Group Project 1

BUAN6320.504

Group Members:

Li Zhang	lxz200000@utdallas.edu
Umapathi Saravanan	uxs210009@utdallas.edu
Darsh Rakesh Pancholi	dxp210036@utdallas.edu
Niharika Byarply Yathiraju	nby210001@utdallas.edu
Haneesh Goutham Balaji	hxb210033@utdallas.edu

1: Choose a Dataset

Dataset: Lyft Dataset Boston

Business: Lyft

Source of the dataset: Kaggle.com

2: Business Understanding

Based on the investigation of the order delivery system of our company, Lyft, we found out that many drivers got a plethora of orders during peak hours whereas on the other hand, drivers were not getting enough orders during non-rush hours. To balance the order delivery amount, we investigate and analyze the customer's Lyft ordering patterns and improve the experience of both the customers as well as the drivers. We have chosen the orders records of the Boston area collected from November 2018 to December 2018.

Based on the available dataset, we have decided the following:
First of all, we will keep these columns from the original table based on our business goal: ***Datetime***, ***Source***, ***Destination***, ***Product_id***, ***Price***, ***Distance***, ***Surge Multiplier***, ***Weather***, ***Day*** and ***Hour***.
Temperature. Then, for the column ***Hours***, we will separate the hours into four categories: *Midnight* (0-5), *Morning* (6-11), *Afternoon* (12-17), *Night* (18-23) as a new column ***Time***. In order to figure out which time period has more orders.
Third, based on the ***Short_summary*** which already classifies the data into nine types of weather. To find out which kind of weather customers tend to use Lyft more.
Next, through the data visualization, we will use the bar chart to figure out how different weather (***Weather***), location (***Source*** and ***Destination***) and day time (***Time)*** will affect the Total Orders Amount.
Also, we will do the basic statistical analysis of all continuous variables we have, for example, ***Price***, ***Distance***, ***Temperature***. Besides that, we will use the box plot to figure out whether there are outliers for those variables.

3: Data understanding

Data description:

The columns delineate various aspects of the dataset such as the id, time of the day, the pick-up and drop location of the customers, the type of vehicle used such as Lyft Luxury, Luxury XL or was the ride a carpool or a shared ride, the pricing for each ride, distance, temperature, and the weather forecast for the day. All this information is lucrative in making strategies for pricing and rolling out offers for the users which can be helpful for both the customers as well as the company to grow.
- Off the 57 columns in the dataset, 29 columns have a float value, 17 columns have an integer value and 11 columns have an object value

Dataset analysis:

Statistical analysis

The range of order ***Price*** is from $2.5 to $97.5, and highest standard deviation (10.02), which means the records from ***Price*** are the most separate around it's mean ($17.35), compared to other two, ***Distance***

(1.09) and ***Temperature*** (6.73). And the range of ***Distance*** and ***Temperature*** are from 0.39 to 6.33 and 18.91 to 57.22, and the means are 2.18 and 39.60 separately.

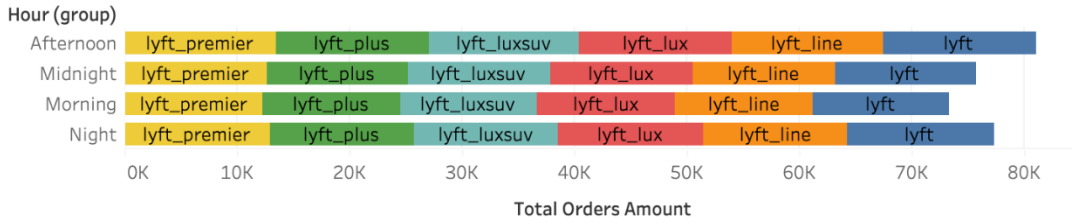
Summary Statistics

The MEANS Procedure

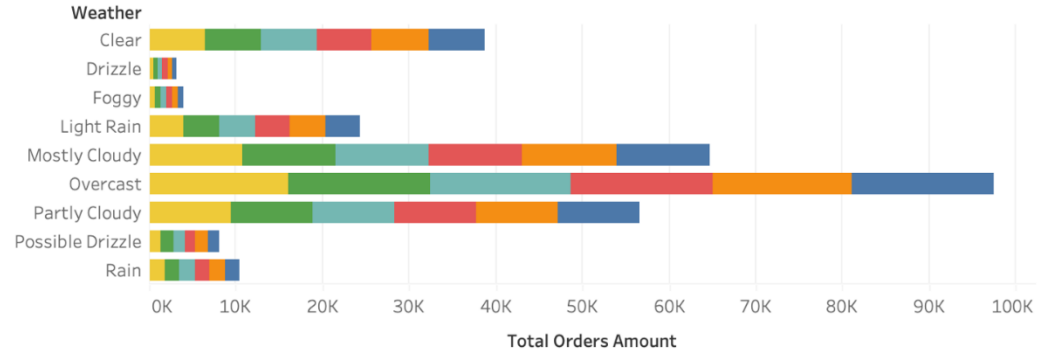
Variable	N	Mean	Mode	Std Dev	Minimum	Median	Maximum
price	307408	17.3513961	16.5000000	10.0191708	2.5000000	16.5000000	97.5000000
distance	307408	2.1869756	1.0600000	1.0866217	0.3900000	2.1400000	6.3300000
temperature	307408	39.5963842	37.9200000	6.7304249	18.9100000	40.4900000	57.2200000

Daily - Total Orders Amount

From the bar chart which uses the ***Time*** (which is grouped from ***Hours***) vs ***Total Orders Amount*** (which is from the count of ***Product_id***). We can see that the afternoon has the highest amount of orders among the 4 time periods of the day. And the morning has the least.



