Ziye Zhang 260766101

External Data Info

The weather data I used was from climate.weather.gc.ca; the holiday data I collected was from Morland and Talwar from GitHub.

The external data is used to get pattern of how weathers and days off correlate with customer's passion and willingness to access rides and facilities.

For weather, I set 5 weather conditions: Rainy, Snowy, Moderate, Very Hot, and Very Cold.

- Rainy days (ID=2) are those having Total Rain > 15mm,
- Snowy days (ID=3) are those having Total Snow > 5cm,
- Very Cold days (ID=4) are those having mean temperature < -10C,
- Very Hot days (ID=5) are those having mean temperature > 29C,
- All others that do not fulfill the conditions above are Moderate days (ID=1).

Limitations:

- The metrics or cut-offs were a combination of multiple weather websites, but people can perceive differently and set other cut-offs; for example, people may perceive very Cold days to have a temperature lower than -15C.
- Other conditions are not included such as cloudy or windy. We make the assumption that they have less impact than the extreme 5 weather conditions above.
- Daily weather changes are not captured in this data. However, this original dataset has accurate numbers such as Total Rain (mm) and Total Snow (cm) so that further continuous-variable statistical analysis can be implemented.

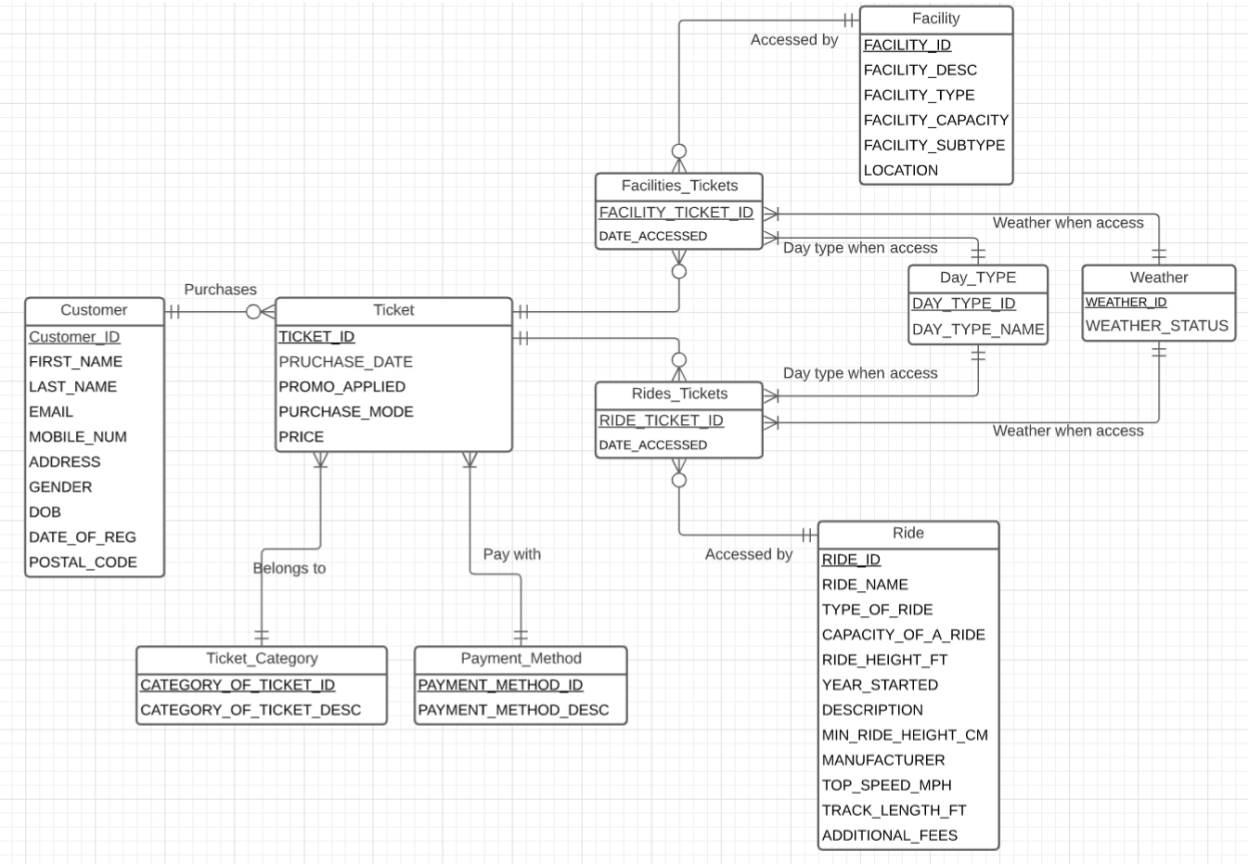
For day type, I set three types:

- Weekday (ID=1),
- Weekend (ID=2),
- Holiday (ID=3).

