183	Nw8lWJRGybAsciao9...	Bronze Cafe at Th...
174	JXUX_oiCrfHm6b1sb...	Poke Express
116	GmzpzmxinfLMw5OXQ...	Little Miss BBQ
115	BMl0s3R8_M7ux2TQg...	Ike's Love & Sand...
73	rUfhe6qibE1W-80Pq...	Desert Roots Kitchen
68	sJpv6MouGQhQOt0V0...	Layla Grill & Hookah
58	Q6exnM7RmHdNMZqeg...	Simply Dentistry
52	hkLODtco5ITL3DjZi...	Trader Joe's
51	OjrgRcLvYttRGQCew...	Tasty Crepes
51	7EY1sCIfoYvfU6Hs6...	McAdams Dental, Inc
51	8ktREOIxt3kRdJpJK...	Santos Lucha Libre
46	1JXCQLdUZSWJqghmL...	Cafe Cornucopia
46	CMdNwkfXQanQ3asGk...	Quan Chiropractic
42	w16-9GyhR3srZ8Nod...	Sonoran Desert De...
39	_tYT5_N3zCwrpkdfV...	The Brush Bar
37	Pr3LIe9bCTfhk5_hB...	Hanedan
37	az63c_MoXG_-Xs73O...	The Joint ...The ...
37	G3_wu9kGC5KQe0Irz...	World Class Driving
34	5hc4-EyYjo1yJZzQa...	English Garden Fl...
34	23sB_DG-bQO3luNlZ...	Flora Unique Florist
only showing top 20 rows

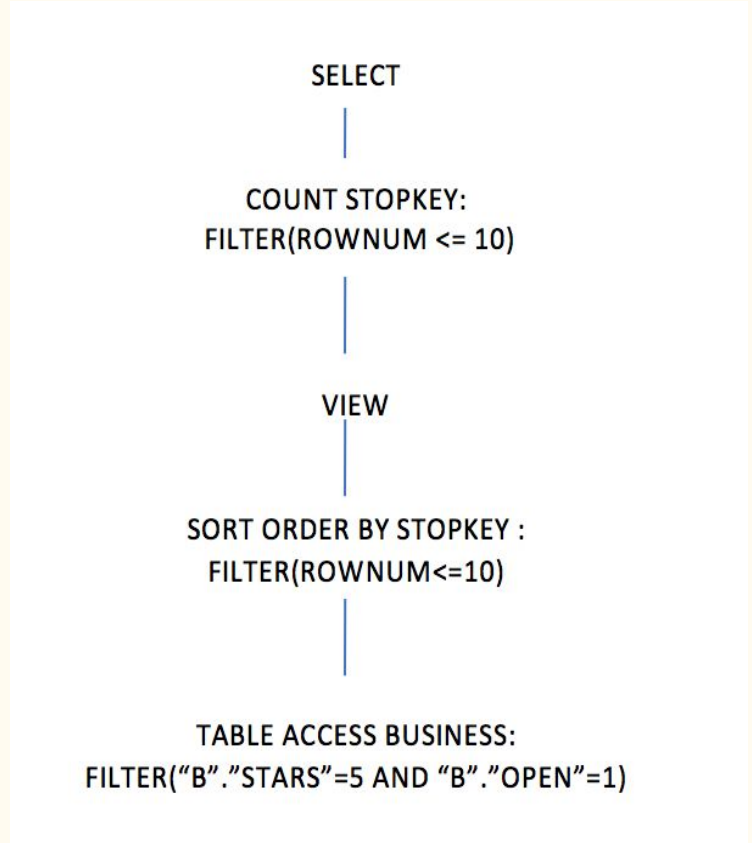
Hive

Query Plan


Π
z
σ
ROWNUM>10
Π(B.review_count B.business_id, B.name) →(z),
τ
B.review_count desc
σ
B.stars=5 ∧ B.open =1
Business B

Plan Table

```
SQL> explain plan for (SELECT * FROM (SELECT B.REVIEW_COUNT, B.BUSINESS_ID, B.NAME FROM BUSINESS B WHERE B.STARS=5 AND B.OPEN=1 ORDER BY 1 DESC) WHERE ROWNUM <=10);

Explained.

SQL> select plan_table_output from table (dbms_xplan.display());

PLAN_TABLE_OUTPUT
--------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Plan hash value: 4180548888

-------------------------------------------------------------------------------
| Id  | Operation              | Name     | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT       |          |    10 |  1420 |   138   (2)| 00:00:02 |
|* 1  |  COUNT STOPKEY         |          |       |       |            |          |
|   2 |   VIEW                 |          |  1141 |   158K|   138   (2)| 00:00:02 |
|* 3  |    SORT ORDER BY STOPKEY|         |  1141 | 60473 |   138   (2)| 00:00:02 |
|* 4  |     TABLE ACCESS FULL  | BUSINESS |  1141 | 60473 |   137   (1)| 00:00:02 |
-------------------------------------------------------------------------------

PLAN_TABLE_OUTPUT
--------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   1 - filter(ROWNUM<=10)
   3 - filter(ROWNUM<=10)
   4 - filter("B"."STARS"=5 AND "B"."OPEN"=1)

18 rows selected.
```

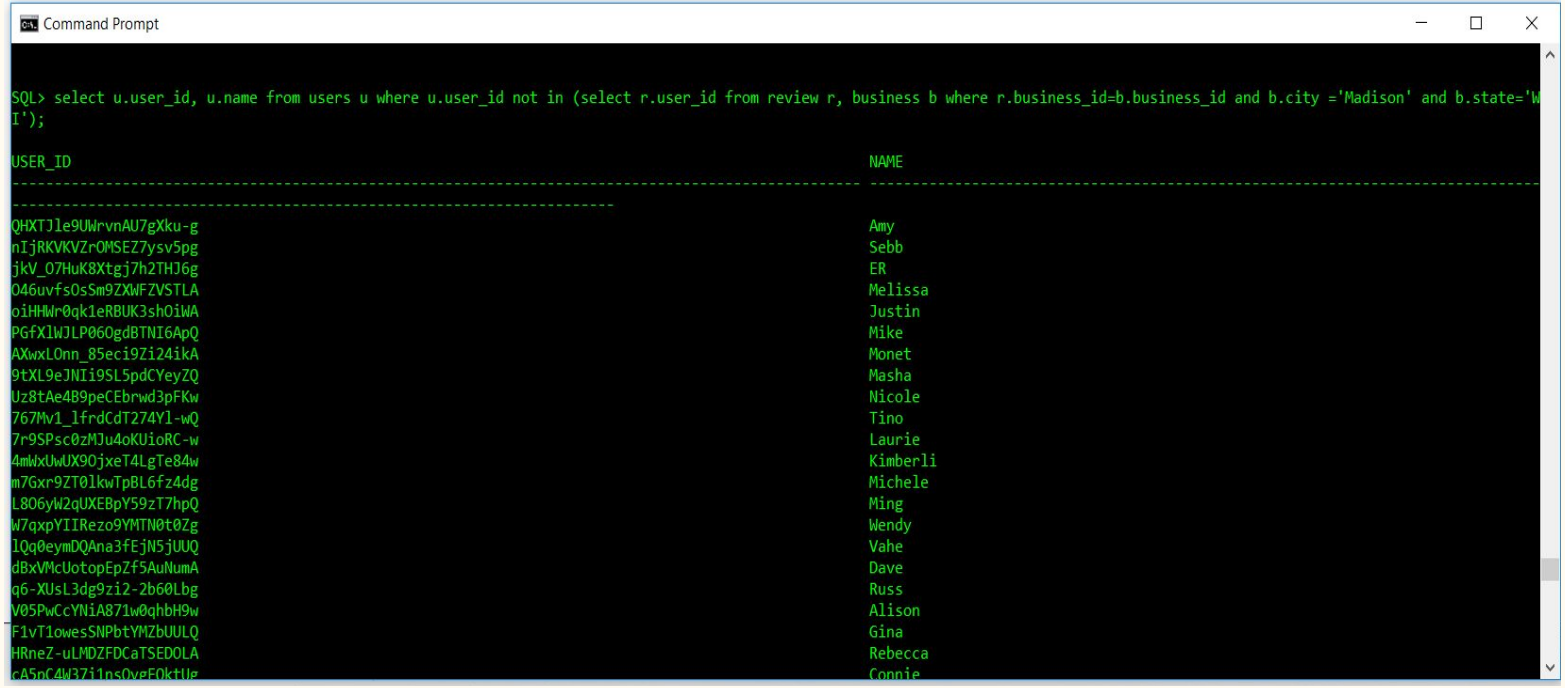


Execution Time of Single core - 0.01 secs

```
SQL> SELECT * FROM (SELECT B.REVIEW_COUNT, B.BUSINESS_ID, B.NAME FROM BUSINESS B WHERE B.STARS=5 AND B.OPEN=1 ORDER BY 1 DESC) WHERE ROWNUM <=10;

10 rows selected.

Elapsed: 00:00:00.01
```

Execution Time of Quad core - 0.00 secs

```
SQL> SELECT * FROM (SELECT B.REVIEW_COUNT, B.BUSINESS_ID, B.NAME FROM BUSINESS B WHERE B.STARS=5 AND B.OPEN=1 ORDER BY 1 DESC) WHERE ROW
NUM <=10;

10 rows selected.

Elapsed: 00:00:00.00
```



Oracle

Hive

```
scala> time {
     | sparkHive.sql("SELECT U.USER_ID, U.NAME FROM YELP_USERS U WHERE U.USER_ID NOT IN( SELECT R.USER_ID FROM YELP_REVIEW R, YELP_BUSINESS B WHERE R.BUSINESS_ID = B.BUSINESS_ID AND B.CITY ='Madison' AND B.STATE='WI') ").show()
     | }
org.apache.spark.sql.AnalysisException:
Unsupported language features in query: SELECT U.USER_ID, U.NAME FROM YELP_USERS U WHERE U.USER_ID NOT IN( SELECT R.USER_ID FROM YELP_REVIEW R, YELP_BUSINESS B WHERE R.BUSINESS_ID = B.BUSINESS_ID AND B.CITY ='Madison' AND B.STATE='WI')
TOK_QUERY 1, 0,73, 30
  TOK_FROM 1, 11,15, 30
```

Spark Hive-Execution Time of Query : 28.35 sec

```
scala> time {
     | sparkHive.sql("SELECT b.REVIEW_COUNT, B.BUSINESS_ID, B.NAME FROM YELP_BUSINESS B  WHERE B.STARS=5 AND B.OPEN=1  ORDER BY b.review_count DESC").show();
     | }
+------------+--------------------+--------------------+
|REVIEW_COUNT|         BUSINESS_ID|                NAME|
+------------+--------------------+--------------------+
|         183|Nw8lWJRGybAsciao9...|Bronze Cafe at Th...|
|         174|JXUX_oiCrfHm6b1sb...|        Poke Express|
|         116|GmzpzmxinfLMw5OXQ...|     Little Miss BBQ|
|         115|BMl0s3R8_M7ux2TQg...|Ike's Love & Sand...|
|          73|rUfhe6qibE1W-80Pq...|Desert Roots Kitchen|
|          68|sJpv6MouGQhQOt0V0...|Layla Grill & Hookah|
|          58|Q6exnM7RmHdNMZqeg...|    Simply Dentistry|
|          52|hkLODtco5ITL3DjZi...|        Trader Joe's|
|          51|OjrgRcLvYttRGQCew...|        Tasty Crepes|
|          51|7EY1sCIfoYvfU6Hs6...| McAdams Dental, Inc|
|          51|8ktREOIxt3kRdJpJK...|  Santos Lucha Libre|
|          46|1JXCQLdUZSWJqghmL...|     Cafe Cornucopia|
|          46|CMdNwkfXQanQ3asGk...|   Quan Chiropractic|
|          42|wi6-9GyhR3srZ8Nod...|Sonoran Desert De...|
|          39|_tYT5_N3zCwrpkdfV...|       The Brush Bar|
|          37|Pr3LIe9bCTfhk5_hB...|             Hanedan|
|          37|az63c_MoXG_-Xs73O...|The Joint ...The ...|
|          37|G3_wu9kGC5KQe0Irz...| World Class Driving|
|          34|5hc4-EyYjo1yJZzQa...|English Garden Fl...|
|          34|23sB_DG-bQO3luNlZ...|Flora Unique Florist|
+------------+--------------------+--------------------+
only showing top 20 rows

time: 28.352081332 sec
```