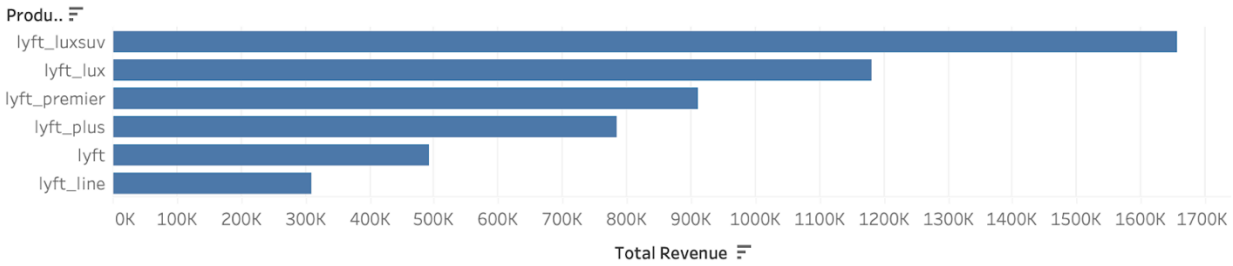
Weather - Total Orders Amount



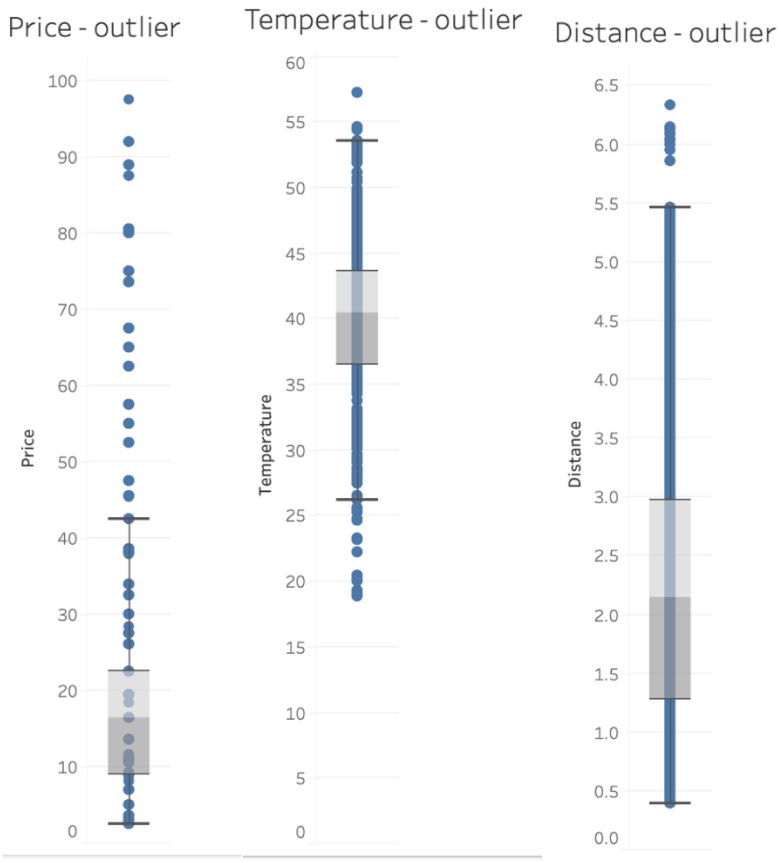
For the bar chart of ***Weather*** vs ***Total Orders Amount*** (which is from the count of ***Product_id***), when the weather is overcast, people are more likely to use Lyft, followed by mostly cloudy and partly cloudy. Also, the drizzle and foggy weather has the least number of orders.

Product - price

From the bar chart of total revenue and product_id, lytf_luxsuv has the highest revenue amount of all six types of Lyft product, followed by lyft_lux. And the lyft_line is the least welcomed product for this time period.



Outliers: price, temperature, and distance

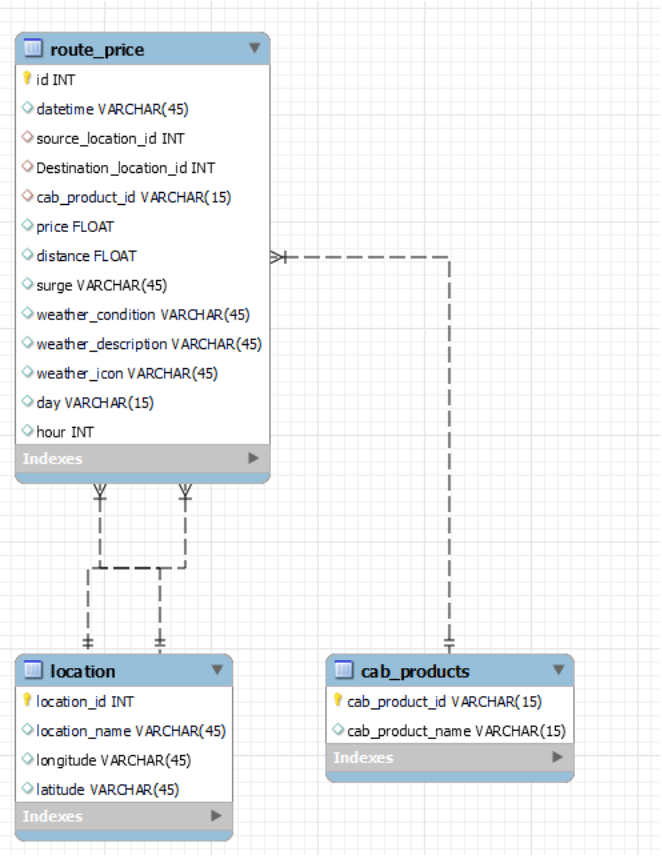


For the outlier, we are using the box plot to figure out the range of three continuous variables, ***Price***, ***Temperature*** and ***Distance***. From the box plot, we could see the variables of all three columns are all in a reasonable range, even though there are some variables are out of the maximum and minimum range of the box plot