We predict people have different levels of preference towards amusement park on different types of days. For example, people may like to visit La Ronde during holidays and weekends.

All processing for the additional data is done through Excel. Take holidays/weekends for example, the formula to determine day type id is
'=IF(COUNTIF($I$1:$I$20,D2)>0,$M$4,IF(OR(WEEKDAY(D2)=1,WEEKDAY(D2)=7),$M$3,$M$2))'

ERD



Logical Model

Customer (**CUSTOMER_ID**, FIRST_NAME, LAST_NAME, EMAIL, MOBILE_NUM, ADDRESS, GENDER, DOB, DATE_OF_REG, POSTAL_CODE)

PAYMENT_METHOD (**PAYMENT_METHOD_ID**, PAYMENT_METHOD_DESC)

TICKET_CATEGORY (**CATEGORY_OF_TICKET_ID**, CATEGORY_OF_TICKET_DESC)

Ticket (**TICKET_ID**, CUSTOMER_ID, PURCHASE_DATE, PROMO_APPLIED, CATEGORY_OF_TICKET_ID, PAYMENT_METHOD_ID, PURCHASE_MODE, PRICE)

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_FT, YEAR_STARTED, DESCRIPTION, MIN_RIDE_HEIGHT_CM, MANUFACTURER, TOP_SPEED_MPH, TRACK_LENGTH_FT, ADDITIONAL_FEES)

Weather (**WEATHER_ID**, WEATHER_TYPE)

DAY_TYPE (**DAY_TYPE_ID**, DAY_TYPE)

Facilities_Tickets (**FACILITY_TICKET_ID**, FACILITY_ID, TICKET_ID, DATE_ACCESSED, DAY_TYPE_ID, WEATHER_ID)

Rides_Tickets (**RIDE_TICKET_ID**, RIDE_ID, TICKET_ID, DATE_ACCESSED, DAY_TYPE_ID, WEATHER_ID)

Additional Assumptions:

- A date must have one and only one day type, i.e., either weekday or weekend or holiday.
- A date must have one and only one weather, i.e., one of moderate, rainy, snowy, too hot, and too cold.

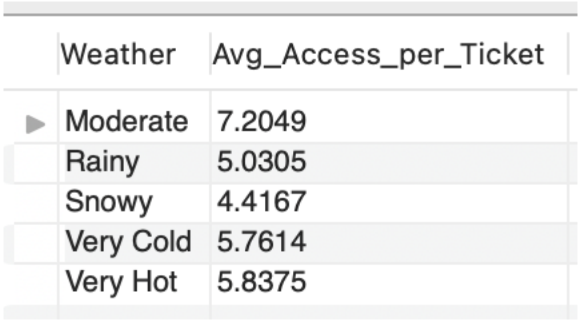
Query: 1
Objective:
Find out if customers' accesses to rides and facilities are affected by weathers conditions.

Code:

```
SELECT
  weather.weather_type AS Weather,
  rtft.access_per_Ticket AS Avg_Access_per_Ticket
FROM
  ((SELECT
    weather_ID,
      COUNT(Ticket_ID) / COUNT(DISTINCT (Ticket_ID)) AS access_per_ticket
  FROM
    (SELECT
    Ride_ID AS service_ID, Ticket_ID, Weather_ID
  FROM
    rides_tickets UNION ALL SELECT
    Facility_ID, Ticket_ID, Weather_ID
  FROM
    facilities_tickets) rft
  GROUP BY weather_ID) rtft)
```

```
    JOIN
  weather ON weather.weather_ID = rtft.Weather_ID;
```

Output screenshot:



Weather	Avg_Access_per_Ticket
Moderate	7.2049
Rainy	5.0305
Snowy	4.4167
Very Cold	5.7614
Very Hot	5.8375

Query: 2

Objective:

Find out people of which gender are affected more by the bad weathers.