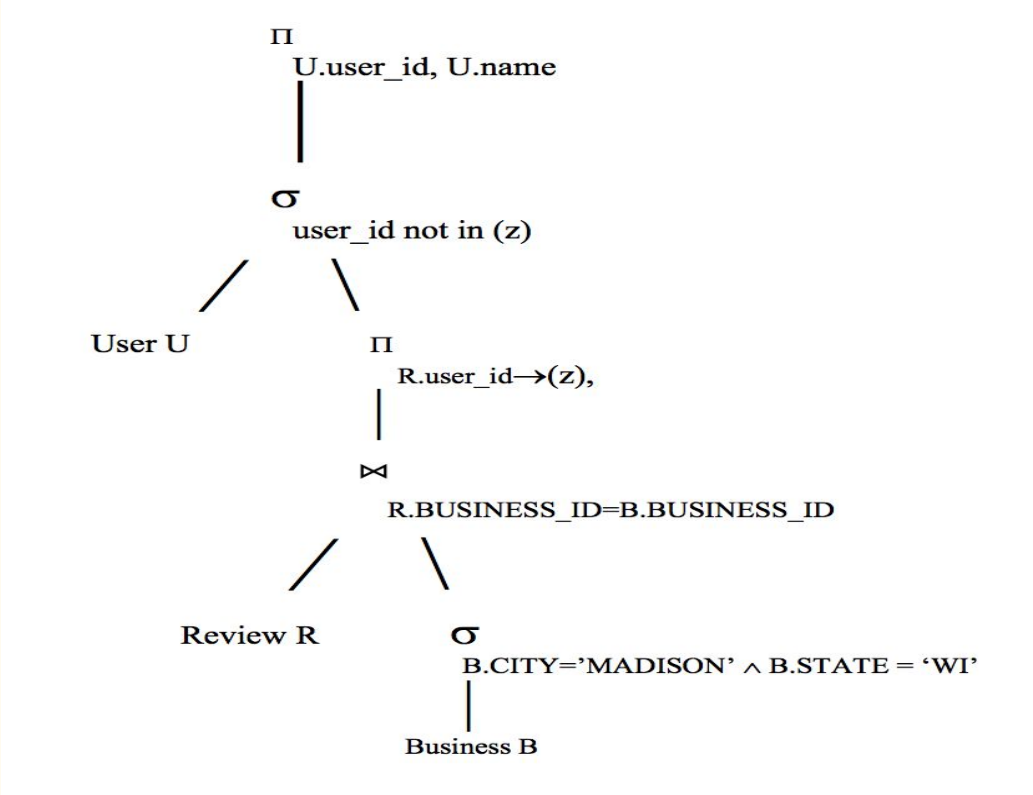
Query on Single table query having two way join SUBQUERY with NOT IN operator

3. Find the users who never reviewed any business in Madison, WI.

Ans:

```
SELECT U.USER_ID, U.NAME
    FROM USERS U
WHERE U.USER_ID NOT IN( SELECT R.USER_ID
                            FROM REVIEW R, BUSINESS B
                        WHERE R.BUSINESS_ID = B.BUSINESS_ID
                            AND B.CITY ='Madison'
                            AND B.STATE='WI') ;
```

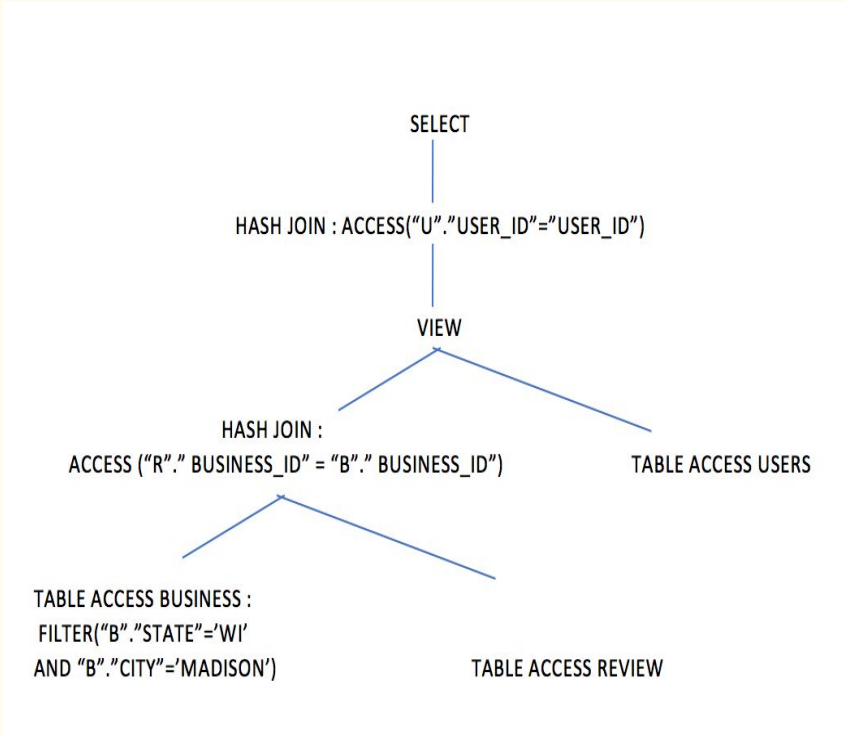
Query Plan


Π
U.user_id, U.name
σ
user_id not in (z)
User U
Π
R.user_id→(z),
⋈
R.BUSINESS_ID=B.BUSINESS_ID
Review R
σ
B.CITY='MADISON' ∧ B.STATE = 'WI'
Business B

Plan table

Command Prompt - sqlplus

```
SQL> select plan_table_output from table(dbms_xplan.display());

PLAN_TABLE_OUTPUT
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------
Plan hash value: 1074373672

-------------------------------------------------------------------------------
| Id  | Operation               | Name      | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT        |           |   211K|    16M| 27667   (1)| 00:05:33 |
|*  1 |  HASH JOIN RIGHT ANTI NA|           |   211K|    16M| 27667   (1)| 00:05:33 |
|   2 |   VIEW                  | VW_NSO_1  |     1 |    52 | 27218   (1)| 00:05:27 |
|*  3 |    HASH JOIN            |           |     1 |    83 | 27218   (1)| 00:05:27 |
|*  4 |     TABLE ACCESS FULL   | BUSINESS  |     1 |    37 |   137   (1)| 00:00:02 |
|   5 |     TABLE ACCESS FULL   | REVIEW    |   826K|    36M| 27075   (1)| 00:05:25 |

PLAN_TABLE_OUTPUT
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------
|   6 |   TABLE ACCESS FULL     | USERS     |   211K|  6181K|   447   (1)| 00:00:06 |
-------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   1 - access("U"."USER_ID"="USER_ID")
   3 - access("R"."BUSINESS_ID"="B"."BUSINESS_ID")
   4 - filter("B"."STATE"='wi' AND "B"."CITY"='Madison')

20 rows selected.

SQL>
```



Execution Time of Single core - 9.70 secs

```
SQL> select u.user_id, u.name from users u where u.user_id not in (select r.user_id from review r, business b
  2  where r.business_id = b.business_id and b.city = 'Madison' and b.state = 'WI');

203545 rows selected.

Elapsed: 00:00:09.70
```

Execution Time of Quad core - 4.10 secs

```
SQL> SELECT U.USER_ID, U.NAME
  2       FROM USERS U
  3  WHERE U.USER_ID NOT IN( SELECT R.USER_ID
  4        FROM REVIEW R, BUSINESS B
  5   WHERE R.BUSINESS_ID = B.BUSINESS_ID
  6           AND B.CITY ='Madison'
  7           AND B.STATE='WI') ;

203545 rows selected.

Elapsed: 00:00:04.10
```

Query on Two way join having predicates

4. Find the businesses which are elite businesses and reviewed between the dates “1-JAN-2006” and “31-DEC-2010” and the business is 3 star. Elite businesses are the one’s who has review count more than 30.

Ans:

```
SELECT DISTINCT B.BUSINESS_ID, B.NAME
   FROM BUSINESS B, REVIEW R
WHERE R.BUSINESS_ID=B.BUSINESS_ID
    AND B.REVIEW_COUNT >=30
    AND R.DATE_OF_REVIEW BETWEEN ‘1-JAN-2006’ AND ‘31-DEC-2016’
    AND B.STARS =3;
```

Command Prompt - sqlplus

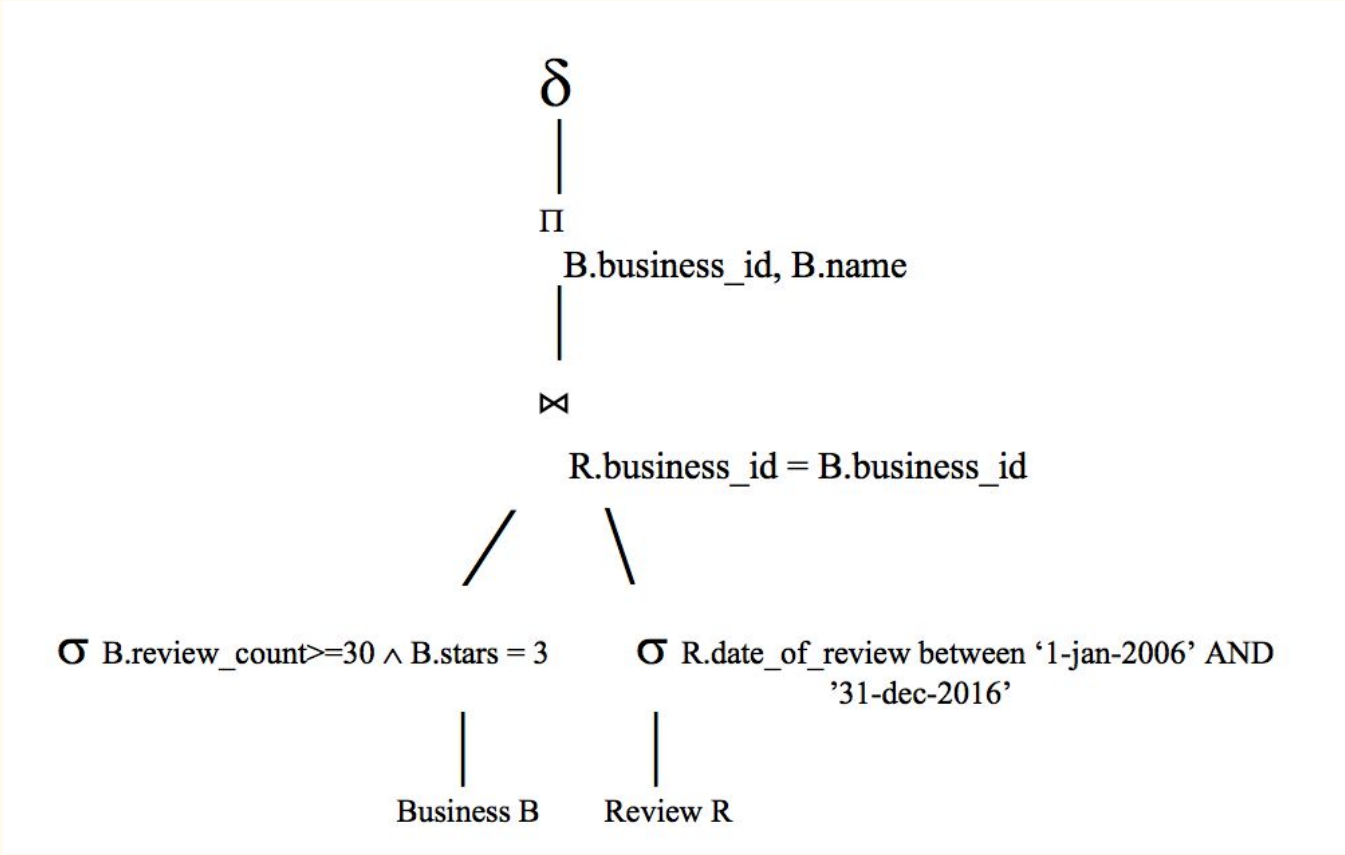
```
SQL> select distinct b.business_id, b.name from business b, review r where r.business_id=b.business_id and b.review_count >=30 and r.date_of_review bet
ween '1-jan-2006' and '31-dec-2016' and b.stars=3;

BUSINESS_ID                                                                      NAME
-------------------------------------------------------------------------------- --------------------------------------------------
_yfprBETaYgySkKyl8ZWMQ                                                           Hotel San Carlos
-lOSaCuBRAvX5JBifx-EMw                                                           Mi Amigo's Mexican Grill
jzPMl-NoB5F2fwoHX3hN8A                                                           Macayo's Mexican Kitchen
2unQHR9uaRxk3dsW6f3MGA                                                           Lao Ching Hing
3N__CAwM5VZOhRKmxMtImw                                                           Eastwind Restaurant
sh1-BzB29lIOA2qQ_PxN-A                                                           Waffle House
rHFWPJML2WlGbad0Tqevxw                                                           Ichi Ban Japanese
Q-qvDFHD3FCy82KIWoc3tA                                                           Ajo Al's
XA-mMT47rFZNE1lPBCYWQQ                                                           Chipotle Mexican Grill
0tqrgESNOpN2CNoEtCGYpQ                                                           Joe's Crab Shack
NKSA0em3dHSGTyo6Eu-Waw                                                           Indian Maharaja Palace
y6uO4ydAwBHUujfiSktxZg                                                           Arriba Mexican Grill
vgOpnyxW5d8pvIdFBXdBtg                                                           Pasta Cucina
JSURpnFu8jy4etZ_My4Erw                                                           Chipotle Mexican Grill
PWhVs3aarTbbknYGB5DgPA                                                           Aurelio's the Family Pizzeria
q28CkzJaV5yKPXysY8xewA                                                           Buffet @ Asia
jsO9c15l1KJ5ncBid3pnmA                                                           Roadrunner Saloon
b1Rosp3eIyRKgSbr6Axg1Q                                                           Paisan's
aP9wnNPiOIWCyk27ZTeRqQ                                                           Avenue Bar
m79L5WLftprDSKdXEtdrIg                                                           State Street Brats
QGZiJsrHVKFO_B3GLkph8A                                                           Top Shelf
z6vO1HR8L2iW4avr610XKw                                                           The Grapevine
7_zL7NX_rDFwhbLp98PwZg                                                           RA Sushi Bar Restaurant
YPjYXym_ELaoo8c2cdVPBA                                                           Library Bar & Grill
MwaFQkJxzqg8G5_hTip99A                                                           Joe's Crab Shack
nnPrX9mNtV5jKpo2cMjPWQ                                                           Gameworks
K9Bv1h5BQPXZFl2Q0FnXrw                                                           Red Lobster
ynK2ljcx5S1qSxouBrN5Xg                                                           Stratosphere
```