4: Database design

Schema design and Schema normalization

ER diagram before normalization



BCNF -Normalization:

1. **Functional dependency**

Initial FDs:
F = {
{Hour_id} → {Time_category},
{Cab_products_id} → {Cab_products_name},
{Weather_condition}, {Weather_icon}-> {weather_description},
{Source_location_id, Destination_location_id} → { Location_name },
{Location_id} → {Location_name},
{datetime} → {day},
}

Infer FDs:
{Source_location_id} → {Location_name}

⇒ {Source_location_id, Location_id} → {Location_name}

Check BCNF condition for all FDs:
F = {
{Hour_id} → {Time_catefory} ⇒ in the same table ×
{Cab_products_id} → {Cab_products_name} ⇒ all in table Cab_products, and {Cab_products_id} is a key √
{Weather_condition, Weather_icon } → {Weather_description} ⇒ in the same table ×
{Source_location_id} → {Location_name} ⇒ not in the same table √
{Location_id} → {Location_name} ⇒ all in table location, and {Location_id} is a key √
{datetime} → {day} ⇒ in the same table ×
{Source_location_id, Location_id} → {Location_name} ⇒ not in the same table √

Decomposing BCNF:
A = {Id, datetime, Source_location_id, Destination_location_id, Cab_products_id, price, distance, surge, Weather_condition, Weather_description, Weather_icon, day, Hour_id, Time_category}

{Hour_id} → {Time_category}
⇒ violates BCNF ×
⇒ decompose into {Hour_id, Time_category}
and {Id, datetime, Source_location_id, Destination_location_id, Cab_products_id, distance,price, surge, Weather_condition, Weather_description, Weather_icon, day, Hour_id}

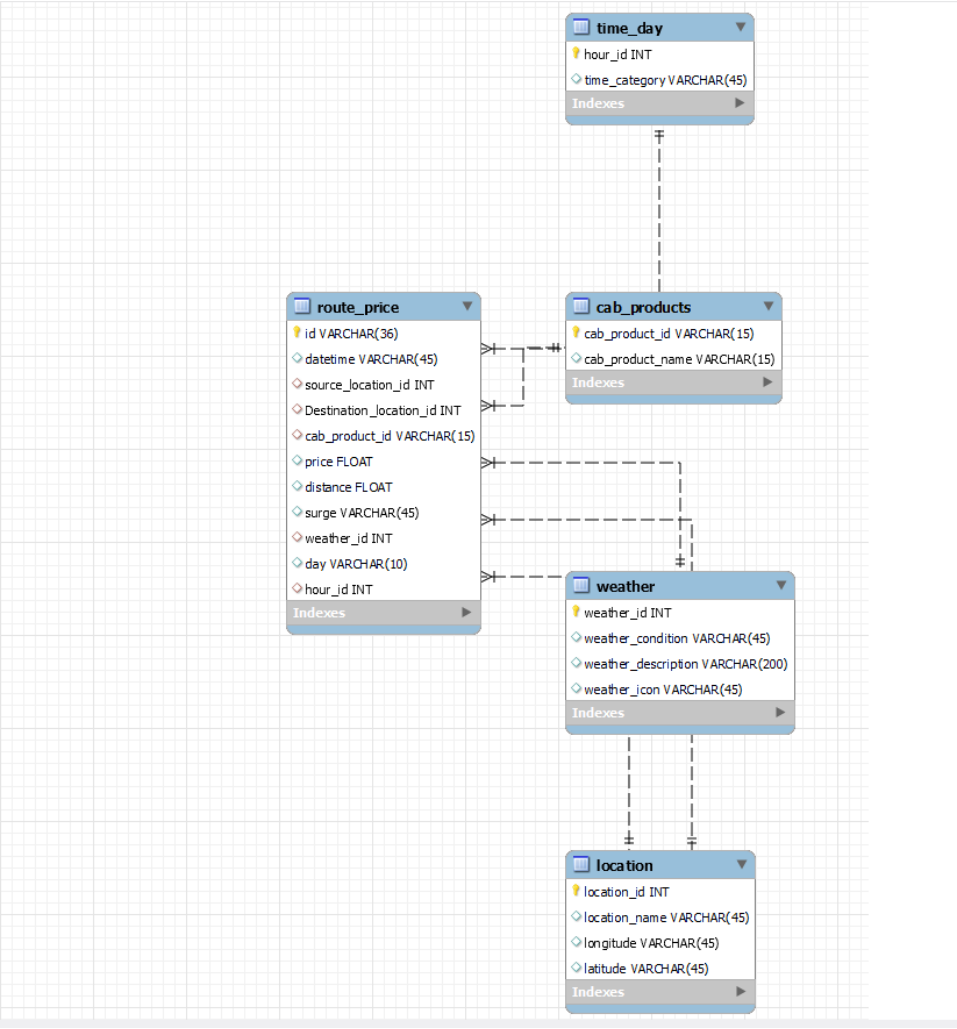
{Weather_condition,weather_description} → {Weather_icon}
⇒ violates BCNF ×
⇒ decompose into {Weather_condition, weather_id , weather_description ,Weather_icon}
and {Id, datetime, Source_location_id, Destination_location_id, Cab_products_id, distance,price, surge, weather_id, day, Hour_id}

{datetime} → {day}
⇒ violates BCNF ×
⇒ decompose into {datetime, day} and {Id, datetime, Source_location_id, Destination_location_id, Cab_products_id, distance,price, surge, weather_id,Hour_id}

