Shawn Yat Sin, Ethan Lim, Jing-Ling Chen, Lucas Chew

Group 49

Professor ElSaadawy and Professor Kemme

Comp 421

March 20th, 2024

I. Relational Schema

Entities

RegularAccount(<u>email</u>, username, password, phoneNumber)

- Everything should be not null

BusinessAccount(<u>email</u>, username, password, phoneNumber)

- Everything should be not null

Restaurant(<u>restaurantId</u>, name, phoneNumber, <u>ownerEmail</u>)

- Everything should be not null
- ownerEmail is the foreign key to BusinessAccount

Review(<u>reviewId</u>, rating, comment, postedAt, <u>accountEmail</u>, <u>restaurantId</u>)

- Everything should be not null
- accountEmail is the foreign key to RegularAccount
- restaurantId is the foreign key to Restaurant

MenuItem(<u>itemId</u>, name, price, description)

- Everything should be not null

Weak Entities

Address(street, <u>unit</u>, city, province, <u>zipCode</u>, <u>restaurantId</u>)

- Everything should be not null
- restaurantId is the foreign key to Restaurant

BusinessHour(<u>day</u>, openTime, closeTime, <u>restaurantId</u>)

- openTime and closeTime may be null if the restaurant is closed
- Everything else should be not null
- restaurantId is the foreign key to Restaurant

Menu(<u>menuName</u>, <u>restaurantId</u>, isValid)

- Everything should be not null
- restaurantId is the foreign key to Restaurant

Relationships

AccountFollowing(account, follower)

- account and follower are both foreign keys for RegularAccount

Recommendation(accountEmail, restaurantId)

- accountEmail is the foreign key to RegularAccount
- restaurantId is the foreign key to Restaurant

Reservation(date, numberOfPeople, isValid, accountEmail, restaurantId)

- accountEmail is the foreign key to RegularAccount
- restaurantId is the foreign key to Restaurant

ContainsMenuItem(menuName, restaurantId, itemId)

- menuName is the foreign key for Menu
- restaurantId is the foreign key to Restaurant
- itemId is the foreign key for MenuItem

II. Stored Procedures

Stored Procedure 1

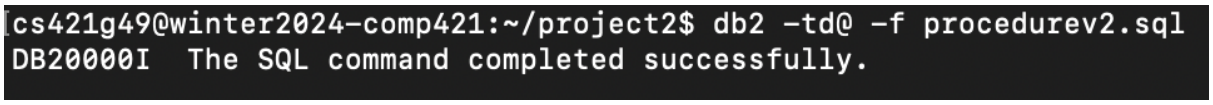
(a) What does the procedure do?

We want to enforce that the rating of a review has a value between 1-5. Ratings smaller than 1 will be corrected with a value of 1. Similarly, ratings higher than 5 will be corrected with a value of 5.

(b) Listing of Procedure Code

```
CREATE OR REPLACE PROCEDURE UPDATEINVALIDRATINGS2
LANGUAGE SQL
BEGIN
        DECLARE rID varchar(30);
        DECLARE currRating int;
        DECLARE end_table INT default 0;
        DECLARE my_cursor CURSOR FOR
                SELECT reviewId, rating FROM review;
        DECLARE CONTINUE HANDLER FOR NOT FOUND SET end_table = 1;
        OPEN my_cursor;
        FETCH my_cursor into rID, currRating;
        WHILE end_table = 0 DO IF currRating < 1
                THEN SET currRating = 1;
        END IF;
        IF currRating > 5
                THEN SET currRating = 5;
        END IF;
        UPDATE Review SET rating = currRating WHERE reviewId = rID;
        FETCH my_cursor into rID, currRating; END WHILE;
        CLOSE my_cursor;
END@
```

(c) *Screenshot of the db2 command*

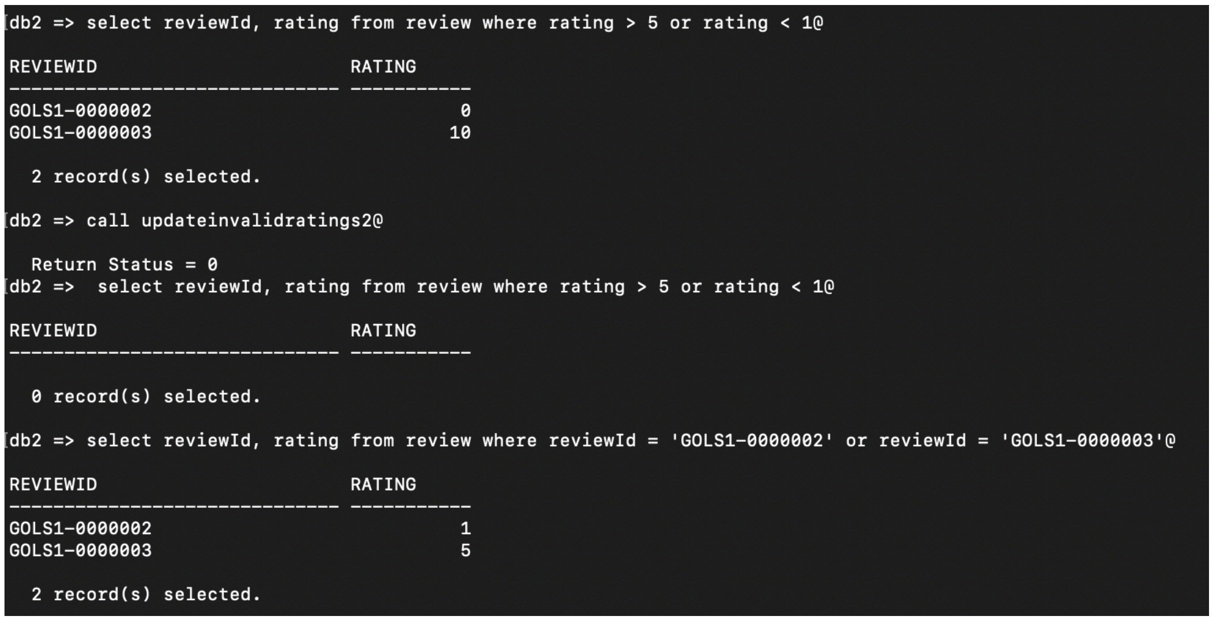


```
cs421g49@winter2024-comp421:~/project2$ db2 -td@ -f procedurev2.sql
DB20000I  The SQL command completed successfully.
```

to execute: call updateinvalidratings2 (screenshot in (d))

The filename has been updated to updateinvalidratings2.sql, and the procedure itself is called updateinvalidratings2.