Oracle

Hive

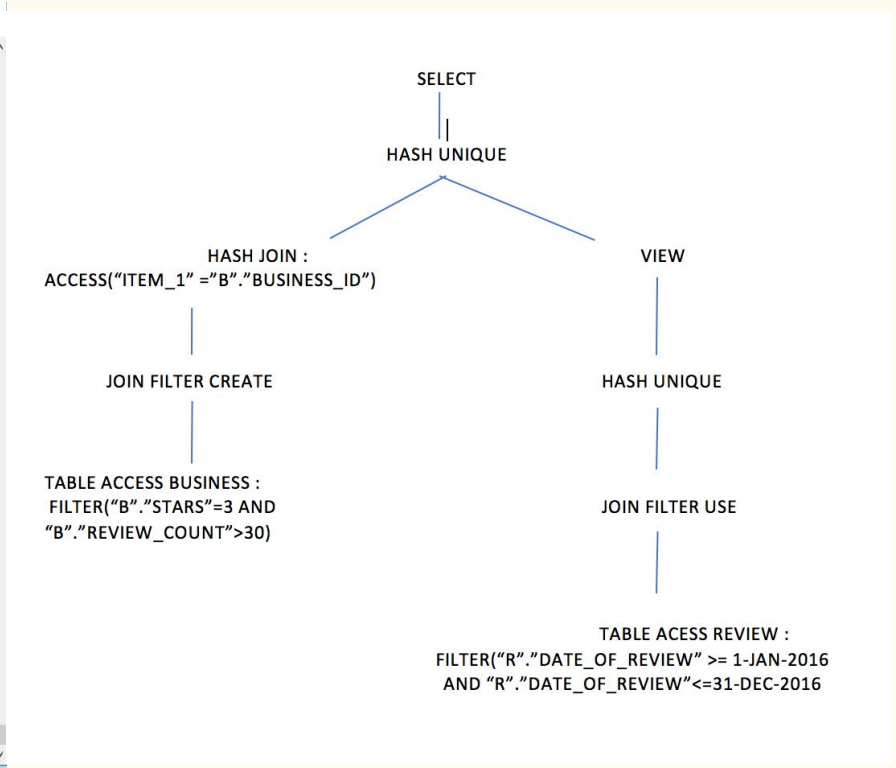
```
scala> sparkHive.sql("SELECT  distinct b.BUSINESS_ID, b.NAME FROM Yelp_BUSINESS B, Yelp_review r WHERE r.BUSINESS_ID=B.BUSINESS_ID AND b.review_count >=30 AND r.date_of_review BETWEEN '1-JAN-2006' AND '31-DEC-2016' AND B.STARS =3").show(
)
+--------------------+--------------------+
|         BUSINESS_ID|                NAME|
+--------------------+--------------------+
|B54ozdLlH5ozkwnwe...|     Paradise Bakery|
|vGeat2M50_z7Qumsq...|Scottsdale Marrio...|
|_F2DNPjsqgkAd50zu...|     Harold's Corral|
|ocCzxkuTYjUtujlML...|       Lahaina Grill|
|PSFWA3pgyB0-uE6hG...|Capital Seafood R...|
|tg1xGdWUMX02XD6AV...|   Gray's Tied House|
|VhraACjnyT90dKMCt...|Temple Bar Sports...|
|YY1UCrnuOLCTxK3mK...|           Cafe Rio|
|Q2AMxNME0cg44U5Vw...|Monteiths Restaurant|
|d6CUe9LjkFFqYtt7W...|   La Parrilla Suiza|
|NmYTwkqyaCtEg3wif...|           Wong's Jr|
|XmDiFI8SuougNUi02...|Five Guys Burgers...|
|eyT6w_RVpjRnbz4hT...|Bawarchi Indian C...|
|LbcDWyqGgQdlwWfGO...|         Barrio Cafe|
|6LNuxmnwiTvxXEDk0...|     Babystacks Cafe|
|0aKii4YS6MYZlPCsK...|Garbanzo Mediterr...|
|qVYpYB4vSeijG8JO1...|Ruth's Chris Stea...|
|Xi1AxyoKZ37PGzorO...|   Ko'sin Restaurant|
|x6nqtTyGA-jvUyy_5...|               Qdoba|
|AeMfb2FtQNnPXE9n9...|    Coach & Willie's|
+--------------------+--------------------+
only showing top 20 rows
```

Query Plan


δ
Π
B.business_id, B.name
⋈
R.business_id = B.business_id
σ B.review_count>=30 ∧ B.stars = 3
σ R.date_of_review between '1-jan-2006' AND '31-dec-2016'
Business B
Review R

Plan Table

```
Command Prompt - sqlplus
SQL> explain plan for (select distinct b.business_id, b.name from business b, review r where r.business_id=b.business_id and b.review_count >=30 and r.
date_of_review between '1-jan-2006' and '31-dec-2016' and b.stars=3);

Explained.

SQL> select plan_table_output from table(dbms_xplan.display());

PLAN_TABLE_OUTPUT
--------------------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------------------------------
Plan hash value: 2269482666

-------------------------------------------------------------------------------------------
| Id  | Operation              | Name            | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT       |                 |  2271 |   161K| 27265   (1)| 00:05:28 |
|   1 |  HASH UNIQUE           |                 |  2271 |   161K| 27265   (1)| 00:05:28 |
|*  2 |   HASH JOIN            |                 |  2271 |   161K| 27263   (1)| 00:05:28 |
|   3 |    JOIN FILTER CREATE  | :BF0000         |  2271 |   110K|   137   (1)| 00:00:02 |
|*  4 |     TABLE ACCESS FULL  | BUSINESS        |  2271 |   110K|   137   (1)| 00:00:02 |
|   5 |    VIEW                | VW_DTP_EF284C12 | 20378 |   457K| 27126   (1)| 00:05:26 |
|   6 |     HASH UNIQUE        |                 | 20378 |   616K| 27126   (1)| 00:05:26 |
|   7 |      JOIN FILTER USE   | :BF0000         |   724K|    21M| 27080   (1)| 00:05:25 |
|*  8 |       TABLE ACCESS FULL| REVIEW          |   724K|    21M| 27080   (1)| 00:05:25 |
-------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   2 - access("ITEM_1"="B"."BUSINESS_ID")
   4 - filter("B"."STARS"=3 AND "B"."REVIEW_COUNT">=30)
   8 - filter("R"."DATE_OF_REVIEW">=TO_DATE(' 2006-01-01 00:00:00', 'syyyy-mm-dd
              hh24:mi:ss') AND "R"."DATE_OF_REVIEW"<=TO_DATE(' 2016-12-31 00:00:00',
              'syyyy-mm-dd hh24:mi:ss') AND SYS_OP_BLOOM_FILTER(:BF0000,"R"."BUSINESS_ID"))

24 rows selected.

SQL>
```



Execution Time of Single core - 8.34 secs

```
SQL> select b.business_id, b.name from business b, review r where r.business_id = b.business_id
  2  and b.review_count >= 30 and r.date_of_review between '1-JAN-2006' AND '31-DEC-2016' and b.stars = 3;

79759 rows selected.

Elapsed: 00:00:08.34
```

Execution Time of Quad core - 5.01 secs

```
SQL> SELECT B.BUSINESS_ID, B.NAME
  2  FROM BUSINESS B, REVIEW R
  3  WHERE R.BUSINESS_ID=B.BUSINESS_ID
  4  AND B.REVIEW_COUNT >=30
  5  AND R.date_of_review BETWEEN '1-JAN-2006' AND '31-DEC-2016'
  6  AND B.STARS =3;

79759 rows selected.

Elapsed: 00:00:05.01
```

Spark Hive-Execution Time of Query : 30.99 sec

```
scala> time {
     | sparkHive.sql("SELECT  distinct b.BUSINESS_ID, b.NAME FROM Yelp_BUSINESS B, Yelp_review r WHERE r.BUSINESS_ID=B.BUSINESS_ID AND b.review_count >=30 AND r.date_of_review BETWEEN '1-JAN-2006' AND '31-DEC-2016' AND B.STARS =3").show(
)
     | }
+--------------------+--------------------+
|         BUSINESS_ID|                NAME|
+--------------------+--------------------+
|VhraACjnyT90dKMCt...|Temple Bar Sports...|
|YY1UCrnuOLCTxK3mK...|            Cafe Rio|
|Q2AMxNME0cg44U5Vw...|Monteiths Restaurant|
|d6CUe9LjkFFqYtt7W...|   La Parrilla Suiza|
|NmYTwkqyaCtEg3wif...|           Wong's Jr|
|XmDiFI8SuougNUi02...|Five Guys Burgers...|
|eyT6w_RVpjRnbz4hT...|Bawarchi Indian C...|
|LbcDWyqGgQdlWfGO...|         Barrio Cafe|
|6LNuxmnwiTvxXEDk0...|     Babystacks Cafe|
|0aKii4YS6MYZlPCsK...|Garbanzo Mediterr...|
|B54ozdLlH5ozkwnwe...|     Paradise Bakery|
|vGeat2M50_z7Qumsq...|Scottsdale Marrio...|
|_F2DNPjsqgkAd50zu...|     Harold's Corral|
|ocCzxkuTYjUtujlML...|       Lahaina Grill|
|PSFWA3pgyB0-uE6hG...|Capital Seafood R...|
|tg1xGdWUMX02XD6AV...|   Gray's Tied House|
|qVYpYB4vSeijG8J01...|Ruth's Chris Stea...|
|Xi1AxyoKZ37PGzorO...|   Ko'sin Restaurant|
|x6nqtTyGA-jvUyy_5...|               Qdoba|
|AeMfb2FtQNnPXE9n9...|    Coach & Willie's|
+--------------------+--------------------+
only showing top 20 rows

time: 30.992691868 sec
```

Query on Three way join with predicates on each table and order by clause

5. Give the user name & ID of top 10 users order by review count who have found the businesses in WI which have friends count greater than 100 and votes more than 10.

