DATABASE DESIGN PART II

IMPACT OF THE COVID-19 EPEDEMIC ON THE REAL ECONOMY

Xintong Li | INSY 661 | 2021-9-10

A Brief Introduction for this project

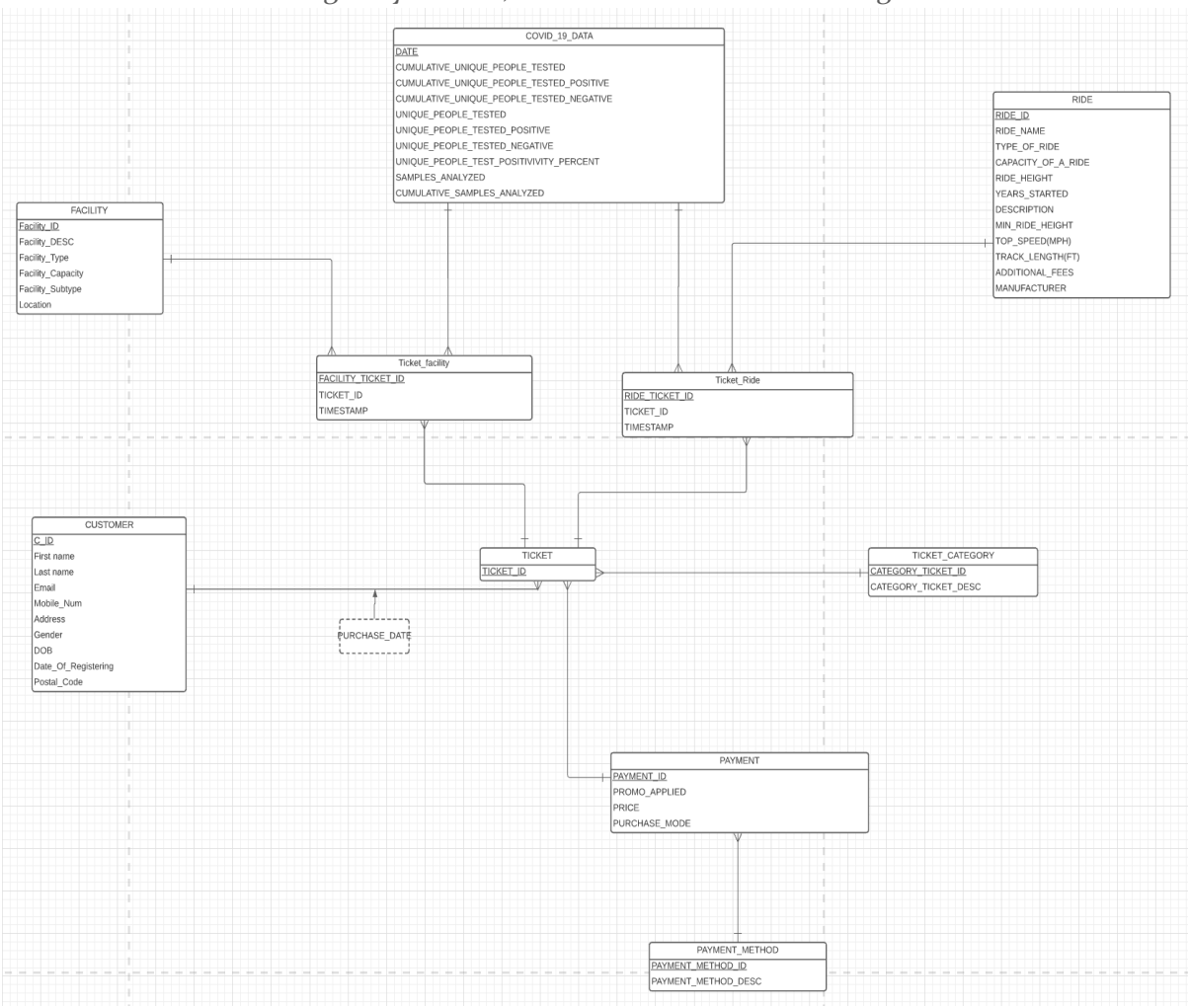
In this project, I will show how to clean and process the data crawled by the website and the data provided by the company/amusement park, how to build a ERD using them and how to implement the ERD to a real-world database from scratch.

This report is for the second part of the project, in this part I will introduce how to get the opensource covid-19 data, and how to connect it with our current La Ronde database, to analyze the impact of the epidemic on a real economy like amusement parks.

