University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 4

Date: Dec 1,2023

Group Number: 123

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Kentaro Lim	44267326	klim10	kentarolim10@gmail.com
Riley Baines	98686033	rbaine01	rileybaines@outlook.com
Tony Liu	42641019	tl0226	tl0226yn@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

University of British Columbia, Vancouver
Department of Computer Science

Description

This project is an application that allows the user to read and write reviews for restaurants. However, the user is also allowed to read and write reviews for the food items themselves, giving a better understanding on recommendations and foods to avoid. The implementation of the project contains users, restaurants, menu items and reviews, and allows the viewing, creation, modifying and deleting of each of them.

Changed Schema

- Get rid of the Restaurant_Rating table. All the ratings were already located in Restaurant_Review so we felt that the Restaurant_Rating table was not necessary.
- Restaurant_Review no longer has a foreign key referencing Restaurant_Rating.
- Get rid of the Menu_Item_Rating table. All the ratings were already located in Menu_Item_Review so we felt that the table was not necessary.
- Menu_Item_Review no longer has a foreign key referencing Menu_Item_Rating.
- Adds a new foreign key into Menu_Item_Review (MenuItemName) referencing the name of the menu item being reviewed, as in the Menu_Item table the primary key also includes the name
- Get rid of the Cost_Map table. We felt that people could already get a sense of how expensive a food item was based on its price alone.
- Menu_Item no longer has a foreign key referencing Cost_Map.
    - This also meant we got rid of price_category attribute in Menu_Item

Copy of Schema

- City(Name, Province_State, Country)
- Review_User(UserId, Name, **CityName (NOT NULL)**, **Province_State(NOT NULL)**)
- Follows(**UserId1**, **UserId2**)
- Verified_User(**UserId** (NOT NULL), VerificationNumber (UNIQUE))
- Restaurant(BusinessNumber, **StreetName**, **HouseNumber**, **PostalCode**, Name, **OwnerId** (NOT NULL))
- Postal_Area(PostalCode, Province_State)
- Address(HouseNumber, StreetName, PostalCode, CityName (NOT NULL))
- Owner(SIN, Name)
- Menu_Item(**BusinessNumber**, Name, Cost, **Type** (NOT NULL))
- Menu_Item_Type(Type)
- offers(**Business Number**, **Type**)
- Review(ReviewNumber, Content, **UserId**)
- Restaurant_Review(**ReviewNumber** (NOT NULL), **BusinessNumber** (NOT NULL), AmbienceRating, CleanlinessRating, ServiceRating)
- Menu_Item_Review(**ReviewNumber** (NOT NULL), **Business Number** (NOT NULL), **Menu_Item_Name** (NOT NULL), PresentationRating, TasteRating, PortionSizeRating)

- Helpful_Tag(TagID, HelpfulnessRating (NOT NULL), **ReviewNumber**, **UserId**)

Screenshots of data in relations after running SQL initialization script

- City

```
SQL> SELECT * FROM CITY
  2  ;

NAME                           PROVINCE_STATE
------------------------------ ------------------------------
COUNTRY
------------------------------
Vancouver                      British Columbia
Canada

Richmond                       British Columbia
Canada

Toronto                        Ontario
Canada


NAME                           PROVINCE_STATE
------------------------------ ------------------------------
COUNTRY
------------------------------
Ottawa                         Ontario
Canada

Los Angeles                    California
United States of America
```

- Review_User

```
SQL> SELECT * FROM Review_User;

USERID                         NAME
------------------------------ ------------------------------
CITYNAME                       PROVINCE_STATE
------------------------------ ------------------------------
user1                          John
Toronto                        Ontario

user2                          Jane
Los Angeles                    California

user3                          Kentaro
Vancouver                      British Columbia


USERID                         NAME
------------------------------ ------------------------------
CITYNAME                       PROVINCE_STATE
------------------------------ ------------------------------
user4                          Tony
Vancouver                      British Columbia

user5                          Riley
Vancouver                      British Columbia

user6                          ChatGPT
Vancouver                      British Columbia
```

- Follows

```
SQL> SELECT * FROM Follows;

USERID                         USERID2
------------------------------ ------------------------------
user1                          user2
user1                          user3
user3                          user4
user4                          user2
user5                          user6
```

- Verified_User

```
SQL> SELECT * FROM Verified_User;

USERID                         VERIFICATIONNUMBER
------------------------------ ------------------
user1                                           1
user2                                           2
user3                                           3
user4                                           4
user5                                           5
```

- Restaurant*

```
SQL> SELECT * FROM Restaurant;

BUSINESSNUMBER HOUSENUMBER STREETNAME
-------------- ----------- ------------------------------------------------------
POSTALC NAME                                                 OWNERID
------- -------------------------------------------------- ----------
           123         601 W Hastings Street
V6B 1M8 Bob's Generic Pizza Place                         123456789

           234        2205 Lower Mall
V6T 1Z4 The Point                                         789123456

           345        6363 Agronomy Road
V6T 1Z4 Orchard Commons                                   456789123


BUSINESSNUMBER HOUSENUMBER STREETNAME
-------------- ----------- ------------------------------------------------------
POSTALC NAME                                                 OWNERID
------- -------------------------------------------------- ----------
           456        6363 Agronomy Road
V6T 1Z4 McDonalds                                         654321987

           999           1 Wellington Street
K1A 0A9 Bob's Underground Food Court                      987654321

           909         290 Bremner Blvd
M5V 3L9 Bob's Sky High Food Court                         987654321
```

- Postal_Area*

```
SQL> SELECT * FROM Postal_Area;

POSTALC PROVINCE_STATE
------- ------------------------------
V6T 1Z4 British Columbia
T2X 2L9 Alberta
V6B 1M8 British Columbia
K1A 0A9 Ontario
M5V 3L9 Ontario
```

- Address*

```
SQL> SELECT * FROM ADDRESS;

HOUSENUMBER STREETNAME                                         POSTALC
----------- -------------------------------------------------- -------
CITYNAME
------------------------------
        601 W Hastings Street                                  V6B 1M8
Vancouver

       2205 Lower Mall                                         V6T 1Z4
Vancouver

       6363 Agronomy Road                                      V6T 1Z4
Vancouver


HOUSENUMBER STREETNAME                                         POSTALC
----------- -------------------------------------------------- -------
CITYNAME
------------------------------
        290 Bremner Blvd                                       M5V 3L9
Toronto

          1 Wellington Street                                  K1A 0A9
Ottawa
```

- Owner

```
SQL> SELECT * FROM OWNER;

       SIN NAME
---------- ------------------------------
 123456789 Bob
 456789123 Jick
 789123456 Jane
 987654321 Bob
 654321987 Jane
```

- Menu_Item

```
SQL> SELECT * FROM Menu_Item;

BUSINESSNUMBER NAME
-------------- --------------------------------------------------
TYPE                           COST
------------------------------ ----------
           123 Pepperoni Pizza
Pizza                            15

           234 Cheeseburger
Burgers                        19.99

           234 Lasagna
Pasta                            25


BUSINESSNUMBER NAME
-------------- --------------------------------------------------
TYPE                           COST
------------------------------ ----------
           345 Dumplings
Dim Sum                          30

           999 Shrimp wrappers
Dim Sum                        7.99

           999 California Roll
Sushi                          4.99


BUSINESSNUMBER NAME
-------------- --------------------------------------------------
TYPE                           COST
------------------------------ ----------
           909 Premium Spaghetti
Pasta                            40
```

