



DESAUTELS

**Faculty of Management
Faculté de gestion**

INDIVIDUAL PROJECT

Due on September 10th, 2021

INSY 661: Database and Distributed Systems for Analytics

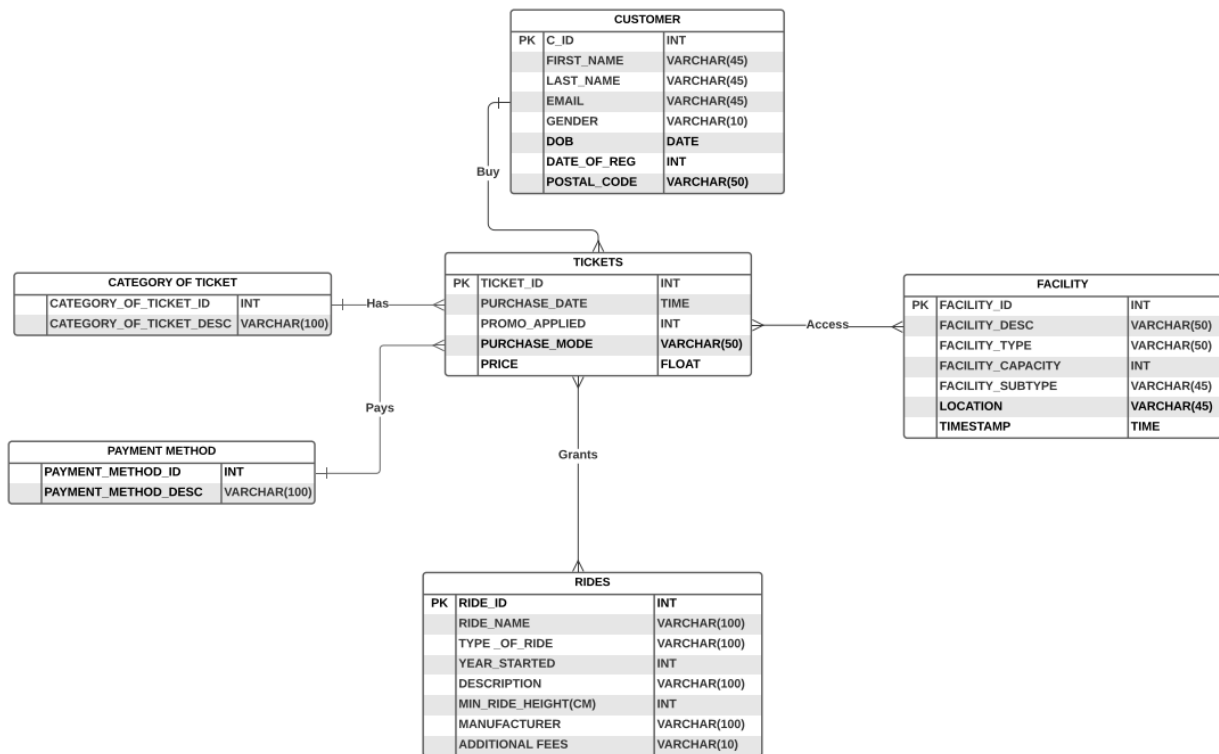
Professor Animesh Animesh

Summer 2021

Student Name	Student ID
Mesaye Bahiru	260634934

SECTION-1 ERD

La Ronde DATABASE ERD



Assumptions: Category of Ticket and Payment Method should be seen as a weak entity because they lacked unique primary keys. Tickets can be seen as a parent for these entities.