COVID 19 Data Reference:
https://github.com/ccodwg/Covid19Canada/tree/master/official_datasets/qc

1. ERD

- I extracted the ticket_facility, and ticket_ride to be tables this time, as I need to use them to connect with Covid19-data. Note before, they are many-to-many relationships. As learned in class, this is an legitimate alternative way of representing many-to-many relationship.
- In my Covid 19-dataset, there is covid data everyday for Quebec province, so every element in the date column is distinct, hence I use date as the primary key for covid19-data table. (the data is found in https://github.com/ccodwg/Covid19Canada)
- The dotted boxes represent attributes for relationships, the arrow on it shows to which relationship it belongs to.
- Assume that one can purchase many tickets in one payment, but one ticket can only belong to one payment.
- The underline attributes represents primary key.
- I did not include foreign key in ERD, but it will be shown in the logical model.



2. Logical Model

- **Entities:**

CUSTOMER(C_ID, FIRST_NAME, LAST_NAME, EMAIL, MOBILE_NUM, ADDRESS, GENDER, DOB, DATE_OF_REGISTERING, POSTAL_CODE)

FACILITY(FACILITY_ID, FACILITY_DESC, FACILITY_TYPE, FACILITY_CAPACITY, FACILITY_SUBTYPE, LOCATION)

RIDE(RIDE_ID, RIDE_NAME, TYPE_OF_RIDE, CAPACITY_OF_A_RIDE, RIDE_HEIGHT, YEARS_STARTED, DESCRIPTION, MIN_RIDE_HEIGHT, TOP_APEED(MPH), TRACK_LENGTH(FT), ADDITIONAL_FEES, MANUFACTURER)

TICKET_CATEGORY(CATEGORY_TICKET_ID, CATEGORY_TICKET_DESC)

PAYMENT(PAYMENT_ID, PAYMENT_METHOD_ID, PROMO_APPLIED, PRICE, PURCHASE_MODE)

PAYMENT_METHOD(PAYMENT_METHOD_ID, PAYMENT_METHOD_DESC)

TICKET(TICKET_ID, C_ID, CATEGORY_TICKET_ID, PAYMENT_METHOD_ID, PURCHASE_DATE)

TICKET_FACILITY(FACILITY_TICKET_ID, DATE, TICKET_ID, TIMESTAMP)

TICKET_RIDE(RIDE_TICKET_ID, DATE, TICKET_ID, TIMESTAMP)

COVID19_DATA(DATE, CUMULATIVE_UNIQUE_PEOPLE_TESTED, CUMULATIVE_UNIQUE_PEOPLE_TESTED_POSITIVE, CUMULATIVE_UNIQUE_PEOPLE_TESTED_NEGATIVE, UNIQUE_PEOPLE_TESTED, UNIQUE_PEOPLE_TESTED_POSITIVE, UNIQUE_PEOPLE_TESTED_NEGATIVE, UNIQUE_PEOPLE_TEST_POSITIVITY_PERCENT, SAMPLES_ANALYZED, CUMULATIVE_SAMPLES_ANALYZED)