Assumption:

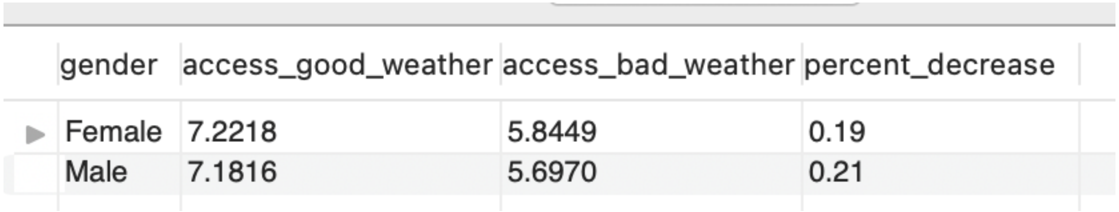
Bad weathers include all weathers except moderate.

Code:

```
select a.gender,a.access_per_ticket as access_good_weather, b.access_per_ticket as
access_bad_weather,
format((a.access_per_ticket-b.access_per_ticket)/a.access_per_ticket,2) as percent_decrease
from (
select gender,count(ticket_ID)/count(distinct(ticket_ID)) as access_per_ticket from
(select c.customer_ID,c.gender,t.ticket_ID,rft.service_ID,rft.Weather_ID from ticket t join
customer c on c.customer_ID=t.customer_ID join (SELECT
    Ride_ID AS service_ID, Ticket_ID, Weather_ID
  FROM
    rides_tickets UNION ALL SELECT
    Facility_ID, Ticket_ID, Weather_ID
  FROM
    facilities_tickets) rft on rft.Ticket_ID=t.ticket_ID where weather_ID=1) w group by
gender) a join
(select gender,count(ticket_ID)/count(distinct(ticket_ID)) as access_per_ticket from
(select c.customer_ID,c.gender,t.ticket_ID,rft.service_ID,rft.Weather_ID from ticket t join
customer c on c.customer_ID=t.customer_ID join (SELECT
    Ride_ID AS service_ID, Ticket_ID, Weather_ID
  FROM
    rides_tickets UNION ALL SELECT
    Facility_ID, Ticket_ID, Weather_ID
  FROM
```

```
    facilities_tickets) rft on rft.Ticket_ID=t.ticket_ID where weather_ID!=1) w group by
gender) b on a.gender=b.gender;
```

Output screenshot:



gender	access_good_weather	access_bad_weather	percent_decrease
Female	7.2218	5.8449	0.19
Male	7.1816	5.6970	0.21

Query: 3

Objective: Determine if weather affects more accesses to rides than facilities (since many rides are outdoor and many facilities are indoor)

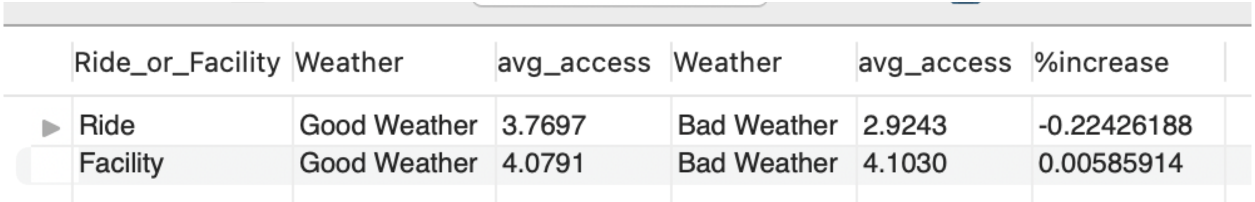
Code:

```
create view xxRide as
select 'Ride','Good Weather' as Weather,count(ticket_ID)/count(distinct(ticket_ID)) as
avg_access from Rides_tickets where weather_ID=1  union
select 'Ride','Bad Weather',count(ticket_ID)/count(distinct(ticket_ID)) as avg_access  from
Rides_tickets where weather_ID!=1;

create view xxFacility as
select 'Facility','Good Weather' as Weather,count(ticket_ID)/count(distinct(ticket_ID)) as
avg_access from Facilities_tickets where weather_ID=1 union
select 'Facility','Bad Weather',count(ticket_ID)/count(distinct(ticket_ID)) from
Facilities_tickets where weather_ID!=1;

(select x.Ride as Ride_or_Facility,x.Weather,x.avg_access,y.Weather,y.avg_access,
(y.avg_access-x.avg_access)/x.avg_access as '%increase' from xxRide x join xxRide y on
x.Ride=y.Ride and x.Weather!=y.Weather limit 1) union (
select x.Facility,x.Weather,x.avg_access,y.Weather,y.avg_access, (y.avg_access-
x.avg_access)/x.avg_access as '%increase' from xxFacility x join xxFacility y on
x.Facility=y.Facility and x.Weather!=y.Weather limit 1);
```

Output screenshot:



Ride_or_Facility	Weather	avg_access	Weather	avg_access	%increase
Ride	Good Weather	3.7697	Bad Weather	2.9243	-0.22426188
Facility	Good Weather	4.0791	Bad Weather	4.1030	0.00585914

Query: 4

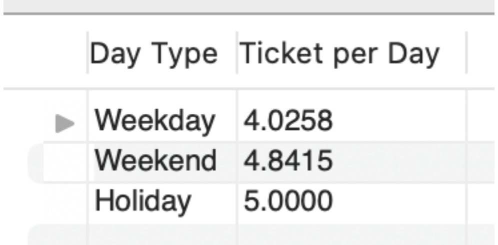
Objective:

Find out if people are more willing to buy tickets on holidays and weekends than on weekdays.

Code:

```
SELECT
  Day_Type.Day_type 'Day Type',
  dt.ticket_per_day 'Ticket per Day'
FROM
  (SELECT
    Date_type_ID,
      COUNT(DISTINCT (Ticket_ID)) / COUNT(DISTINCT (Date_accessed)) ticket_per_day
  FROM
    (SELECT
    Ride_ID AS service_ID,
      Ticket_ID,
      Date_accessed,
      Date_type_ID
  FROM
    rides_tickets UNION ALL SELECT
    Facility_ID, Ticket_ID, Date_accessed, Date_type_ID
  FROM
    facilities_tickets) rft
  GROUP BY Date_Type_ID) dt
    JOIN
  Day_type ON Day_type.Day_type_ID = dt.Date_type_ID;
```

Output screenshot:



Day Type	Ticket per Day
Weekday	4.0258
Weekend	4.8415
Holiday	5.0000

Query: 5

Objective: Find out the most popular activity for each of the three types of day

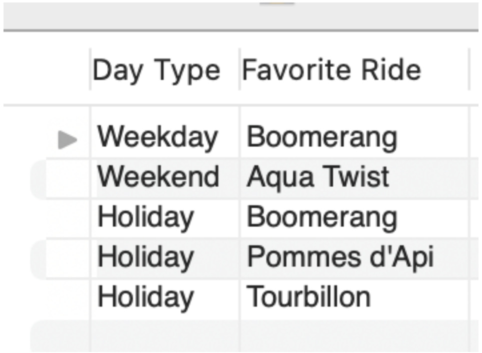
Assumptions:

Include activities that have the same number.

Code:

```
select dt.day_type 'Day Type', r.ride_name 'Favorite Ride' from (
select a.Date_Type_ID, a.max_count,b.ride_ID from (
select Date_Type_ID, max(count) max_count from (select Date_Type_ID,
ride_id,count(ride_id) count from Rides_Tickets group by date_type_id, ride_id) a group by
Date_Type_ID)a
join (select Date_Type_ID,
ride_id,count(ride_id) count from Rides_Tickets group by date_type_id, ride_id) b
on a.Date_Type_ID=b.Date_Type_ID and a.max_count=b.count) c
join day_type dt on dt.day_type_id=c.date_type_id
join Ride r on r.Ride_ID=c.Ride_ID;
```

Output screenshot:



Day Type	Favorite Ride
Weekday	Boomerang
Weekend	Aqua Twist
Holiday	Boomerang
Holiday	Pommes d'Api
Holiday	Tourbillon