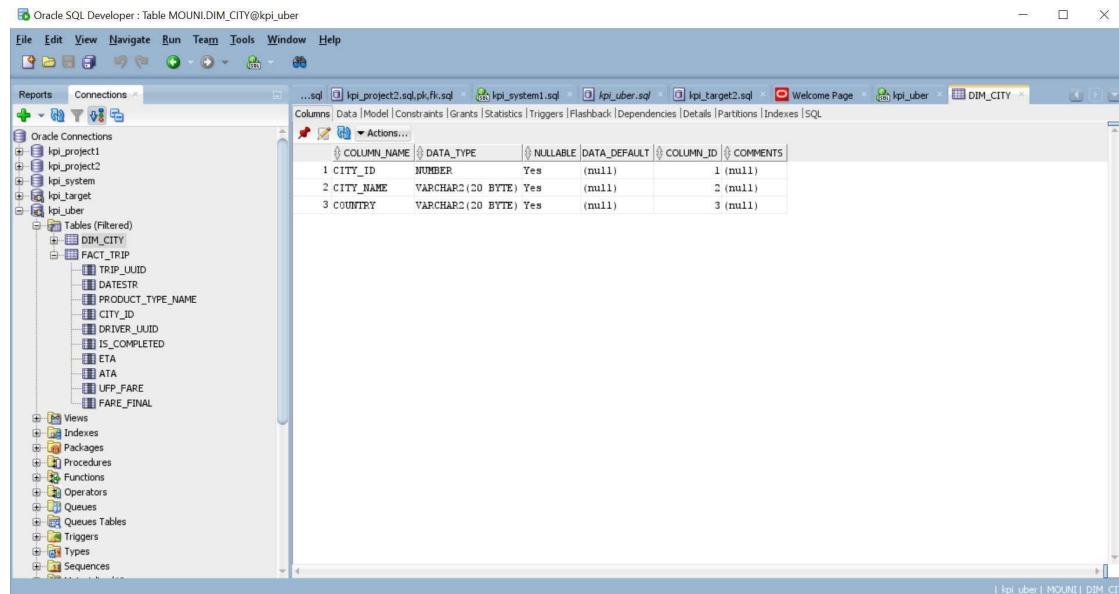


## INTRODUCTION:

Uber is a prominent Taxi Aggregator that caters to commuters needs. Commuters can use Uber app to request a taxi for their commute needs. With ever increasing smart phones, Uber has become a go to option for most of the travellers

## A BRIEF DESCRIPTION OF THE DATA USED:

We are having two data sets 1. dim city and 2. fact trip, in the fact table we can see 3 columns City\_id, city name, country. And in the fact trip we have trip\_uuid, datastr, product\_type\_name, city\_id, driver\_uuid, is\_completed, ETA, ATA, UFF\_fare, fare final this columns will provide all the data by using the this 2 data sets we can solve the customer requirements, and Uber provides services across lot of cities and there are various products catered to the traveller's needs. Uber seeks our help to understand which of the products are profitable and how many times were they able to meet the ETA so they can fine tune the service offerings.



Oracle SQL Developer

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
TRIP_UUID	VARCHAR2(100 BYTE)	Yes	(null)	1 (null)	
DATESTR	DATE	Yes	(null)	2 (null)	
PRODUCT_TYPE_NAME	VARCHAR2(100 BYTE)	Yes	(null)	3 (null)	
CITY_ID	NUMBER	Yes	(null)	4 (null)	
DRIVER_UUID	VARCHAR2(100 BYTE)	Yes	(null)	5 (null)	
IS_COMPLETED	VARCHAR2(100 BYTE)	Yes	(null)	6 (null)	
ETA	NUMBER	Yes	(null)	7 (null)	
ATA	NUMBER	Yes	(null)	8 (null)	
UFF_FARE	NUMBER	Yes	(null)	9 (null)	
FARE_FINAL	NUMBER	Yes	(null)	10 (null)	

## Anomalies:

In the given data set I didn't get any anomalies.

