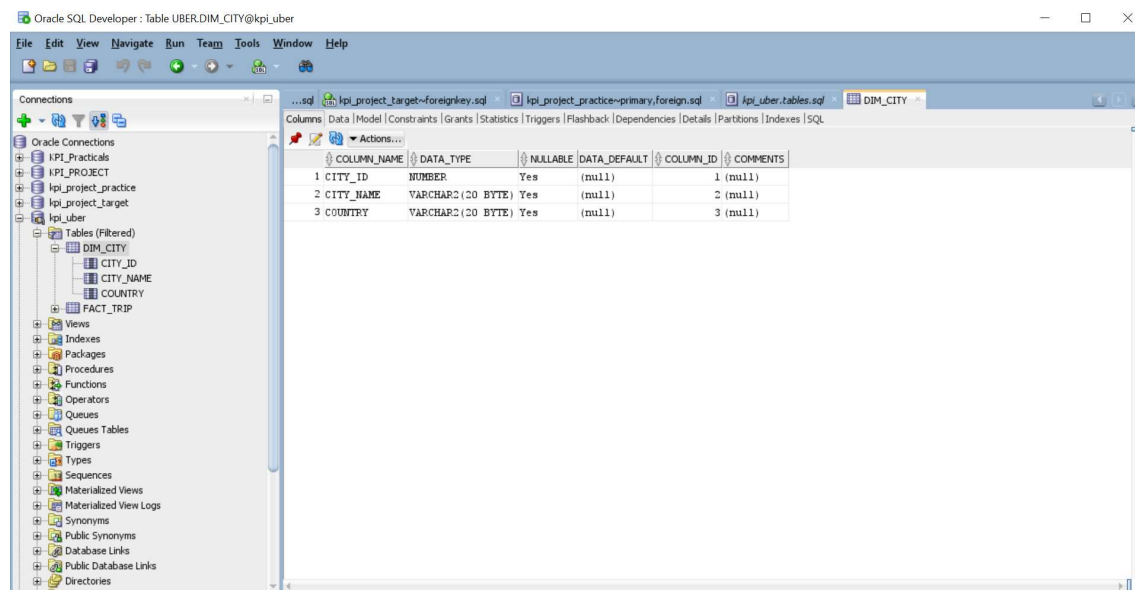


## INTRODUCTION:

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:

Here we are having two data sets 1. Dim\_city and 2. fact\_trip where Dim\_city is a dimension which lists all the cities that Uber provides services to. Fact\_trip provides details of all the trip transactions. In the dim\_city we are having 3 columns City\_id, city\_name, country. And in the fact\_trip we are having 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 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.



The screenshot shows the Oracle SQL Developer interface. The left pane displays the 'Connections' tree with the 'kpi\_uber' connection selected. The right pane shows the 'Columns' tab for the 'DIM\_CITY' table. The table structure is as follows:

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
CITY_ID	NUMBER	Yes	(null)	1	(null)
CITY_NAME	VARCHAR2(20 BYTE)	Yes	(null)	2	(null)
COUNTRY	VARCHAR2(20 BYTE)	Yes	(null)	3	(null)

Oracle SQL Developer: Table UBER.FACT\_TRIP@kpi\_uber

File Edit View Navigate Run Team Tools Window Help

Connections

Oracle Connections

- KPI\_Practicals
  - KPI\_PROJECT
  - kpi\_project\_practice
  - kpi\_project\_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
        - UFP\_FARE
        - FARE\_FINAL
  - Views
  - Indexes
  - Packages
  - Procedures
  - Functions
  - Operators
  - Queues
  - Queues Tables

Columns: Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

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

## ANOMALIES:

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

Oracle SQL Developer: C:\Users\TanujaNekkanti\kpi\_uber.tables.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections

Oracle Connections

- KPI\_Practicals
  - KPI\_PROJECT
  - kpi\_project\_practice
  - kpi\_project\_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
        - UFP\_FARE
        - FARE\_FINAL
    - Views
    - Indexes
    - Packages
    - Procedures
    - Functions
    - Operators
    - Queues
    - Queues Tables