Ans:

```
SELECT *
   FROM (SELECT DISTINCT U.USER_ID, U.NAME, U.REVIEW_COUNT
               FROM USERS U, REVIEW R, BUSINESS B
            WHERE R.USER_ID = U.USER_ID
                AND B.BUSINESS_ID = R.BUSINESS_ID
                AND B.STATE = 'WI'
                AND U.FRIENDS_COUNT >100
                AND R.VOTES>10
             ORDER BY U.REVIEW_COUNT DESC)
WHERE ROWNUM <= 10;
```

Select Command Prompt - sqlplus

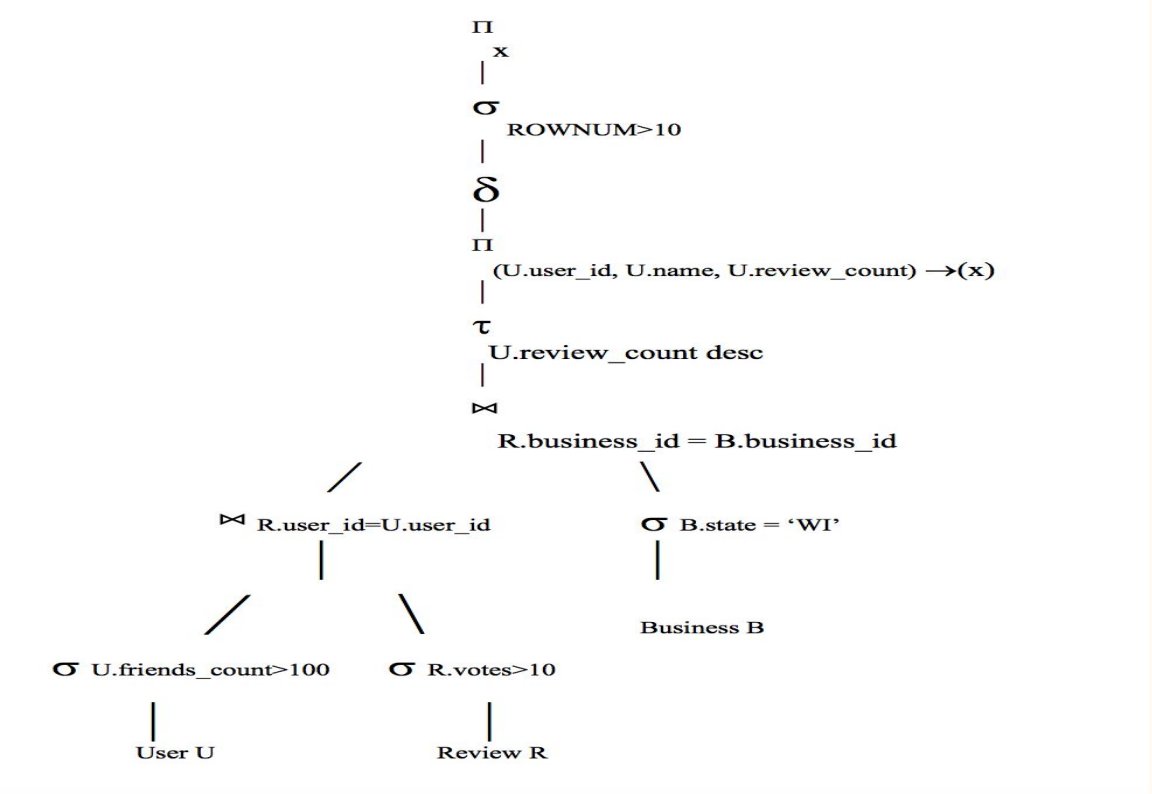
```
SQL> select * from (select distinct u.user_id, u.name, u.review_count from users u, review r, business b where r.user_id=u.user_id and b.business_id=r.business_id and b.state='WI' and u.friends_count > 100 and r.votes >10 order by u.review_count desc) where rownum<=10;

USER_ID                                                                  NAME
                                                                         REVIEW_COUNT
------------------------------------------------------------------------ ------------------------------
------------------------------------------------------------------------ ------------
xNb8pFe99ENj8BeMsCBPcQ                                                   Anthony
                                                                         2318
F9T6m1YdRFreyKDufcyoOQ                                                   Tiffany
                                                                         1836
7zpDhrRZRTGCkAh3SHbEww                                                   Candice
                                                                         1735
ID0f8slG5-Ch9TlFDrFK1Q                                                   Eli
                                                                         1289
Bvh3IHeMzSPnJAe5QlBsSw                                                   Jim
                                                                         1222
9fTTesCwn9EO-d82ODJAnw                                                   Anthony
                                                                         1093
jlpbR7IurKHCnSrGxELVlg                                                   Richie
                                                                         1093
GDi9_3Rn0N7LHHPXdJ3fFQ                                                   Daniel
                                                                         1054
qWLezzHxOXN-GQdInixZzw                                                   Carolynne
                                                                         1019
GzMh3V8oVBoII38x1GpajQ                                                   Annie
                                                                          888
```

← Oracle

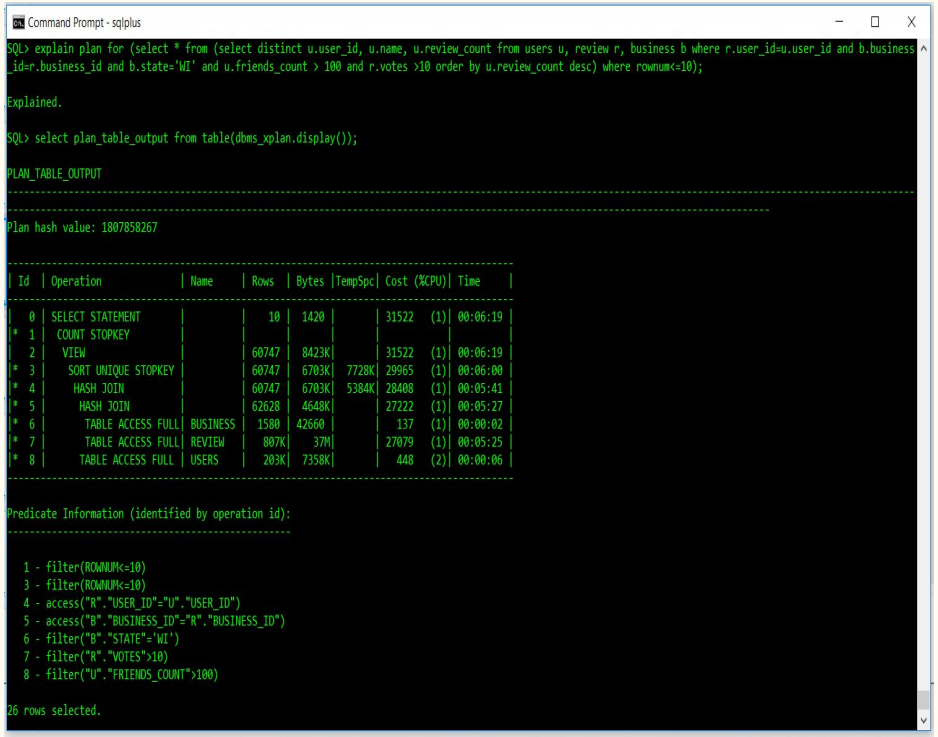
Hive ↓

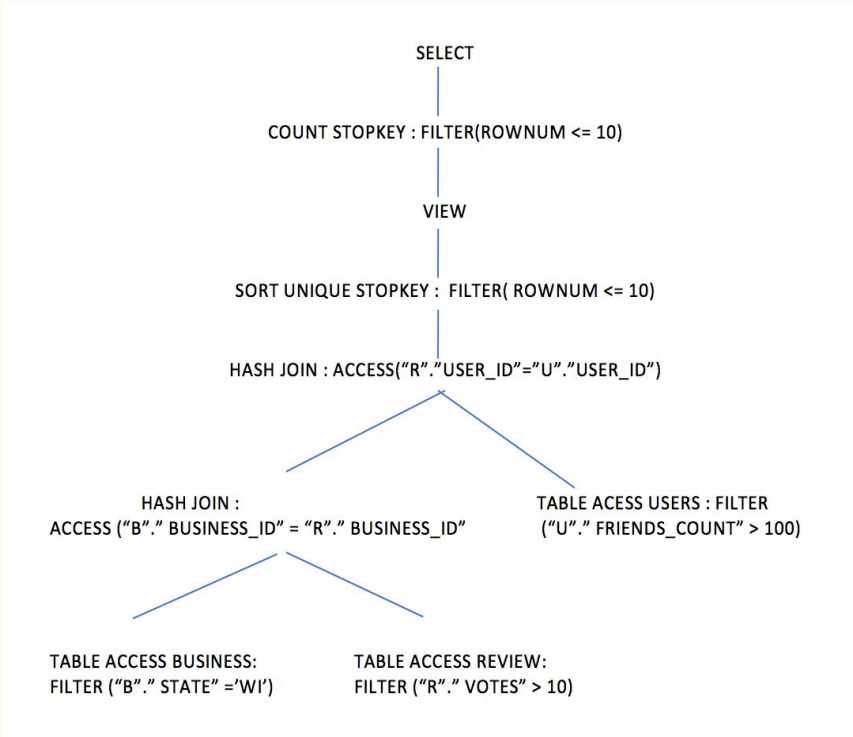
```
scala> sparkHive.sql("select distinct u.user_id,u.NAME,u.review_count from yelp_user u, yelp_review r, yelp_business b where r.user_id=u.user_id and b.business_id=r.business_id and b.state='WI' and u.FRIENDS_COUNT >100 and r.VOTES>10 order by u.review_count desc").show()
+--------------------+---------+------------+
|             user_id|     NAME|review_count|
+--------------------+---------+------------+
|xNb8pFe99ENj8BeMs...|  Anthony|        2318|
|F9T6m1YdRFreyKDuf...|  Tiffany|        1836|
|7zpDhrRZRTGCkAh3S...|  Candice|        1735|
|ID0f8slG5-Ch9TlFD...|      Eli|        1289|
|Bvh3IHeMzSPnJAe5Q...|      Jim|        1222|
|9fTTesCwn9EO-d82O...|  Anthony|        1093|
|jlpbR7IurKHCnSrGx...|   Richie|        1093|
|GDi9_3Rn0N7LHHPXd...|   Daniel|        1054|
|qWLezzHxOXN-GQdIn...|Carolynne|        1019|
|GzMh3V8oVBoII38x1...|    Annie|         888|
|KWfpXiJWFN7F_ZwqW...|    Angie|         868|
|CqGcXOmFF_ktXCXa0...| Michelle|         861|
|lC0KGXmIhyjzghBUl...|    Corey|         824|
|C5YT9IID8c-x20EXs...|     Elsa|         810|
|nEYPahVwXGD2Pjvgk...|   Rachel|         804|
|hN5DcUpA39GRux8BR...|  Jeffrey|         804|
|Q1fpW3qiL3oOe3b0s...|   Rachel|         801|
|knWADdyhOIQk5S1U5...|  Melanie|         769|
|oFhtzLhXS1iENWoBM...|    Steve|         756|
|w2w9Jocc9crXZeij3...|   Aurore|         749|
+--------------------+---------+------------+
only showing top 20 rows
```

Query Plan


Π
x
σ
ROWNUM>10
δ
Π
(U.user_id, U.name, U.review_count) →(x)
τ
U.review_count desc
⋈
R.business_id = B.business_id
⋈ R.user_id=U.user_id
σ B.state = 'WI'
Business B
σ U.friends_count>100
σ R.votes>10
User U
Review R

Plan Table


Command Prompt - sqlplus
SQL> explain plan for (select * from (select distinct u.user_id, u.name, u.review_count from users u, review r, business b where r.user_id=u.user_id and b.business_id=r.business_id and b.state='WI' and u.friends_count > 100 and r.votes >10 order by u.review_count desc) where rownum<=10);
Explained.
SQL> select plan_table_output from table(dbms_xplan.display());
PLAN_TABLE_OUTPUT
Plan hash value: 1807858267
Id	Operation	Name	Rows	Bytes	TempSpc	Cost (%CPU)	Time
0	SELECT STATEMENT		10	1420		31522 (1)	00:06:19
1	COUNT STOPKEY						
2	VIEW		60747	8423K		31522 (1)	00:06:19
3	SORT UNIQUE STOPKEY		60747	6703K	7728K	29965 (1)	00:06:00
4	HASH JOIN		60747	6703K	5384K	28408 (1)	00:05:41
5	HASH JOIN		62628	4648K		27222 (1)	00:05:27
6	TABLE ACCESS FULL	BUSINESS	1580	42660		137 (1)	00:00:02
7	TABLE ACCESS FULL	REVIEW	807K	37M		27079 (1)	00:05:25
8	TABLE ACCESS FULL	USERS	203K	7358K		448 (2)	00:00:06
Predicate Information (identified by operation id):
1 - filter(ROWNUM<=10)
3 - filter(ROWNUM<=10)
4 - access("R"."USER_ID"="U"."USER_ID")
5 - access("B"."BUSINESS_ID"="R"."BUSINESS_ID")
6 - filter("B"."STATE"='WI')
7 - filter("R"."VOTES">10)
8 - filter("U"."FRIENDS_COUNT">100)
26 rows selected.