Oracle SQL Developer : C:\Users\MONISHA\kpi\_uber.sql

SQL Worksheet: History

Worksheet: Query Builder

Script Output: Query Result

SQL | Fetched 50 rows in 0.009 seconds

TRIP_UUID	DATESTR	PRODUCT_TYPE_NAME	CITY_ID	DRIVER_UUID	IS_COMPLETED	ETA	ATA	UFF_FARE	FARE_FIN
1 AAA49807	06-01-18	Helium	10	bfc85d1e-b403-422d-b47c-b005947eb3d4	TRUE	1893	1325	8.46	5
2 AAA21475	06-01-18	uberX	9	c6ca70e4-ff5e-4e98-9a53-23810a2fe43b	TRUE	1235	988	1.17	1.
3 AAA52773	06-01-18	uberX	19	4ad9a375-7198-437f-b148-f1935f4d079d	FALSE	5552	4997	3.7	3
4 AAA04695	06-01-18	uberPOOL	19	e4e4d46e-e410-43ad-8de8-c304b9d41be4	TRUE	5664	7363	54.75	45.
5 AAA20580	06-01-18	Helium	18	e7bc1169-47af-4577-9e41-7f0bbeca552e	TRUE	5621	3935	54.27	45.
6 AAA78933	06-01-18	Helium	2	ac7450f4-1eb8-4eb3-be5b-7a0e090e8eb2	TRUE	3737	3737	37.55	41.
7 AAA62409	06-01-18	uberX	11	8ebe4897-ea17-4ea0-be38-ba9f2b157649	TRUE	4672	4672	4.34	4.
8 AAA47490	06-01-18	uberX	7	5347c86a-7e1e-4283-aalc-7c34ba405d4d	TRUE	2285	2513	42.94	35.
9 AAA59137	06-01-18	uberX	8	de53af08-fd2d-421e-9ad0-083d4de94913	TRUE	9512	7609	39.06	43
10 AAA65846	06-01-18	Helium	8	cc541373-11a3-4805-9e4f-bf3bd44d087	FALSE	8642	6914	1.69	2.
11 AAA1293	06-02-18	Helium	8	ac7450f4-1eb8-4eb3-be5b-7a0e090e8eb2	TRUE	5453	4908	17.73	16.
12 AAA25770	06-02-18	Helium	15	00c05886-5359-489f-bfe1-33d9c8f8b45d	TRUE	2567	2310	7.15	8.
13 AAA84449	06-02-18	uberX	2	e27cafb6-42c1-49e4-5de8-1abe0d04afd4	TRUE	8887	9776	51.83	43.
14 AAA32758	06-02-18	Helium	9	874479cb-eb0d-46dd-51d6-2206bfaca05e	TRUE	2158	1942	8.83	8.
15 AAA78788	06-02-18	uberX	9	bed8d33e-90b2-421d-bd3d-6ddf3ff51020	TRUE	3646	4375	22.41	20.
16 AAA50727	06-02-18	uberX	1	ec0ff3f9-9a1e-4e08-b5b5-22fd2ee2e07e	TRUE	7177	5024	29.67	37.
17 AAA48965	06-02-18	Helium	3	e7bc1169-47af-4577-9e41-7f0bbeca552e	FALSE	7533	8286	26.64	22.