SQL Worksheet: History

Worksheet

Query Builder

select \* from fact\_trip;

Script Output x Query Result x

SQL All Rows Fetched: 73 in 0.01 seconds

TRIP_UUID	DATESTR	PRODUCT_TYPE_NAME	CITY_ID	DRIVER_UUID	IS_COMPLETED	ETA	ATA	UFP_FARE	FARE_FINAL
1 AAA49807	06-01-18	Helium	10	bfc85d1e-b403-422d-b47c-b005947eb3dd	TRUE	1093	1325	0.46	9.4
2 AAA21475	06-01-18	uberX	9	c6ca70e4-ffe5-4690-5a53-23810a2fe43b	TRUE	1235	988	1.17	1.06
3 AAA52773	06-01-18	uberX	19	4ad5a375-7190-437e-b140-f1935f4d079d	FALSE	5552	4997	3.7	3.7
4 AAA84695	06-01-18	uberPOOL	19	e4e4d46e-e410-43ad-8de9-c304b9d41be4	TRUE	5664	7363	54.75	45.63
5 AAA20580	06-01-18	Helium	10	e7bc1169-47a6-4577-9e41-7f0bb6ca5526	TRUE	5621	3935	54.27	45.23
6 AAA70933	06-01-18	Helium	2	a27450f4-1eb8-4eb3-be5b-7a0e090e8eb2	TRUE	3737	3737	37.55	41.73
7 AAA62409	06-01-18	uberX	11	8ebed097-ea17-4ea0-be38-ba9f2db157649	TRUE	4672	4672	4.34	4.82
8 AAA47490	06-01-18	uberX	7	5347c86a-7e1e-4283-aa1c-7c34aba405dd	TRUE	2285	2513	42.94	35.78
9 AAA59137	06-01-18	uberX	8	de53af08-e4d2-421e-9ad0-083da4e94913	TRUE	9512	7609	39.06	43.4
10 AAA65846	06-01-18	Helium	8	cc541373-11a3-4805-9e4f-bf3b444e4087	FALSE	8642	6914	1.69	2.12
11 AAA1293	06-02-18	Helium	8	a27450f4-1eb8-4eb3-be5b-7a0e090e8eb2	TRUE	5453	4908	17.73	16.12
12 AAA25770	06-02-18	Helium	15	00c05806-5399-409f-bfe1-33d9c0f9b49d	TRUE	2567	2310	7.15	8.93
13 AAA84449	06-02-18	uberX	2	e27c9fb6-42c1-49e4-9de0-1abe0e04afdd	TRUE	8807	9776	51.83	43.19
14 AAA32758	06-02-18	Helium	9	874479cb-eb0d-46dd-51d6-2206bfce09e	TRUE	2158	1942	8.83	8.03
15 AAA78788	06-02-18	uberX	9	bcd8433e-90b2-4214-bd34-6ddcf3ef51020	TRUE	3646	4375	22.41	20.38
16 AAA50727	06-02-18	uberX	1	ec0ff3f9-9ale-4e08-b5b5-22fd2ce2f87f	TRUE	7177	5024	29.67	37.08
17 AAA48965	06-02-18	Helium	3	e7bc1169-47a6-4577-9e41-7f0bb6ca5526	FALSE	7533	8286	26.64	22.2
18 AAA41301	06-02-18	Helium	1	2fa0e110-1d30-4b0c1-8686-afbf59f26d45	TRUE	5436	6523	21.49	21.49

Oracle SQL Developer: C:\Users\TanujaNekkanti\kpi\_uber.tables.sql

SQL Worksheet: History

Worksheet: Query Builder

Script Output: x Query Result: x

SQL: All Rows Fetched: 20 in 0.002 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 and Outputs:

- a. A brief description of your understanding of data.