SELECT
COUNT STOPKEY : FILTER(ROWNUM <= 10)
VIEW
SORT UNIQUE STOPKEY : FILTER(ROWNUM <= 10)
HASH JOIN : ACCESS("R"."USER_ID"="U"."USER_ID")
HASH JOIN :
ACCESS ("B"." BUSINESS_ID" = "R"." BUSINESS_ID"
TABLE ACESS USERS : FILTER
("U"." FRIENDS_COUNT" > 100)
TABLE ACCESS BUSINESS:
FILTER ("B"." STATE" ='WI')
TABLE ACCESS REVIEW:
FILTER ("R"." VOTES" > 10)

Execution Time of Single core - 8.04 secs

```
SQL> select * from ( select distinct u.user_id, u.name, u.review_count from users u, review r, business b
  2  where r.user_id = u.user_id and b.business_id = r.business_id and b.state = 'WI'
  3  and u.friends_count > 100 and r.votes > 10 order by u.review_count desc) where rownum <= 10;

10 rows selected.

Elapsed: 00:00:08.04
```

Execution Time of Quad core - 0.23 secs

```
SQL> SELECT * FROM (SELECT DISTINCT U.USER_ID, U.NAME, U.REVIEW_COUNT FROM USERS U, REVIEW R, BUSINESS B WHERE R.USER_ID = U.USER_ID AND
  2  B.BUSINESS_ID = R.BUSINESS_ID AND B.STATE = 'WI' AND U.FRIENDS_COUNT >100 AND R.VOTES>10 ORDER BY U.REVIEW_COUNT DESC) WHERE ROWNUM <= 10;

10 rows selected.

Elapsed: 00:00:00.23
```

Spark Hive-Execution Time of Query : 33.44 sec

```
scala> time {
     | sparkHive.sql("select distinct u.user_id,u.NAME,u.review_count from yelp_user u, yelp_review r, yelp_business b where r.user_id=u.user_id and b.business_id=r.business_id and b.state='WI' and u.FRIENDS_COUNT >100 and r.VOTES>10 ord
er by u.review_count desc").show()
     | }
+--------------------+---------+------------+
|             user_id|     NAME|review_count|
+--------------------+---------+------------+
|xNb8pFe99ENj8BeMs...|  Anthony|        2318|
|F9T6m1YdRFreyKDuf...|  Tiffany|        1836|
|7zpDhrRZRTGCkAh3S...|  Candice|        1735|
|ID0f8slG5-Ch9TlFD...|      Eli|        1289|
|Bvh3IHeMzSPnJAe5Q...|      Jim|        1222|
|9fTTesCwn9EO-d82O...|  Anthony|        1093|
|jlpbR7IurKHCnSrGx...|   Richie|        1093|
|GDi9_3Rn0N7LHHPXd...|   Daniel|        1054|
|qWLezzHxOXN-GQdIn...|Carolynne|        1019|
|GzMh3V8oVBoII38x1...|    Annie|         888|
|KWfpXiJWFN7F_ZwqW...|    Angie|         868|
|CqGcXOmFF_ktXCXa0...| Michelle|         861|
|lC0KGXmIhyjzghBUl...|    Corey|         824|
|C5YT9IID8c-x20EXs...|     Elsa|         810|
|nEYPahVwXGD2Pjvgk...|   Rachel|         804|
|hN5DcUpA39GRux8BR...|  Jeffrey|         804|
|Q1fpW3qiL3oOe3b0s...|   Rachel|         801|
|knWADdyhOIQk5S1U5...|  Melanie|         769|
|oFhtzLhXS1iENWoBM...|    Steve|         756|
|w2w9Jocc9crXZeij3...|   Aurore|         749|
+--------------------+---------+------------+
only showing top 20 rows

time: 33.440373858 sec
```

Query on Three way join with oracle's with in-built function EXTRACT

6. Get the 5-Star businesses in Madison , that have been reviewed by more than 10 ANCIENT users. ANCIENT users are those who have been yelping till 2010. Return business ID, name for particular business.

Ans:

```
SELECT R.BUSINESS_ID, B.NAME
   FROM BUSINESS B, USERS U, REVIEW R
WHERE B.BUSINESS_ID=R.BUSINESS_ID
    AND R.USER_ID= U.USER_ID
    AND B.CITY='Madison'
    AND R.STARS=5
    AND EXTRACT(YEAR FROM U.YELPING_SINCE) <= 2010
 GROUP BY R.BUSINESS_ID, B.NAME
HAVING COUNT(R.USER_ID)>10 ;
```

Command Prompt - sqlplus

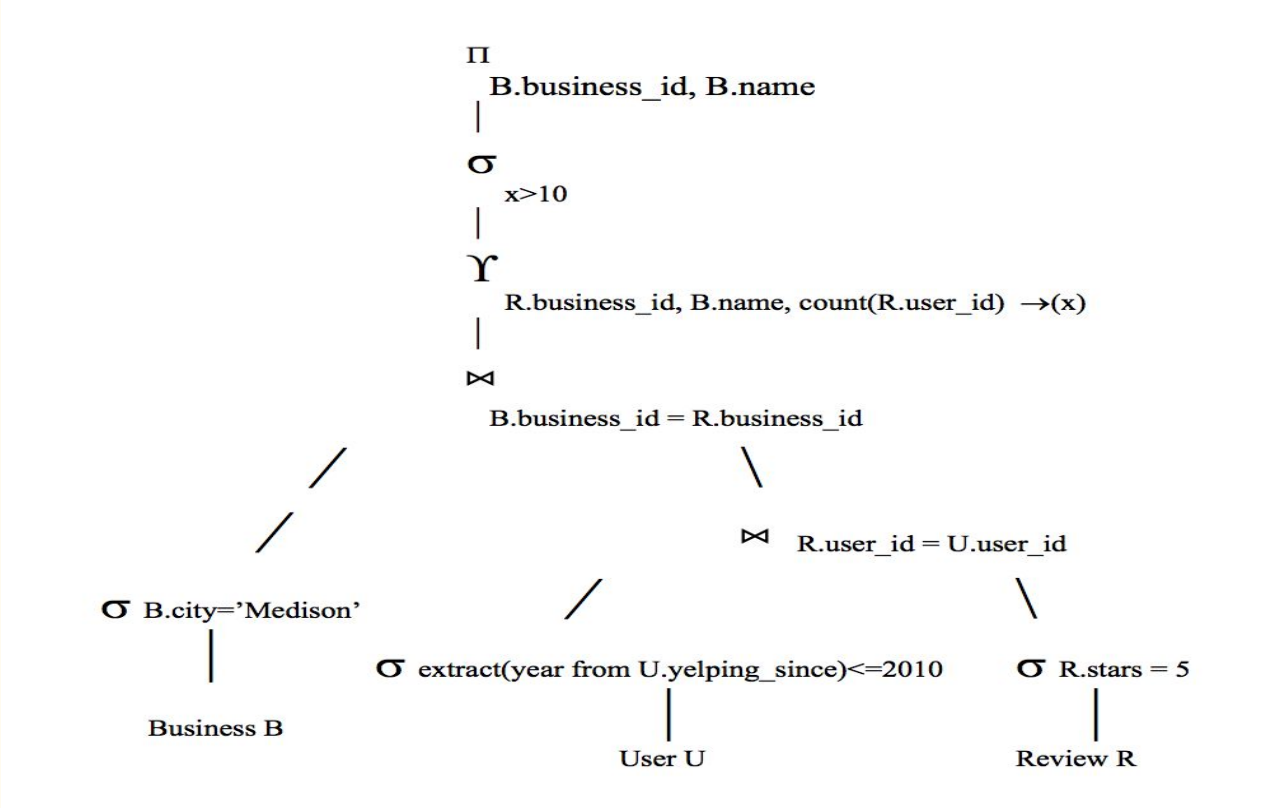
```
SQL> select r.business_id, b.name from business b, users u, review r where b.city='Madison' and r.stars = 5 and extract (year from u.yelping_since)<=2010 and b.business_
id=r.business_id and r.user_id=u.user_id group by r.business_id, b.name having count(r.user_id)>10;

BUSINESS_ID                                                                                                      NAME
---------------------------------------------------------------------------------------------------------------- ----------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------
VkQYdpbna8eTYaHZ3fAMbg                                                                                           Johnson Public House
Htwla8cEt8Lec_yJBmwlZg                                                                                           Indie Coffee
l_0JhWjpL2BesMEIDAM4Fg                                                                                           Pizza Brutta
cd5bMHS4w56rLfjcLGTobg                                                                                           Maharaja Restaurant
ka2hxeDmMCiechV3GfAJvw                                                                                           Dane County Regional Airport
0pXk2JkijBpi2Iz0D5pFXA                                                                                           Kabul Restaurant
xHJPgPO3wF-F3Ep8Vz17ng                                                                                           La Taguara
DZV4_RtUL6h1reDupspxpQ                                                                                           The Madison Concourse Hotel and Governor's Club
fEX1IcPNrhJAdwk4DgyVVw                                                                                           Tornado Steak House
fjaQ3Ixkofh8xGhklUtDnA                                                                                           Monty's Blue Plate Diner
IVPXZZXklwOc5cbZV-mLRg                                                                                           Restaurant Magnus
oGE8GPmBvHXQskP-m2cdLw                                                                                           L'Etoile
aCMhR8XTu-f8GgMHY10ipA                                                                                           Sophia's Bakery & Cafe
JQrF4iHYZkF5IBTvOQBpIw                                                                                           La Brioche True Food
fjaMSLUCjCFKdFgq5QiFkA                                                                                           Cafe Soleil
th4uGmjyjPnQpN-ukf20PA                                                                                           Ha Long Bay Vietnamese & Thai Bistro
RSus8g2uybiZVmcTK0PeIw                                                                                           Red Sushi
3SnqTPne49E5XSYv7BvYUw                                                                                           Lazy Jane's
-lM-2nuo2PWM_VbG6h1g1Q                                                                                           Kushi Bar Muramoto
r4HOnHnQWKaq0e7dlP1fRA                                                                                           DLUX
v8cTq3_Ll2E8_EvD0dCsow                                                                                           Daisy Cafe and Cupcakery
HpSBHeixIP2YxLy4L1f-HA                                                                                           Ella's Deli
vGCetjJwnPU5bqoNxFM3sw                                                                                           Opus Lounge
kIFYMeaYAQ44wQVQSO2qqA                                                                                           Fugu
rgSSCx0tb9AQyCkAvMSIoQ                                                                                           Buraka Restaurant
UIjO8lq1Lh0H2l9VfIfifQ                                                                                           Forequarter
xajae0ukR4_hguNobwEawA                                                                                           Lombardinos Italian Restaurant and Bar
YCKOz3ja7i3nbavwdf6lWQ                                                                                           Greenbush Bar
xgku6TQ_HzM-vZxaos6JWw                                                                                           Cafe Porta Alba
FHqwLwpLjUjeLP-IdG_o_w                                                                                           The Old Fashioned
Agr28_3DohGpTAD0NGS95Q                                                                                           Crema Cafe
GTAOZFZqZmis3TuM9H_t1w                                                                                           43 North
```