SECTION-2: RELATIONAL MODEL

Foreign keys represented in *

CUSTOMER (C_ID, FIRST_NAME, LAST_NAME, EMAIL, GENDER, DOB,
DATE_OF_REG, POSTAL_CODE)
PK: C_ID
FK: N/A

CATEGORY OF TICKET (CATEGORY_OF_TICKET_ID,
CATEGORY_OF_TICKET_DESC)
PK: N/A
FK: N/A

PAYMENT METHOD (PAYMENT_METHOD_ID, PAYMENT_METHOD_DESC)
PK: N/A
FK: N/A

TICKETS (TICKET_ID, PURCHASE_DATE, PROMO_APPLIED, PURCHASE_MODE,
PRICE, C_ID*, CATEGORY_OF_TICKET_ID*, PAYMENT_METHOD_ID*)

PK: TICKET_ID
FK: C_ID references CUSTOMER (C_ID)

CATEGORY_OF_TICKET_ID references
CATEGORY_OF_TICKET (CATEGORY_OF_TICKET_ID)

PAYMENT_METHOD_ID references PAYMENT_METHOD (PAYMENT_METHOD_ID)

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

PK: FACILITY_ID
FK: N/A

TICKETS_FACILITY (TICKET_ID*, FACILITY_ID*)

PK: TICKET_ID, FACILITY_ID
FK: TICKET_ID references TICKETS (TICKET_ID)
FACILITY_ID references FACILITY (FACILITY_ID)

RIDES (RIDE_ID, RIDE_NAME, TYPE_OF_RIDE, YEAR_STARTED, DESCRIPTION,
MIN_RIDE_HEIGHT(CM), MANUFACTURER, ADDITIONAL FEES)
PK: RIDE_ID
FK: N/A

TICKETS_RIDES (TICKET_ID*, RIDE_ID*)

PK: TICKET_ID, RIDE_ID
FK: TICKET_ID references TICKETS (TICKET_ID)
RIDE_ID references RIDES (RIDE_ID)

DDL

```
CREATE TABLE `CUSTOMER` (  
  `C_ID` INT,  
  `FIRST_NAME` VARCHAR (45),  
  `LAST_NAME` VARCHAR (45),  
  `EMAIL` VARCHAR (45),  
  `GENDER` VARCHAR (10),  
  `DOB` DATE,  
  `DATE_OF_REG` INT,  
  `POSTAL_CODE` VARCHAR (50),  
  PRIMARY KEY (`C_ID`)  
);
```

```
CREATE TABLE `CATEGORY OF TICKET` (  
  `CATEGORY_OF_TICKET_ID` INT,  
  `CATEGORY_OF_TICKET_DESC` VARCHAR (100),  
);
```

```
CREATE TABLE `PAYMENT METHOD` (  
  `PAYMENT_METHOD_ID` INT,  
  `PAYMENT_METHOD_DESC` VARCHAR (100),  
);
```

```
CREATE TABLE `TICKETS` (  
  `TICKET_ID` INT,  
  `PURCHASE_DATE` TIME,  
  `PROMO_APPLIED` INT,  
  `PURCHASE_MODE` VARCHAR (50),  
  `PRICE` FLOAT,  
  `C_ID` INT,  
  `CATEGORY_OF_TICKET_ID` INT,  
  `PAYMENT_METHOD_ID` INT,
```

```
PRIMARY KEY (`TICKET_ID`),  
FOREIGN KEY(`C_ID`) REFERENCES CUSTOMER (C_ID)  
FOREIGN KEY(`CATEGORY_OF_TICKET_ID`) REFERENCES `CATEGORY OF  
TICKET` (`CATEGORY_OF_TICKET_ID`)  
FOREIGN KEY(`PAYMENT_METHOD_ID`) REFERENCES `PAYMENT METHOD`  
(`PAYMENT_METHOD_ID`)
```

```
);
```

```
CREATE TABLE `FACILITY` (  
  `FACILITY_ID` INT,  
  `FACILITY_DESC` VARCHAR (50),  
  `FACILITY_TYPE` VARCHAR (50),  
  `FACILITY_CAPACITY` INT,  
  `FACILITY_SUBTYPE` VARCHAR (45),  
  `LOCATION` VARCHAR (100),  
  `TIMESTAMP` TIME,  
  PRIMARY KEY (`FACILITY_ID`)  
);
```

```
CREATE TABLE `RIDES` (  
  `RIDE_ID` INT,  
  `RIDE_NAME` VARCHAR (100),  
  `TYPE_OF_RIDE` VARCHAR (100),  
  `YEAR_STARTED` INT,  
  `DESCRIPTION` VARCHAR (100),  
  `MIN_RIDE_HEIGHT(CM)` INT,  
  `MANUFACTURER` VARCHAR (100),  
  `ADDITIONAL FEES` VARCHAR (10),  
  PRIMARY KEY (`RIDE_ID`)  
);
```

```
CREATE TABLE `TICKETS_FACILITY` (  
  `TICKET_ID` INT,  
  `FACILITY_ID` INT,  
  PRIMARY KEY (TICKET_ID, FACILITY_ID),  
  FOREIGN KEY (TICKET_ID) REFERENCES TICKETS (TICKET_ID),  
  FOREIGN KEY (FACILITY_ID) REFERENCES FACILITY (FACILITY_ID)  
);
```

```
CREATE TABLE `TICKETS_RIDES` (
  `TICKET_ID` INT,
  `RIDE_ID` INT,
  PRIMARY KEY (TICKET_ID, RIDE_ID),
  FOREIGN KEY (TICKET_ID) REFERENCES TICKETS (TICKET_ID),
  FOREIGN KEY (RIDE_ID) REFERENCES RIDES (RIDE_ID)
);
```