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

Oracle SQL Developer: C:\Users\TanujaNekkanti\kpi\_uber.tables.sql

SQL Worksheet: History

Worksheet: Query Builder

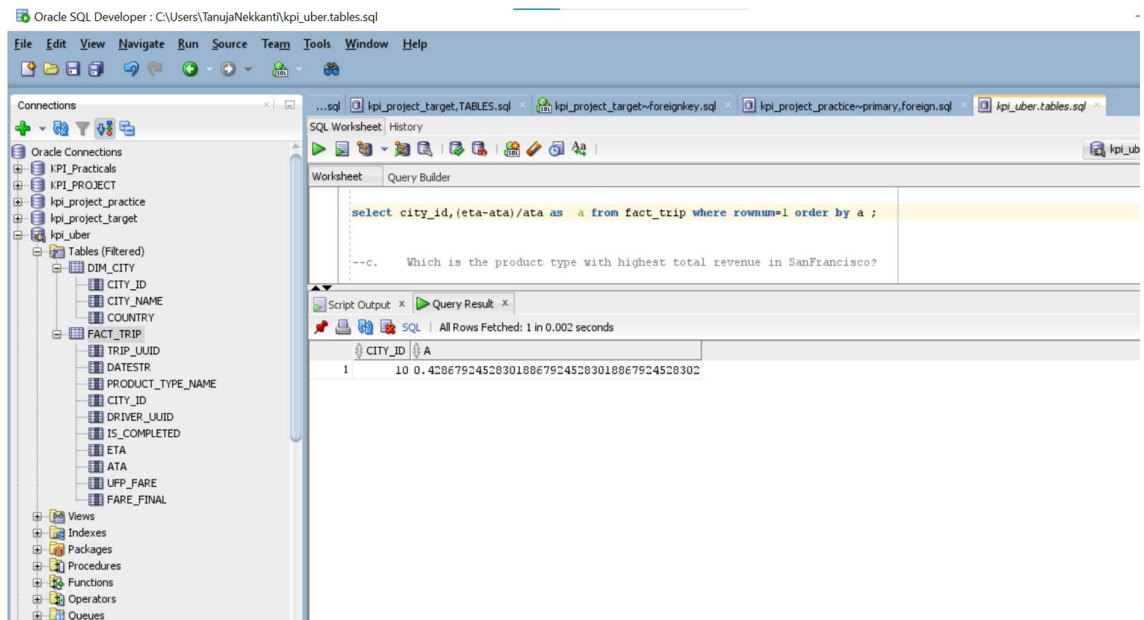
Script Output: x Query Result: x

SQL: All Rows Fetched: 1 in 0.003 seconds

COUNT(DISTINCT(CITY_ID))
11

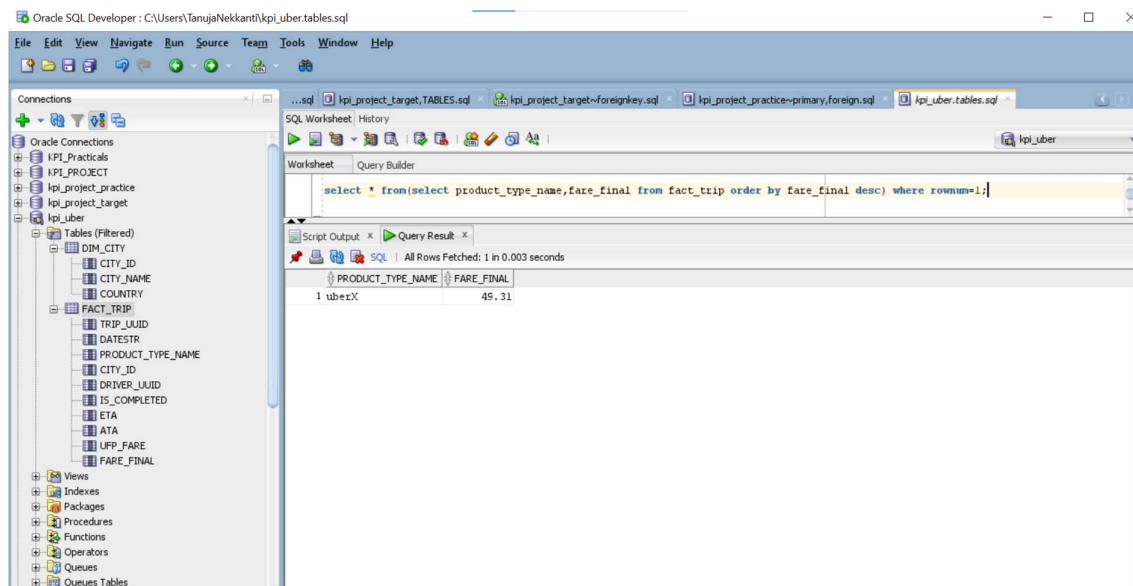
- b. Which city\_id has the highest error in ETA (where error in ETA =  $\{(eta - ata)/ata\}$ ) for the given time period?