(d) Intended effect



```
db2 => select reviewId, rating from review where rating > 5 or rating < 1@

REVIEWID                       RATING
------------------------------ -----------
GOLS1-0000002                            0
GOLS1-0000003                           10

  2 record(s) selected.

db2 => call updateinvalidratings2@

  Return Status = 0
db2 =>  select reviewId, rating from review where rating > 5 or rating < 1@

REVIEWID                       RATING
------------------------------ -----------

  0 record(s) selected.

db2 => select reviewId, rating from review where reviewId = 'GOLS1-0000002' or reviewId = 'GOLS1-0000003'@

REVIEWID                       RATING
------------------------------ -----------
GOLS1-0000002                            1
GOLS1-0000003                            5

  2 record(s) selected.
```

Stored Procedure 2

(a) *What does the procedure do?*

This stored procedure updates the values of isValid in the Menus table, filtering on the average of the ratings of the restaurants' reviews that the menus belong to. The threshold for the average rating to filter on is given as a parameter. This could aid in the process of finding "recommended" menus, or at least menus that are above/below a certain average rating.

(b) *Listing of Procedure Code*

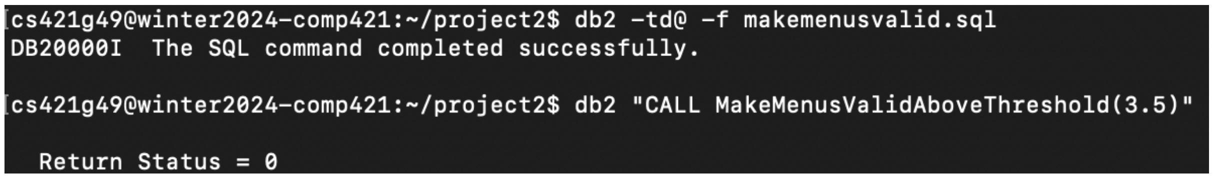
```
CREATE OR REPLACE PROCEDURE MakeMenusValidAboveThreshold(IN ratingThreshold
DECIMAL(3,2))
BEGIN
        DECLARE v_restaurantId VARCHAR(30);
        DECLARE v_avgRating DECIMAL(10,2);
        DECLARE done INT DEFAULT 0;
        DECLARE cur1 CURSOR FOR
                SELECT restaurantId FROM Review;

        DECLARE CONTINUE HANDLER FOR NOT FOUND
                SET done = 1;

        OPEN cur1;

        read_loop: LOOP
                FETCH cur1 INTO v_restaurantId;
                IF done = 1 then
                        LEAVE read_loop;
                END IF;
                SELECT AVG(rating) INTO v_avgRating
                FROM Review
                WHERE restaurantId = v_restaurantId;
                IF v_avgRating >= ratingThreshold THEN
                        UPDATE MENU
                        SET isValid = 1
                        WHERE restaurantId = v_restaurantId;
                ELSE
                        UPDATE MENU
                        SET isValid = 0
                        WHERE restaurantId = v_restaurantId;
                END IF;
        END LOOP;
        CLOSE cur1;
END@
```

(c) *Execution of the stored procedure*



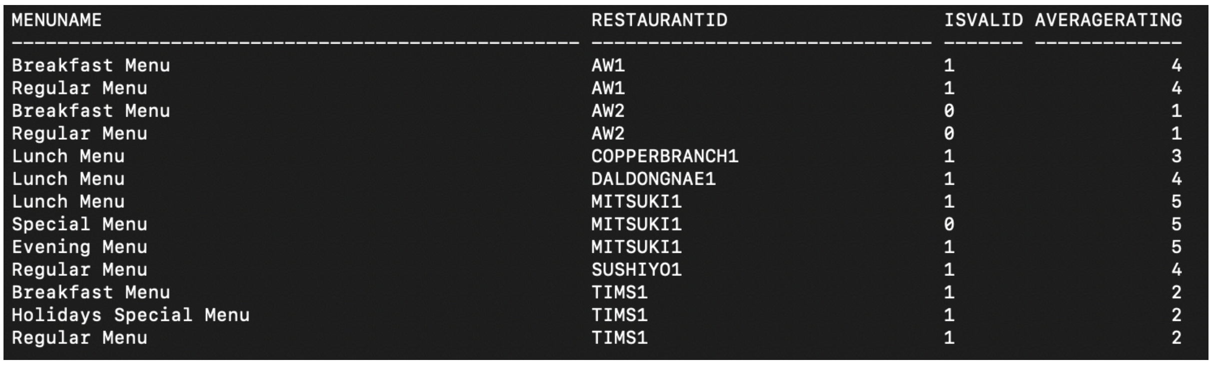
```
cs421g49@winter2024-comp421:~/project2$ db2 -td@ -f makemenusvalid.sql
DB20000I  The SQL command completed successfully.

cs421g49@winter2024-comp421:~/project2$ db2 "CALL MakeMenusValidAboveThreshold(3.5)"

  Return Status = 0
```

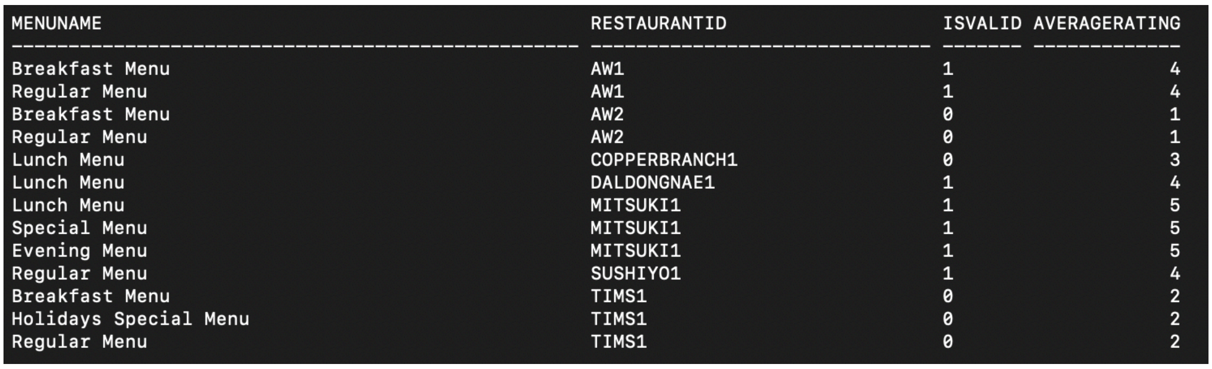
(d) *Evidence of effect (note that the threshold passed into the procedure is an average rating of 3.5):*

Before Execution:



```
MENUNAME                                           RESTAURANTID                   ISVALID AVERAGERATING
-------------------------------------------------- ------------------------------ ------- -------------
Breakfast Menu                                     AW1                            1                   4
Regular Menu                                       AW1                            1                   4
Breakfast Menu                                     AW2                            0                   1
Regular Menu                                       AW2                            0                   1
Lunch Menu                                         COPPERBRANCH1                  1                   3
Lunch Menu                                         DALDONGNAE1                    1                   4
Lunch Menu                                         MITSUKI1                       1                   5
Special Menu                                       MITSUKI1                       0                   5
Evening Menu                                       MITSUKI1                       1                   5
Regular Menu                                       SUSHIYO1                       1                   4
Breakfast Menu                                     TIMS1                          1                   2
Holidays Special Menu                              TIMS1                          1                   2
Regular Menu                                       TIMS1                          1                   2
```