SECTION-3: Populate the data

Normalization Process:

- Used python pandas to perform data cleaning steps
 - See the attached python file

Populate data


































- Using MAMP built-in function to insert normalized data into the dedicated database
- See the attached SQL file to verify the populated of data.

SECTION-4: Queries

#1

Business objective:

To send targeted adverting, promotion, or discount for students. We can check the email address with educational domain. This information is used to determine and locate potential clients.

<pre>1 SELECT DISTINCT EMAIL, C_ID, FIRST_NAME 2 FROM CUSTOMER1 3 WHERE EMAIL LIKE "%edu"</pre>				
<input checked="" type="checkbox"/> Enable foreign key checks <input type="button" value="Go"/> <input type="button" value="Cancel"/>				
<input type="checkbox"/> Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]				
<input type="checkbox"/> Show all Number of rows: 25 Filter rows: Search this table Sort by key: None				
<div> <div>+ Options</div> <div> <div> <div></div> <div>▼ EMAIL</div> </div> <div> <div>C_ID</div> <div>FIRST_NAME</div> </div> </div> </div>				
<input type="checkbox"/>	 Edit	 Copy	 Delete	tbridewell1@tamu.edu
<input type="checkbox"/>	 Edit	 Copy	 Delete	afeaney2@stanford.edu
<input type="checkbox"/>	 Edit	 Copy	 Delete	gwetherill@cmu.edu
<input type="checkbox"/>	 Edit	 Copy	 Delete	macocks@washington.edu
<input type="checkbox"/>	 Edit	 Copy	 Delete	wfranzolt@berkeley.edu
<input type="checkbox"/>	 Edit	 Copy	 Delete	kdelhanty15@columbia.edu
<input type="checkbox"/>	 Edit	 Copy	 Delete	jmeachem1g@psu.edu
<input type="checkbox"/>	 Edit	 Copy	 Delete	mwofford1x@washington.edu
<input type="checkbox"/>	 Edit	 Copy	 Delete	cjra2a@harvard.edu
<input type="checkbox"/>	 Edit	 Copy	 Delete	bhanuward3d@psu.edu
<input type="checkbox"/>	 Edit	 Copy	 Delete	tsannes3s@cornell.edu

Business objective:

To check which payment type clients frequently use. The result shows online payment is commonly used as a form of payment. Interestingly, cash is the second form of payment used by the customers.