Oracle SQL Developer: C:\Users\MONISHA\kpi\_uber.sql

File Edit View Navigate Run Source Team Tools Window Help

Reports Connections

Oracle Connections

- kpi\_project1
- kpi\_project2
- kpi\_system
- kpi\_target
- kpi\_uber

Tables (Filtered)

- DIM\_CITY
  - CITY\_ID
  - CITY\_NAME
  - COUNTRY
- FACT\_TRIP
  - TRIP\_UUID
  - DATESTR
  - PRODUCT\_TYPE\_NAME
  - CITY\_ID
  - DRIVER\_UUID
  - IS\_COMPLETED
  - ETA
  - ATA
  - UPF\_FARE
  - FARE\_FINAL

Views

Indexes

Packages

Procedures

Functions

Operators

Queues

Tables

SQL Worksheet: History

Worksheet Query Builder

select \* from dim\_city

Script Output x Query Result x

SQL All Rows Fetched: 20 in 0.003 seconds

CITY_ID	CITY_NAME	COUNTRY
1	San Francisco	US
2	Paris	FR
3	London	UK
4	New Delhi	IN
5	Bengaluru	IN
6	Mumbai	IN
7	New York	US
8	Amsterdam	NL
9	Chicago	US
10	San Diego	US
11	Hyderabad	IN
12	Lisbon	PT
13	Milan	IT
14	Cairo	EG
15	Dubai	AE
16	San Antonio	US
17	Islamabad	PK
18	Berlin	DE

## Query's & outputs:

a. How many city\_ids does uberPOOL operate in?

```
select count(distinct(city_id))
from fact_trip
where product_type_name='uberPOOL';
```

Oracle SQL Developer: C:\Users\MONISHA\kpi\_uber.sql

File Edit View Navigate Run Source Team Tools Window Help

Reports Connections

Oracle Connections

- kpi\_project1
- kpi\_project2
- kpi\_system
- kpi\_target
- kpi\_uber

Tables (Filtered)

- DIM\_CITY
- FACT\_TRIP
  - TRIP\_UUID
  - DATESTR
  - PRODUCT\_TYPE\_NAME
  - CITY\_ID
  - DRIVER\_UUID
  - IS\_COMPLETED
  - ETA
  - ATA
  - UPF\_FARE
  - FARE\_FINAL

Views

Indexes

Packages

Procedures

Functions

Operators

Queues

Tables

Triggers

Types

Sequences

SQL Worksheet: History

Worksheet Query Builder

--3 a. How many city\_ids does uberPOOL operate in?----

```
select count(distinct(city_id))
from fact_trip
where product_type_name='uberPOOL';
```