{Source_location_id, Destination_location_id} → {location_name}
⇒ no violation (Source_location_id, Destination_location_id location_name) √

Final Database Schema:
{distance, price},
{Hour_id, Time_category},
{Weather_condition, Weather_icon,weather_description, weather_id},
{datetime, day},
{location_id,location_Name}
{Id, datetime, Source_location_id, Destination_location_id, Cab_products_id, surge, Weather_condition, distance,price, Weather_description, Hour_id}

ER- Diagram after schema normalization:



Errors and fixes:

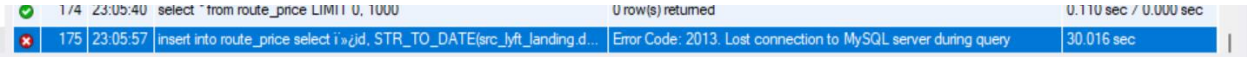
Error 1: *Alter table add AutoIncr option for Weather _id after all table creation with foreign key constraints added for Route_price table column in referencing weather_ID*

Fix: Foreign key constraint error when Modify column option is used in Weather table. Solved by ignoring forignkey constraint and set it back

SET FOREIGN_KEY_CHECKS=0;
Alter table weather modify column weather_id int AutoIncrement;
SET FOREIGN_KEY_CHECKS=1;

Error 2*: Inserting data to main table Route_price , got Lost connection error @ 30 sec.*
insert into route_price select id, STR_TO_DATE(src_lyft_landing.datetime, '%m/%d/%Y %T') as datetime, (select Location_id from Location where location_name = source) as source_location_id,(select

Location_id from Location where location_name = destination) as destination_location_id,product_id,price,distance,surge_multiplier,
(select weather_id from weather where weather_condition = short_summary and weather_icon = icon) as weather_id,dayname(STR_TO_DATE(src_lyft_landing.datetime, '%m/%d/%Y %T')) as day, hour from src_lyft_landing;



Fix: Updated timeout interval to 6000 in system preference, solved the issue

Error 3: *Inserting data to main table Route_price using above mentioned query at Error 3*
"lock wait timeout exceeded try restarting transaction"
Fix: After so many attempts identified that query goes to deadlock because of missing column names in insert statement INSERT INTO *table_name* (*column1*, *column2*, *column3*, ...)
Updated Query:
insert into route_price (id,datetime,source_location_id,destination_location_id,
cab_product_id,price,distance,surge,weather_id,day,hour_id) select id,
STR_TO_DATE(src_lyft_landing.datetime, '%m/%d/%Y %T') as datetime,
(select Location_id from Location where location_name = source) as source_location_id,
(select Location_id from Location where location_name = destination) as
destination_location_id,product_id,price,distance,surge_multiplier,
(select weather_id from weather where weather_condition = short_summary and weather_icon = icon)
as weather_id,dayname(STR_TO_DATE(src_lyft_landing.datetime, '%m/%d/%Y %T')) as day, hour from
src_lyft_landing;

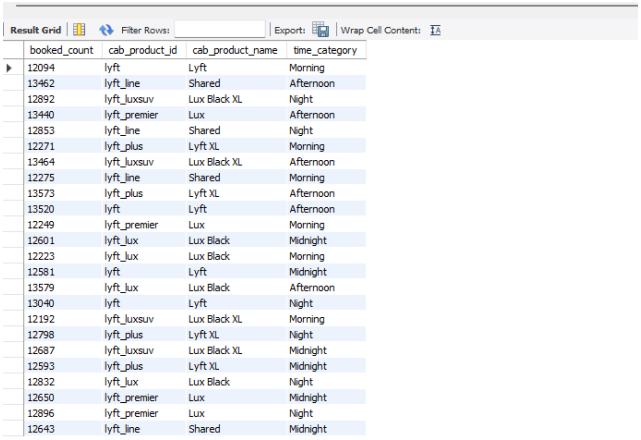
Error 4*: Inserting main data took lot of time and execution went beyond 1 hour.*
Fix: Fixed this using main setting at my.ini file. "innodb_buffer_pool_size" to 2G.
3 lakh records got inserted in 95 sec.

5: Data cleaning and database testing

Database testing and querying

- **Route booked and revenue by product and time of day**

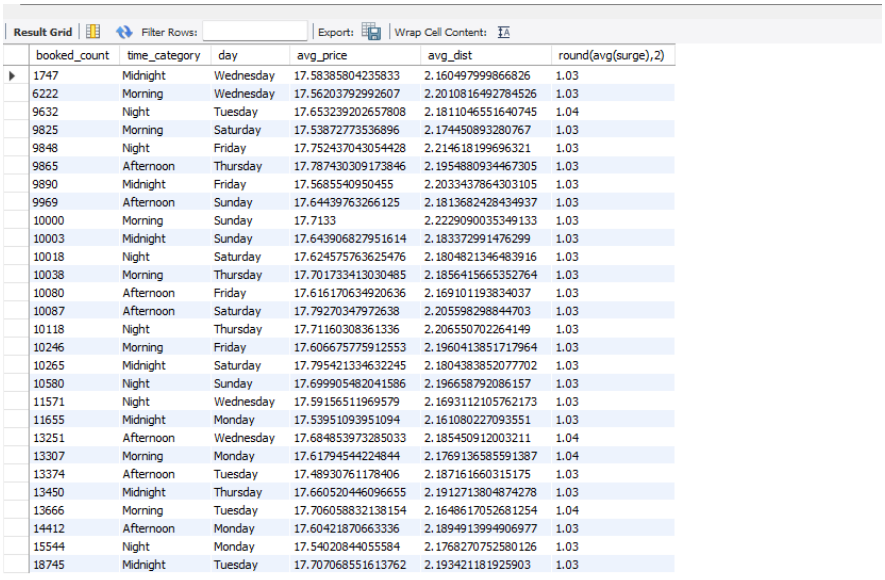
select count(rp.id) as booked_count, rp.cab_product_id, cp.cab_product_name,td.time_category from route_price rp
left join cab_products cp on rp.cab_product_id = cp.cab_product_id
left join time_day td on rp.hour_id = td.hour_id
group by cp.cab_product_id,td.time_category;