SELECT PAYMENT_METHOD_DESC, COUNT(*) AS "Payment Form" FROM payment_method GROUP BY PAYMENT_METHOD_DESC

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Refresh](#)]

☐ Show all | Number of rows: 25 ▼ Filter rows:

+ Options

PAYMENT_METHOD_DESC	Payment Form
Cash	258
Credit Card	242
Debit Card	236
Online Payment	264

#3

Business objective:

To provide detail information on specific facility. This information can be used to organize an event and schedule activities accordingly. In this case if we want to see the summary of FAC102

[illegible]

#4

Business objective:

Using advanced subquery to see customers who paid more than the average price. This information can help decide on future price determination process.

```
1 SELECT CUSTOMER_ID, PURCHASE_DATE, PRICE
2 FROM 'TICKET1'
3 WHERE PRICE > (SELECT AVG(PRICE) AS "average price" FROM TICKET1)
4 ORDER BY PURCHASE_DATE DESC
```

☒ Enable foreign key checks

Go Cancel

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

1 > >> ☐ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

				CUSTOMER_ID	PURCHASE_DATE	PRICE
<input type="checkbox"/>	Edit	Copy	Delete	CD0027	31-07-2020	700
<input type="checkbox"/>	Edit	Copy	Delete	CD0053	31-07-2020	800
<input type="checkbox"/>	Edit	Copy	Delete	CD0009	31-05-2020	800
<input type="checkbox"/>	Edit	Copy	Delete	CD0133	31-05-2020	800
<input type="checkbox"/>	Edit	Copy	Delete	CD0097	31-03-2020	700
<input type="checkbox"/>	Edit	Copy	Delete	CD0035	31-01-2020	800
<input type="checkbox"/>	Edit	Copy	Delete	CD0106	30-11-2019	700
<input type="checkbox"/>	Edit	Copy	Delete	CD0025	30-07-2020	700
<input type="checkbox"/>	Edit	Copy	Delete	CD0114	30-07-2020	700
<input type="checkbox"/>	Edit	Copy	Delete	CD0082	30-06-2020	800

#5

Business objective:

To know more about the tickets sold and which category of ticket sold the most.

```
SELECT COUNT(CATEGORY_OF_TICKET_ID) AS "total ticket sold", CATEGORY_OF_TICKET_DESC FROM category_of_ticket GROUP BY 2 ASC ORDER BY 1
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 Filter rows: Search this table

total ticket sold	CATEGORY_OF_TICKET_DESC
313	Parking ticket
343	Annual pass
344	Daily Pass

#6

Business objective:

For more depth knowledge of our client, it is important to check the gender distribution. This provides an opportunity to tailor our products based on the needs.

```
select count(case when gender='Male' then 1 end) as male_count, count(case when gender='Female' then 1 end) as female_count, count(*) as total_gender_count from CUSTOMER1
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 Filter rows: Search this table

male_count	female_count	total_gender_count
61	89	150

#7

Business objective: To maximize profit, it is important to check which top clients who are paying more for our products, how they are paying, and which category of tickets brings more revenue for the park. The data shows annual pass is essential for the business success.

```
SELECT TICKET1.CUSTOMER_ID, TICKET1.PURCHASE_MODE, TICKET1.CATEGORY_OF_TICKET_DESC, TICKET1.PRICE FROM TICKET1 LEFT JOIN CUSTOMER1 ON TICKET1.CUSTOMER_ID = CUSTOMER1.C_ID WHERE PRICE > 500 LIMIT 10
```

☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

+ Options

CUSTOMER_ID	PURCHASE_MODE	CATEGORY_OF_TICKET_DESC	PRICE
CD0143	Online	Annual pass	700
CD0046	Online	Annual pass	800
CD0078	Online	Annual pass	800
CD0147	Offline	Annual pass	700
CD0131	Offline	Annual pass	800
CD0135	Online	Annual pass	800
CD0048	Offline	Annual pass	700
CD0142	Online	Annual pass	700
CD0028	Online	Annual pass	800
CD0002	Online	Annual pass	700