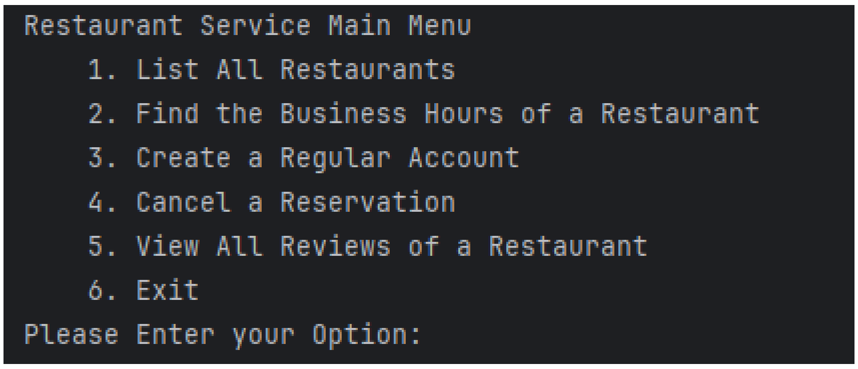
After Execution:



```
MENUNAME                                           RESTAURANTID                   ISVALID AVERAGERATING
-------------------------------------------------- ------------------------------ ------- -------------
Breakfast Menu                                     AW1                            1                   4
Regular Menu                                       AW1                            1                   4
Breakfast Menu                                     AW2                            0                   1
Regular Menu                                       AW2                            0                   1
Lunch Menu                                         COPPERBRANCH1                  0                   3
Lunch Menu                                         DALDONGNAE1                    1                   4
Lunch Menu                                         MITSUKI1                       1                   5
Special Menu                                       MITSUKI1                       1                   5
Evening Menu                                       MITSUKI1                       1                   5
Regular Menu                                       SUSHIYO1                       1                   4
Breakfast Menu                                     TIMS1                          0                   2
Holidays Special Menu                              TIMS1                          0                   2
Regular Menu                                       TIMS1                          0                   2
```

III. Application Program

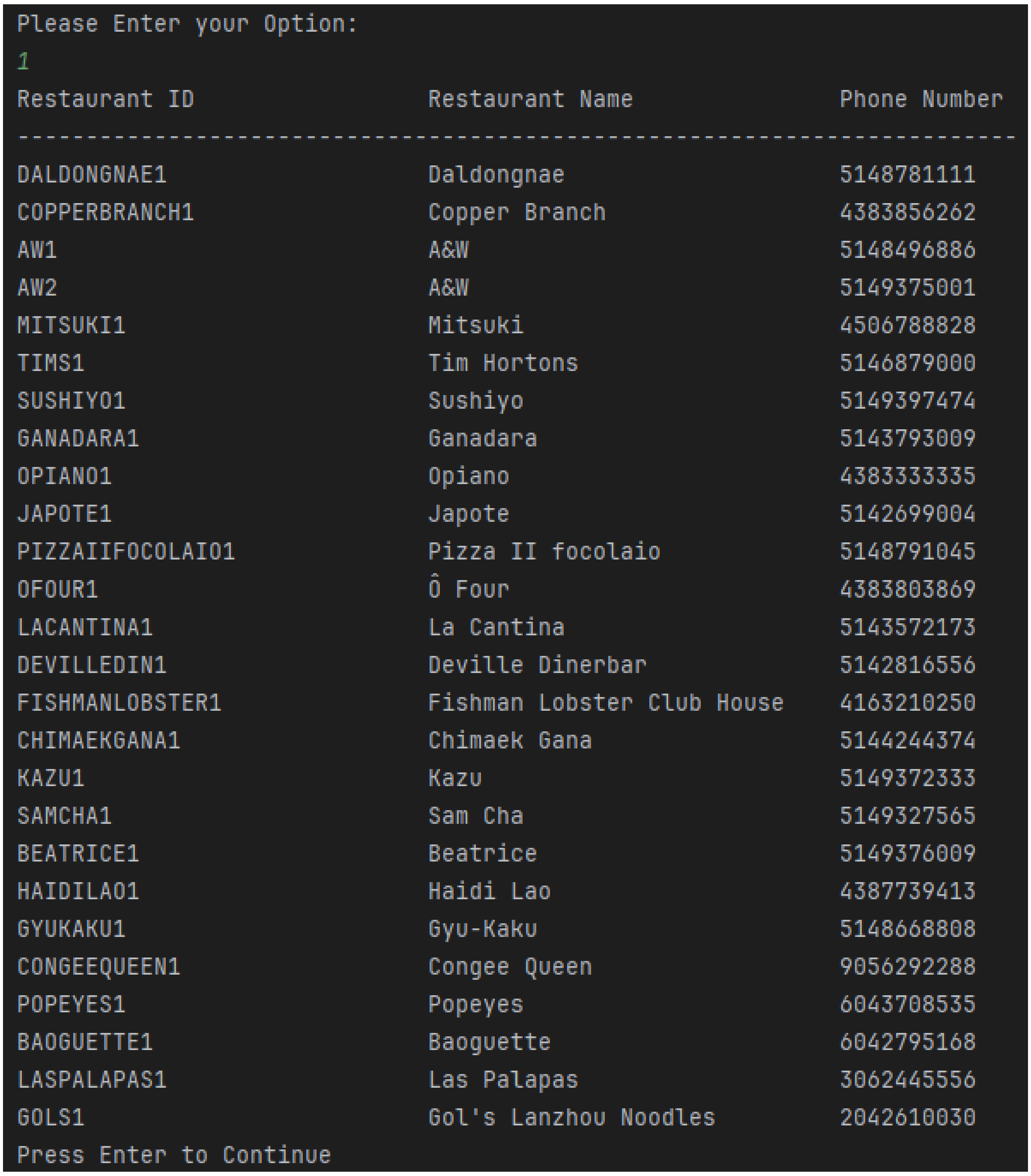
Menu



```
Restaurant Service Main Menu
    1. List All Restaurants
    2. Find the Business Hours of a Restaurant
    3. Create a Regular Account
    4. Cancel a Reservation
    5. View All Reviews of a Restaurant
    6. Exit
Please Enter your Option:
```

