INSY 661

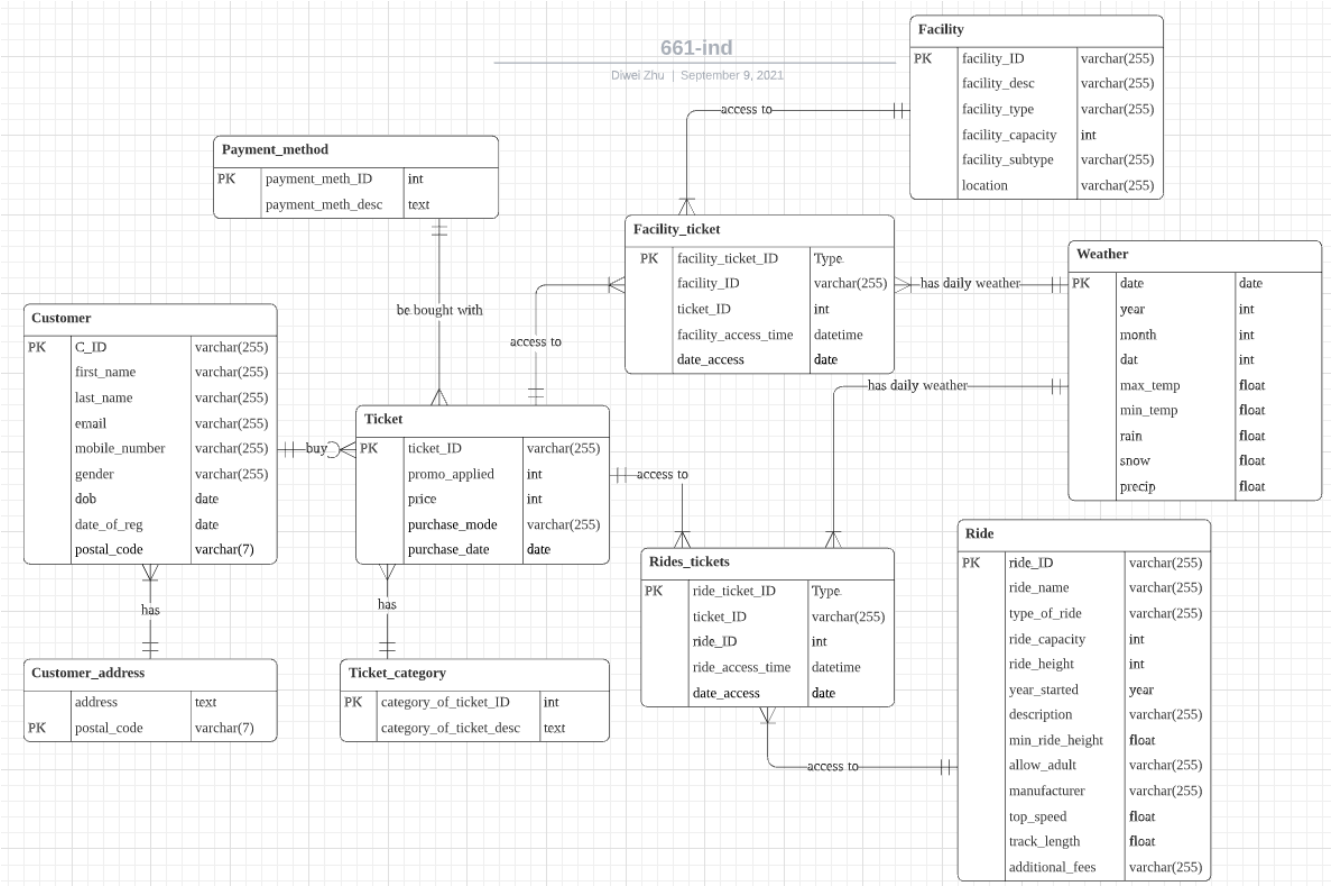
Individual Project – Report 2: External Data

1. Introduction

The external data I used is the historical weather data retrived from the online databased of Canadian government at https://climate.weather.gc.ca/historical_data/search_historic_data_e.html.