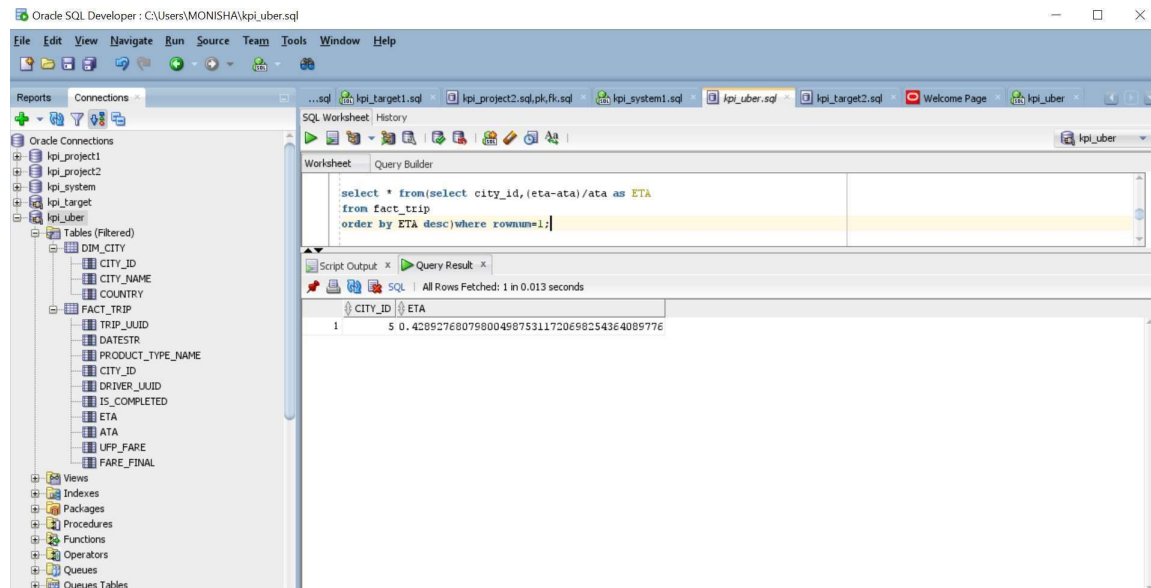
Script Output x Query Result x

SQL All Rows Fetched: 1 in 0.003 seconds

COUNT(DISTINCT(CITY_ID))
11

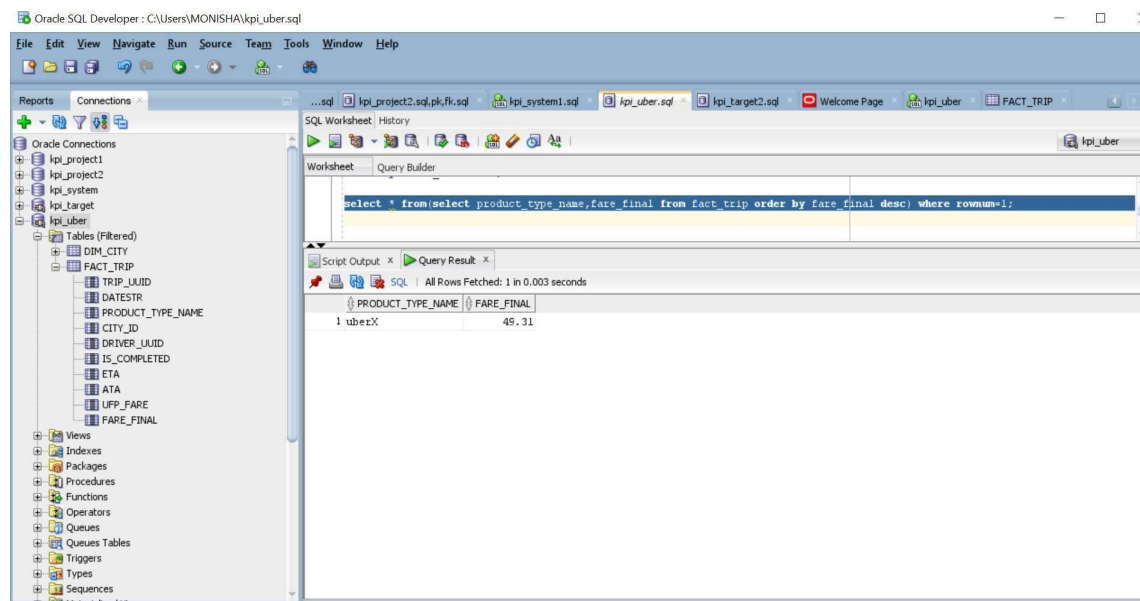
b. Which city\_id has the highest error in ETA (where error in ETA =  $\{(eta - ata)/ata\}$ ) for the given time period?

```
select * from (select city_id, (eta-ata)/ata as
ETA from fact_trip
order by ETA desc) where rownum=1;
```