Option 1

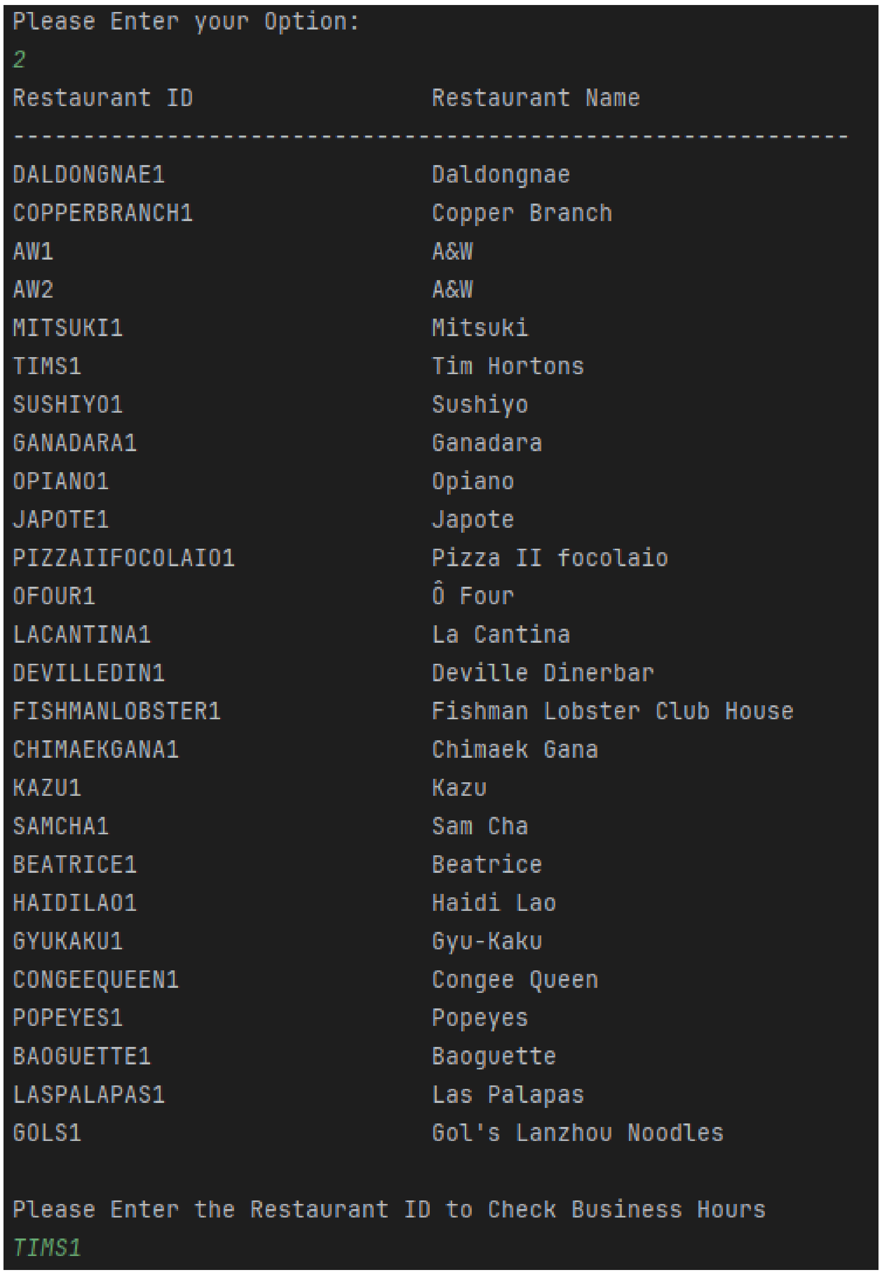
This option is a query of all the restaurants in the database. It shows the restaurantId, name, and phoneNumber.



```
Please Enter your Option:
1
Restaurant ID                 Restaurant Name               Phone Number
---------------------------------------------------------------------------
DALDONGNAE1                   Daldongnae                    5148781111
COPPERBRANCH1                 Copper Branch                 4383856262
AW1                           A&W                           5148496886
AW2                           A&W                           5149375001
MITSUKI1                      Mitsuki                       4506788828
TIMS1                         Tim Hortons                   5146879000
SUSHIYO1                      Sushiyo                       5149397474
GANADARA1                     Ganadara                      5143793009
OPIANO1                       Opiano                        4383333335
JAPOTE1                       Japote                        5142699004
PIZZAIIFOCOLAIO1              Pizza II focolaio             5148791045
OFOUR1                        Ô Four                        4383803869
LACANTINA1                    La Cantina                    5143572173
DEVILLEDIN1                   Deville Dinerbar              5142816556
FISHMANLOBSTER1               Fishman Lobster Club House    4163210250
CHIMAEKGANA1                  Chimaek Gana                  5144244374
KAZU1                         Kazu                          5149372333
SAMCHA1                       Sam Cha                       5149327565
BEATRICE1                     Beatrice                      5149376009
HAIDILAO1                     Haidi Lao                     4387739413
GYUKAKU1                      Gyu-Kaku                      5148668808
CONGEEQUEEN1                  Congee Queen                  9056292288
POPEYES1                      Popeyes                       6043708535
BAOGUETTE1                    Baoguette                     6042795168
LASPALAPAS1                   Las Palapas                   3062445556
GOLS1                         Gol's Lanzhou Noodles         2042610030
Press Enter to Continue
```

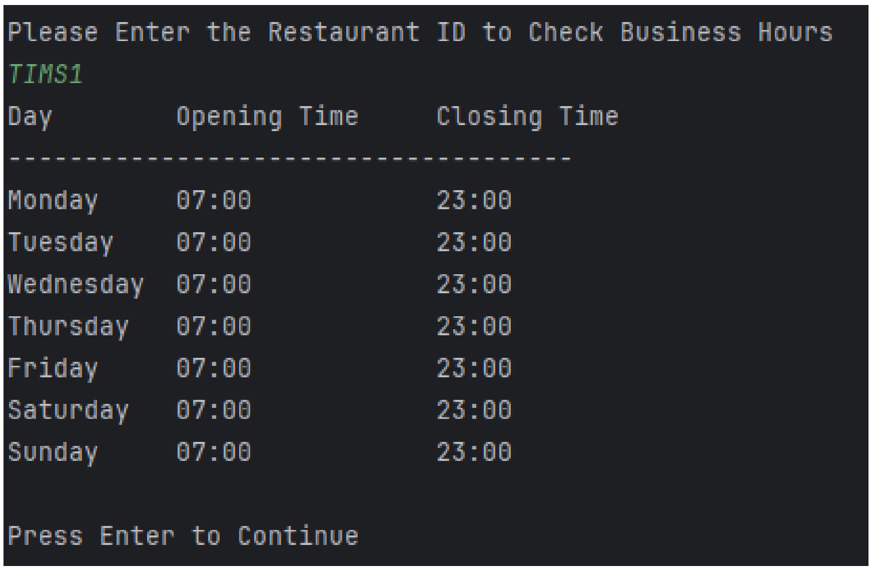
Option 2

This option displays all the restaurants and then asks for the ID of the restaurant from the list. It then shows the businessHour of that restaurant.



```
Please Enter your Option:
2
Restaurant ID                 Restaurant Name
------------------------------------------------------------
DALDONGNAE1                   Daldongnae
COPPERBRANCH1                 Copper Branch
AW1                           A&W
AW2                           A&W
MITSUKI1                      Mitsuki
TIMS1                         Tim Hortons
SUSHIYO1                      Sushiyo
GANADARA1                     Ganadara
OPIANO1                       Opiano
JAPOTE1                       Japote
PIZZAIIFOCOLAIO1              Pizza II focolaio
OFOUR1                        Ô Four
LACANTINA1                    La Cantina
DEVILLEDIN1                   Deville Dinerbar
FISHMANLOBSTER1               Fishman Lobster Club House
CHIMAEKGANA1                  Chimaek Gana
KAZU1                         Kazu
SAMCHA1                       Sam Cha
BEATRICE1                     Beatrice
HAIDILAO1                     Haidi Lao
GYUKAKU1                      Gyu-Kaku
CONGEEQUEEN1                  Congee Queen
POPEYES1                      Popeyes
BAOGUETTE1                    Baoguette
LASPALAPAS1                   Las Palapas
GOLS1                         Gol's Lanzhou Noodles

Please Enter the Restaurant ID to Check Business Hours
TIMS1
```

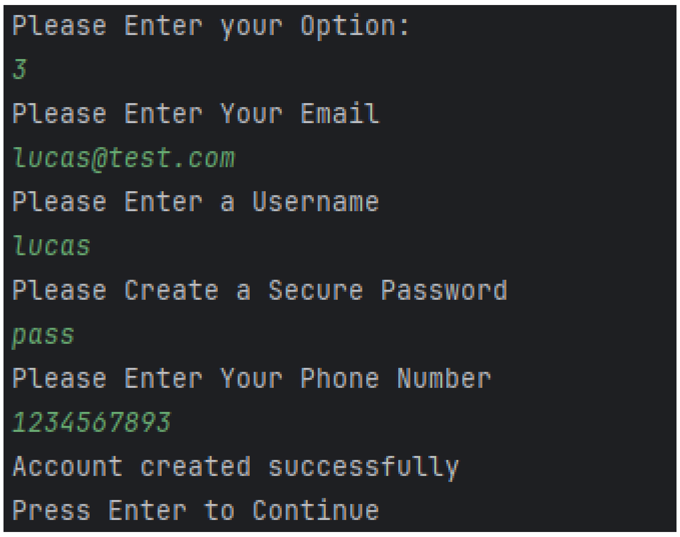


```
Please Enter the Restaurant ID to Check Business Hours
TIMS1
Day         Opening Time     Closing Time
-------------------------------------
Monday      07:00            23:00
Tuesday     07:00            23:00
Wednesday   07:00            23:00
Thursday    07:00            23:00
Friday      07:00            23:00
Saturday    07:00            23:00
Sunday      07:00            23:00

Press Enter to Continue
```

Option 3

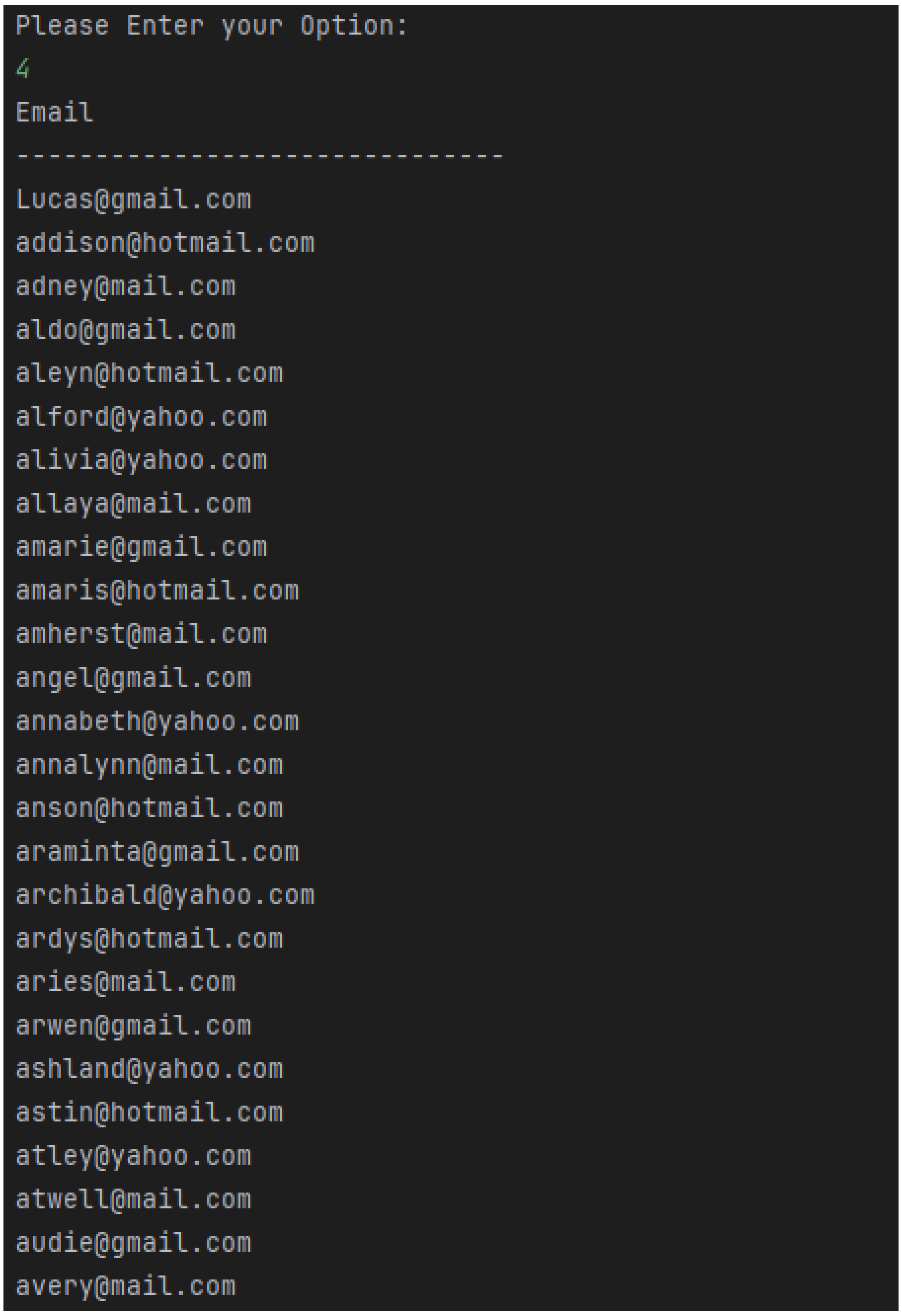
This option allows the user to create a regular account and insert it into the database regularAccount.



```
Please Enter your Option:
3
Please Enter Your Email
lucas@test.com
Please Enter a Username
lucas
Please Create a Secure Password
pass
Please Enter Your Phone Number
1234567893
Account created successfully
Press Enter to Continue
```