Oracle

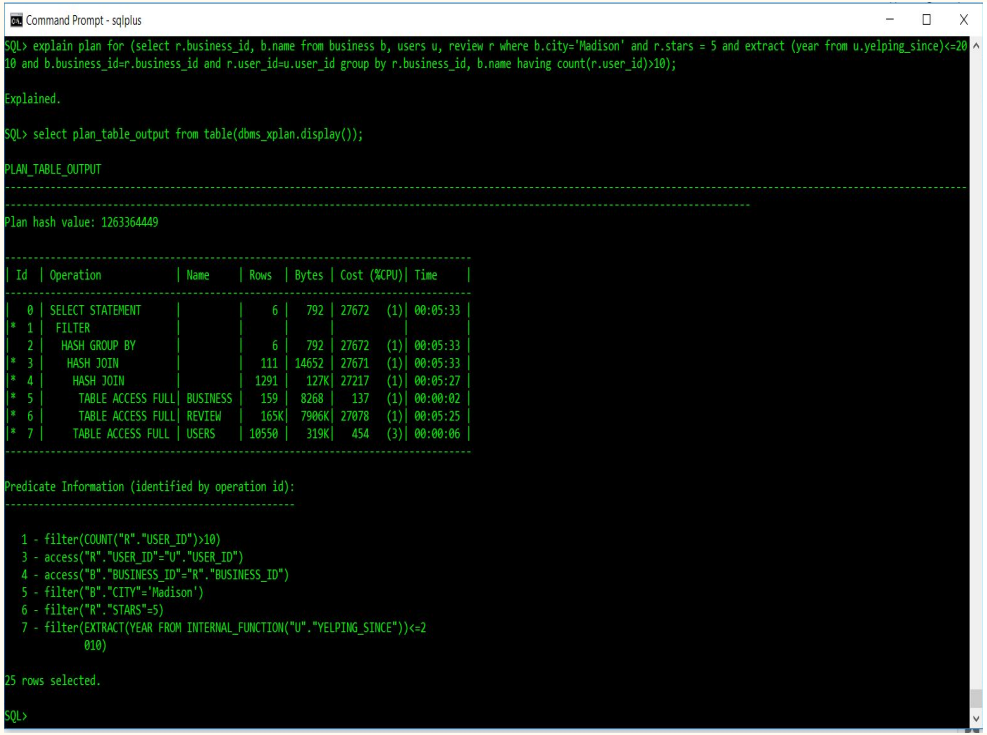
Hive

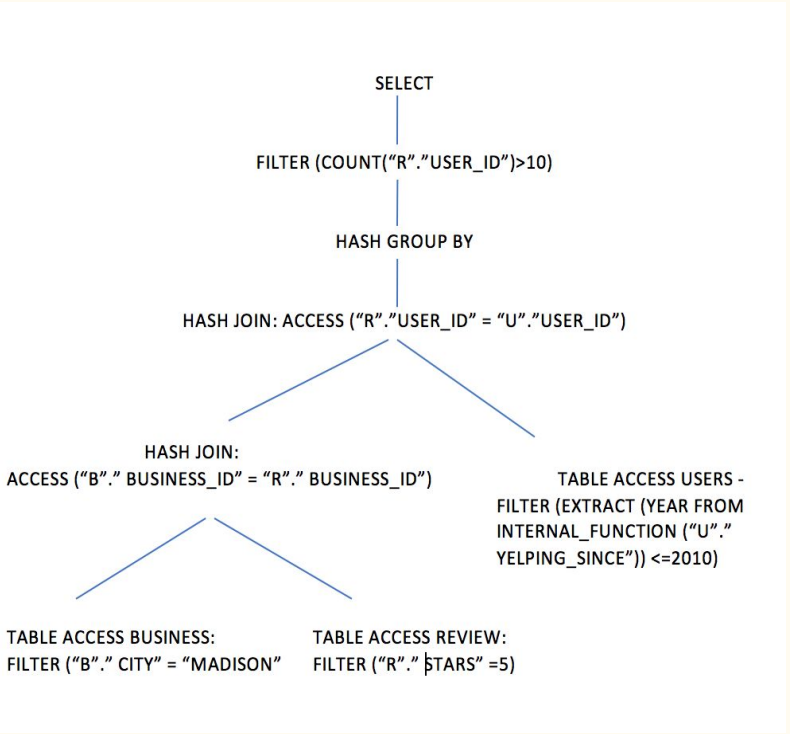
```
scala> sparkHive.sql("SELECT R.BUSINESS_ID, B.NAME FROM BUSINESS B, USERS U, REVIEW R WHERE B.BUSINESS_ID=R.BUSINESS_ID  AND R.USER_ID= U.USER_ID   AND B.CITY='Madison' AND R.STARS=5 AND EXTRACT(YEAR FROM U.YELPING_SINCE) <= 2010 GROUP B
Y R.BUSINESS_ID, B.NAME ").show()
NoViableAltException(118@[147:1: selectExpression : ( expression | tableAllColumns );])
        at org.antlr.runtime.DFA.noViableAlt(DFA.java:158)
        at org.antlr.runtime.DFA.predict(DFA.java:116)
        at org.apache.hadoop.hive.ql.parse.HiveParser_SelectClauseParser.selectExpression(HiveParser_SelectClauseParser.java:4217)
        at org.apache.hadoop.hive.ql.parse.HiveParser.selectExpression(HiveParser.java:44592)
        at org.apache.hadoop.hive.ql.parse.HiveParser_IdentifiersParser.function(HiveParser_IdentifiersParser.java:4556)
        at org.apache.hadoop.hive.ql.parse.HiveParser_IdentifiersParser.atomExpression(HiveParser_IdentifiersParser.java:6759)
        at org.apache.hadoop.hive.ql.parse.HiveParser_IdentifiersParser.precedenceFieldExpression(HiveParser_IdentifiersParser.java:6862)
        at org.apache.hadoop.hive.ql.parse.HiveParser_IdentifiersParser.precedenceUnaryPrefixExpression(HiveParser_IdentifiersParser.java:7247)
        at org.apache.hadoop.hive.ql.parse.HiveParser_IdentifiersParser.precedenceUnarySuffixExpression(HiveParser_IdentifiersParser.java:7307)
```

Query Plan


Π
B.business_id, B.name
σ
x>10
ϒ
R.business_id, B.name, count(R.user_id) →(x)
⋈
B.business_id = R.business_id
⋈
R.user_id = U.user_id
σ B.city='Medison'
σ extract(year from U.yelping_since)<=2010
σ R.stars = 5
Business B
User U
Review R

Plan Table


Command Prompt - sqlplus
SQL> explain plan for (select r.business_id, b.name from business b, users u, review r where b.city='Madison' and r.stars = 5 and extract (year from u.yelping_since)<=20
10 and b.business_id=r.business_id and r.user_id=u.user_id group by r.business_id, b.name having count(r.user_id)>10);
Explained.
SQL> select plan_table_output from table(dbms_xplan.display());
PLAN_TABLE_OUTPUT
Plan hash value: 1263364449
Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		6	792	27672 (1)	00:05:33
* 1	FILTER					
2	HASH GROUP BY		6	792	27672 (1)	00:05:33
* 3	HASH JOIN		111	14652	27671 (1)	00:05:33
* 4	HASH JOIN		1291	127K	27217 (1)	00:05:27
* 5	TABLE ACCESS FULL	BUSINESS	159	8268	137 (1)	00:00:02
* 6	TABLE ACCESS FULL	REVIEW	165K	7906K	27078 (1)	00:05:25
* 7	TABLE ACCESS FULL	USERS	10550	319K	454 (3)	00:00:06
Predicate Information (identified by operation id):
1 - filter(COUNT("R"."USER_ID")>10)
3 - access("R"."USER_ID"="U"."USER_ID")
4 - access("B"."BUSINESS_ID"="R"."BUSINESS_ID")
5 - filter("B"."CITY"='Madison')
6 - filter("R"."STARS"=5)
7 - filter(EXTRACT(YEAR FROM INTERNAL_FUNCTION("U"."YELPING_SINCE"))<=2
010)
25 rows selected.
SQL>


SELECT
FILTER (COUNT("R"."USER_ID")>10)
HASH GROUP BY
HASH JOIN: ACCESS ("R"."USER_ID" = "U"."USER_ID")
HASH JOIN:
ACCESS ("B"." BUSINESS_ID" = "R"." BUSINESS_ID")
TABLE ACCESS USERS -
FILTER (EXTRACT (YEAR FROM
INTERNAL_FUNCTION ("U"."
YELPING_SINCE")) <=2010)
TABLE ACCESS BUSINESS:
FILTER ("B"." CITY" = "MADISON"
TABLE ACCESS REVIEW:
FILTER ("R"." STARS" =5)

Execution Time of Single core - 7.95 secs

```
SQL> select r.business_id, b.name from business b, users u, review r where b.city = 'Madison' and r.stars = 5
  2  and extract(year from u.yelping_since)<=2010 and b.business_id = r.business_id and r.user_id = u.user_id
  3  group by r.business_id , b.name having count(r.user_id)>10;

94 rows selected.

Elapsed: 00:00:07.95
```

Execution Time of Quad core - 0.53 secs

```
SQL> SELECT R.BUSINESS_ID, B.NAME FROM BUSINESS B, USERS U, REVIEW R WHERE B.CITY='Madison' AND R.STARS=5
  2  AND B.BUSINESS_ID=R.BUSINESS_ID AND R.USER_ID= U.USER_ID AND EXTRACT(YEAR FROM U.YELPING_SINCE) <= 2010 GROUP BY R.BUSINESS_ID, B.NAME HAVING COUNT(R.USER_ID)>10 ;

94 rows selected.

Elapsed: 00:00:00.53
```

Query on Three way join having INLINE VIEW with group by and having clauses

7. List all “5 stars” business that have been reviewed by any user who has been yelping for 10 to 15 years.

5 star businesses are the one’s who have average rating of 5.

Ans:

```
SELECT B.BUSINESS_ID,ROUND(AVG(R.STARS)) AVG_RATING
   FROM BUSINESS B, REVIEW R, (SELECT USER_ID
                                 FROM USERS
                                WHERE ((SYSDATE-YELPING_SINCE)/365) BETWEEN 10 AND 15 ) U
WHERE R.BUSINESS_ID=B.BUSINESS_ID
    AND R.USER_ID=U.USER_ID
GROUP BY R.BUSINESS_ID
HAVING ROUND(AVG(R.STARS)) =5;
```

Oracle

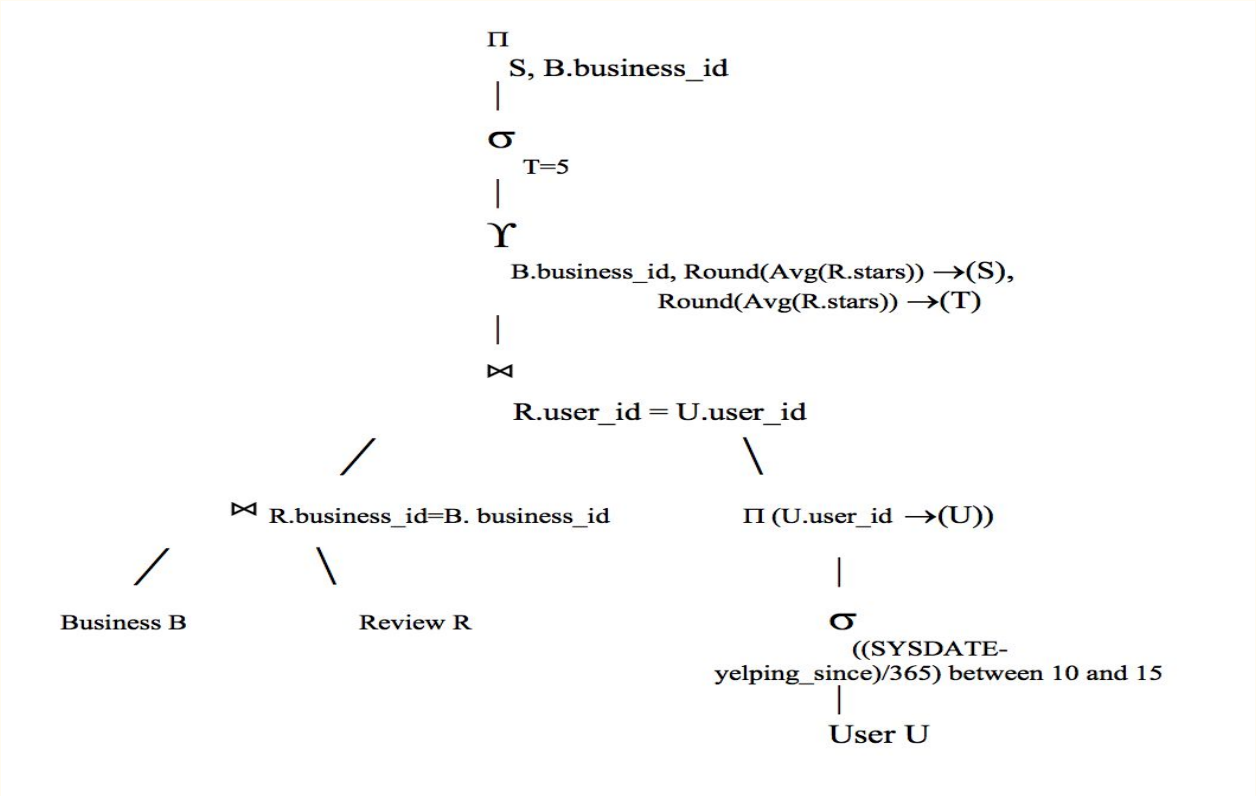
Command Prompt - sqlplus

```
SQL> select b.business_id, round(avg(r.stars)) avg_rating from business b, review r, (select user_id from users where
((sysdate-yelping_since)/365) between 10 and 15 ) U where r.business_id=b.business_id and r.user_id=u.user_id group by
 r.business_id having round(avg(r.stars))=5;

BUSINESS_ID                                                                                                AVG_RATING
---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ----------
cgCqxPhadIAlsqnD1MoWuQ                                                                                           5
MBLOaMM3g4TrLNpHUA5EtQ                                                                                           5
FrT0OQFkUZBtfoW3V1Wpfw                                                                                           5
NYBx4sdSVrN-I5Q8RfLMPg                                                                                           5
Cp6JGY5YIRncTV_My9nf9g                                                                                           5
TPyGIwybbUsOvwERNzNYnw                                                                                           5
GGhyhajTNKm5yTxtAoaxIw                                                                                           5
Sy-hmzSH1OmWpHmAfZtxRA                                                                                           5
ZPkomfeRH2pflacWf7U3-A                                                                                           5
CkQKNhlm8DOzxbXm_zb3bg                                                                                           5
0gQrhOYWdO0XVZEKtOrvaA                                                                                           5
cM_Pgssgemse4PefTNnvrw                                                                                           5
JU2tsuqG-YspQBQObhxoQA                                                                                           5
P2i4eojfX61pZn8i7RZZxA                                                                                           5
_9h-jH3YOkAU9-2A-dy6YQ                                                                                           5
j5tNIkM_UbyaiAGPqqLJcQ                                                                                           5
PEezgevlEHjujMN1qTQH1g                                                                                           5
kmiYqdQYJ5JmQQHgFdHCug                                                                                           5
frO7LlaQZBgixjClIPBeAA                                                                                           5
6SMQl2vR37HvjYWwSl1V3w                                                                                           5
uy1Ek9LC3uDGH-DHtGx00Q                                                                                           5
TQasUKgKxJGMINAXuNE63A                                                                                           5
bzSzpGAn4kjGRtsgMM1pvg                                                                                           5
hr41l2QX7OHh3k_x0Zk2Vw                                                                                           5
```

