

## A brief description of your understanding of data

- There are two data sets 1. dim city and 2. fact trip.
- In the dim\_city we have 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 by using the this 2 data sets we can solve the customer requirements
- 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

Any anomalies you identified in the provided dataset and a brief description of how you identified them and why do you think they are anomalies

There are no anomalies in the dataset

## Queries

A. How many city\_ids does uberpool operate in?

```
SELECT COUNT(D.CITY_ID) FROM DIM_CITY D,FACT_TRIP F
WHERE D.CITY_ID=F.CITY_ID AND PRODUCT_TYPE_NAME='UBERPOOL';
```

<pre> SELECT COUNT(D.CITY_ID) FROM DIM_CITY D,FACT_TRIP F WHERE D.CITY_ID=F.CITY_ID AND PRODUCT_TYPE_NAME='UBERPOOL'; </pre>					
<div> <div>Script Output x</div> <div>Query Result x</div> <div>Query Result 1 x</div> <div>Query Result 2 x</div> <div>Query Result 3 x</div> </div> <div> <div>SQL</div> <div>All Rows Fetched: 1 in 0.003 seconds</div> </div> <table> <thead> <tr> <th></th><th>COUNT(D.CITY_ID)</th></tr> </thead> <tbody> <tr> <td>1</td><td>0</td></tr> </tbody> </table>			COUNT(D.CITY_ID)	1	0
	COUNT(D.CITY_ID)				
1	0				

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;**

**SELECT CITY\_ID,(ETA-ATA)/ATA AS A FROM FACT\_TRIP WHERE ROWNUM=1 ORDER BY A ;**

```
SELECT CITY_ID, (ETA-ATA)/ATA AS A FROM FACT_TRIP;
SELECT CITY_ID, (ETA-ATA)/ATA AS A FROM FACT_TRIP WHERE ROWNUM=1 ORDER BY A ;
```

Script Output x


Query Result x

Query Result 1 x

Query Result 2 x

Query Result 3 x

Query Result 4 x

 SQL

All Rows Fetched: 1 in 0.001 seconds

	CITY_ID	A
1	10	0.4286792452830188679245283018867924528302

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;
```

The screenshot shows a SQL Developer window with a query editor at the top containing the SQL statement: `SELECT * FROM (SELECT PRODUCT_TYPE_NAME, FARE_FINAL FROM FACT_TRIP ORDER BY FARE_FINAL DESC) WHERE ROWNUM=1;`. Below the editor, the 'Query Result' tab is active, displaying a table with two columns: 'PRODUCT\_TYPE\_NAME' and 'FARE\_FINAL'. The table contains one row with the value 'uberX' under 'PRODUCT\_TYPE\_NAME' and '49.31' under 'FARE\_FINAL'. The status bar indicates 'All Rows Fetched: 1 in 0.003 seconds'.

PRODUCT_TYPE_NAME	FARE_FINAL
1 uberX	49.31

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

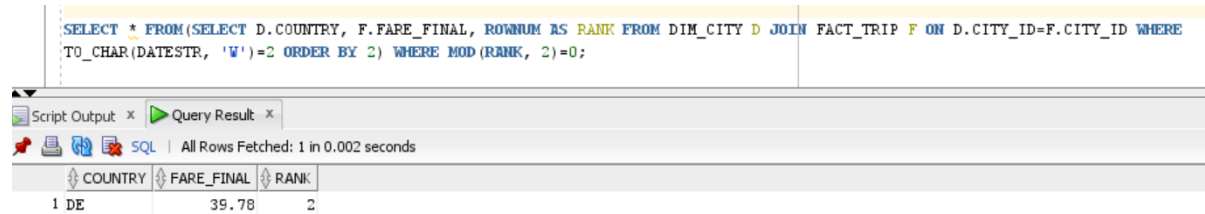
```
SELECT * FROM (SELECT PRODUCT_TYPE_NAME, SUM(FARE_FINAL) AS TOTAL_REV  
FROM FACT_TRIP  
GROUP BY PRODUCT_TYPE_NAME) WHERE TOTAL_REV > 1000;
```

The screenshot shows a SQL Developer window with a query editor at the top containing the SQL statement: `SELECT * FROM (SELECT PRODUCT_TYPE_NAME, SUM(FARE_FINAL) AS TOTAL_REV FROM FACT_TRIP GROUP BY PRODUCT_TYPE_NAME) WHERE TOTAL_REV > 1000;`. Below the editor, the 'Query Result' tab is active, displaying a table with two columns: 'PRODUCT\_TYPE\_NAME' and 'TOTAL\_REV'. The table contains three rows: 'Helium' with '447.42', 'uberX' with '875.92', and 'uberPOOL' with '549.49'. The status bar indicates 'All Rows Fetched: 3 in 0.002 seconds'.

PRODUCT_TYPE_NAME	TOTAL_REV
1 Helium	447.42
2 uberX	875.92
3 uberPOOL	549.49

E. Get to 2nd highest country by uber revenue (fare\_final) for 2nd week of june 2018 across product

```
SELECT * FROM (SELECT D.COUNTRY, F.FARE_FINAL, ROWNUM AS RANK FROM DIM_CITY D JOIN FACT_TRIP F ON D.CITY_ID=F.CITY_ID WHERE TO_CHAR(DATESTR, 'W')=2 ORDER BY 2) WHERE MOD(RANK, 2)=0;
```

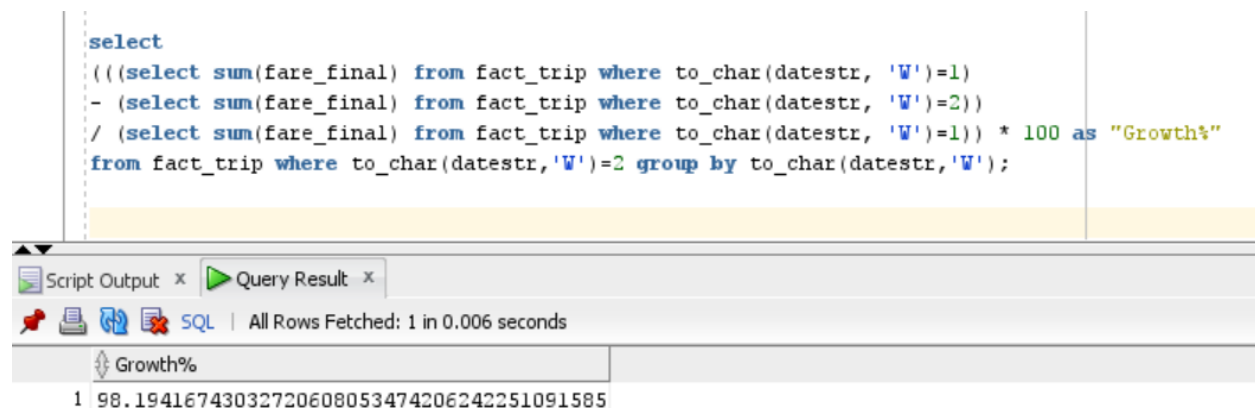


The screenshot shows a SQL query editor with a query window and a results window. The query is: `SELECT * FROM (SELECT D.COUNTRY, F.FARE_FINAL, ROWNUM AS RANK FROM DIM_CITY D JOIN FACT_TRIP F ON D.CITY_ID=F.CITY_ID WHERE TO_CHAR(DATESTR, 'W')=2 ORDER BY 2) WHERE MOD(RANK, 2)=0;` The results window shows one row with columns COUNTRY, FARE\_FINAL, and RANK. The values are DE, 39.78, and 2 respectively.

	COUNTRY	FARE_FINAL	RANK
1	DE	39.78	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');
```



The screenshot shows a SQL query editor with a query window and a results window. The query is: `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');` The results window shows one row with the column Growth%. The value is 98.19416743032720608053474206242251091585.

	Growth%
1	98.19416743032720608053474206242251091585

G. Growth % = ((Current week fare final - previous week fare final) / previous week fare final) \* 100

SELECT

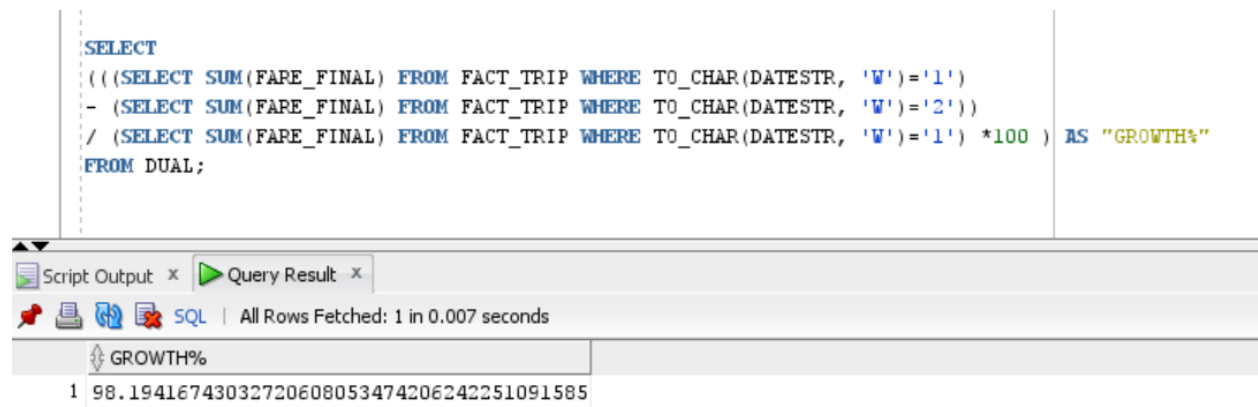
(( (SELECT SUM(FARE\_FINAL) FROM FACT\_TRIP WHERE TO\_CHAR(DATESTR, 'W')='1')

- (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;



```
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;
```

GROWTH%	
1	98.19416743032720608053474206242251091585