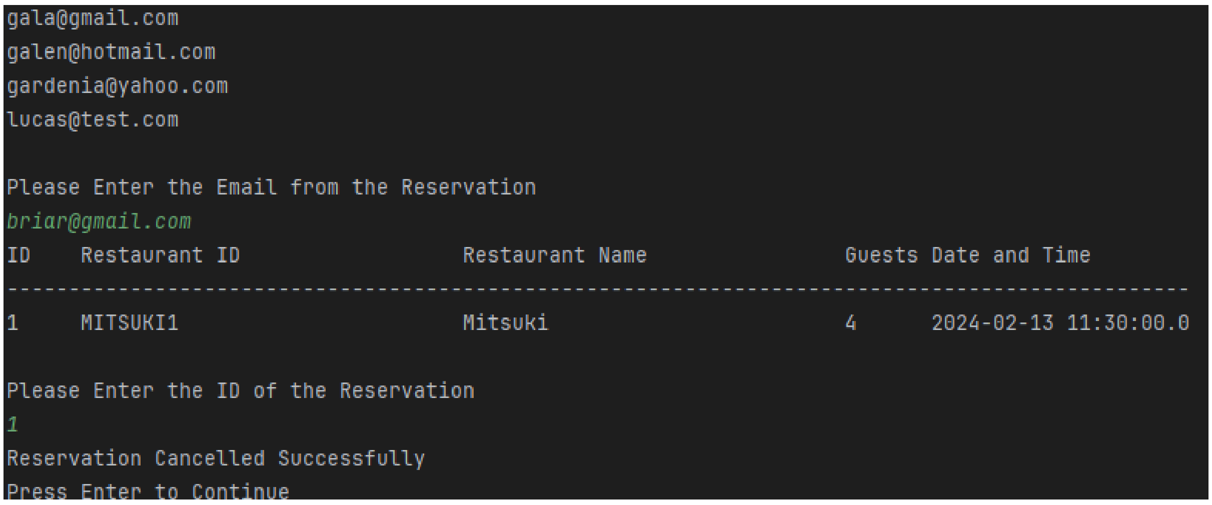
Option 4

This option allows the user to delete a reservation. It first shows all the emails in the table of regularAccount and then allows the user to input the email of an account. It then gives them a list of reservations that the entered account has. It then prompts the user to enter the ID of the reservation they would like to cancel.



```
Please Enter your Option:
4
Email
-------------------------------
Lucas@gmail.com
addison@hotmail.com
adney@mail.com
aldo@gmail.com
aleyn@hotmail.com
alford@yahoo.com
alivia@yahoo.com
allaya@mail.com
amarie@gmail.com
amaris@hotmail.com
amherst@mail.com
angel@gmail.com
annabeth@yahoo.com
annalynn@mail.com
anson@hotmail.com
araminta@gmail.com
archibald@yahoo.com
ardys@hotmail.com
aries@mail.com
arwen@gmail.com
ashland@yahoo.com
astin@hotmail.com
atley@yahoo.com
atwell@mail.com
audie@gmail.com
avery@mail.com
```

This is followed by all the emails in the database.



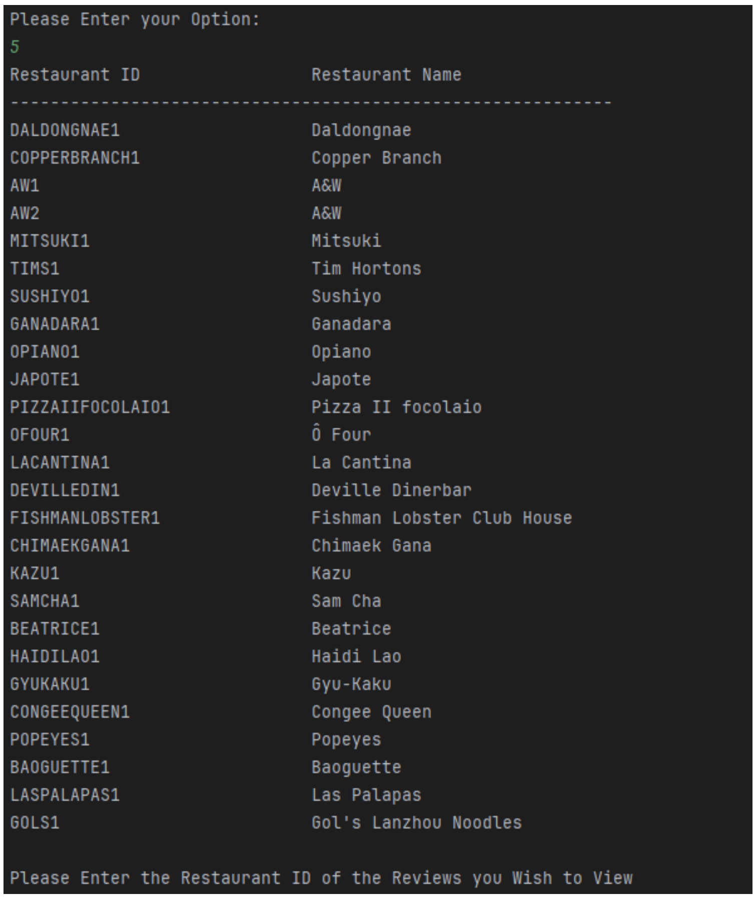
```
gala@gmail.com
galen@hotmail.com
gardenia@yahoo.com
lucas@test.com

Please Enter the Email from the Reservation
briar@gmail.com
ID    Restaurant ID                 Restaurant Name               Guests Date and Time
-----------------------------------------------------------------------------------------------
1     MITSUKI1                      Mitsuki                       4      2024-02-13 11:30:00.0

Please Enter the ID of the Reservation
1
Reservation Cancelled Successfully
Press Enter to Continue
```