booked_count	cab_product_id	cab_product_name	time_category
12094	lyft	Lyft	Morning
13462	lyft_line	Shared	Afternoon
12892	lyft_luxsuv	Lux Black XL	Night
13440	lyft_premier	Lux	Afternoon
12853	lyft_line	Shared	Night
12271	lyft_plus	Lyft XL	Morning
13464	lyft_luxsuv	Lux Black XL	Afternoon
12275	lyft_line	Shared	Morning
13573	lyft_plus	Lyft XL	Afternoon
13520	lyft	Lyft	Afternoon
12249	lyft_premier	Lux	Morning
12601	lyft_lux	Lux Black	Midnight
12223	lyft_lux	Lux Black	Morning
12581	lyft	Lyft	Midnight
13579	lyft_lux	Lux Black	Afternoon
13040	lyft	Lyft	Night
12192	lyft_luxsuv	Lux Black XL	Morning
12798	lyft_plus	Lyft XL	Night
12687	lyft_luxsuv	Lux Black XL	Midnight
12593	lyft_plus	Lyft XL	Midnight
12832	lyft_lux	Lux Black	Night
12650	lyft_premier	Lux	Midnight
12896	lyft_premier	Lux	Night
12643	lyft_line	Shared	Midnight

- **Route booked and revenue by day and time of day**

```
select * from (select count(rp.id) as booked_count,td.time_category,rp.day, avg(rp.price) avg_price,
avg(rp.distance) avg_dist,round(avg(surge),2) from route_price rp
left join time_day td on rp.hour_id = td.hour_id
group by td.time_category,rp.day) a order by
a.booked_count,time_category,FIELD(a.day,'Sunday','Monday','Tuesday','Wednesday','Thursday','Friday
','Saturday');
```



booked_count	time_category	day	avg_price	avg_dist	round(avg(surge),2)
1747	Midnight	Wednesday	17.58385804235833	2.160497999866826	1.03
6222	Morning	Wednesday	17.56203792992607	2.2010816492784526	1.03
9632	Night	Tuesday	17.653239202657808	2.1811046551640745	1.04
9825	Morning	Saturday	17.538727735368996	2.174450893280767	1.03
9848	Night	Friday	17.752437043054428	2.214618199696321	1.03
9865	Afternoon	Thursday	17.787430309173846	2.1954880934467305	1.03
9890	Midnight	Friday	17.5685540950455	2.2033437864303105	1.03
9969	Afternoon	Sunday	17.64439763266125	2.1813682428434937	1.03
10000	Morning	Sunday	17.7133	2.2229090035349133	1.03
10003	Midnight	Sunday	17.643906827951614	2.183372991476299	1.03
10018	Night	Saturday	17.624575763625476	2.1804821346483916	1.03
10038	Morning	Thursday	17.701733413030485	2.1856415665352764	1.03
10080	Afternoon	Friday	17.616170634920636	2.169101193834037	1.03
10087	Afternoon	Saturday	17.79270347972638	2.205598298844703	1.03
10118	Night	Thursday	17.71160308361336	2.206550702264149	1.03
10246	Morning	Friday	17.606675775912553	2.1960413851717964	1.03
10265	Midnight	Saturday	17.795421334632245	2.1804383852077702	1.03
10580	Night	Sunday	17.699905482041586	2.196658792086157	1.03
11571	Night	Wednesday	17.59156511969579	2.1693112105762173	1.03
11655	Midnight	Monday	17.53951093951094	2.161080227093551	1.03
13251	Afternoon	Wednesday	17.684853973285033	2.185450912003211	1.04
13307	Morning	Monday	17.61794544224844	2.1769136585591387	1.04
13374	Afternoon	Tuesday	17.48930761178406	2.187161660315175	1.03
13450	Midnight	Thursday	17.660520446096655	2.1912713804874278	1.03
13666	Morning	Tuesday	17.706058832138154	2.1648617052681254	1.04
14412	Afternoon	Monday	17.60421870663336	2.1894913994906977	1.03
15544	Night	Monday	17.54020844055584	2.176827075258012 6	1.03
18745	Midnight	Tuesday	17.707068551613762	2.193421181925903	1.03

- **Route booked and revenue by day**

```
select count(rp.id) as booked_count,rp.day, sum(rp.price) tot_price, sum(rp.distance)
tot_dist,avg(rp.price) avg_price,
avg(rp.distance) avg_dist,round(avg(surge),2) from route_price rp
left join time_day td on rp.hour_id = td.hour_id
group by rp.day order by count(rp.id),
td.time_category,FIELD(rp.day,'Sunday','Monday','Tuesday','Wednesday','Thursday','Friday','Saturday');
```

booked_count	day	tot_price	tot_dist	avg_price	avg_dist	round(avg(surge),2)
32791	Wednesday	577884	71530.03008010983	17.623250282089597	2.1813921527281828	1.03
40064	Friday	706548	87965.81014472246	17.63548322683706	2.195632242030812	1.03
40195	Saturday	711026	87838.1201159954	17.689414106232118	2.1852996670231475	1.03
40552	Sunday	716787	89056.08010226488	17.675749654764253	2.1960958794206173	1.03
43471	Thursday	769903	95396.44015979767	17.710726691357458	2.194484602603981	1.03
54918	Monday	965222	119547.13020849228	17.57569467205652	2.176829640709645	1.03
55417	Tuesday	977828	120960.18020299077	17.644910406553947	2.182726964703805	1.03