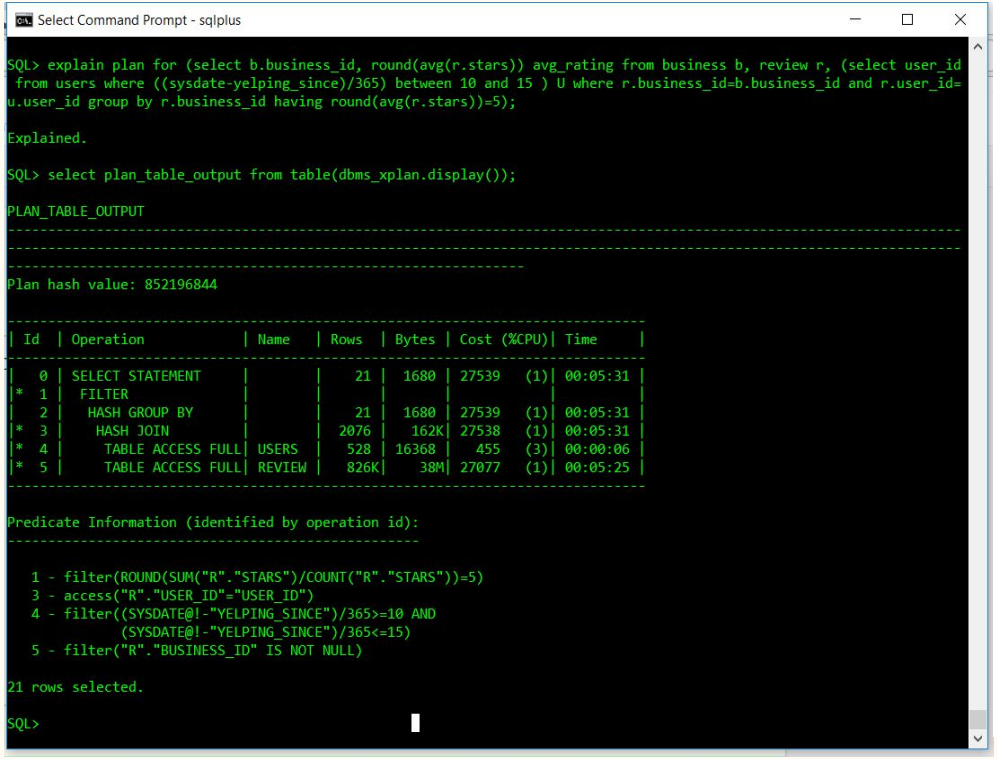
Hive

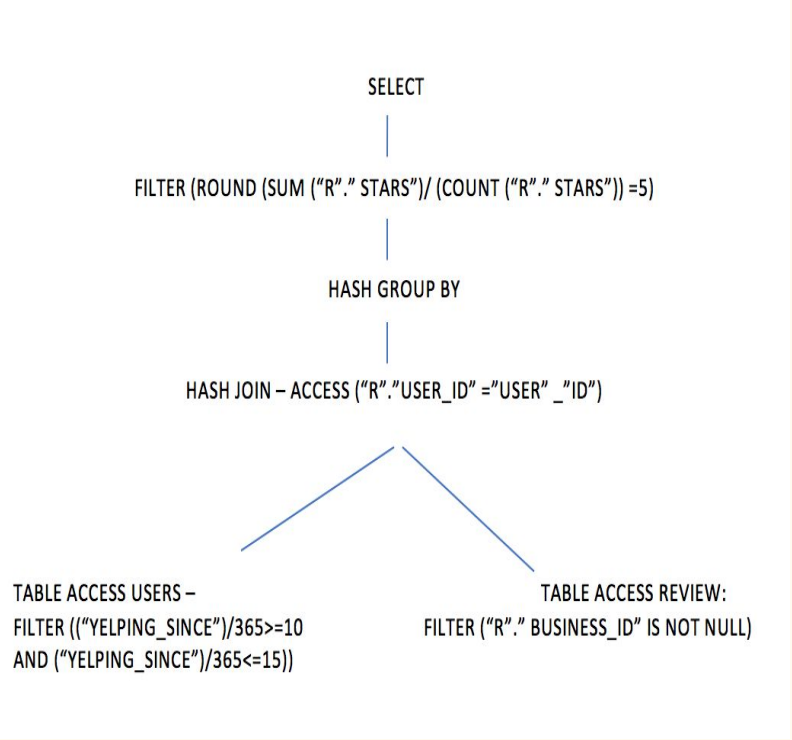
```
scala> sparkHive.sql("Select b.business_id,round(avg(r.avg)) from yelp_business b, yelp_review r,(select user_id from yelp_user where ((current_date - yelping_since)/365) between 10 and 15) u  where r.business_id=b.business_id and r.user
_id=u.user_id group by r.business_id  having round(avg(r.star))=5").show()
org.apache.spark.sql.AnalysisException: cannot resolve '(currentdate() - yelping_since)' due to data type mismatch: '(currentdate() - yelping_since)' requires (numeric or calendarinterval) type, not date; line 1 pos 129
        at org.apache.spark.sql.catalyst.analysis.package$AnalysisErrorAt.failAnalysis(package.scala:42)
        at org.apache.spark.sql.catalyst.analysis.CheckAnalysis$$anonfun$checkAnalysis$1$$anonfun$apply$2.applyOrElse(CheckAnalysis.scala:61)
        at org.apache.spark.sql.catalyst.analysis.CheckAnalysis$$anonfun$checkAnalysis$1$$anonfun$apply$2.applyOrElse(CheckAnalysis.scala:53)
        at org.apache.spark.sql.catalyst.trees.TreeNode$$anonfun$transformUp$1.apply(TreeNode.scala:293)
        at org.apache.spark.sql.catalyst.trees.TreeNode$$anonfun$transformUp$1.apply(TreeNode.scala:293)
        at org.apache.spark.sql.catalyst.trees.CurrentOrigin$.withOrigin(TreeNode.scala:51)
        at org.apache.spark.sql.catalyst.trees.TreeNode.transformUp(TreeNode.scala:292)
        at org.apache.spark.sql.catalyst.trees.TreeNode$$anonfun$5.apply(TreeNode.scala:290)
```

Query Plan


Π
S, B.business_id
σ
T=5
ϒ
B.business_id, Round(Avg(R.stars)) →(S),
Round(Avg(R.stars)) →(T)
⋈
R.user_id = U.user_id
⋈ R.business_id=B. business_id
Π (U.user_id →(U))
Business B
Review R
σ
((SYSDATE-
yelping_since)/365) between 10 and 15
User U

Plan Table


Select Command Prompt - sqlplus
SQL> explain plan for (select b.business_id, round(avg(r.stars)) avg_rating from business b, review r, (select user_id from users where ((sysdate-yelping_since)/365) between 10 and 15) U where r.business_id=b.business_id and r.user_id=u.user_id group by r.business_id having round(avg(r.stars))=5);
Explained.
SQL> select plan_table_output from table(dbms_xplan.display());
PLAN_TABLE_OUTPUT
Plan hash value: 852196844
Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		21	1680	27539 (1)	00:05:31
* 1	FILTER					
2	HASH GROUP BY		21	1680	27539 (1)	00:05:31
* 3	HASH JOIN		2076	162K	27538 (1)	00:05:31
* 4	TABLE ACCESS FULL	USERS	528	16368	455 (3)	00:00:06
* 5	TABLE ACCESS FULL	REVIEW	826K	38M	27077 (1)	00:05:25
Predicate Information (identified by operation id):
1 - filter(ROUND(SUM("R"."STARS")/COUNT("R"."STARS"))=5)
3 - access("R"."USER_ID"="USER_ID")
4 - filter((SYSDATE@!-"YELPING_SINCE")/365>=10 AND (SYSDATE@!-"YELPING_SINCE")/365<=15)
5 - filter("R"."BUSINESS_ID" IS NOT NULL)
21 rows selected.
SQL>


SELECT
FILTER (ROUND (SUM (“R”.” STARS”)/ (COUNT (“R”.” STARS”)) =5)
HASH GROUP BY
HASH JOIN – ACCESS (“R”.”USER_ID” =”USER” _”ID”)
TABLE ACCESS USERS –
FILTER ((“YELPING_SINCE”)/365>=10
AND (“YELPING_SINCE”)/365<=15))
TABLE ACCESS REVIEW:
FILTER (“R”.” BUSINESS_ID” IS NOT NULL)

Execution Time of Single core - 8.98 secs

```
SQL> SELECT B.BUSINESS_ID,ROUND(AVG(R.STARS)) AVG_RATING FROM BUSINESS B, REVIEW R, (SELECT USER_ID FROM USERS WHERE ((S
YSDATE-YELPING_SINCE)/365) BETWEEN 10 AND 15 ) U
  2  WHERE R.BUSINESS_ID=B.BUSINESS_ID AND R.USER_ID=U.USER_ID GROUP BY R.BUSINESS_ID HAVING ROUND(AVG(R.STARS)) =5;

1438 rows selected.

Elapsed: 00:00:08.98
```

Execution Time of Quad core - 8.05 secs

```
SQL> SELECT B.BUSINESS_ID,ROUND(AVG(R.STARS)) AVG_RATING FROM BUSINESS B, REVIEW R, (SELECT USER_ID FROM USERS WHERE ((SYSDATE-YELPING_SINCE)/365) BETWEEN 10 AND 15 ) U
  2  WHERE R.BUSINESS_ID=B.BUSINESS_ID AND R.USER_ID=U.USER_ID GROUP BY R.BUSINESS_ID HAVING ROUND(AVG(R.STARS)) =5;

1438 rows selected.

Elapsed: 00:00:08.50
```

Query on Two way join having GROUP BY clause and AGGREGATE function

8. Query on find the average rating across all reviews written by a particular user.

Ans:

```
SELECT AVG(R1.STARS) AS AVG_RATING,U.USER_ID AS REVIEW_USER
   FROM USERS U, REVIEW R1
WHERE R1.USER_ID=U.USER_ID
GROUP BY U.USER_ID
ORDER BY AVG(R1.STARS) DESC;
```

Command Prompt - sqlplus

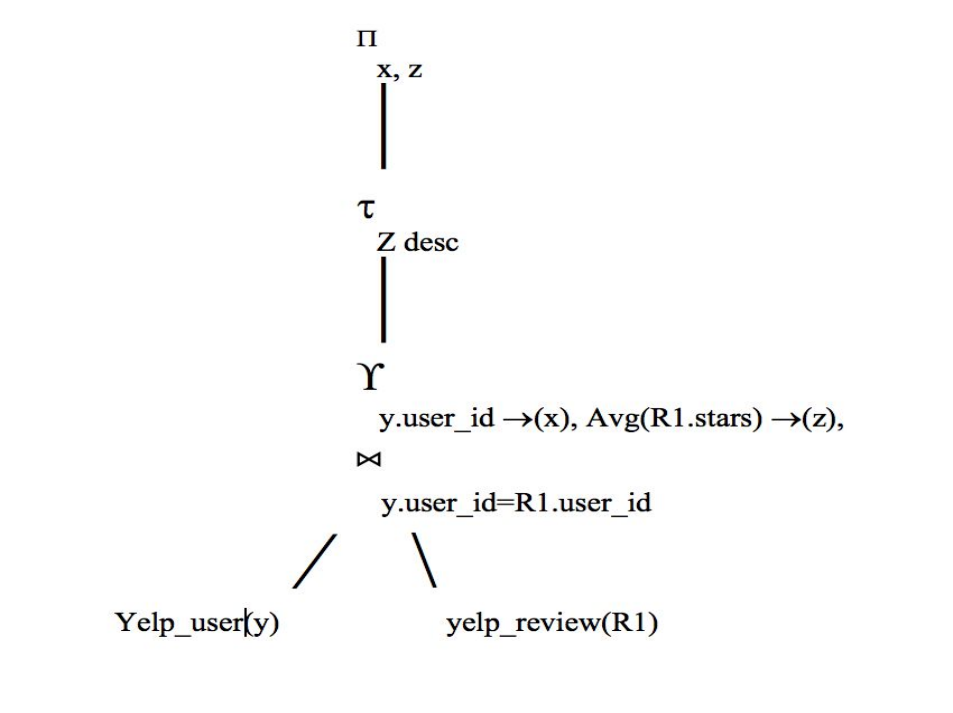
```
SQL> select avg(r1.stars) as avg_rating, u.user_id as review_user from users u, review r1 where r1.user_id=u.user_id group by u.user_i
d order by avg(r1.stars) desc;

AVG_RATING REVIEW_USER
---------- ----------------------------------------------------------------------------------------------------
         5 5C0646lN2A88IYer-KewQA
         5 yK2lI82Va7jWcp7-AEGNQA
         5 hNSBSa6ZCndHcthNGndkGg
         5 n82J25ep8Q-GBN12Mkc6xw
         5 ApgbgjqFPCGnkiR0jwBMfQ
         5 CtZf_MOoDx5MUPMylAswRg
         5 xBhkL-DTwmfqUfwWEg1nfQ
         5 KNgG3Ldg6g-Nj0dgQ9j7hA
         5 XpCad7cdor-F2BkgGNld6A
         5 6nTT0ueMIPJ1jPmdN9JFPA
         5 _sQdV_RmGfkm3IW1uhwtgA
         5 BHsPv8pElh9zFgnsIDAYzQ
         5 ZjIS-fUfaFC-mL81pBGqbQ
         5 J5XFTWCDpFTG7Fn-JAd_wA
         5 BdvzXubYzaYhDEyRtEwX8g
         5 aidabdi5I_wErGkr1eEPsg
         5 -fia27fQ9Qwk3IsFwu0EpQ
         5 u9XAV7qxaOCMIlXjBV3K5g
         5 Bt9J7fHTnZuPhmBOD6xjiQ
         5 zNe9mqSVGnO6rhOZGHyWYA
         5 aDeWL749QPMdSyZiw2pJmw
         5 a0t2qdjEGbeOfbAZNQcIkA
         5 BYocB37x4HDB4u5mp30yqg
         5 TzE2ktbwHRezlkYU5Fr9bg
```