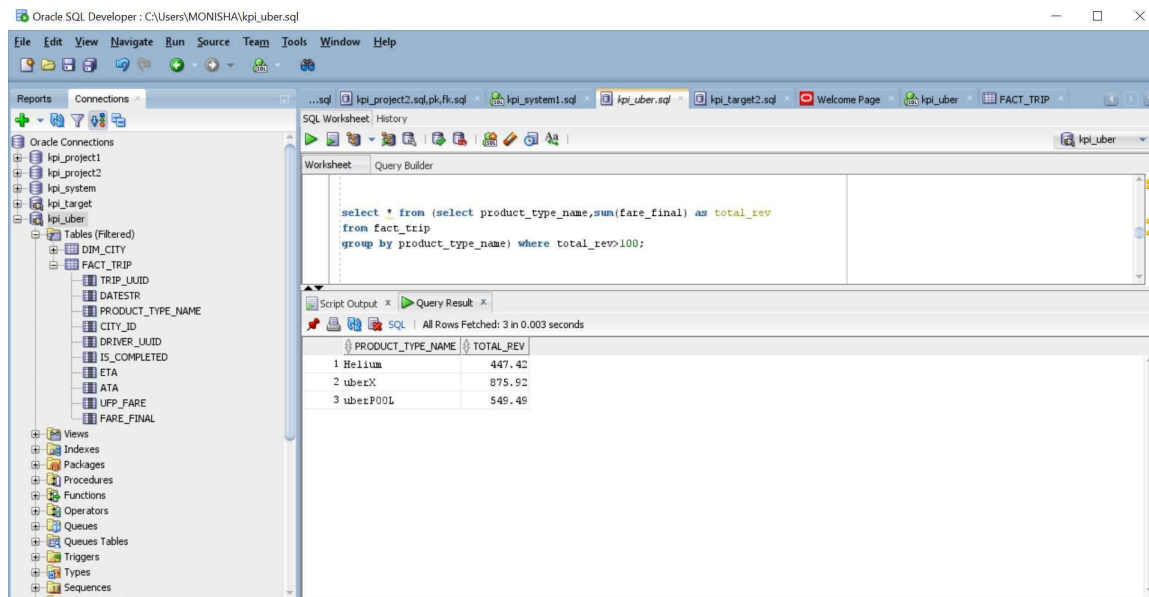
c. Which is the product type with highest total revenue in San Francisco?

```
select * from (select product_type_name, fare_final from fact_trip order by fare_final desc)
where rownum=1;
```



d. Which are the products in each city where total revenue(fare\_final) > \$1000?

```
select * from (select product_type_name,sum(fare_final) as
total_rev from fact_trip
group by product_type_name) where total_rev>100;
```



e. Get to 2nd highest country by Uber Revenue (fare\_final) for 2nd week of June 2018 across

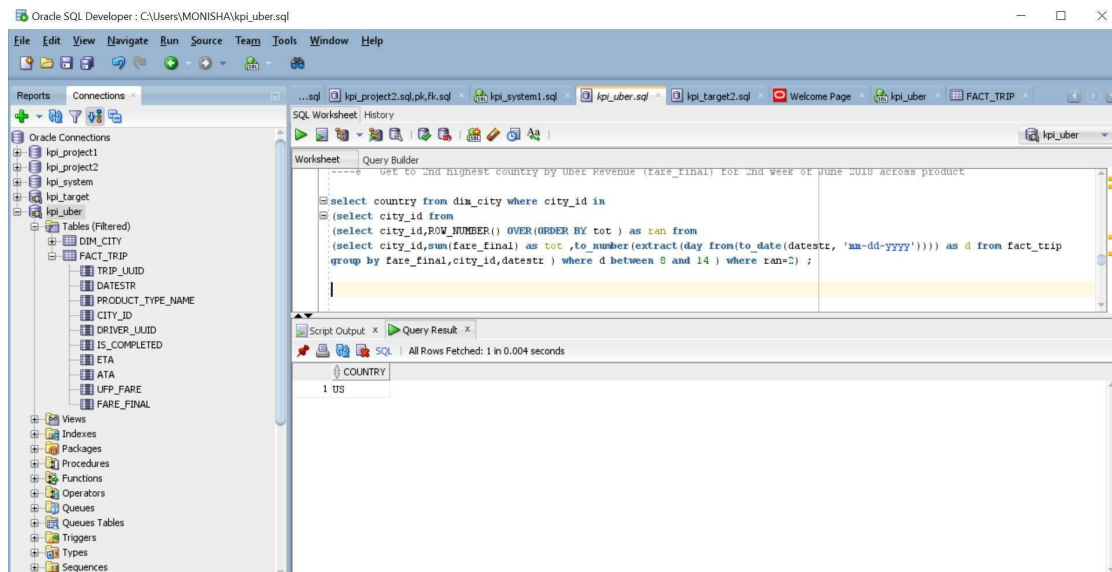
product select country from dim\_city where city\_id in