- **Route booked and revenue by weather**

```
select count(rp.id) as booked_count,w.weather_condition,w.weather_icon, sum(rp.price) price
,avg(rp.price) avg_price , round(sum(rp.distance)) dist,avg(rp.distance) avg_dist,
 round(avg(surge),2) surge from route_price rp
left join weather w on rp.weather_id = w.weather_id
group by w.weather_id order by count(rp.id);
```

booked_count	weather_condition	weather_icon	price	avg_price	dist	avg_dist	surge
3111	Drizzle	rain	54889	17.643522982963677	6775	2.177743493859234	1.03
4002	Foggy	fog	71240	17.80109945027486	8810	2.2014592715140164	1.03
8072	Possible Drizzle	rain	141546	17.535431119920712	17650	2.186551043536695	1.03
10443	Rain	rain	183895	17.609403428133678	22865	2.1895518567379915	1.03
11954	Clear	clear-day	210875	17.640538731805254	26059	2.1799088212829263	1.03
24265	Partly Cloudy	partly-cloudy-day	428842	17.67327426334226	53128	2.1894795002530536	1.03
24328	Light Rain	rain	428974	17.63293324564288	53148	2.1846308812403494	1.03
26699	Clear	clear-night	469854	17.598187198022398	58162	2.1784280346879314	1.03
26870	Mostly Cloudy	partly-cloudy-day	475320	17.68961667286937	58763	2.1869490171193102	1.03
32398	Partly Cloudy	partly-cloudy-night	572995	17.686122600160505	71295	2.2006071393101942	1.03
37850	Mostly Cloudy	partly-cloudy-night	670360	17.710964332893	83382	2.2029529753449415	1.03
97416	Overcast	cloudy	1716408	17.619364375461938	212257	2.1788720572663296	1.03

- **Route booked and revenue by weather and cab product**

```
select count(rp.id) as booked_count, rp.cab_product_id, cp.cab_product_name,sum(rp.price) price
,w.weather_icon,w.weather_condition,avg(rp.price) avg_price , round(sum(rp.distance))
dist,avg(rp.distance) avg_dist,
 round(avg(surge),2) surge from route_price rp
left join cab_products cp on rp.cab_product_id = cp.cab_product_id
left join weather w on rp.weather_id = w.weather_id
group by cp.cab_product_id,w.weather_id;
```

booked_count	cab_product_id	cab_product_name	price	weather_icon	weather_condition	avg_price	dist	avg_dist	surge
4439	lyft	Lyft	43411	clear-night	Clear	9.779454832169408	9631	2.1696395626621867	1.04
4076	lyft_line	Shared	25002	rain	Light Rain	6.133954857703631	8945	2.1945215936117717	1
6281	lyft_luxsuv	Lux Black XL	205686	partly-cloudy-night	Mostly Cloudy	32.747333227193124	13846	2.20447858983953	1.04
5407	lyft	Lyft	52579	partly-cloudy-night	Partly Cloudy	9.724246347327538	11774	2.177501390574268	1.04
2005	lyft_premier	Lux	36744	clear-day	Clear	18.326184538653365	4407	2.1980099786992677	1.04
4513	lyft_line	Shared	27472	partly-cloudy-day	Mostly Cloudy	6.087303345889652	9762	2.163073346048508	1
4102	lyft_plus	Lyft XL	64414	rain	Light Rain	15.703071672354948	8956	2.183300832142079	1.04
16327	lyft_luxsuv	Lux Black XL	531085	cloudy	Overcast	32.528021069394256	35618	2.1815128349579576	1.04
4488	lyft_line	Shared	27346	clear-night	Clear	6.0931372549019605	9860	2.1968939436363875	1
4447	lyft_plus	Lyft XL	70106	partly-cloudy-day	Mostly Cloudy	15.764785248482124	9741	2.1904407500992846	1.04
4000	lyft_line	Shared	24450	partly-cloudy-day	Partly Cloudy	6.1125	8794	2.1984175034463407	1
16268	lyft	Lyft	158873	cloudy	Overcast	9.765982296533071	35583	2.187295307086761	1.04
4074	lyft	Lyft	39673	partly-cloudy-day	Partly Cloudy	9.738095238095237	8888	2.1816028509440817	1.04
16148	lyft_line	Shared	98416	cloudy	Overcast	6.094624721327719	35140	2.1760899215209504	1
4490	lyft_premier	Lux	81722	clear-night	Clear	18.200890868596883	9743	2.1699287339945412	1.04
1714	lyft_lux	Lux Black	40101	rain	Rain	23.39614935822637	3771	2.199900821946744	1.03
5455	lyft_lux	Lux Black	128085	partly-cloudy-night	Partly Cloudy	23.480293308890925	12020	2.203532541929091	1.04
6274	lyft_lux	Lux Black	147995	partly-cloudy-night	Mostly Cloudy	23.588619700350655	13801	2.1997625149328086	1.04
4466	lyft_lux	Lux Black	105390	partly-cloudy-day	Mostly Cloudy	23.598298253470666	9782	2.1902463088922315	1.04
3992	lyft	Lyft	39074	rain	Light Rain	9.78807615230461	8725	2.185501004656952	1.04
16268	lyft_lux	Lux Black	380607	cloudy	Overcast	23.396053602163757	35422	2.1773930447845222	1.04
652	lyft	Lyft	6401	fog	Foggy	9.817484662576687	1447	2.2194018402805358	1.04
4056	lyft_plus	Lyft XL	64136	partly-cloudy-day	Partly Cloudy	15.812623274161735	8959	2.208870811780999	1.04
4439	lyft_plus	Lyft XL	69724	clear-night	Clear	15.707141248028835	9641	2.1718990798938056	1.04
4051	lyft_premier	Lux	73222	rain	Light Rain	18.07504319921007	8763	2.163241178349076	1.04
1279	lyft_lux	Lux Black	30038	rain	Possible Drizzle	23.485535574667708	2800	2.189202508509765	1.04
1991	lyft_lux	Lux Black	46463	clear-day	Clear	23.336514314414867	4287	2.1531541995191503	1.04
662	lyft_line	Shared	4019	fog	Foggy	6.070996978851964	1452	2.1938217547454864	1
6337	lyft_premier	Lux	116306	partly-cloudy-night	Mostly Cloudy	18.353479564462268	14014	2.2115164936689315	1.04
5418	lyft_luxsuv	Lux Black XL	176481	partly-cloudy-night	Partly Cloudy	32.573089700996675	11882	2.193111852867527	1.04