3. Queries

- DDL part

- **Note:** You are not able to get all the raw data in your database just by simply running these queries, because the data given by the company is so messy that I must use python, excel and SQL to cleaned and preprocessed them for each table. After that, I also manually imported the cleaned csv files. So it is not possible that you could run these sql queries and get everything done in one second, please understand this.
- But I will also submit the cleaned data, too. They are in csv format, they are named in this format : '**XXX_with_external_data.csv**'. if you want to see my data, you could first run the create table part then import those csv files I submitted.
- Besides these, I will also submit the python file which are used to clean the data incase if you would like to look them. The python files for this part **contain "COVID" in their names**, that's how you could find them.

```sql
CREATE TABLE CUSTOMER (
  C_ID VARCHAR(100) NOT NULL,
  FIRST_NAME VARCHAR(100),
  LAST_NAME VARCHAR(100),
  EMAIL VARCHAR(100),
  MOBILE_NUM VARCHAR(100),
  ADDRESS VARCHAR(100),
  GENDER VARCHAR(100),
  DOB DATE,
  DATE_OF_REG DATE,
  POSTAL_CODE VARCHAR(100),
  PRIMARY KEY (C_ID)
);

CREATE TABLE FACILITY (
  FACILITY_ID VARCHAR(100) NOT NULL,
  FACILITY_DESC VARCHAR(1000),
  FACILITY_TYPE VARCHAR(100),
  FACILITY_CAPACITY INT,
  FACILITY_SUBTYPE VARCHAR(1000),
  LOCATION VARCHAR(100),
  PRIMARY KEY (FACILITY_ID)
);

CREATE TABLE PAYMENT_METHOD (
  PAYMENT_METHOD_ID INT NOT NULL,
  PAYMENT_METHOD_DESC VARCHAR(100),
  PRIMARY KEY (PAYMENT_METHOD_ID)
);
```

```sql
CREATE TABLE PAYMENT (
  PAYMENT_ID INT NOT NULL,
  PAYMENT_METHOD_ID INT,
  PROMO_APPLIED INT(1),
  PRICE INT,
  PURCHASE_MODE VARCHAR(100),
  PRIMARY KEY (PAYMENT_ID),
  FOREIGN KEY (PAYMENT_METHOD_ID) REFERENCES PAYMENT_METHOD(PAYMENT_METHOD_ID)
);

CREATE TABLE TICKET_CATEGORY (
  CATEGORY_OF_TICKET_ID INT(1) NOT NULL,
  CATEGORY_OF_TICKET_DESC VARCHAR(100),
  PRIMARY KEY (CATEGORY_OF_TICKET_ID)
);

CREATE TABLE RIDE (
  RIDE_ID VARCHAR(100) NOT NULL,
  RIDE_NAME VARCHAR(100),
  TYPE_OF_RIDE VARCHAR(100),
  CAPACITY_OF_A_RIDE DOUBLE,
  RIDE_HEIGHT DOUBLE,
  YEARS_STARTED DOUBLE,
  DESCRIPTION VARCHAR(10000),
  MIN_RIDE_HEIGHT VARCHAR(1000),
   MANUFACTURER VARCHAR(100),
  `TOP_SPEED(MPH)` DOUBLE,
  `TRACK_LENGTH(FT)` DOUBLE,
  ADDITIONAL_FEES VARCHAR(1),
  PRIMARY KEY (RIDE_ID)
);
```

```sql
CREATE TABLE TICKET (
  TICKET_ID INT NOT NULL,
  C_ID VARCHAR(100),
  PAYMENT_ID INT NOT NULL,
  `CATEGORY_OF_TICKET_ID` INT(1),
    PURCHASE_DATE DATE,
  PRIMARY KEY (TICKET_ID, C_ID, PAYMENT_ID, `CATEGORY_OF_TICKET_ID`, PURCHASE_DATE),
  FOREIGN KEY (PAYMENT_ID) REFERENCES PAYMENT(PAYMENT_ID),
  FOREIGN KEY (C_ID) REFERENCES CUSTOMER(C_ID),
  FOREIGN KEY (`CATEGORY_OF_TICKET_ID`) REFERENCES `ticket_category`(`CATEGORY_OF_TICKET_ID`)
);



CREATE TABLE COVID19_DATA(
`date` DATE,
`CUMULATIVE_UNIQUE_PEOPLE_TESTED` INT,
`CUMULATIVE_UNIQUE_PEOPLE_TESTED_POSITIVE` INT,
`CUMULATIVE_UNIQUE_PEOPLE_TESTED_NEGATIVE` INT,
`UNIQUE_PEOPLE_TESTED` INT,
`UNIQUE_PEOPLE_TESTED_POSITIVE` INT,
`UNIQUE_PEOPLE_TESTED_NEGATIVE` INT,
`UNIQUE_PEOPLE_TESTED_POSITIVITY_PERCENT` DOUBLE,
`SAMPLES_ANALYZED` INT,
 `CUMULATIVE_SAMPLES_ANALYZED` INT,
PRIMARY KEY (`DATE`)
);
```

```sql
CREATE TABLE TICKET_RIDE(
RIDE_TICKET_ID INT NOT NULL,
RIDE_ID VARCHAR(100),
TICKET_ID INT,
TIMESTAMP DATETIME,
`date` DATE,
PRIMARY KEY (RIDE_TICKET_ID),
FOREIGN KEY (TICKET_ID) REFERENCES TICKET(TICKET_ID),
FOREIGN KEY (RIDE_ID) REFERENCES RIDE(RIDE_ID),
FOREIGN KEY (`date`) REFERENCES COVID19_DATA(`date`)
);

CREATE TABLE TICKET_FACILITY(
FACILITY_TICKET_ID INT NOT NULL,
TICKET_ID INT,
FACILITY_ID VARCHAR(100),
TIMESTAMP DATETIME,
`date` DATE,
PRIMARY KEY (FACILITY_TICKET_ID),
FOREIGN KEY (TICKET_ID) REFERENCES TICKET(TICKET_ID),
FOREIGN KEY (FACILITY_ID) REFERENCES FACILITY(FACILITY_ID),
FOREIGN KEY (`date`) REFERENCES COVID19_DATA(`date`)
);
```