(select city\_id from

(select city\_id,ROW\_NUMBER() OVER(ORDER BY tot ) as ran from

(select city\_id,sum(fare\_final) as tot ,to\_number(extract(day from(to\_date(datestr, 'mm-dd-yyyy')))) as d from fact\_trip

group by fare\_final,city\_id,datestr ) where d between 8 and 14 ) where ran=2) ;



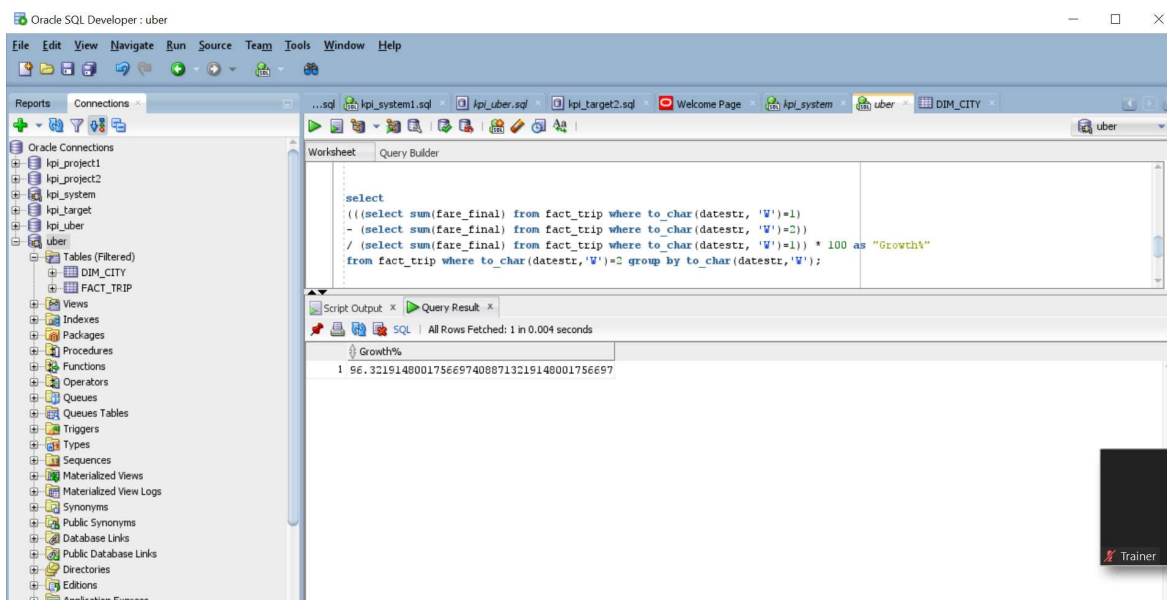
f. Get WOW growth % for US region for June Month. WOW- Week over week .

select

```

(((select sum(fare_final) from fact_trip where to_char(datestr, 'W')=1)
- (select sum(fare_final) from fact_trip where to_char(datestr, 'W')=2))
/ (select sum(fare_final) from fact_trip where to_char(datestr, 'W')=1)) * 100 as
"Growth%" from fact_trip where to_char(datestr, 'W')=2 group by
to_char(datestr, 'W');

```



$$g.\text{Growth \%} = ((\text{Current week fare final} - \text{previous week fare final}) / \text{previous week fare final}) * 100$$

```
select
(((select sum(fare_final) from fact_trip where to_char(datestr, 'W')='1')
- (select sum(fare_final) from fact_trip where to_char(datestr, 'W')='2'))
/ (select sum(fare_final) from fact_trip where to_char(datestr, 'W')='1') *100 ) as
"Growth%" from dual;
```