- **Route booked by source and destination:**

select count(rp.id) as booked_count,l.location_name as source, l1.location_name as destination
from route_price rp left join location l on rp.source_location_id =l.location_id
left join location l1 on rp.Destination_location_id =l1.location_id group by
rp.source_location_id,rp.Destination_location_id order by rp.source_location_id;

booked_count	source	destination
4236	Haymarket Square	Back Bay
4314	Haymarket Square	North Station
4308	Haymarket Square	Theatre District
4218	Haymarket Square	Beacon Hill
4506	Haymarket Square	Financial District
4032	Haymarket Square	West End
4236	Back Bay	Haymarket Square
4212	Back Bay	Fenway
4548	Back Bay	North End
4271	Back Bay	Northeastern University
4164	Back Bay	Boston University
4224	Back Bay	South Station
4314	North Station	Haymarket Square
4284	North Station	Fenway
4194	North Station	North End
4254	North Station	Northeastern University
4230	North Station	Boston University
4050	North Station	South Station
4212	Fenway	Back Bay
4284	Fenway	North Station
4068	Fenway	Theatre District
4200	Fenway	Beacon Hill
4326	Fenway	Financial District
4530	Fenway	West End
4308	Theatre District	Haymarket Square
4068	Theatre District	Fenway
4164	Theatre District	North End
4254	Theatre District	Northeastern University
4416	Theatre District	Boston University
4320	Theatre District	South Station

- **Total Revenue by source and destination:**

select sum(rp.price) as booked_count,l.location_name as source, l1.location_name as destination
from route_price rp left join location l on rp.source_location_id =l.location_id
left join location l1 on rp.Destination_location_id =l1.location_id
group by rp.source_location_id,rp.Destination_location_id order by rp.source_location_id;

booked_count	source	destination
75549	Haymarket Square	Back Bay
54491	Haymarket Square	North Station
60817	Haymarket Square	Theatre District
57526	Haymarket Square	Beacon Hill
58826	Haymarket Square	Financial District
51376	Haymarket Square	West End
81300	Back Bay	Haymarket Square
62401	Back Bay	Fenway
94574	Back Bay	North End
59796	Back Bay	Northeastern University
63420	Back Bay	Boston University
71098	Back Bay	South Station
56272	North Station	Haymarket Square
83964	North Station	Fenway
57419	North Station	North End
88436	North Station	Northeastern University
85220	North Station	Boston University
65419	North Station	South Station
62338	Fenway	Back Bay
92880	Fenway	North Station
81090	Fenway	Theatre District
75285	Fenway	Beacon Hill
111852	Fenway	Financial District
89003	Fenway	West End
70592	Theatre District	Haymarket Square
89052	Theatre District	Fenway
69413	Theatre District	North End
81352	Theatre District	Northeastern University
106579	Theatre District	Boston University
58100	Theatre District	South Station

Clean up:

1. Dataset had typo/weak data on weather description. Eg: Clear condition had 3 different long summary. Used avg to find mostly used description and used the same

Before cleanup:

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

short_summary	long_summary	icon
Mostly Cloudy	Rain throughout the day.	partly-cloudy-night
Rain	Rain until morning, starting again in the evening.	rain
Clear	Light rain in the morning.	clear-night
Clear	Partly cloudy throughout the day.	clear-night
Partly Cloudy	Mostly cloudy throughout the day.	partly-cloudy-night
Overcast	Light rain in the morning and overnight.	cloudy
Overcast	Rain until morning, starting again in the evening.	cloudy
Light Rain	Light rain until evening.	rain
Foggy	Foggy in the morning.	fog
Light Rain	Light rain in the morning.	rain
Clear	Mostly cloudy throughout the day.	clear-day
Overcast	Mostly cloudy throughout the day.	cloudy
Clear	Rain throughout the day.	clear-night
Mostly Cloudy	Light rain in the morning.	partly-cloudy-day
Mostly Cloudy	Mostly cloudy throughout the day.	partly-cloudy-night
Overcast	Rain throughout the day.	cloudy
Partly Cloudy	Partly cloudy throughout the day.	partly-cloudy-day
Overcast	Partly cloudy throughout the day.	cloudy
Mostly Cloudy	Mostly cloudy throughout the day.	partly-cloudy-day
Partly Cloudy	Light rain in the morning and overnight.	partly-cloudy-day
Possible Drizzle	Rain until morning, starting again in the evening.	rain
Partly Cloudy	Light rain in the morning and overnight.	partly-cloudy-night
Overcast	Foggy in the morning.	cloudy
Partly Cloudy	Foggy in the morning.	partly-cloudy-day
Overcast	Overcast throughout the day.	cloudy
Possible Drizzle	Rain throughout the day.	rain
Light Rain	Light rain in the morning and overnight.	rain
Partly Cloudy	Mostly cloudy throughout the day.	partly-cloudy-day
Overcast	Light rain until evening.	cloudy
Partly Cloudy	Partly cloudy throughout the day.	partly-cloudy-night
Possible Drizzle	Light rain until evening.	rain