/*the following is the query to manipulate Covid19_data table*/
/* I first use python to upload covid data to covid19_data_2 table,
then use sql to insert covid19_data from covid19_data_2*/
/*I will also submit all python files I used*/

```sql
INSERT INTO
RIDE
(RIDE_ID,`RIDE_NAME`,`TYPE_OF_RIDE`,`CAPACITY_OF_A_RIDE`,`RIDE_HEIGHT`
,`YEARS_STARTED`,
`DESCRIPTION`,`MIN_RIDE_HEIGHT`,`MANUFACTURER`,`TOP_SPEED(MPH)`,`TRA
CK_LENGTH(FT)`,`ADDITIONAL_FEES`)
SELECT
RIDE_ID,`RIDE_NAME` ,`TYPE_OF_RIDE` ,`CAPACITY_OF_A_RIDE`,`RIDE_HEIGHT
`,`YEARS_STARTED` ,
`DESCRIPTION`,`MIN_RIDE_HEIGHT`,`MANUFACTURER`,`TOP_SPEED(MPH)`,`TRA
CK_LENGTH(FT)`,`ADDITIONAL_FEES`
FROM la_ronde_amusement_park.ride2;
```

```
INSERT INTO
covid19_data
(`date`,`cumulative_unique_people_tested`,`cumulative_unique_people_tested_p
ositive`,`cumulative_unique_people_tested_negative`,`unique_people_tested`,`u
nique_people_tested_positive`,`unique_people_tested_negative`,`unique_peopl
e_tested_positivity_percent`,`samples_analyzed`,`cumulative_samples_analyzed`
)
SELECT
`date`,`cumulative_unique_people_tested`,`cumulative_unique_people_tested_p
ositive`,`cumulative_unique_people_tested_negative`,`unique_people_tested`,`u
nique_people_tested_positive`,`unique_people_tested_negative`,`unique_peopl
e_tested_positivity_percent`,`samples_analyzed`,`cumulative_samples_analyzed`
FROM covid19_data_2;
```

- **5 complex queries part**
- **NOTE:** it is impossible to get these queries to run unless you already have the correct the tables and data in the database, please understand this. For information about how to get the correct data in the database, please refer to the previous DDL section. If you are still having trouble getting the data into your database, feel free to leave me a message, I will be very happy to help you import them and get the queries run on your computer. (if needed, please leave me a message using this number: 4389277966)

1. Find the total number of visit to dinning facility, and amount of cumulative unique people tested positive, for each month from 2020 Jan to 2020 Aug. The restaurant industry should be the most seriously affected by the epidemic, and you can see if this is really the case with this query.
   - **Code:**

```
select x.TotalVisitToDinning, x.perMonth, y.cumulative_positive_cases from (
select sum(FACILITY_TICKET_ID) as TotalVisitToDinning, month(date) as
perMonth from(
select ticket_facility.date, facility.FACILITY_TYPE,
ticket_facility.FACILITY_TICKET_ID from ticket_facility
inner join facility on ticket_facility.FACILITY_ID = facility.FACILITY_ID
```

```
where facility.FACILITY_TYPE = 'Dinning'
) as a
group by month(date)
) as x
inner join (
select max(cumulative_unique_people_tested_positive) as
cumulative_positive_cases, month(date) as perMonth
 from covid19_data
 group by month(date)
)as y
on x.perMonth = y.perMonth;
```

- ## Output:

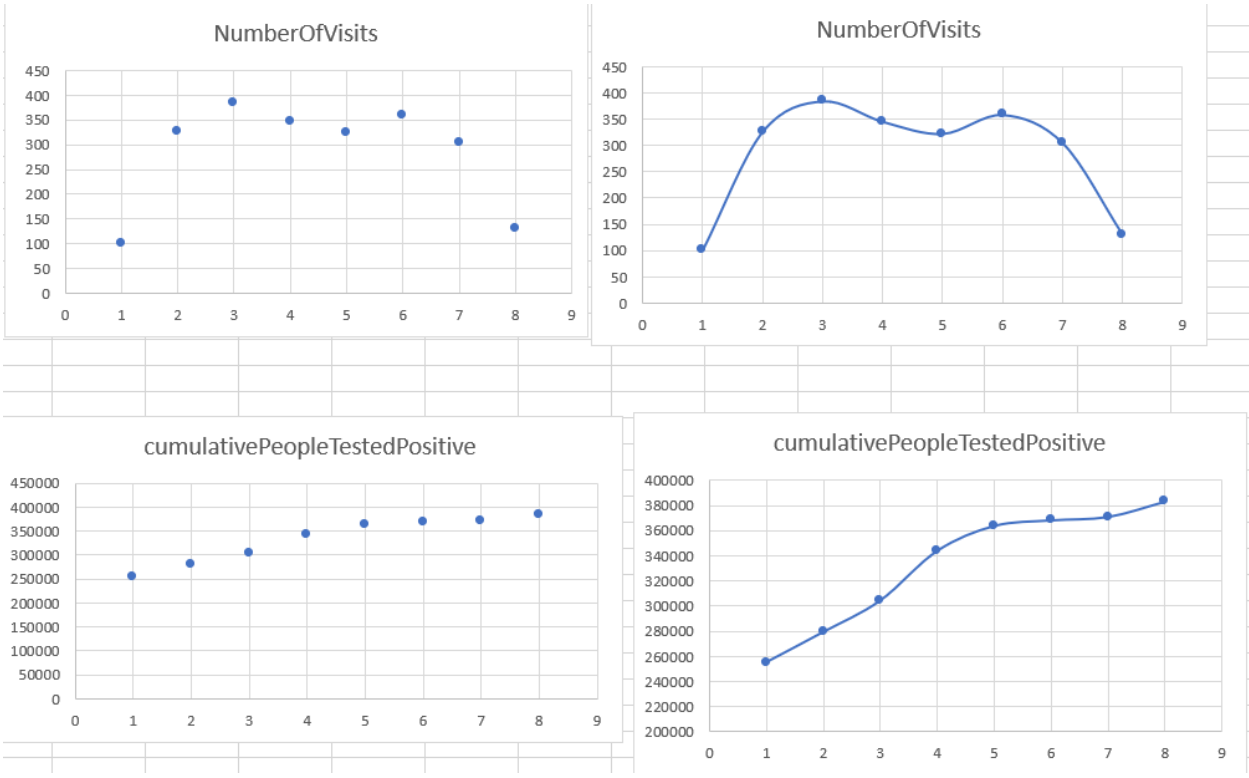
	perMonth	TotalVisitToDinning	cumulative_positive_cases
▶	1	212261	255086
	2	685732	279214
	3	789230	304270
	4	717819	343892
	5	655706	363749
	6	684799	368200
	7	618327	370910
	8	239953	383097

- ## Conclusion:

I visualized the data I found above:
The first one below is about the trend of Number of visits to dinning facility, the second one is about the trend of cumulative people tested positive (I know that in theory I should use scatterplot rather than line chart, because there might not be linear relation between them, but I want us to see the trends more clearly, so I also included the line chart version)

As we can see from the graphs below, the period of the sharpest rise of Number of Visits to dinning facility is Month 1-2, which is the least severe period of the epidemic.
And the first decline of Number of Visits to dinning is Month 3-5, which also happens to be the period of beginning of the increasing rise of number of cumulative tested positive.

Lastly, the deepest decline of Number of visits to dinning facility happens in month 7-8, which is also the period that the number of cumulative tested positive reached its peak.
In summary, it is difficult to deny the hypothesis that the Covid-19 epidemic has no impact on the restaurant industry, which means that the Covid-19 epidemic is likely to lead to a depression in the restaurant industry.