Oracle

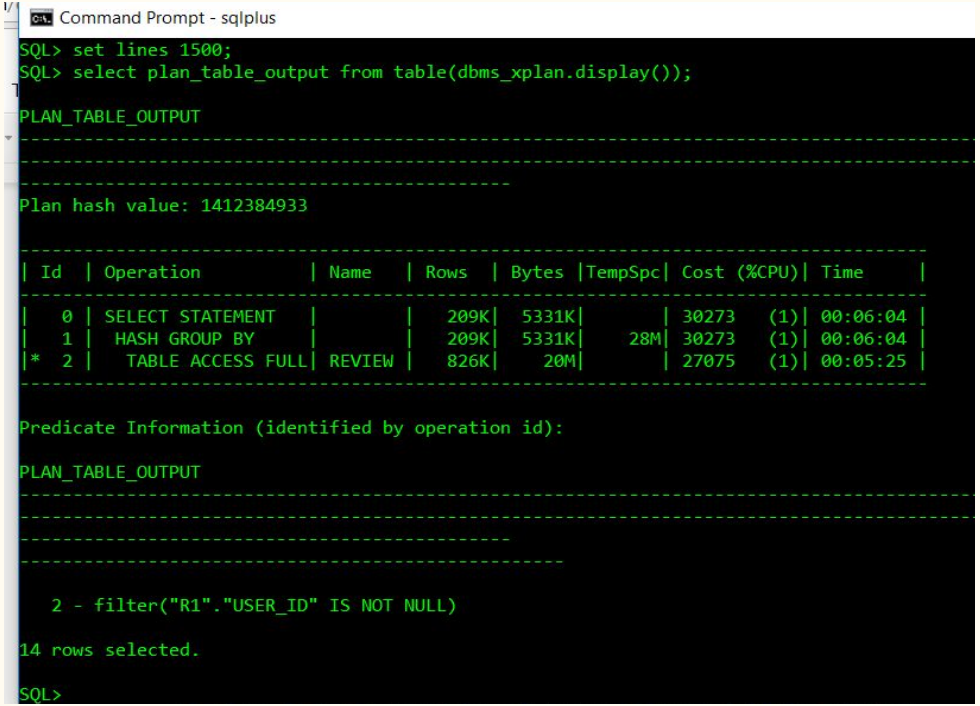
```
scala> sparkHive.sql("SELECT avg(r1.stars),y.user_id from yelp_user y, yelp_review r1 where r1.user_id = y.user_id group by (y.user_id) order by (avg(r1.stars)) desc").show()
+---+--------------------+
|_c0|             user_id|
+---+--------------------+
|5.0|29wH8TzbIrxJghgnP...|
|5.0|2CBkfGj6_I46WTp9v...|
|5.0|2DvQAyLPIX8sa4Twk...|
|5.0|6AzKyBQAJoFcqCJ6r...|
|5.0|-AWOPGwfLbC3x5Zap...|
|5.0|-F8zttwaxu7vlGPM6...|
|5.0|-cacOnSayh_PL-7bC...|
|5.0|-fkyRGlayuZ8gHMi5...|
|5.0|3CjK2kV0FMRThm0zS...|
|5.0|00uXtdqcIqXOVdK7g...|
|5.0|3TFWP6yVAaT1F55PS...|
|5.0|0KQr7T0FrT_PozYjh...|
|5.0|3lIU1IyTH95mi2QlJ...|
|5.0|0wekFc0vMC9EfbRuc...|
|5.0|3rTR2IS7GSj4d2tW0...|
|5.0|-1U7FLmptAHm1B1av...|
|5.0|41SJmKpGYe-JW_VwN...|
|5.0|59VVVF8Y9UD1tEpXD...|
|5.0|4Ia-vVMfyuhNAfOBA...|
|5.0|0bUUw5s1V49OnU0cM...|
+---+--------------------+
only showing top 20 rows
```

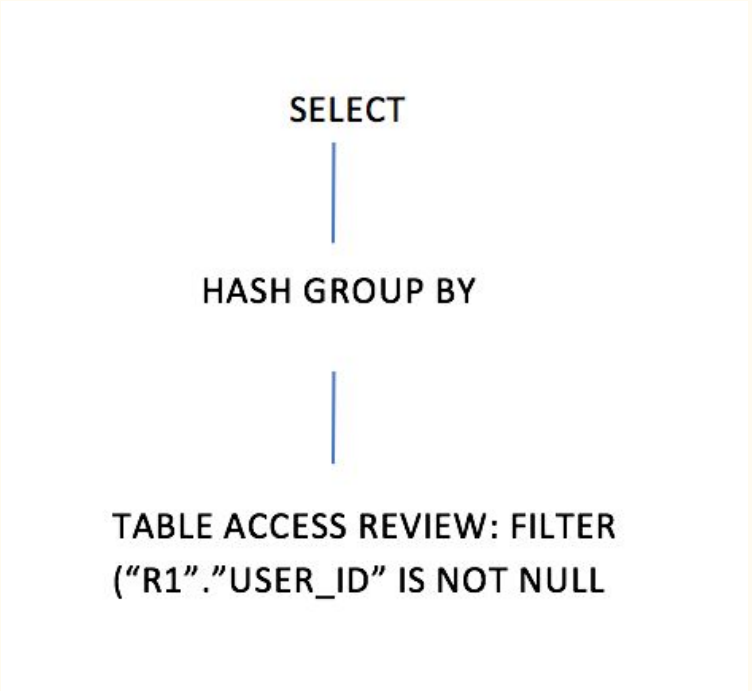
Hive

Query Plan


Π
x, z
τ
Z desc
ϒ
y.user_id →(x), Avg(R1.stars) →(z),
⋈
y.user_id=R1.user_id
Yelp_user(y)
yelp_review(R1)

Plan Table


Command Prompt - sqlplus
SQL> set lines 1500;
SQL> select plan_table_output from table(dbms_xplan.display());
PLAN_TABLE_OUTPUT
Plan hash value: 1412384933
Id	Operation	Name	Rows	Bytes	TempSpc	Cost (%CPU)	Time
0	SELECT STATEMENT		209K	5331K		30273 (1)	00:06:04
1	HASH GROUP BY		209K	5331K	28M	30273 (1)	00:06:04
* 2	TABLE ACCESS FULL	REVIEW	826K	20M		27075 (1)	00:05:25
Predicate Information (identified by operation id):
PLAN_TABLE_OUTPUT
2 - filter("R1"."USER_ID" IS NOT NULL)
14 rows selected.
SQL>


SELECT
HASH GROUP BY
TABLE ACCESS REVIEW: FILTER
("R1"."USER_ID" IS NOT NULL

Execution Time of Single core - 8.61secs

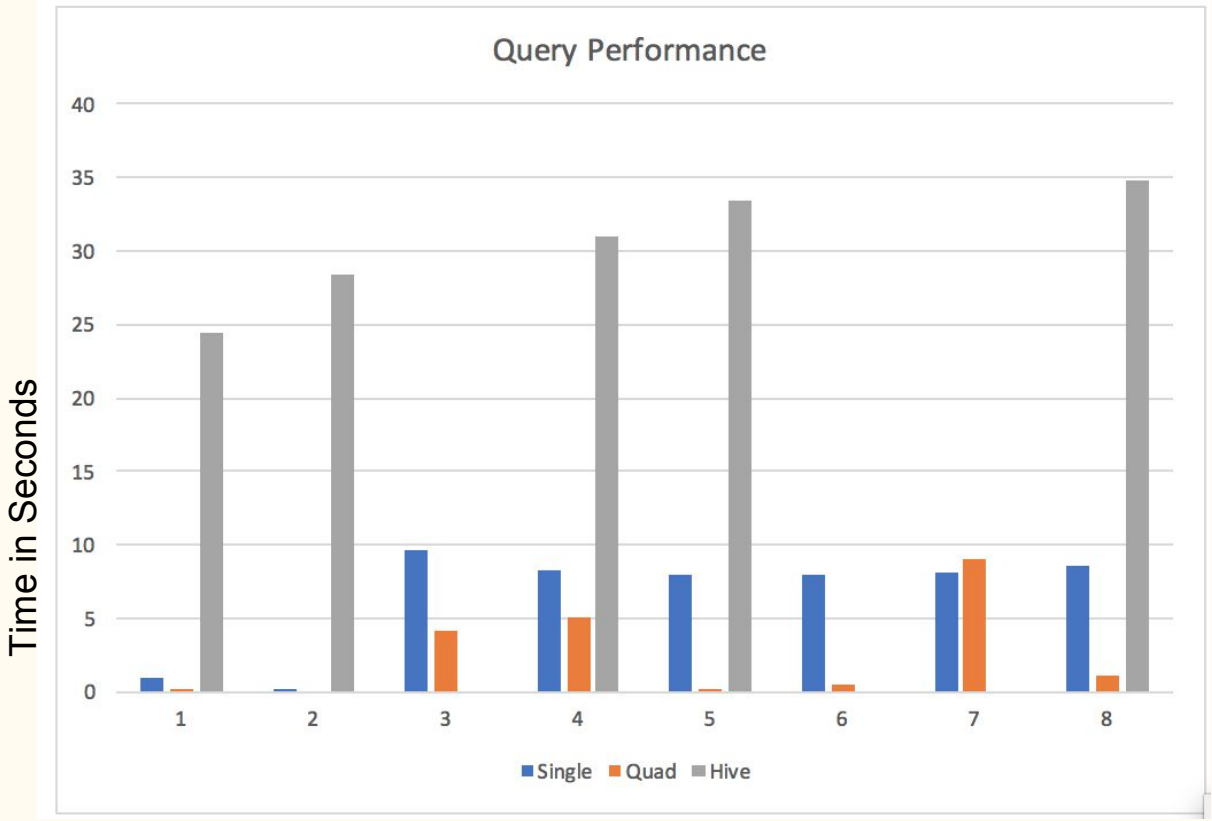
```
SQL> select avg(r1.stars) as avg_rating, u.user_id as review_user from users u,
  2  review r1 where r1.user_id=u.user_id group by u.user_id order by avg(r1.stars)
  3  desc;

210966 rows selected.

Elapsed: 00:00:08.61
```

Execution Time of Quad core - 1.14 secs

```
SQL> SELECT AVG(R1.stars) AS AVG_RATING,u.user_id AS REVIEW_USER FROM USERS u, REVIEW R1
  2  WHERE R1.user_id=u.user_id GROUP BY u.user_ID ORDER BY AVG(R1.stars) DESC;

210966 rows selected.

Elapsed: 00:00:01.14
```

Spark Hive-Execution Time of Query : 34.79 sec

```
scala> time {
     | sparkHive.sql("SELECT AVG(R1.stars) AS AVG_RATING,u.user_id AS REVIEW_USER FROM YELP_USER u, Yelp_REVIEW R1 WHERE R1.user_id=u.user_id GROUP BY u.user_ID ORDER BY AVG(R1.stars) DESC").show()
     | }
+----------+--------------------+
|AVG_RATING|         REVIEW_USER|
+----------+--------------------+
|       5.0|23iddgKmhtImswSQV...|
|       5.0|2R5bHjfrjXOc7lnBU...|
|       5.0|4hDpu3ILIgzBK8ZV1...|
|       5.0|2VolxqB_H5m-pLwxe...|
|       5.0|-NxMj8-GuWTom3LTb...|
|       5.0|2WALGYpaEdsHOluKb...|
|       5.0|-Z7LF1NQYMpprOljw...|
|       5.0|-Qq59ncADTtn9l9_j...|
|       5.0|-ZKGlp5bczSA5iI-M...|
|       5.0|0DTLj7xNH2Y-4kMWW...|
|       5.0|1tUHmcTXgJ9o4JUTn...|
|       5.0|0pe2Qw8nbuDLD_ZHI...|
|       5.0|37BJGmBKDWlh8SEDQ...|
|       5.0|1VVouWm0WXYPMhct-...|
|       5.0|3Lola6l2ixxfhOcK8...|
|       5.0|-0yauhuaZY_eyqg_M...|
|       5.0|3NYaWWFzUDn1pZRZz...|
|       5.0|-KKQ2qoOvG0aHLf9D...|
|       5.0|3ai9os2lH-1aDLNds...|
|       5.0|0TexIhmvZexai28gI...|
+----------+--------------------+
only showing top 20 rows

time: 34.793607099 sec
```


Query Performance
Time in Seconds
40
35
30
25
20
15
10
5
0
1
2
3
4
5
6
7
8
Single
Quad
Hive

CONCLUSION

- The time taken to execute the queries on Quad core processor is less than that on single core processor. The time taken to run the queries on Hive is more.

- 5 out of 8 queries executed successfully on Hive.

The End

Thank you!