After Cleanup:

insert into weather (weather_condition,weather_description,weather_icon)
select short_summary,long_summary,icon from
(select distinct short_summary,long_summary,icon,avg(TemperatureMin),avg(temperatureMax) from src_lyft_landing group by short_summary,icon) a;

short_summary	long_summary	icon	avg(TemperatureMin)	avg(temperatureMax)
Mostly Cloudy	Rain throughout the day.	partly-cloudy-night	31.904423249667804	43.85028824306477
Rain	Rain until morning, starting again in the evening.	rain	39.683693383129544	46.344232500235904
Clear	Light rain in the morning.	clear-night	29.369600359564892	42.74187010749777
Partly Cloudy	Mostly cloudy throughout the day.	partly-cloudy-night	34.12685752206854	46.62503024877893
Overcast	Light rain in the morning and overnight.	cloudy	34.17374989735552	45.4804468465191
Light Rain	Light rain until evening.	rain	36.83301052284618	47.94656034199222
Foggy	Foggy in the morning.	fog	41.99593453273189	53.101564217891216
Clear	Mostly cloudy throughout the day.	clear-day	29.290577212649318	44.6215367241119
Mostly Cloudy	Light rain in the morning.	partly-cloudy-day	31.965853368067354	44.31025976926093
Partly Cloudy	Partly cloudy throughout the day.	partly-cloudy-day	31.44725200906756	44.48855429631445
Possible Drizzle	Rain until morning, starting again in the evening.	rain	37.150646679881575	46.86935827551908
Drizzle	Rain until morning, starting again in the evening.	rain	35.669427836710945	42.57522018643464

2. CSV to DB import: All text fields had space at the end in their data. Cleaned it across all tables.
Note: This was identified after loading the data across all normalized tables.
Fix: Remove trailing and preceding spaces for each varchar field.

Before cleanup:

```
SELECT * FROM homeworks2.weather;
select  quote(weather_icon),quote(weather_description),quote(weather_condition) from weather;
```

quote(weather_icon)	quote(weather_description)	quote(weather_condition)
' partly-cloudy-night '	' Rain throughout the day. '	' Mostly Cloudy '
' rain '	' Rain until morning, starting again in the evenin...	' Rain '
' clear-night '	' Light rain in the morning. '	' Clear '
' partly-cloudy-night '	' Mostly cloudy throughout the day. '	' Partly Cloudy '
' cloudy '	' Light rain in the morning and overnight. '	' Overcast '
' rain '	' Light rain until evening. '	' Light Rain '
' fog '	' Foggy in the morning. '	' Foggy '
' clear-day '	' Mostly cloudy throughout the day. '	' Clear '
' partly-cloudy-day '	' Light rain in the morning. '	' Mostly Cloudy '
' partly-cloudy-day '	' Partly cloudy throughout the day. '	' Partly Cloudy '
' rain '	' Rain until morning, starting again in the evenin...	' Possible Drizzle '
' rain '	' Rain until morning, starting again in the evenin...	' Drizzle '

After Cleanup:

```
SELECT * FROM homeworks2.weather;
select  quote(weather_icon),quote(weather_description),quote(weather_condition) from weather;
update weather set weather_icon = rtrim(ltrim(weather_icon)),weather_description=rtrim(ltrim(weather_description)),weather_condition=rtrim(ltrim(weather_condition));
```

quote(weather_icon)	quote(weather_description)	quote(weather_condition)
'partly-cloudy-night'	'Rain throughout the day.'	'Mostly Cloudy'
'rain'	'Rain until morning, starting again in the evening.'	'Rain'
'clear-night'	'Light rain in the morning.'	'Clear'
'partly-cloudy-night'	'Mostly cloudy throughout the day.'	'Partly Cloudy'
'cloudy'	'Light rain in the morning and overnight.'	'Overcast'
'rain'	'Light rain until evening.'	'Light Rain'
'fog'	'Foggy in the morning.'	'Foggy'
'clear-day'	'Mostly cloudy throughout the day.'	'Clear'
'partly-cloudy-day'	'Light rain in the morning.'	'Mostly Cloudy'
'partly-cloudy-day'	'Partly cloudy throughout the day.'	'Partly Cloudy'
'rain'	'Rain until morning, starting again in the evening.'	'Possible Drizzle'
'rain'	'Rain until morning, starting again in the evening.'	'Drizzle'

The statistical analysis has been taken care of in step 3.
The error encountering and fixing part has been taken care of in step 4.

Strategies for business development:

- Based on the weather, cabs are booked more frequently on overcast days, but surge multiplier and the average price remains the same across all weather. So, we suggest a 1.25x surge multiplier will yield 80% profit for overcast day trips.
- Based on the day-to-day data, relatively higher number of cabs were booked during Mondays and Tuesdays from noon till midnight and thus, increasing the surge multiplier to 1.25x would yield more revenue.
- Based on the total pricing it can be concluded that although the booking counts are lower on the weekends, the revenue generated, and the average distance travelled is considerably higher.
- Based on the cab type and the weather data, it is conspicuous that during overcast and mostly cloudy days, regardless of the cab product type, the bookings are more than 50%. Usually, the cab products are normally distributed however, a slightly higher booking fraction is found for the Lyft Luxury vehicle during the aforementioned weathers.