2. Check the total number of elderly customers visiting the amusement park facilities and rides from January to August, as the elderly are more prone to contracting the new coronavirus, and this query can be used to find out whether the elderly have a positive attitude towards outbreak protection. (note: in this question, we regard people older than 50 as elderly customers)

- ## Code:

```
select x.perMonth, y.totalVisitOfOld, x.Cumulative_people_tested_positive
from (
select max(CUMULATIVE_UNIQUE_PEOPLE_TESTED_POSITIVE) AS
Cumulative_people_tested_positive , MONTH(date) as perMonth
from covid19_data
group by month(date)
) as x
```

```
inner join (
select (visitsToFacility + visitsToRide) as totalVisitOfOld, `month(date)` as
perMonth from(
 select sum(FACILITY_TICKET_ID) as visitsToFacility, sum(RIDE_TICKET_ID)
as visitsToRide, month(date) from (
select a.FACILITY_TICKET_ID, a.date, a.RIDE_TICKET_ID,
year(current_date())-year(a.DOB) as age,
covid19_data.CUMULATIVE_UNIQUE_PEOPLE_TESTED_POSITIVE from (
select ticket_facility.FACILITY_TICKET_ID, ticket_facility.date,
ticket_ride.RIDE_TICKET_ID, customer.DOB from ticket inner join
ticket_facility on ticket.TICKET_ID = ticket_facility.TICKET_ID inner join
ticket_ride on ticket_ride.TICKET_ID = ticket.TICKET_ID inner join customer
on ticket.C_ID = customer.C_ID) as a
inner join covid19_data on a.date = covid19_data.date
where (year(current_date())-year(DOB)) > 50
) as b
group by month(date)
) as c
)as y
on x.perMonth = y.perMonth;
```

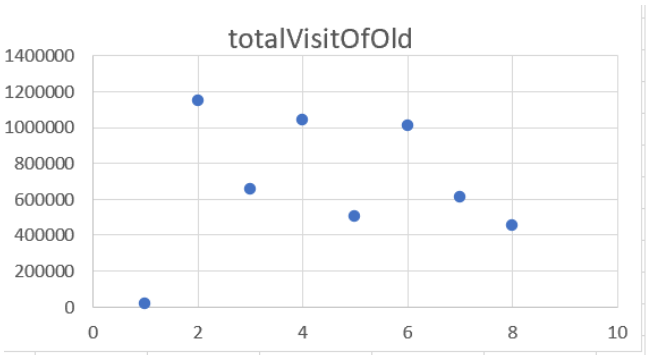
- **Output:**

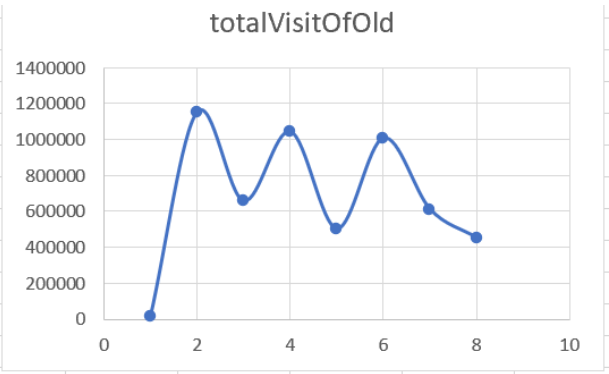
	perMonth	totalVisitOfOld	Cumulative_people_tested_positive
▶	1	18587	255086
	2	1150983	279214
	3	658798	304270
	4	1043590	343892
	5	505141	363749
	6	1008676	368200
	7	612137	370910
	8	454681	383097

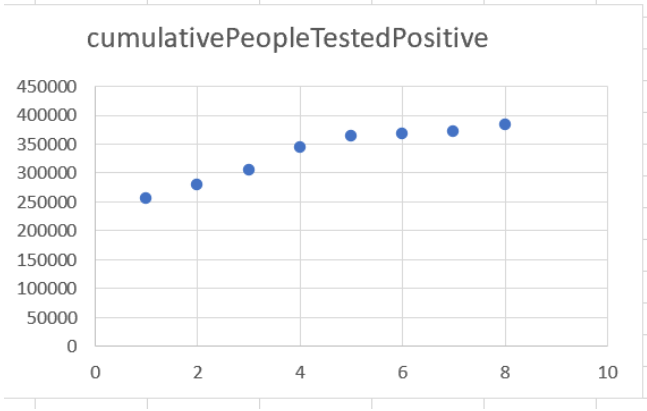
- **Conclusion:**

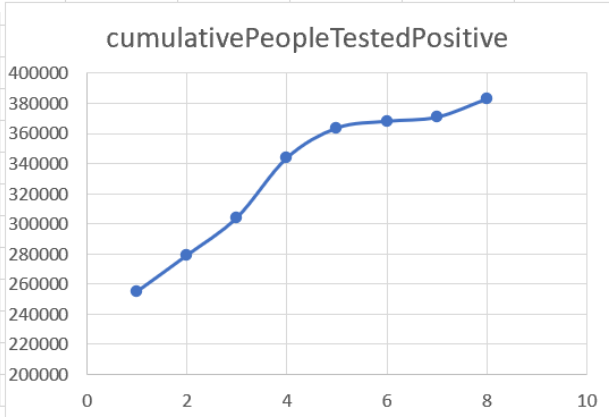
Unlike the previous question, here we cannot see whether there is a significant relationship between the frequency of elderly people going to the amusement

park and the epidemic, and under the assumption that people do not wear masks when they go to the amusement park, the elderly are not doing a good job of protecting themselves from the epidemic. Therefore, it is recommended that amusement parks should be strictly controlled and every customer should be checked to see if they are wearing a mask, and it is also recommended that the government should increase the knowledge of epidemic protection for the elderly.









3. Find the worst period of the epidemic from January to August 2020 and find the top 3 rides that still have the highest visit during that period. With this query we can study the most attractive rides in the amusement park and increase the investment in these two items later after the epidemic is revived.

- **Code:**

```
select RIDE.RIDE_ID, RIDE.RIDE_NAME, RIDE.TYPE_OF_RIDE from (
select SUM(RIDE_TICKET_ID) as Number_Visit_During_CovidPeak, RIDE_ID
from (
select ridesInWorst.date, ride.RIDE_ID, ride.RIDE_NAME,
ride.TYPE_OF_RIDE, ridesInWorst.RIDE_TICKET_ID from (
select * from ticket_ride
where month(date) = (
select worstPeriod from (
```

```
select b.most_cases, c.perMonth as worstPeriod from (
select max(cumulative_positive_cases) as most_cases from (
select max(cumulative_unique_people_tested_positive) as
cumulative_positive_cases, month(date) as perMonth
 from covid19_data
 where month(date)<'9'
 group by month(date)
 ) as a
 ) as b
 inner join (
 select max(cumulative_unique_people_tested_positive) as
cumulative_positive_cases, month(date) as perMonth
 from covid19_data
 group by month(date)
 )as c
 on b.most_cases = c.cumulative_positive_cases
)as worstmonth
 )
) as ridesInWorst
inner join ride
on ridesInWorst.RIDE_ID = RIDE.RIDE_ID
)as R
group by R.RIDE_ID
order by Number_Visit_During_CovidPeak DESC
limit 3
) as N
inner join ride on N.RIDE_ID = RIDE.RIDE_ID;
```

- ## Output:

	RIDE_ID	RIDE_NAME	TYPE_OF_RIDE
▶	R040	Vertigo	Thrill
	R031	Sling Shot	Thrill
	R037	Tour de Ville	Family

- ## Conclusion:

We identified the three rides that remained the most popular during the worst period of the new crown epidemic and recommended that similar rides be added to the amusement park subsequently.