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



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

select \* from(select product\_type\_name,fare\_final from fact\_trip order by fare\_final desc)  
where rownum=1;

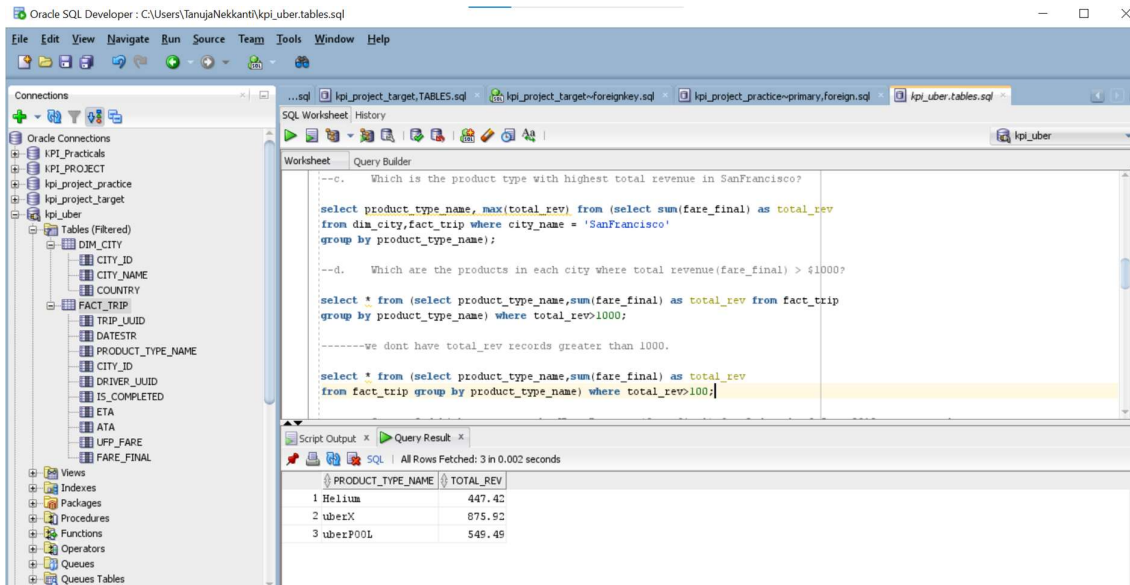


d. Which are the products in each city where total revenue(fare\_final) > \$1000?

select \* from (select product\_type\_name,sum(fare\_final) as total\_rev from fact\_trip  
group by product\_type\_name) where total\_rev>1000;