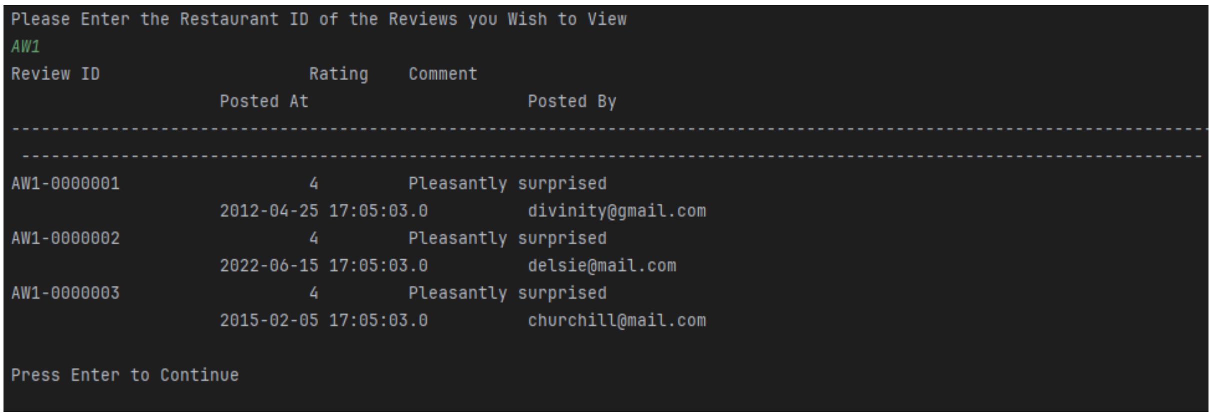
Option 5

This option allows the user to look at the reviews of a specific restaurant. It first shows the restaurantId and the name, and then asks for the input of one of the restaurantIds. Afterwards, it lists all the reviews that the restaurant has, which includes the rating, comment, postedAt date, and postedBy account email.



```
Please Enter your Option:
5
Restaurant ID                 Restaurant Name
------------------------------------------------------------
DALDONGNAE1                   Daldongnae
COPPERBRANCH1                 Copper Branch
AW1                           A&W
AW2                           A&W
MITSUKI1                      Mitsuki
TIMS1                         Tim Hortons
SUSHIYO1                      Sushiyo
GANADARA1                     Ganadara
OPIANO1                       Opiano
JAPOTE1                       Japote
PIZZAIIFOCOLAIO1              Pizza II focolaio
OFOUR1                        Ô Four
LACANTINA1                    La Cantina
DEVILLEDIN1                   Deville Dinerbar
FISHMANLOBSTER1               Fishman Lobster Club House
CHIMAEKGANA1                  Chimaek Gana
KAZU1                         Kazu
SAMCHA1                       Sam Cha
BEATRICE1                     Beatrice
HAIDILAO1                     Haidi Lao
GYUKAKU1                      Gyu-Kaku
CONGEEQUEEN1                  Congee Queen
POPEYES1                      Popeyes
BAOGUETTE1                    Baoguette
LASPALAPAS1                   Las Palapas
GOLS1                         Gol's Lanzhou Noodles

Please Enter the Restaurant ID of the Reviews you Wish to View
```

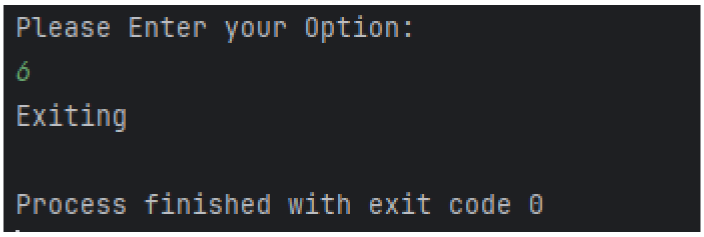


```
Please Enter the Restaurant ID of the Reviews you Wish to View
AW1
Review ID                    Rating    Comment
                     Posted At                    Posted By
------------------------------------------------------------------------------------------------------------------------
 ----------------------------------------------------------------------------------------------------------------------
AW1-0000001                  4         Pleasantly surprised
                     2012-04-25 17:05:03.0          divinity@gmail.com
AW1-0000002                  4         Pleasantly surprised
                     2022-06-15 17:05:03.0          delsie@mail.com
AW1-0000003                  4         Pleasantly surprised
                     2015-02-05 17:05:03.0          churchill@mail.com

Press Enter to Continue
```

Option 6

This option exits the console program.



```
Please Enter your Option:
6
Exiting

Process finished with exit code 0
```

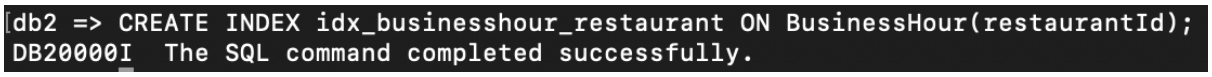
Files

The .java files are attached in the submission.

IV. Indexing

Index 1

(a) *Script*



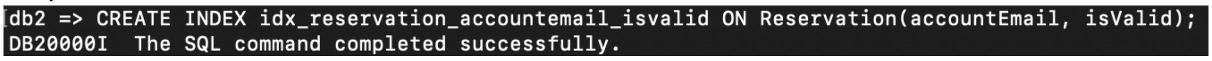
```
db2 => CREATE INDEX idx_businesshour_restaurant ON BusinessHour(restaurantId);
DB20000I  The SQL command completed successfully.
```

(b) *Performance*

Since our application retrieves business hours based on the restaurant ID, creating an index on the *restaurantId* column of the *BusinessHour* table will query faster. When querying the business hours based on the restaurant ID, the database can quickly locate the relevant rows within the *BusinessHour* table without scanning the entire table.

Index 2

(a) *Scripts*



```
db2 => CREATE INDEX idx_reservation_accountemail_isvalid ON Reservation(accountEmail, isValid);
DB20000I  The SQL command completed successfully.
```

(b) *Performance*

When cancelling a reservation in our application, we perform a query that filters on *accountEmail* and *isValid*. This composite index will quicken the filtering process because it allows the database to quickly find all reservations associated with a specific *accountEmail* and *isValid*.