4. See the types of tickets whose sales were most affected in the four months following the outbreak. This is because it is presumed that the number of people buying annual tickets should have plummeted after the outbreak. This query allows us to verify whether our conjecture is correct and, if so, the amusement park can make adjustments accordingly during the outbreak, such as raising the price of daily and parking tickets and lowering the price of annual passes.
   - **Code:**

```
select (NumberSoldInApril - NumberSoldInJan) as IncreaseFromJanToApril,
Jan.CATEGORY_OF_TICKET_DESC from (
SELECT COUNT(TICKET_ID) NumberSoldInApril,
CATEGORY_OF_TICKET_DESC FROM (
SELECT * FROM (
SELECT C.CATEGORY_OF_TICKET_ID, C.MONTH,
C.CUMULATIVE_UNIQUE_PEOPLE_TESTED_POSITIVE,
TICKET_CATEGORY.CATEGORY_OF_TICKET_DESC , C.TICKET_ID FROM (
SELECT A. TICKET_ID, A.CATEGORY_OF_TICKET_ID, A.MONTH,
B.CUMULATIVE_UNIQUE_PEOPLE_TESTED_POSITIVE FROM (
select TICKET_ID, CATEGORY_OF_TICKET_ID, MONTH(PURCHASE_DATE)
AS MONTH
FROM TICKET
)AS A
INNER JOIN (
SELECT CUMULATIVE_UNIQUE_PEOPLE_TESTED_POSITIVE,
MONTH(DATE) AS MONTH
FROM covid19_data
)AS B
ON A.MONTH = B.MONTH
) AS C
INNER JOIN ticket_category ON C.CATEGORY_OF_TICKET_ID =
ticket_category.CATEGORY_OF_TICKET_ID
)AS D
 WHERE MONTH = '4'
 ) AS E
 GROUP BY CATEGORY_OF_TICKET_DESC
 ORDER BY NumberSoldInApril DESC
)as April
inner join (
SELECT COUNT(TICKET_ID) NumberSoldInJan,
CATEGORY_OF_TICKET_DESC FROM (
SELECT * FROM (
```

```
SELECT C.CATEGORY_OF_TICKET_ID, C.MONTH,
C.CUMULATIVE_UNIQUE_PEOPLE_TESTED_POSITIVE,
TICKET_CATEGORY.CATEGORY_OF_TICKET_DESC , C.TICKET_ID FROM (
SELECT A. TICKET_ID, A.CATEGORY_OF_TICKET_ID, A.MONTH,
B.CUMULATIVE_UNIQUE_PEOPLE_TESTED_POSITIVE FROM (
select TICKET_ID, CATEGORY_OF_TICKET_ID, MONTH(PURCHASE_DATE)
AS MONTH
FROM TICKET
)AS A
INNER JOIN (
SELECT CUMULATIVE_UNIQUE_PEOPLE_TESTED_POSITIVE,
MONTH(DATE) AS MONTH
FROM covid19_data
)AS B
ON A.MONTH = B.MONTH
) AS C
INNER JOIN ticket_category ON C.CATEGORY_OF_TICKET_ID =
ticket_category.CATEGORY_OF_TICKET_ID
)AS D
 WHERE MONTH = '1'
 ) AS E
 GROUP BY CATEGORY_OF_TICKET_DESC
 ORDER BY NumberSoldInJan DESC
)as Jan
on April.CATEGORY_OF_TICKET_DESC =
Jan.CATEGORY_OF_TICKET_DESC;
```

- **Output:**

	IncreaseFromJanToApril	CATEGORY_OF_TICKET_DESC
▶	980	Daily Pass
	760	Annual pass
	380	Parking ticket

- **Conclusion:**

From January to April, which is the beginning of the outbreak, we can see a significant increase in the number of people buying daily tickets during this period.
The most affected ticket type is the parking ticket, but there is no clear correlation between the increase of the epidemic and the number of people buying annual passes.
Therefore, it is suggested that the amusement park can raise the price of daily tickets and lower the price of parking tickets appropriately during the epidemic, and the price of annual tickets can remain unchanged.

5. See if people are more likely to use online purchase method to buy tickets in the month when the epidemic is most severe according to the number of cumulative test positive within the period of Jan to Aug, 2020. This is because it is assumed that people's spending patterns were affected by the epidemic and that the epidemic meant a springtime for e-commerce. If so, the epidemic did affect people's spending patterns and the park could make adjustments, such as firing redundant ticketing staff and hiring more technical staff to optimize the park's online ticketing system.

- **CODE:**

```
select count(PAYMENT_ID) AS NumberOfPayments, PURCHASE_MODE
from (
select ticket.PAYMENT_ID, ticket.purchase_date, payment.purchase_mode
from ticket
inner join payment on ticket.PAYMENT_ID = payment.PAYMENT_ID
where month (purchase_date) = (
select worstPeriod from (
select b.most_cases, c.perMonth as worstPeriod from (
select max(cumulative_positive_cases) as most_cases from (
select max(cumulative_unique_people_tested_positive) as
cumulative_positive_cases, month(date) as perMonth
 from covid19_data
 where month(date)<'9'
 group by month(date)
 ) as a
 ) as b
 inner join (
 select max(cumulative_unique_people_tested_positive) as
cumulative_positive_cases, month(date) as perMonth
```

```
 from covid19_data
 group by month(date)
 )as c
 on b.most_cases = c.cumulative_positive_cases
)as worstmonth
)
)as O
group by purchase_mode
order by NumberOfPayments DESC;
```

- Output:

	NumberOfPayments	purchase_mode
▶	20	Online
	16	Offline

- Conclusion:

It's true that more people chose to buy tickets online during the month when the Covid19 epidemic was at its worst. But a significant number of people still continued to use offline ticketing. So it is suggested that the number of ticket sellers can be appropriately reduced on the amusement park side, but there is no need to lay off a large number of staff.