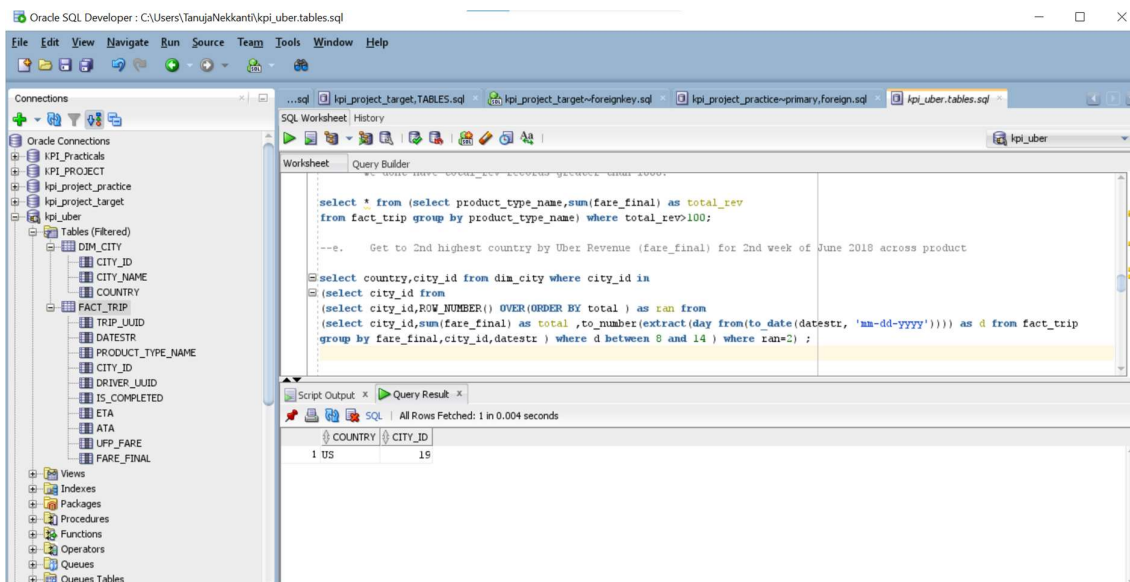
----from the given data sets we don't have the total\_rev records greater than 1000 we  
have greater than 100. So we modified and calculating for greater than 100.

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

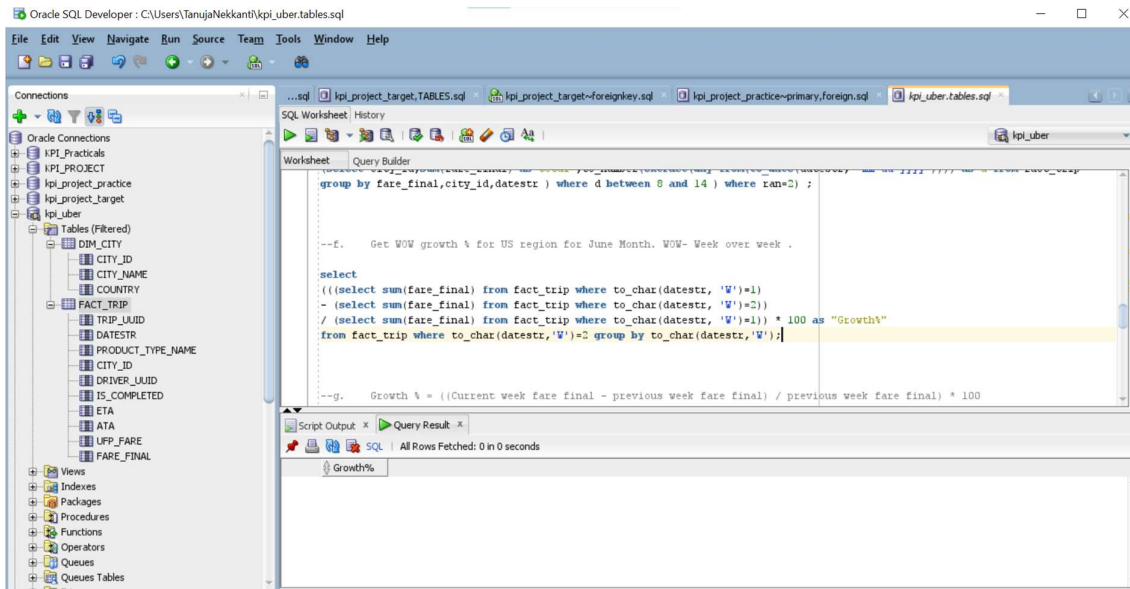
select country,city\_id from dim\_city where city\_id in (select city\_id from  
(select city\_id,ROW\_NUMBER() OVER(ORDER BY total ) as ran from  
(select city\_id,sum(fare\_final) as total ,to\_number(extract(day from(to\_date(datestr, 'mm-dd-  
yyyy')))) as d from fact\_trip group by fare\_final,city\_id,datestr ) 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;