The historical weather dataset included daily weather data of Montreal from 2016 to 2020.

2. ERD



3. Logical Model

4. Queries

Query with External data 1_____

Objective: to calculate the proportion of sunny days in all days across the four years when there was visitor at La Ronde.

Fact: in the provided dataset, there were dates in a year when there was no visitor at all.

Code:

```
SELECT
AVG(CASE WHEN w.precip = 0 THEN 1
   WHEN w.precip > 0 THEN 0 ELSE NULL END ) AS sunny_days_proportion,
1-AVG(CASE WHEN w.precip = 0 THEN 1
```

```
    WHEN w.precip > 0 THEN 0 ELSE NULL END ) AS rainy_snowy_days_proportion
  FROM rides_tickets AS rt
  INNER JOIN weather AS w ON w.date = DATE(rt.ride_access_time);
```

Output:

	sunny_days_proportion	rainy_snowy_days_proportion
▶	0.6090	0.3910

Query with External data 2_____

Objective: respectively generate the number of visitors that visited La Ronde on sunny days, rainy or snowy days, or on days that had no weather record.

Assumption: One visiting record means one customer came to the park on one date.

Code:

```
  SELECT
    COUNT(CASE WHEN w.precip = 0 THEN 1 ELSE NULL END) AS visit_sunny,
      COUNT(CASE WHEN w.precip > 0 THEN 1 ELSE NULL END) AS
visit_snowy_or_rainy,
      COUNT(CASE WHEN w.precip = 0 THEN 1 ELSE NULL END)-COUNT(CASE
WHEN w.precip > 0 THEN 1 ELSE NULL END) AS visit_no_record
  FROM rides_tickets AS rt
  INNER JOIN weather AS w ON w.date = DATE(rt.ride_access_time);
```

Output:

	visit_sunny	visit_snowy_or_rainy	visit_no_record
▶	2419	1553	866

Query with External data 3_____

Objective: to find the visiting records of customers who were not afraid of heatwaves and visited La Ronde on dates where the daily min temperatures were higher than the average min temperature of July, 2020.

Code:

```
SELECT CONCAT(c.first_name, " ", c.last_name) AS customer_name, w.date, w.min_temp,
w.max_temp

FROM rides_tickets AS rt

INNER JOIN Weather AS w ON w.date = DATE(rt.ride_access_time)

INNER JOIN Ticket AS t ON t.ticket_ID = rt.ticket_ID

INNER JOIN Customer AS c ON c.C_ID = t.customer_ID

WHERE w.min_temp > (SELECT AVG(min_temp)

                    FROM Weather

                    WHERE year = 2017 AND month = 7

                                        )

GROUP BY rt.ticket_ID, DATE(rt.ride_access_time)

ORDER BY w.min_temp DESC;
```

Output:

customer_name	date	min_temp	max_temp
Vinnie Brownhill	2020-07-10	24	36.1
Elijah Ruperto	2020-07-10	24	36.1
Michelle Tirrell	2020-07-10	24	36.1
Jaimie Hanster	2020-07-10	24	36.1
Renell Weinmann	2020-07-10	24	36.1
Helaina Biddles	2020-07-10	24	36.1
Kimmi Peddie	2020-07-10	24	36.1
Ania Tibbetts	2020-07-10	24	36.1
Jock Chasier	2020-07-27	23.3	30.5
Jeannine Lanchbery	2020-07-27	23.3	30.5
Akim Preddle	2020-07-27	23.3	30.5
Nicolais Boycott	2020-07-27	23.3	30.5
Cherish Mowbray	2020-07-27	23.3	30.5
Colver Jira	2020-07-27	23.3	30.5
Elora Cuss	2020-07-27	23.3	30.5

Query with External data 4_____

Objective: with the persistent interest in our loyal customer, Berta Drewery, this query aims to find the weather types of the dates when she made her trips to La Ronde.