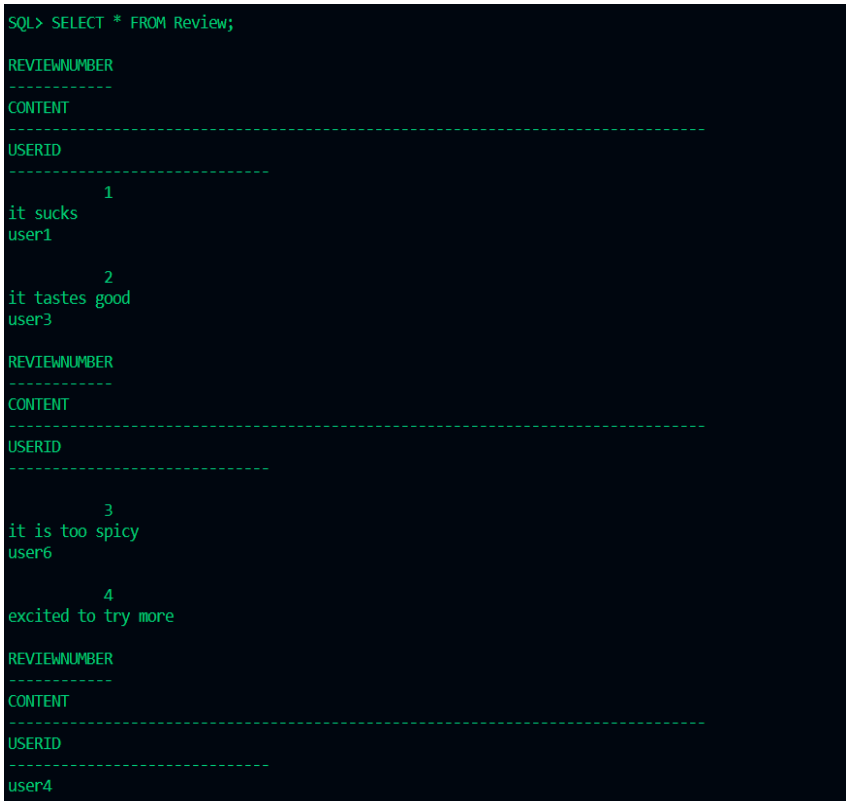
- Menu_Item_Type

```
SQL> SELECT * FROM Menu_Item_Type;

TYPE
------------------------------
Burgers
Dim Sum
Pasta
Pizza
Sushi
```

- offers

```
SQL> SELECT * FROM OFFERS;

BUSINESSNUMBER TYPE
-------------- ------------------------------
           123 Pizza
           234 Burgers
           234 Pasta
           345 Burgers
           345 Dim Sum
           456 Burgers
           909 Pasta
           999 Dim Sum
           999 Sushi
```

- Review (1 of 2 screenshots)



```
SQL> SELECT * FROM Review;

REVIEWNUMBER
------------
CONTENT
--------------------------------------------------------------------------------
USERID
------------------------------
           1
it sucks
user1

           2
it tastes good
user3

REVIEWNUMBER
------------
CONTENT
--------------------------------------------------------------------------------
USERID
------------------------------

           3
it is too spicy
user6

           4
excited to try more

REVIEWNUMBER
------------
CONTENT
--------------------------------------------------------------------------------
USERID
------------------------------
user4
```

- Review (2 of 2 screenshots)

```
          5
way too overpriced
user5

          6

REVIEWNUMBER
------------
CONTENT
--------------------------------------------------------------------------------
USERID
------------------------------
the service was okay
user1

          7
restaurant smelled bad
user2


REVIEWNUMBER
------------
CONTENT
--------------------------------------------------------------------------------
USERID
------------------------------
          8
i cannot seem to understand why they do not have enough staff
user5

          9
I really liked their service
user4

REVIEWNUMBER
------------
CONTENT
--------------------------------------------------------------------------------
USERID
------------------------------

         10
The environment was nice
user6
```

- Restaurant_Review

```
SQL> SELECT * FROM Restaurant_Review;

REVIEWNUMBER BUSINESSNUMBER AMBIENCERATING CLEANLINESSRATING SERVICERATING
------------ -------------- -------------- ----------------- -------------
           6            999              3                 4             2
           7            234              1                 2             5
           8            456              1                 1             1
           9            234              4                 3             5
          10            123              5                 5             5
```

- Menu_Item_Review

```
SQL> SELECT * FROM Menu_Item_Review;

REVIEWNUMBER BUSINESSNUMBER MENUITEMNAME
------------ -------------- ----------------------------------------------------
PRESENTATIONRATING TASTERATING PORTIONSIZERATING
------------------ ----------- -----------------
           1            999 Shrimp wrappers
                 2           2                 2

           2            234 Cheeseburger
                 5           5                 5

           3            345 Dumplings
                 4           2                 3


REVIEWNUMBER BUSINESSNUMBER MENUITEMNAME
------------ -------------- ----------------------------------------------------
PRESENTATIONRATING TASTERATING PORTIONSIZERATING
------------------ ----------- -----------------
           4            123 Pepperoni Pizza
                 4           3                 5

           5            909 Premium Spaghetti
                 1           2                 1
```

- Helpful_Tag

```
SQL> SELECT * FROM Helpful_Tag;

     TAGID HELPFULNESSRATING REVIEWNUMBER USERID
---------- ---------------- ------------ ------------------------------
         1                1            3 user2
         2                0            1 user6
         3                1            7 user4
         4                1            7 user6
         5                0           10 user1
```

*: POSTALC refers to PostalCode

University of British Columbia, Vancouver
Department of Computer Science

All Queries to Satisfy Rubric

Insert:

File: appService.js
Function: insertRestaurant
Line Number: 988
Query:
INSERT INTO Restaurant(BusinessNumber, HouseNumber, StreetName, PostalCode, Name, OwnerId) VALUES (:businessNumber, :houseNumber, :streetName, :postalCode, :name, :ownerId)

Delete:

File: appService.js
Function: deleteMenuItem
Line Number: 845
Query:
DELETE FROM Menu_Item WHERE BusinessNumber=:businessNumber AND Name=:name;

Update:

File: appService.js
Function: editRestaurant
Line Number: 1095
Query:
UPDATE Restaurant
SET Name=:Name, OwnerId=:OwnerId, HouseNumber=:HouseNumber, StreetName=:StreetName, PostalCode=:PostalCode
WHERE BusinessNumber=:BusinessNumber;

Selection:

File: appService.js
Function: ViewFollows
Line Number:

- AND clause: 604
- OR clause: 605
- No userID2 provided: 611
- No userID1 provided: 617
- No userID's provided: 623

Query:

- AND clause:
    - SELECT UserId1, USERID2 FROM Follows WHERE UserId1 = :UserId1 AND UserId2 = :UserId2;
- OR clause:
    - SELECT UserId1, USERID2 FROM Follows WHERE UserId1 = :UserId1 OR UserId2 = :UserId2;
- No userID2 provided:
    - SELECT UserId1, USERID2 FROM Follows WHERE UserId1 = :UserId1;
- No userID1 provided:
    - SELECT UserId1, USERID2 FROM Follows WHERE UserId2 = :UserId2;
- No userID's provided:
    - SELECT UserId1, USERID2 FROM Follows;

Projection:

File: appService.js
Function: viewTables, getAttributes, Project
Line Number: 770, 783-785, 803-804
Query:

```
SELECT table_name FROM user_tables;
SELECT column_name
        FROM all_tab_columns
        WHERE table_name = :tableName;
SELECT ${queryString}
        FROM ${tableName};
```

Join:

File: appService.js
Function: searchUserById
Line Number: 447
Query:

```
SELECT * FROM Review_User NATURAL LEFT OUTER JOIN Verified_User WHERE UPPER(UserId)
LIKE '%'|| :UserId ||'%';
```

Aggregation with Group By:

File: appService.js
Function: viewRestaurantReview
Line Number: 1122-1126
Query:

```
SELECT Re.UserId, U.Name, COUNT(RR.ReviewNumber)
        FROM Restaurant_Review RR, Review Re, Review_User U
```

```
WHERE RR.ReviewNumber = Re.ReviewNumber AND Re.UserId = U.UserId
GROUP BY Re.UserId, U.Name
HAVING Re.UserId = :userId
```

Aggregation with Having:

File: appService.js
Function: viewRestaurantReview
Line Number: 1137-1141
Query:

```
SELECT RR.BusinessNumber, R.Name, COUNT(RR.ReviewNumber)
    FROM Restaurant_Review RR, Restaurant R
    WHERE RR.BusinessNumber = R.BusinessNumber
    GROUP BY RR.BusinessNumber, R.Name
    HAVING RR.BusinessNumber = :businessNumber;
```

Nested Aggregation with Group By:

File: appService.js
Function: getMostExpensiveMenuItemsByType
Line Number: 1232-1250
Query:

```
SELECT
    Type,
    BusinessNumber,
    Name,
    Cost
    FROM
    Menu_Item
    WHERE
    (Type, Cost) IN (
    SELECT
    Type,
    MAX(Cost) AS MaxCost
    FROM
    Menu_Item
    GROUP BY
    Type
    );
```

Division:

File: appService.js

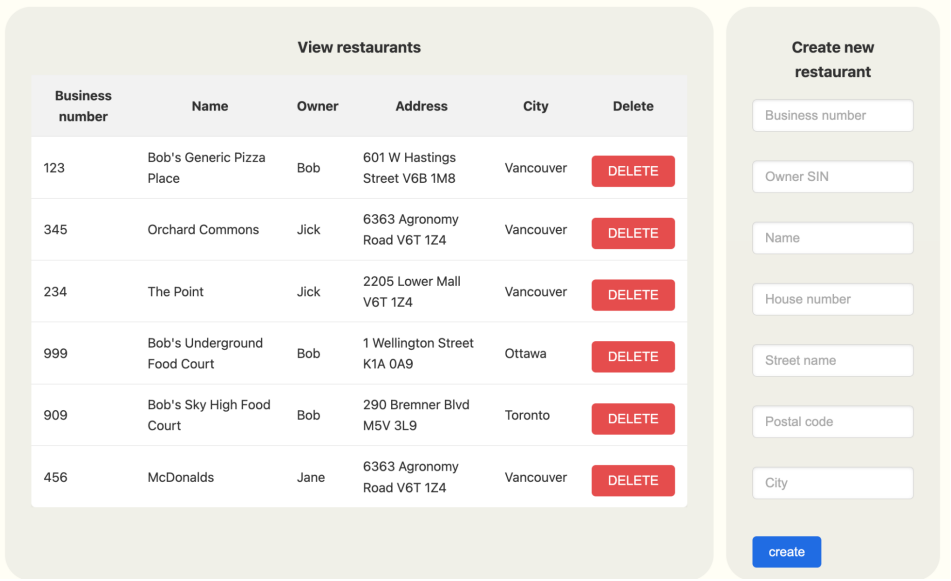
Function: getRestaurantsWithAllMenuItemTypes
Line Number: 1264-1276
Query:

```
SELECT DISTINCT R.BusinessNumber, R.Name
        FROM Restaurant R
        WHERE NOT EXISTS (
        SELECT T.Type
        FROM Menu_Item_Type T
        WHERE NOT EXISTS (
        SELECT 1
        FROM Menu_Item M
        WHERE M.BusinessNumber = R.BusinessNumber AND M.Type = T.Type
        )${selectedTypes.length > 0 ? ` AND T.Type IN ('${selectedTypes.join("','")}')` : ''}
        );
```

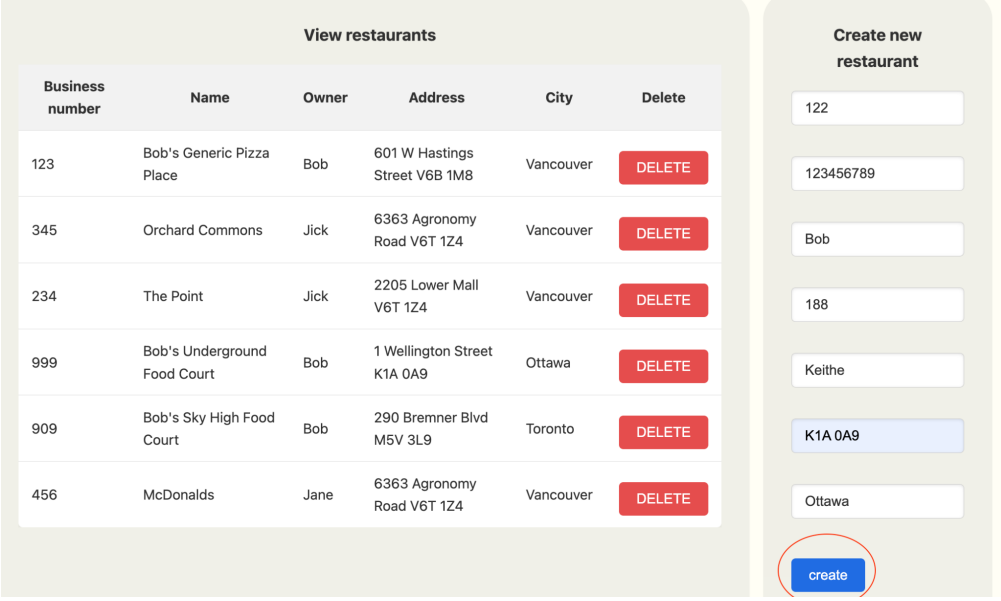
Functionality of Each Query in GUI

Insert:

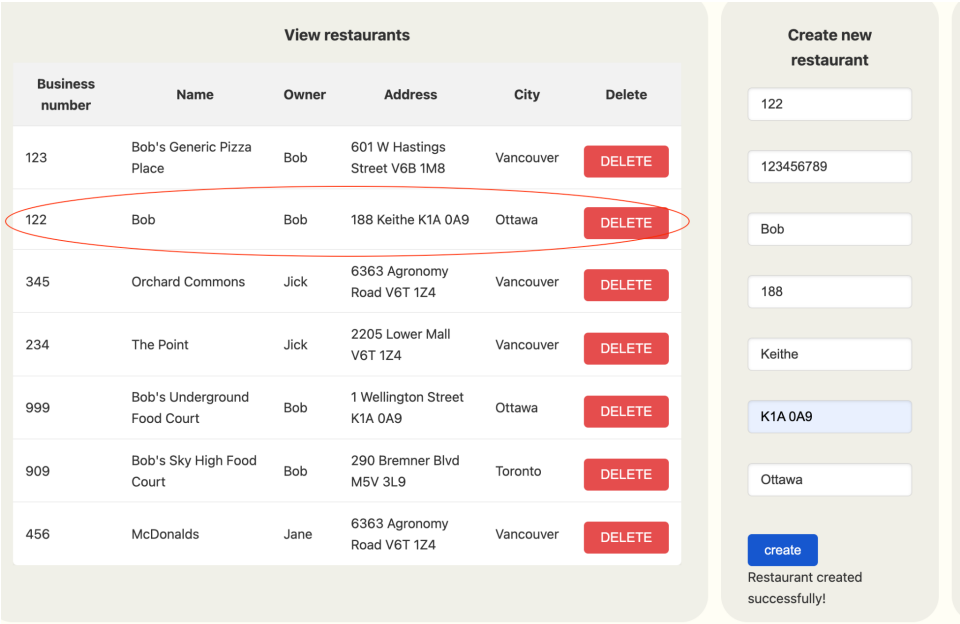
***Before*:**



During:



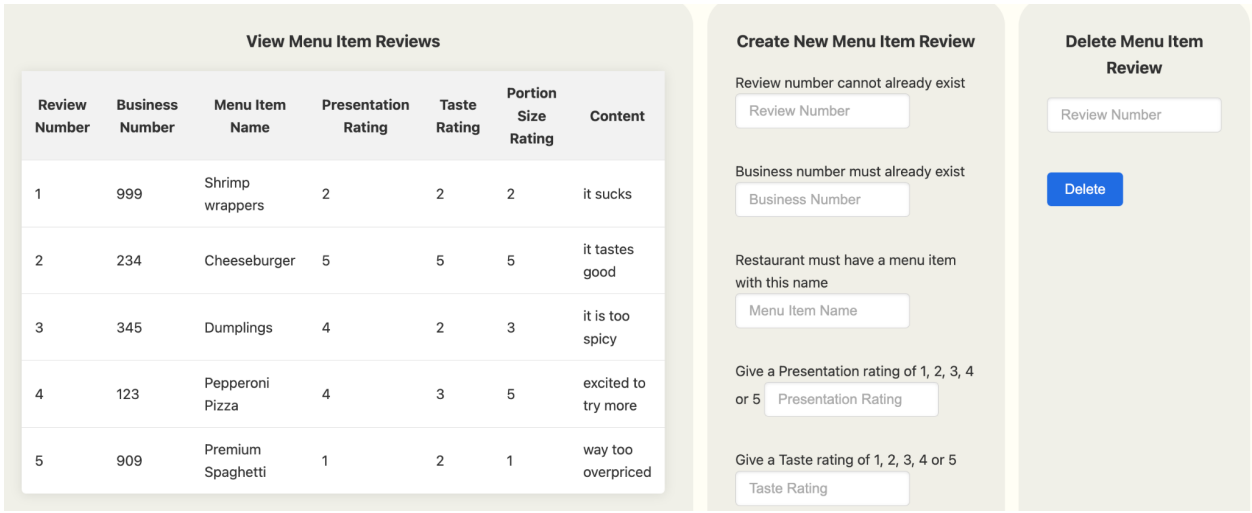
After:



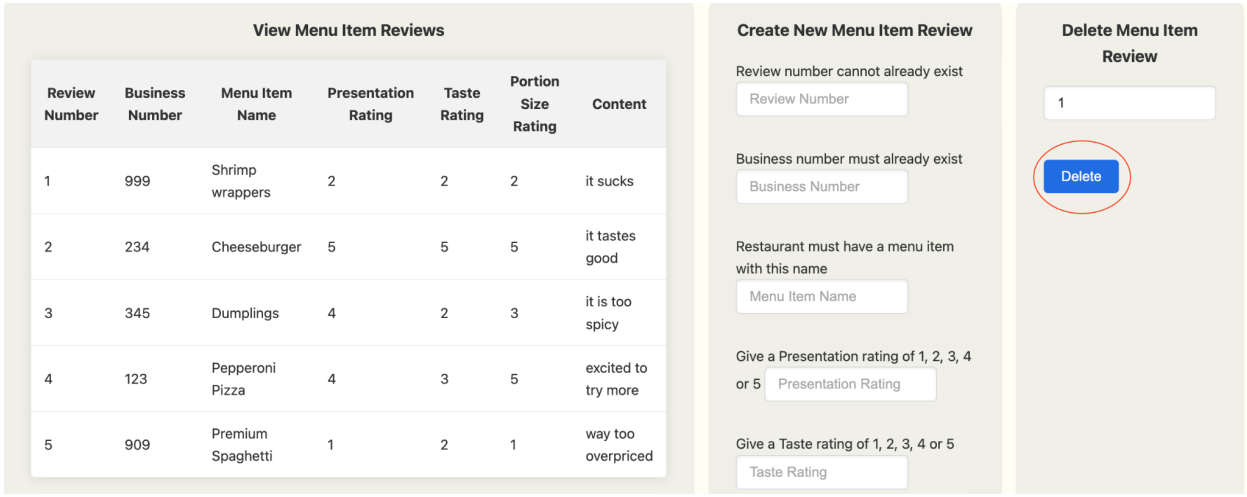
Delete:

University of British Columbia, Vancouver
Department of Computer Science

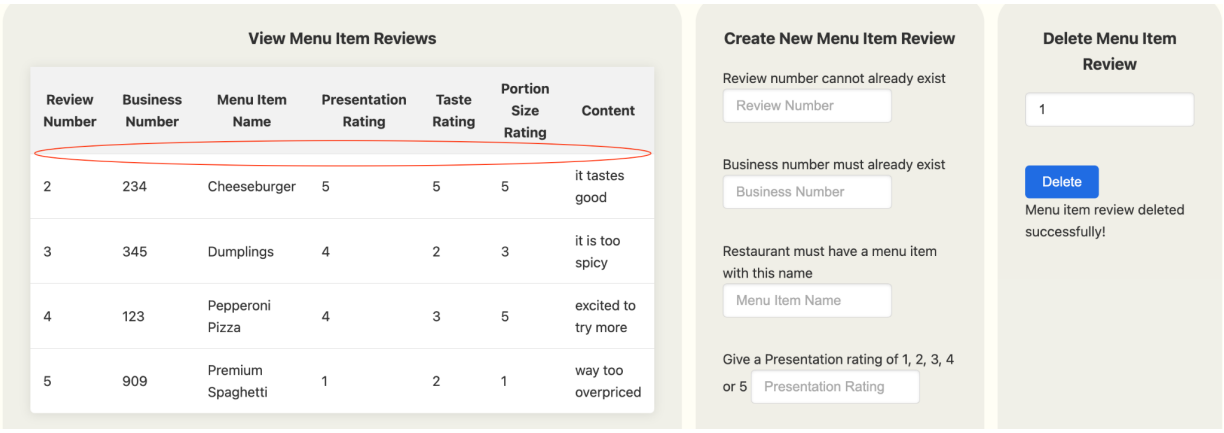
Before:



During:



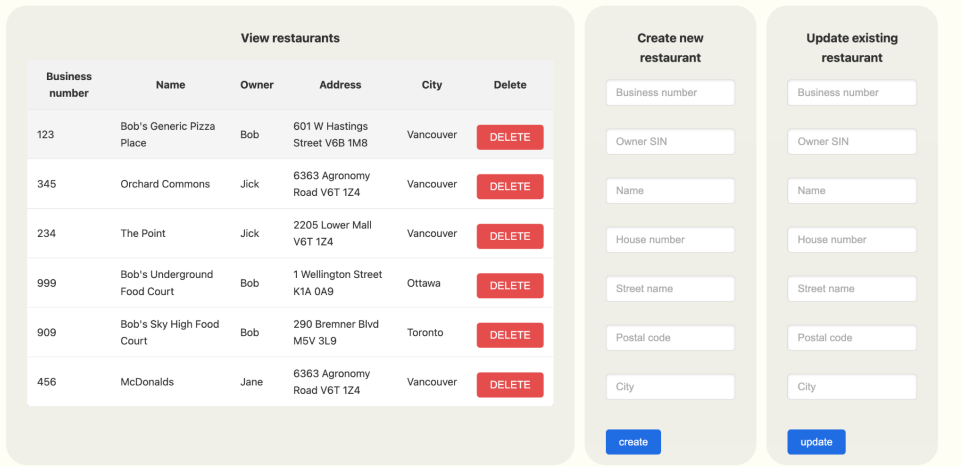
After:



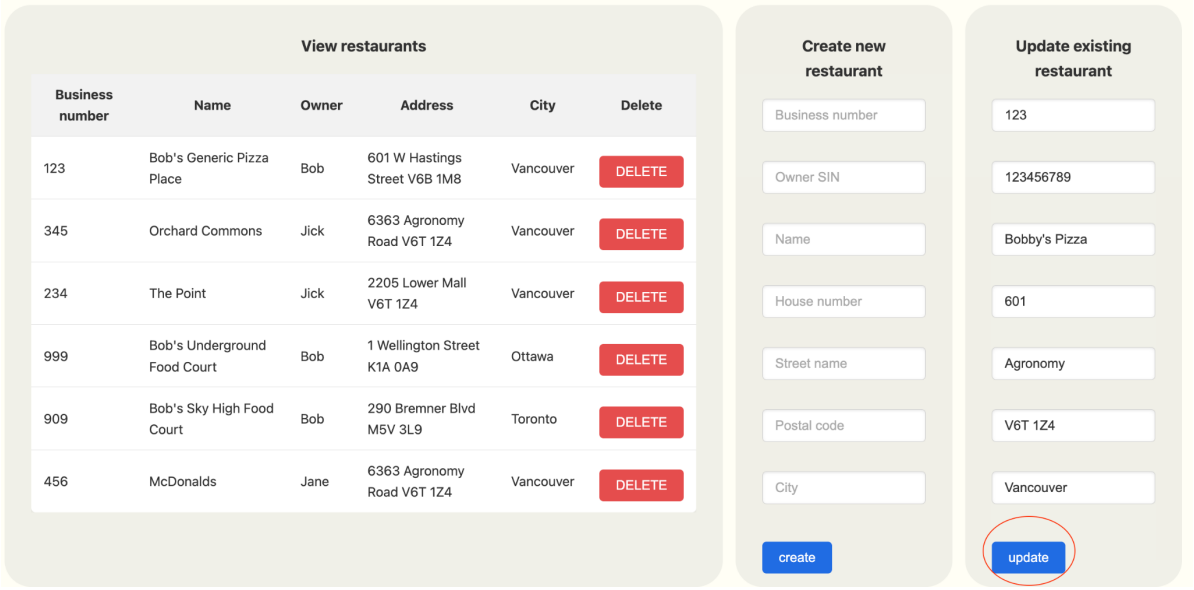
University of British Columbia, Vancouver
Department of Computer Science

Update:

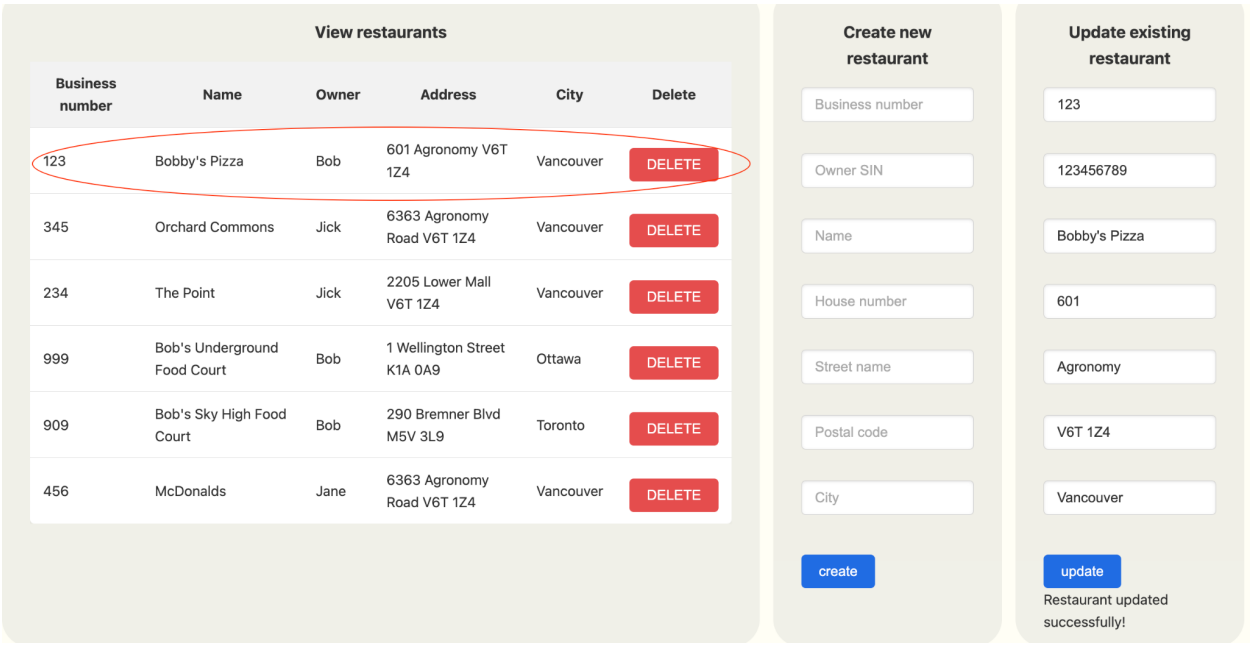
Before:



During:

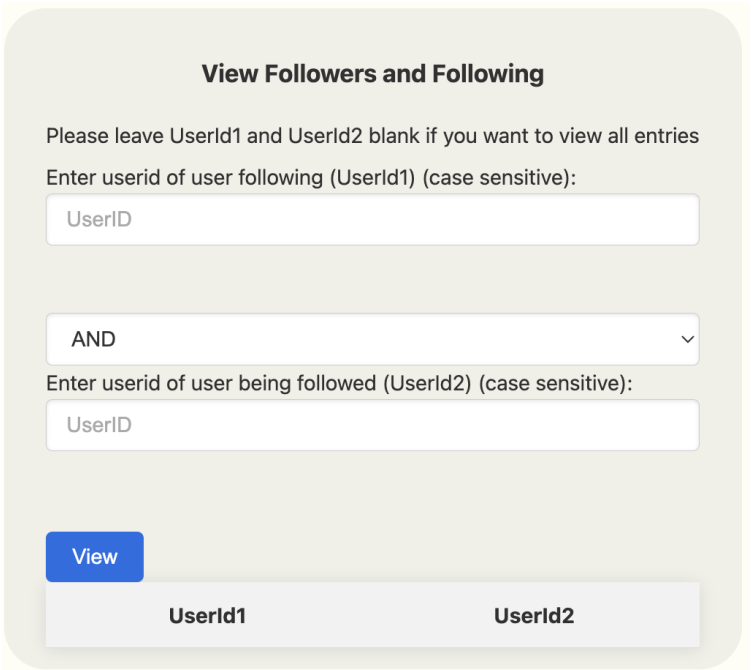


After:

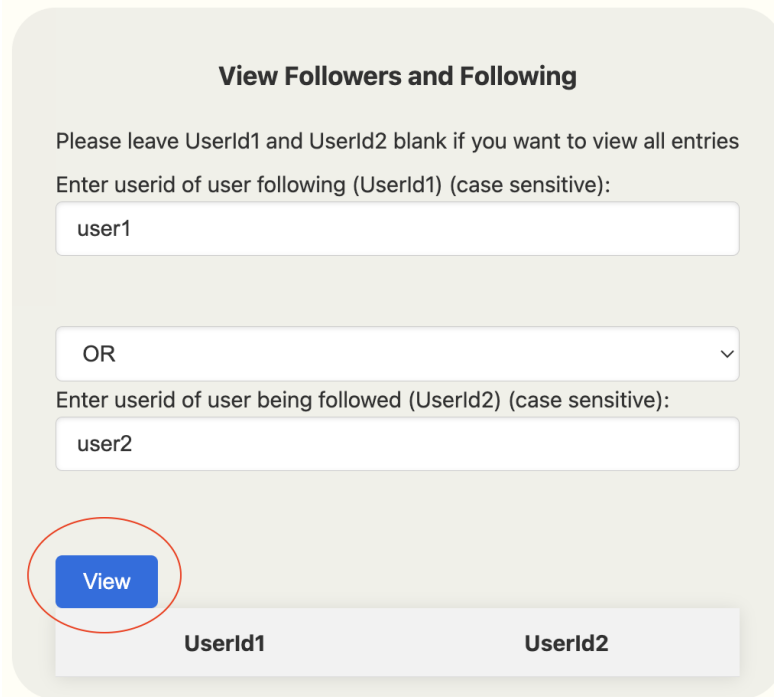


Selection:

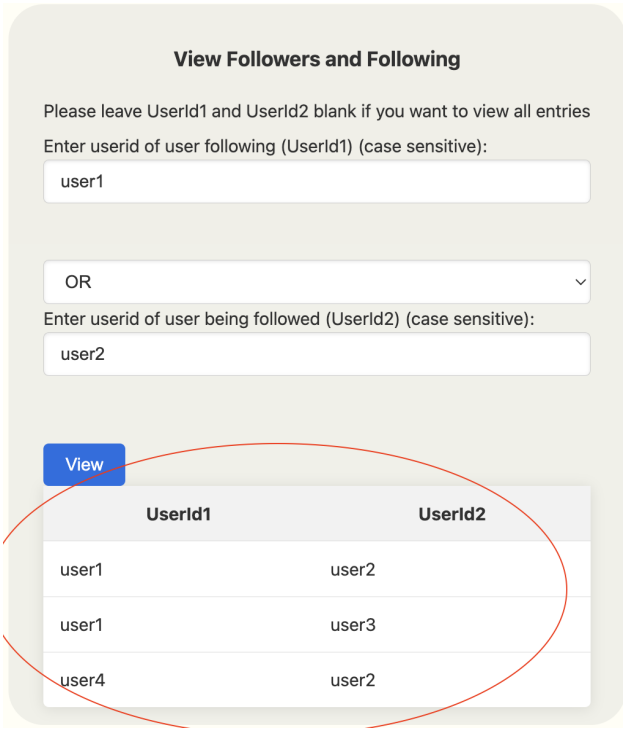
Before:



During:

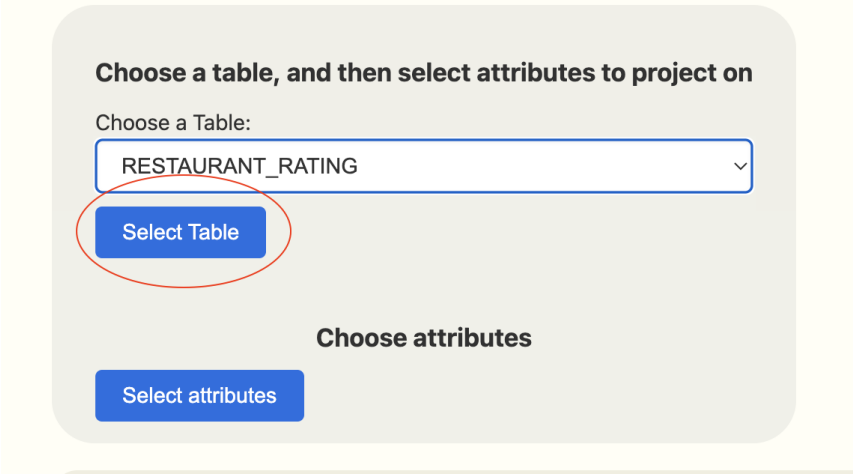


After:

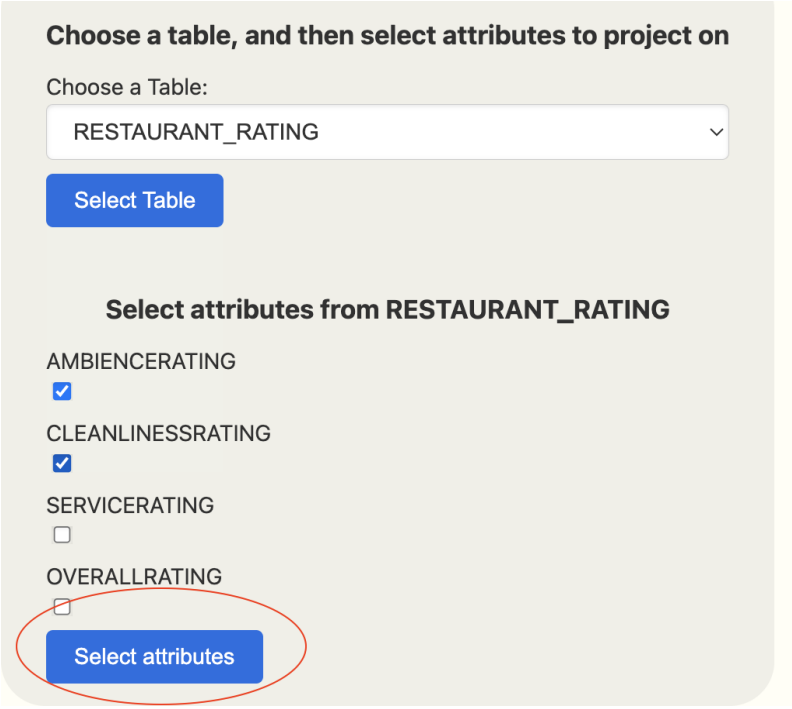


Projection:

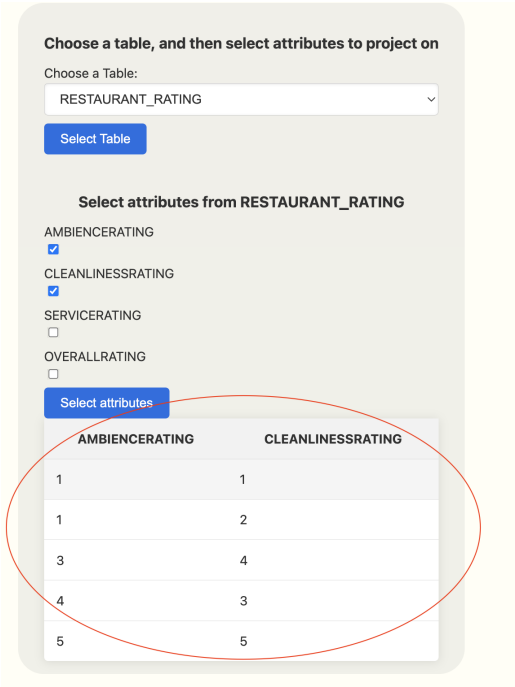
Before:



During:

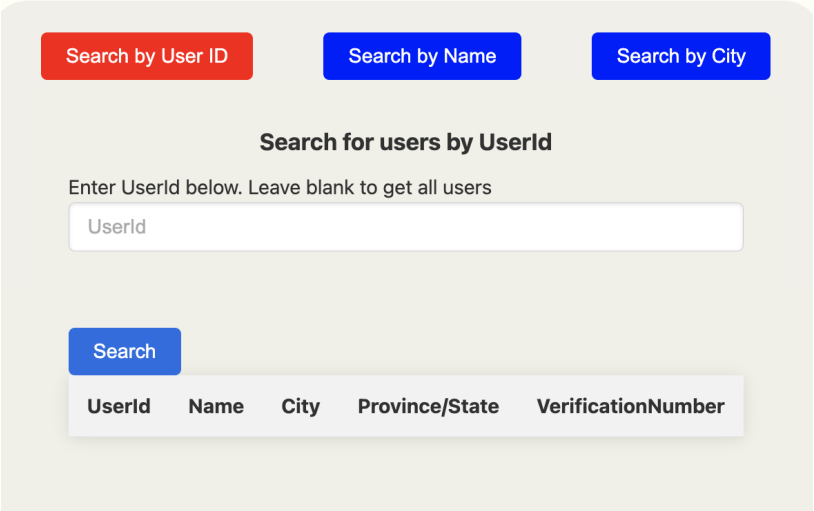


After:

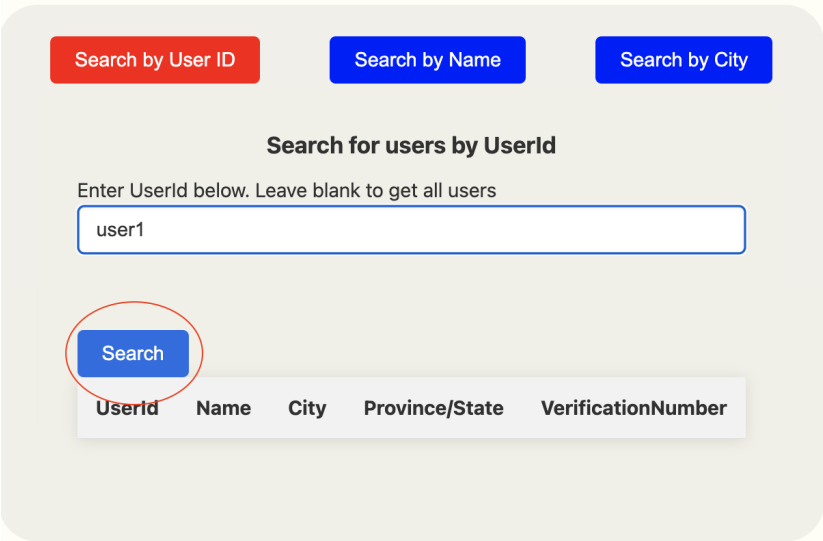


Join

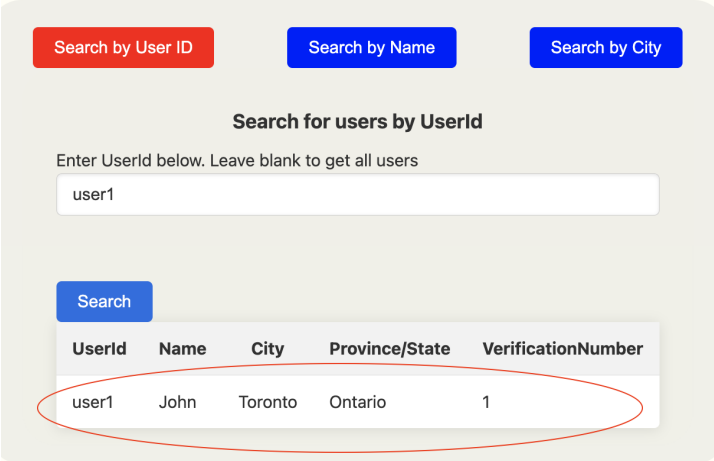
Before:



During:

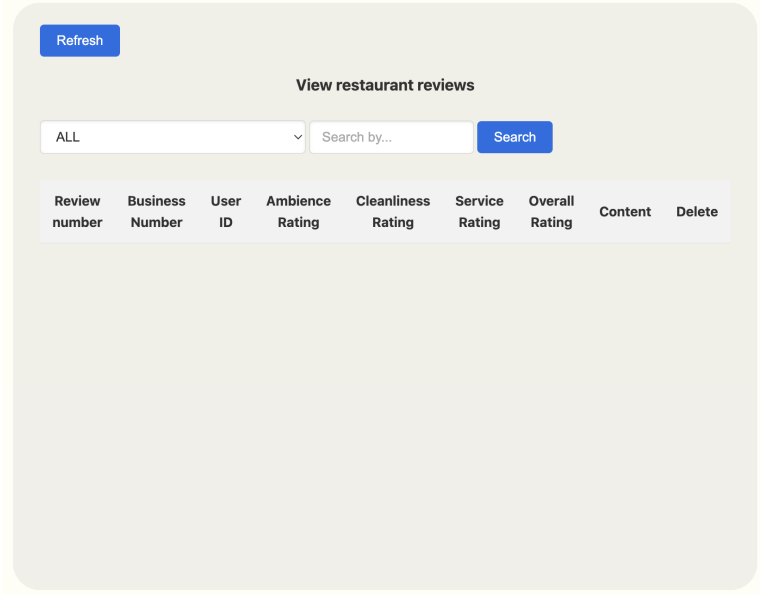


After:

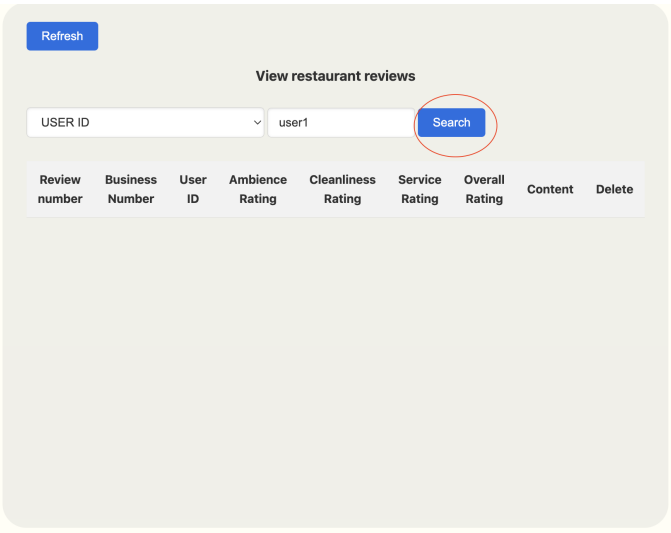


Aggregation with group by:

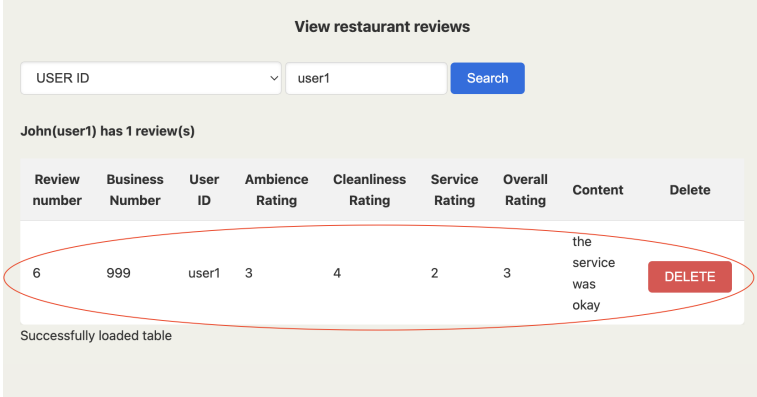
Before:



During:

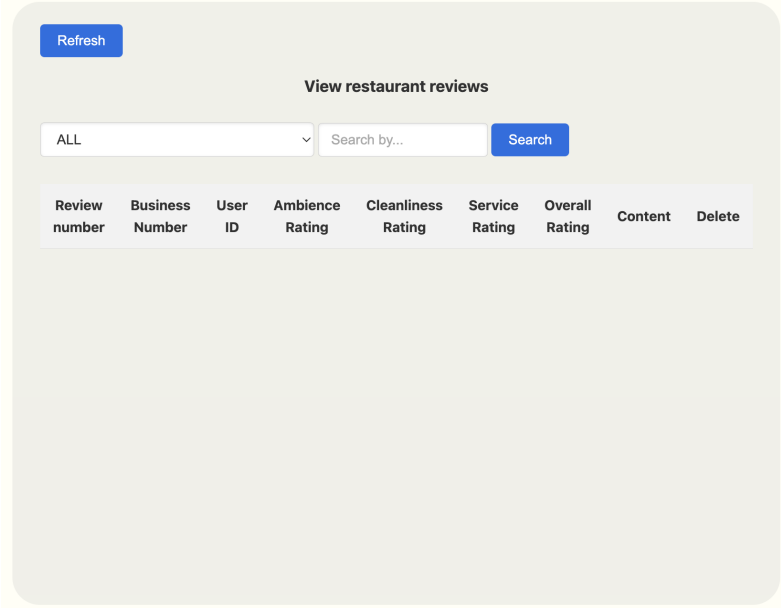


After:

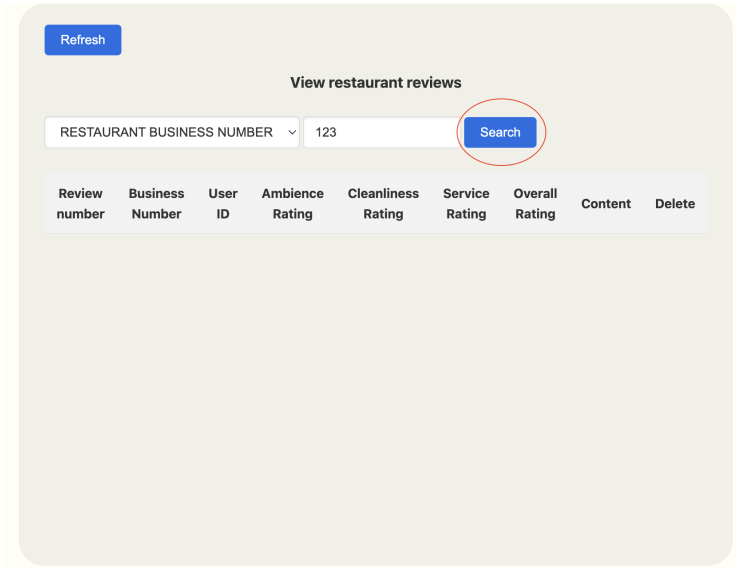


Aggregation with having:

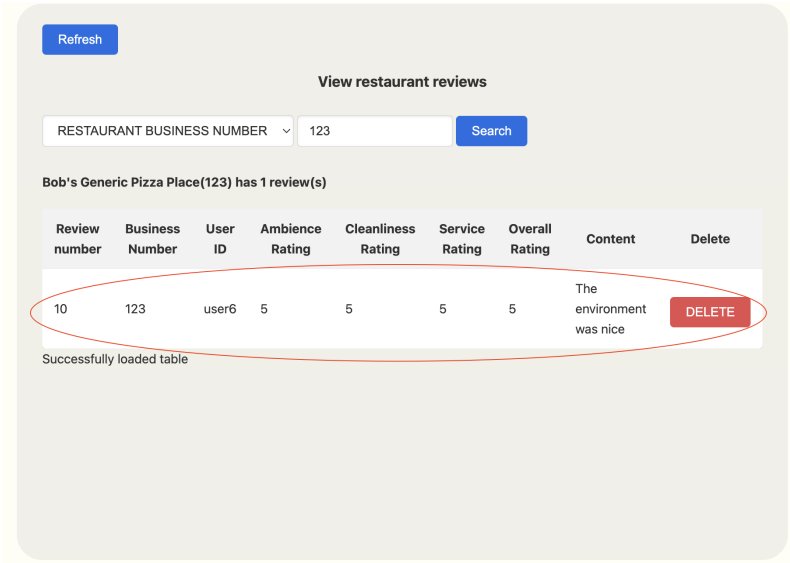
Before:



During:



After:



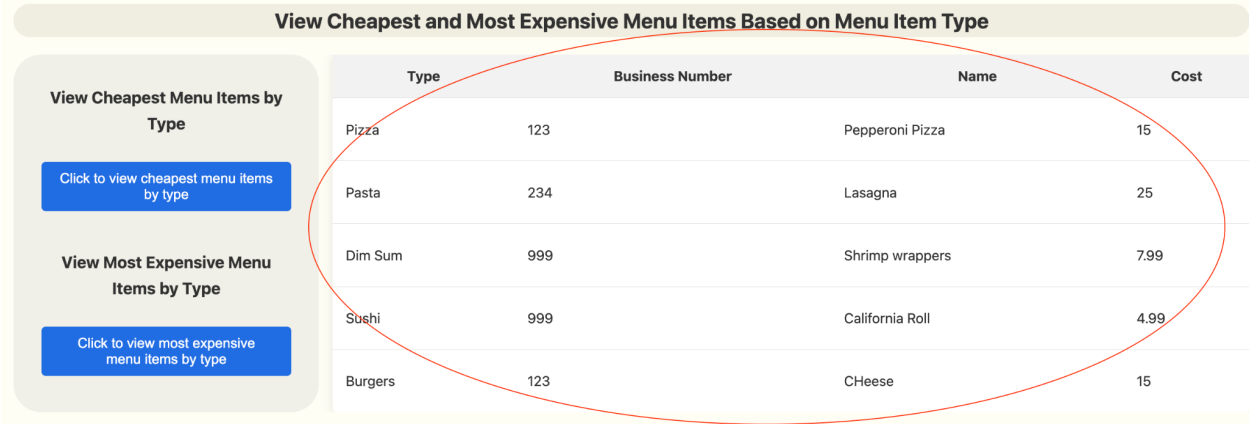
Nested aggregation with group by:

University of British Columbia, Vancouver
Department of Computer Science

Before:

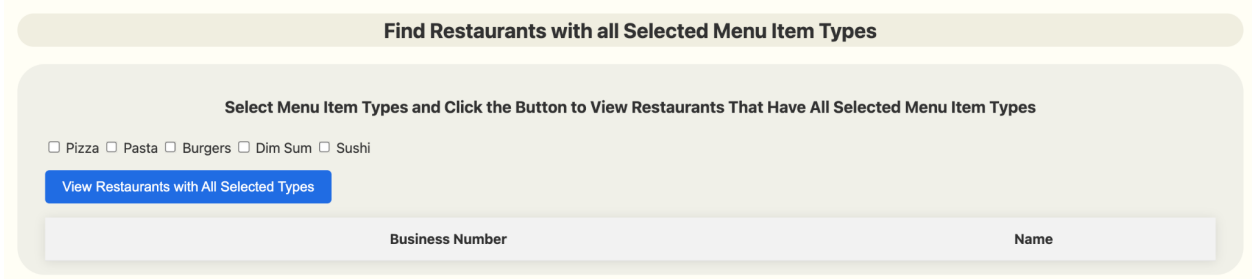
During:

After:

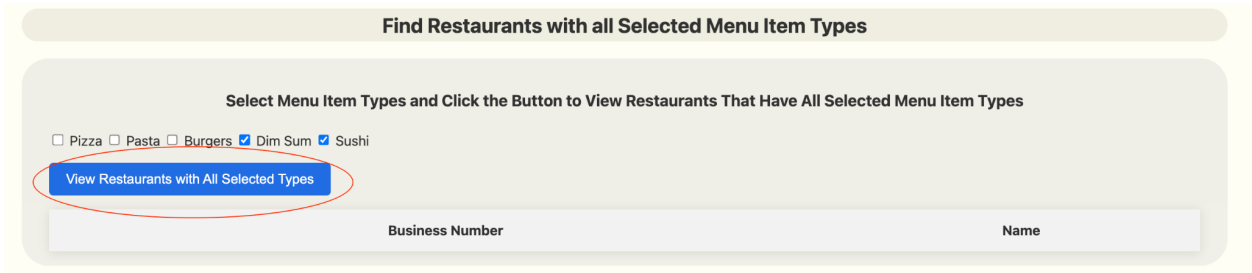


Division:

Before:



During:



After: