CPSC 304 Project Cover Page

Milestone #: 4, 5, 6

Date: 2024.10.25

Group Number: 55

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Jiakang Huang	66382623	t4h5d	jhuang74@student.ubc.ca
Luocheng Shi	64544398	n8h0x	lshi10@ubc.student.ca
texin (Cosine) feng	44781680	o2q5v	kfeng11@student.ubc.ca

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

• A short description of the final project, and what it accomplished.

Our project is a restaurant rating app designed to enhance customer engagement by allowing Users to explore and search for restaurants, rate and review them, and discover dishes using personalized search criteria. The app also provides aggregated insights offering users a deeper understanding of dining options. For restaurant owners, the app provides management tools to add/delete new restaurants, update ratings, or change ownership details.

• A description of how your final schema differed from the schema you turned in. If the final schema differed, explain why.

Added ON DELETE CASCADE to tables with foreign keys. Ensured the consistency of our database after the delete action.

• A list of all SQL queries used to satisfy the rubric items and where each query can be found in the code (file name and line number(s)).

2.1.1 insert

handleAddRestaurant

https://github.students.cs.ubc.ca/CPSC304-2024W-T1/project_n8h0x_o2q5v_t4h5d/blob/662be7fa6537b05b7e27d4c96e3a6c05175b9cbe/project.php#L456

```
SELECT name FROM Restaurant WHERE name = :name
SELECT ID FROM Owner WHERE name = :ownerName
SELECT NVL(MAX(ID), 0) + 1 AS new_id FROM Owner
INSERT INTO Owner (ID, name) VALUES (:id, :name)
INSERT INTO Restaurant (name, owner_ID, rating) VALUES (:name, :ownerId, :rating)
```

2.1.2 update

handleUpdateRestaurantRating

https://github.students.cs.ubc.ca/CPSC304-2024W-T1/project_n8h0x_o2q5v_t4h5d/blob/662be7fa6537b05b7e27d4c96e3a6c05175b9cbe/project.php#L531

```
UPDATE Restaurant SET rating = :newRating WHERE name = :restaurantName
UPDATE Restaurant SET owner_ID = :ownerId WHERE name = :restaurantName
```

2.1.3 delete

handleDeleteRestaurant

https://github.students.cs.ubc.ca/CPSC304-2024W-T1/project_n8h0x_o2q5v_t4h5d/blob/662be7fa6537b05b7e27d4c96e3a6c05175b9cbe/project.php#L607

```
DELETE FROM Restaurant WHERE name = :restaurantName
```

2.1.4 selection

handleSearchDishesRequest

https://github.students.cs.ubc.ca/CPSC304-2024W-T1/project_n8h0x_o2q5v_t4h5d/blob/662be7fa6537b05b7e27d4c96e3a6c05175b9cbe/project.php#L632

```
$query = "SELECT RESTAURANT_NAME, NAME, PRICE FROM Dishes";
if (!empty($conditions)) {
	$query .= " WHERE " . implode(" ", $conditions)
}
```

2.1.5 projection

handleProjectionRequest

https://github.students.cs.ubc.ca/CPSC304-2024W-T1/project_n8h0x_o2q5v_t4h5d/blob/662be7fa6537b05b7e27d4c96e3a6c05175b9cbe/project.php#L731

```
SELECT $attributesList FROM Emp1
```

2.1.6 join

Find Chefs by Restaurant

handleFindChefsByRestaurant

https://github.students.cs.ubc.ca/CPSC304-2024W-T1/project_n8h0x_o2q5v_t4h5d/blob/662be7fa6537b05b7e27d4c96e3a6c05175b9cbe/project.php#L951

```
SELECT DISTINCT H.ID, H.style, H.skill_level
FROM Cook C
JOIN Chef H ON C.chef_id = H.ID
WHERE C.restaurant_name = :restaurantName
```

• For SQL queries 2.1.7 through 2.1.10 inclusive, include a copy of your SQL query and a maximum of 1-2 sentences describing what that query does. You can embed this in your above list of queries. You don't need to include the output of the query. The purpose of this requirement is to allow your TAs time outside of your presentation to verify these more complex queries are well formed.

2.1.7 aggregation with group by

This query will return the average dish Price for all restaurants. Now you can determine where you want to eat based on your budget.

handleAggregationRequest

https://github.students.cs.ubc.ca/CPSC304-2024W-T1/project_n8h0x_o2q5v_t4h5d/blob/662be7fa6537b05b7e27d4c96e3a6c05175b9cbe/project.php#L829

```
SELECT restaurant_name, AVG(price) AS average_price
FROM Dishes
GROUP BY restaurant_name
```

2.1.8 aggregation with having

This will return a list of restaurants which has more than 2 dishes. You always want to eat at a place with more options!

```
handleHavingRequest
https://github.students.cs.ubc.ca/CPSC304-2024W-T1/project_n8h0x_o2q5v_t4h5d/blob/662be7fa6537b05b7e27d4c96e3a6c05175b9cbe/project.php#L852
SELECT restaurant_name, COUNT(*) AS dish_count
FROM Dishes
GROUP BY restaurant_name
HAVING COUNT(*) > 2
```

2.1.9 nested aggregation with group by

This query will return the restaurant with the highest average chef skill level. You want to know which restaurant is most professional!

```
handleNestedAggregationRequest
https://github.students.cs.ubc.ca/CPSC304-2024W-T1/project_n8h0x_o2q5v_t4h5d/blob/662be7fa6537b05b7e27d4c96e3a6c05175b9cbe/project.php#L880
SELECT C.restaurant_name, AVG(H.skill_level) AS avg_skill_level
FROM Cook C, Chef H
WHERE C.chef_id = H.ID
GROUP BY C.restaurant_name
HAVING AVG(H.skill_level) >= ALL (
SELECT AVG(H2.skill_level)
FROM Cook C2, Chef H2
WHERE C2.chef_id = H2.ID
GROUP BY C2.restaurant_name
)
```

2.1.10 division

This query will return a list of users who have eaten all the dishes before. You know who is better than googleMap now.

```
handleDivisionRequest
https://github.students.cs.ubc.ca/CPSC304-2024W-T1/project_n8h0x_o2q5v_t4h5d/blob/662be7fa6537b05b7e27d4c96e3a6c05175b9cbe/project.php#L918
```

```
SELECT U.ID, U.name FROM \"User\" U
WHERE NOT EXISTS (
SELECT D.name FROM Dishes D
MINUS
SELECT E.dish_name FROM Eat E WHERE E.user_id = U.ID
)
```