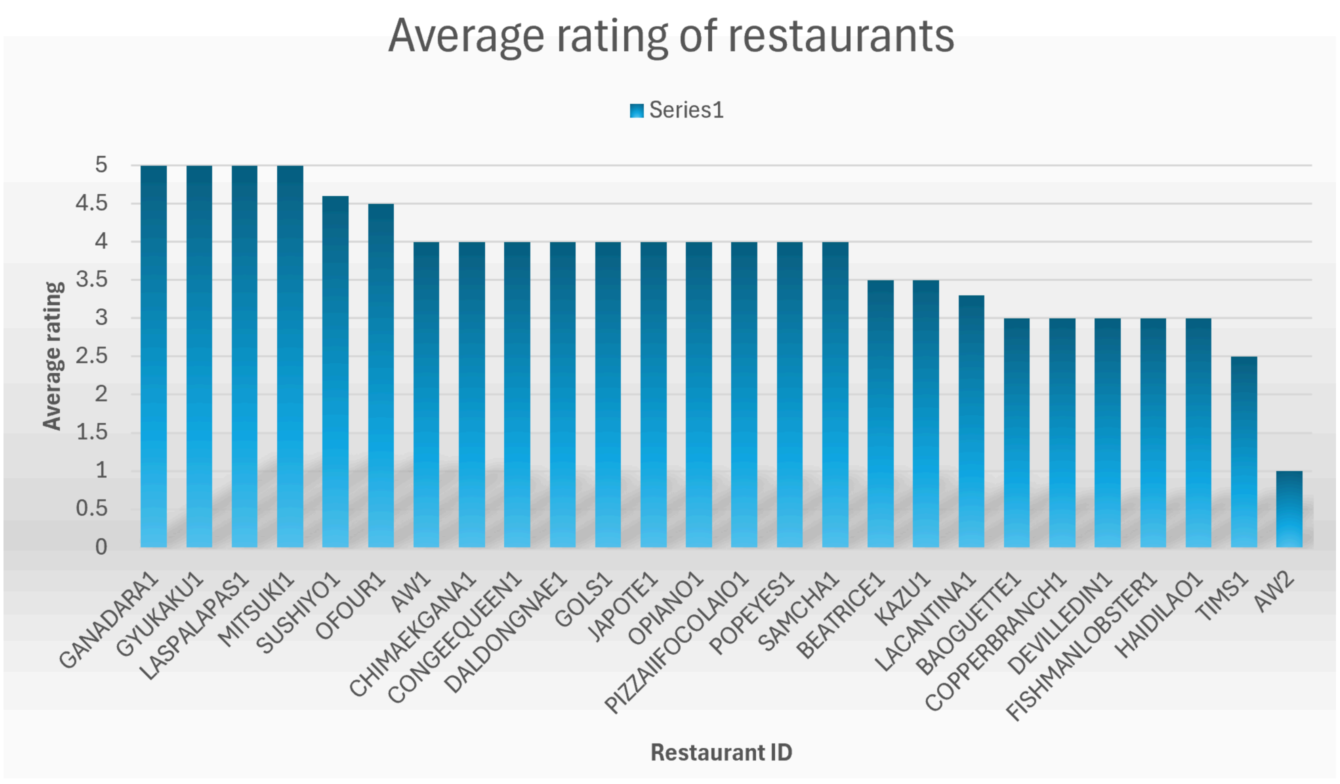
V. Visualization

Visualization 1

(a) *Description*
We visualized the average rating of each restaurant in our app using bars to better display which ones have the highest overall rating compared to those with a lower rating. To make it more realistic, we made our query round the average up to 1 decimal place as it is often seen in website ratings.

(b) *SQL*

```
EXPORT TO avgreviews.csv OF DEL MODIFIED BY NOCHARDEL
SELECT rv.restaurantId, CAST(AVG(rating * 1.0) AS DECIMAL(2,1)) AS average_rating
FROM REVIEW rv
JOIN RESTAURANT r ON rv.restaurantID=r.restaurantID
GROUP BY rv.restaurantID
ORDER BY average_rating DESC;
```

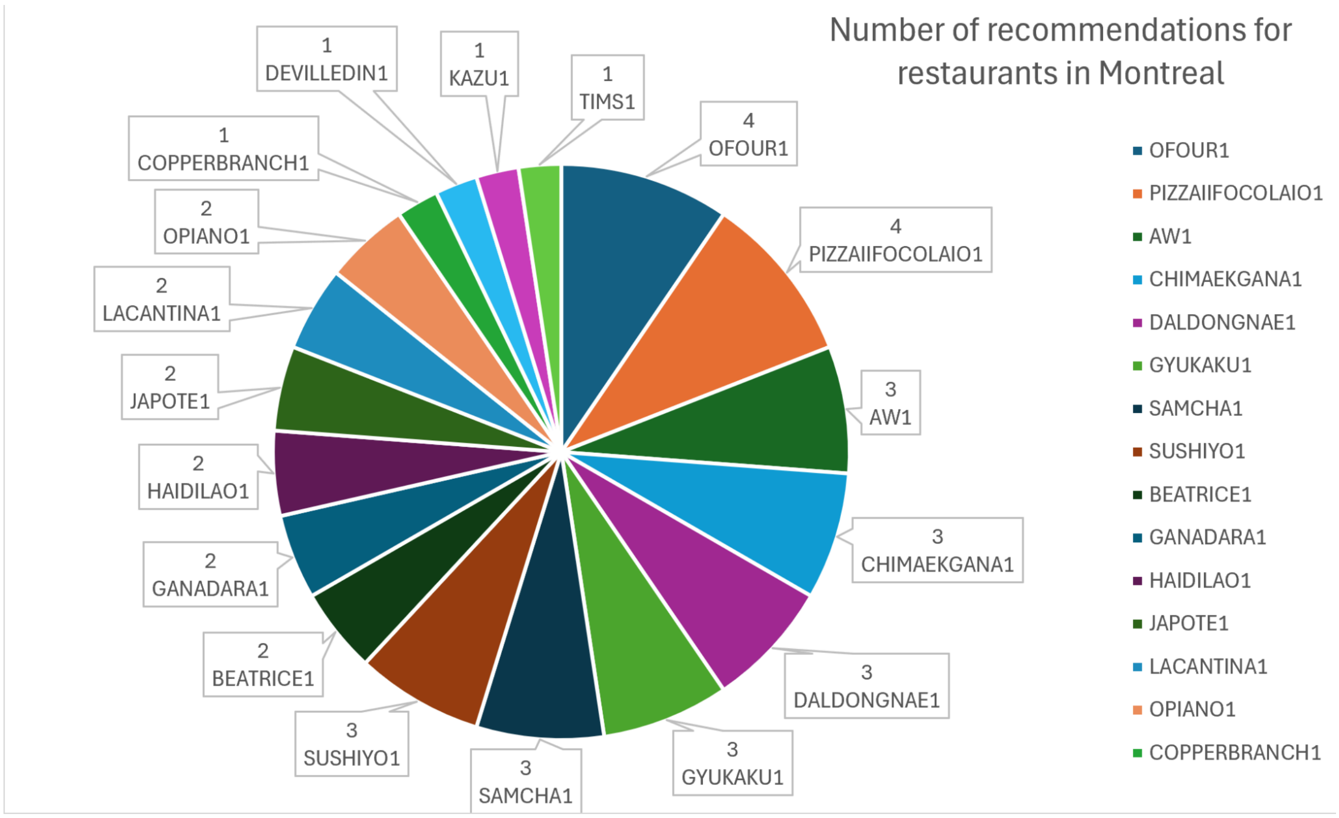


Visualization 2

(a) *Description*
We visualized the number of recommendations restaurants in the city of Montreal have in our app. Restaurants with no recommendations do not appear on our chart. We chose to use a pie chart to better showcase the ratio of each recommended restaurant. The number of recommendations appears next to the restaurant name to give us a better idea of the overall recommendation distribution.

(b) SQL

```
EXPORT TO recommendations.csv OF DEL MODIFIED BY NOCHARDEL
SELECT restaurantID, COUNT(restaurantID) AS number_of_recommendations
FROM (SELECT r.restaurantID
        FROM RECOMMENDATION r
        JOIN ADDRESS a ON r.restaurantID = a.restaurantID WHERE city='Montreal')
GROUP BY restaurantID
ORDER BY number_of_recommendations DESC;
```



VI. Creativity

We created an extra stored procedure to fulfill the creativity requirement (See section II).

VII. Collaboration

We met once over the break to discuss which part of the project each of us was going to do. We also met once after the break to discuss the problems that we encountered and how we wanted our console program to work. Other than in-person meetings, we had an active group chat where we helped each other with issues. Lucas did the console program with help for the connection to the database from the rest of the team. Jing-Ling wrote queries and made the graphs for the visualization part. Shawn did the first stored procedure. Ethan did the second stored procedure and the indexing.