Code:

```
ALTER TABLE Weather
```

```
ADD COLUMN weather_type varchar(255);

SET SQL_SAFE_UPDATES = 0;


UPDATE Weather

SET weather_type = 'sunny' WHERE precip = 0;


UPDATE Weather

SET weather_type = 'rain' WHERE rain > 0 AND snow = 0;


UPDATE Weather

SET weather_type = 'snow' WHERE snow > 0 AND rain = 0;


UPDATE Weather

SET weather_type = 'rain and snow' WHERE snow > 0 AND rain > 0;


SELECT CONCAT(c.first_name, " ", c.last_name) AS customer_name, w.weather_type,
DATE(rt.ride_access_time) AS access_date

FROM rides_tickets AS rt

INNER JOIN Weather AS w ON w.date = DATE(rt.ride_access_time)

INNER JOIN Ticket AS t ON t.ticket_ID = rt.ticket_ID

INNER JOIN Customer AS c ON c.C_ID = t.customer_ID

WHERE c.first_name = 'Berta' AND c.last_name = 'Drewery'

GROUP BY rt.ticket_ID, DATE(rt.ride_access_time);
```

Output:

	customer_name	weather_type	access_date
▶	Berta Drewery	sunny	2020-01-29
	Berta Drewery	sunny	2020-01-30
	Berta Drewery	rain	2019-11-22
	Berta Drewery	rain	2019-11-21
	Berta Drewery	rain	2020-06-01
	Berta Drewery	sunny	2020-06-02
	Berta Drewery	sunny	2020-06-18
	Berta Drewery	rain	2020-04-29
	Berta Drewery	rain	2020-04-30
	Berta Drewery	sunny	2020-04-06
	Berta Drewery	sunny	2020-04-05
	Berta Drewery	rain	2020-04-29
	Berta Drewery	rain	2020-04-30
	Berta Drewery	rain and snow	2020-03-17
	Berta Drewery	sunny	2020-03-16

Query with External data 5_____

Objective: to filter out the facilities of which the popularity is above average across all years.

Assumption: the "average of popularity" is an average value of popularity (visit times) of facilities across all years.

Code:

```
SELECT CONCAT(c.first_name, " ", c.last_name) AS customer_name,
        DATE(rt.ride_access_time) AS access_date,
                w.max_temp,
         w.weather_type,
                w.rain AS rain_precipitation,
                w.snow AS snow_precipitation
FROM rides_tickets AS rt
INNER JOIN Weather AS w ON w.date = DATE(rt.ride_access_time)
INNER JOIN Ticket AS t ON t.ticket_ID = rt.ticket_ID
INNER JOIN Customer AS c ON c.C_ID = t.customer_ID
WHERE w.max_temp < (
            SELECT AVG(max_temp)
            FROM Weather
```

```
        WHERE year = 2016 AND month = 12
    )
AND (w.weather_type = 'snow' OR w.weather_type = 'rain and snow')
GROUP BY rt.ticket_ID, DATE(rt.ride_access_time)
ORDER BY max_temp;
```

Output:

customer_name	access_date	max_temp	weather_type	rain_precipitation	snow_precipitation
Alfonso Gurnell	2020-01-18	-13.3	snow	0	12.4
Doreen Barthelme	2020-01-18	-13.3	snow	0	12.4
Karmen Burbage	2020-01-18	-13.3	snow	0	12.4
Ugo Hanner	2020-01-18	-13.3	snow	0	12.4
Lilian Yatman	2020-01-18	-13.3	snow	0	12.4
Mariska Bruneau	2020-01-18	-13.3	snow	0	12.4
Yelena Triggs	2020-01-18	-13.3	snow	0	12.4
Rik Heppenspall	2020-01-18	-13.3	snow	0	12.4
Anders Wake	2020-01-18	-13.3	snow	0	12.4
Alfonso Gurnell	2020-01-19	-8.6	snow	0	3.8
Doreen Barthelme	2020-01-19	-8.6	snow	0	3.8
Oliy Hotchkin	2020-01-19	-8.6	snow	0	3.8
Cinnamon Bruneton	2020-01-19	-8.6	snow	0	3.8
Filippo Tosdevin	2020-01-19	-8.6	snow	0	3.8
Mariska Bruneau	2020-01-19	-8.6	snow	0	3.8