#8

Business objective:

Nowadays, many businesses are adjusting their venue size due to covid-19 protocol and if we want to select facilities that can host 20 or less people, we can run this query to find the exact information.

```
1 | SELECT Tickets_Facility1.FACILITY_ID, Facility_norm.FACILITY_CAPACITY, Facility_norm.FACILITY_DESC
2 | FROM Tickets_Facility1
3 | LEFT JOIN Facility_norm ON Tickets_Facility1.FACILITY_ID = Facility_norm.FACILITY_ID
4 | WHERE Facility_norm.FACILITY_CAPACITY <= 20
```

☒ Enable foreign key checks

[Go](#) [Cancel](#)

☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

1 > >> Number of rows: 25 Filter rows: Search this table

+ Options

FACILITY_ID	FACILITY_CAPACITY	FACILITY_DESC
FAC115	20	Pizza Ronde 3
FAC116	20	Poulet Etc.
FAC141	20	Popcorn & Cie
FAC102	20	Marchand Du Village
FAC129	20	Halte Gourmande - Au Bol
FAC104	20	Carroussel Du Bonbon
FAC112	20	La Centrale Burgers & Frites
FAC117	20	Restaurant Lafleur
FAC117	20	Restaurant Lafleur
FAC132	20	Amir
FAC104	20	Carroussel Du Bonbon
FAC119	20	Dippin' Dots
FAC104	20	Carroussel Du Bonbon
FAC138	20	Popcorn & Cie
FAC127	20	Bonbon 3

#9

Business objective:

To examine customers behaviour and track their purchasing habits in the last two years. We can run the query as shown below. The data shows people more likely to spend more in summer season.

```
SELECT CUSTOMER_ID, PURCHASE_DATE, SUM(PRICE) AS "Total Price" FROM TICKET1 WHERE PURCHASE_DATE BETWEEN "08-08-2019" AND "08-08-2021" GROUP BY 1,2
```

☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

☐ Show all | Number of rows: 25 Filter rows: Search this table

Options

				CUSTOMER_ID	PURCHASE_DATE	Total Price
<input type="checkbox"/>	Edit	Copy	Delete	CD0001	08-08-2020	700
<input type="checkbox"/>	Edit	Copy	Delete	CD0033	08-08-2020	20
<input type="checkbox"/>	Edit	Copy	Delete	CD0107	08-08-2020	20

#10

Business objective:

Location is important for business success and if we want to see if shopping stores are located strategically, we can run this query below. Research shows that business who are placed at entrance perform better. Thus, this data suggests that we need to add more shopping stores with different subtype close to front entrances.

SELECT

FACILITY_TYPE, FACILITY_SUBTYPE, LOCATION

FROM

Facility_norm

WHERE

FACILITY_TYPE = "Shopping"

Profiling

[\[Edit inline \]](#)
[\[Edit \]](#)
[\[Explain SQL \]](#)
[\[Create PHP code \]](#)
[\[Refresh \]](#)

Show all

Number of rows:

25

Filter rows:

Search this table

























Sort by key:

None

+ Options

←

→

				FACILITY_TYPE	FACILITY_SUBTYPE	LOCATION
<input type="checkbox"/>				Shopping	Toys/Apparel/Sundries	Front entrance
<input type="checkbox"/>				Shopping	Toys/Apparel/Sundries	Secteur Village
<input type="checkbox"/>				Shopping	Photos and Collectables	Near the Goliath
<input type="checkbox"/>				Shopping	Candy	Front entrance
<input type="checkbox"/>				Shopping	Candy	Secteur Village
<input type="checkbox"/>				Shopping	Apparel	Fort Edmonton
<input type="checkbox"/>				Shopping	Apparel	Next to Spirale
<input type="checkbox"/>				Shopping	Apparel	Next to Splash