University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: ___4___

Date: ___29 Nov, 2024____

Group Number: ___97____

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Vamsi Nadella	60336625	p3c3i	nadellavamsi06@gmail.com
Jagathi Moturi	81887028	x4n6s	jagathi.moturi@gmail.com
Hans Chen	46387841	i2c5g	hanschen516@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

PROJECT REPOSITORY INFORMATION

Our project repository can be accessed at the below link:

https://github.students.cs.ubc.ca/orgs/CPSC304-2024W-T1/teams/project_i2c5g_p3c3i_x4n6s

Our full SQL script for setting up the database can be found inside the /src folder. A copy of our project ER diagram can be found in the repository.

PROJECT DESCRIPTION

The Renovation Project Management application is a tool designed to simplify the coordination between different stakeholders of renovation projects. It enables efficient tracking of project timelines, budgets, materials and contractor assignments for both residential and commercial renovations. The application is tailored for project supervisors and property owners to oversee multiple projects seamlessly. It also includes a review system where owners can provide feedback upon project completion. For our final project, we developed an easy to use frontend interface that allows users to interact with the database effectively, ensuring a smooth and user-friendly experience for managing renovation tasks.

As a project supervisor, the user is able to:

- **View Projects and Apply Filters**
    - Access a list of projects they oversee and apply customized filters, such as project status, timeline, or budget range, to find relevant information efficiently.
- **View Material Costs**
    - Analyze the average material cost by project type to identify cost trends and help in optimizing resource allocation.
- **Identify Supervisor working on high-cost projects**
    - Get a list of supervisors working on projects with average total cost higher than average total cost of all projects.
- **Identify Supervisors managing projects of all owner types**
    - Get a list of supervisors managing projects of all owner types.

As a property owner, the user is able to:

- **Customize Project View**
    - Select specific columns to display, enabling a personalized view of project details such as timelines and costs.

- **Add Projects**
    - Add new renovation project to the existing projects.
- **Delete Projects**
    - Delete an existing renovation project from the database
- **Update Projects**
    - Update the information of an existing renovation project from the database.
- **Add Reviews**
    - Provide reviews on completed projects.
- **View Budget Details**
    - Access detailed budget information for each project, including material costs, contractor fees, and total expenses.
- **Analyze Average Project Costs**
    - Calculate the average cost of their projects.

CHANGES TO SCHEMA

For implementing this project, no changes were made to the original ER diagram, but there were very minute changes to the schema (like adding ON DELETE CASCADE) for Residential and Commercial Projects to make sure that the child project gets deleted when the parent project is removed.
Also, additional tuples were added to the database tables to enhance testing and provide more diverse options to interact and query.

SQL QUERIES

1. **INSERT Operation**

Our INSERT query allows project owners to insert a new project into the Project table. The user will be able to input values for the details (Attributes) of the Project such as Project ID, name, start date, end date etc. This query is also demonstrated by allowing owners to insert a new review into the Review table along with values for its attributes.

The implementation for the insert query can be found in lines 614-615 of ownerpage.php file located at src/ownerpage.php.

Link:
https://github.students.cs.ubc.ca/CPSC304-2024W-T1/project_i2c5g_p3c3i_x4n6s/blob/deec483a3c4d35530d5f37edc2f174abcf15d9ca/src/ownerpage.php#L614

2. **DELETE Operation**

Our DELETE operation allows project owners to remove an existing project from the Project table. When a project is deleted, the system ensures that all associated data, such as budget details linked to the project, are also cleaned up to maintain database integrity. This operation helps owners manage their project list effectively by removing completed or canceled projects.

The implementation for the DELETE query can be found in lines 773 of ownerpage.php file located at src/ownerpage.php.

Link:
https://github.students.cs.ubc.ca/CPSC304-2024W-T1/project_i2c5g_p3c3i_x4n6s/blob/deec483a3c4d35530d5f37edc2f174abcf15d9ca/src/ownerpage.php#L773

3. **Update Operation**

Our UPDATE operation allows project owners to modify the details of an existing project in the Project table. This feature enables owners to make changes such as updating the project timeline, supervisor or address.

The implementation for the UPDATE query can be found in lines 867,918, 936,938 of ownerpage.php file located at src/ownerpage.php.

Link:
https://github.students.cs.ubc.ca/CPSC304-2024W-T1/project_i2c5g_p3c3i_x4n6s/blob/deec483a3c4d35530d5f37edc2f174abcf15d9ca/src/ownerpage.php#L867

4. **Selection**

Our SELECT operation allows users to retrieve and view project details from the Project table. Users can apply various filters, such as project status, budget range, or start/end dates to customize the data they view. This functionality provides flexibility in accessing specific project information.

The implementation for the SELECTION query can be found in lines 362-462 of supervisorpage.php file located at src/supervisorpage.php

Link:
https://github.students.cs.ubc.ca/CPSC304-2024W-T1/project_i2c5g_p3c3i_x4n6s/blob/deec483a3c4d35530d5f37edc2f174abcf15d9ca/src/supervisorpage.php#L362

5. **Projection**

Our PROJECTION operation allows users to dynamically select which attributes to view from any existing table in the database. By specifying the desired columns, users can customize their results to display only the information they need, such as project name, budget, supervisor ID or start/end dates.

The implementation for the UPDATE query can be found in lines 248-325 of ownerpage.php file located at src/ownerpage.php.

Link:
https://github.students.cs.ubc.ca/CPSC304-2024W-T1/project_i2c5g_p3c3i_x4n6s/blob/deec483a3c4d35530d5f37edc2f174abcf15d9ca/src/ownerpage.php#L248

6. **Join**

Our JOIN query allows the user to input a Project ID and view combined details of the project and its associated budget details in a single table. It retrieves key project information, such as project ID, name and address, alongside budget details by joining the project and budget tables based on the provided project ID. This provides the user with a comprehensive overview of both the project and its financials in one view.

The implementation for the join query can be found in lines 250-258 of budgetpage.php file located at src/budgetpage.php

Link:
https://github.students.cs.ubc.ca/CPSC304-2024W-T1/project_i2c5g_p3c3i_x4n6s/blob/deec483a3c4d35530d5f37edc2f174abcf15d9ca/src/budgetpage.php#L250

7. **Aggregation with GROUP BY**

Our Aggregation with GROUP BY query calculates the average material cost for projects, grouped by the owner type (residential or commercial). It joins the Project, OwnerEntity, and Budget tables and uses the GROUP BY clause to aggregate the results by the Owner_Type. The query also counts the total number of projects for each owner type and calculates the average material cost.

```
$groupby_sql = "
    SELECT OE.Owner_Type AS Project_Type, COUNT(P.Project_ID) AS Total_Projects, ROUND(AVG(B.BUDGET_MATERIAL_COST),2) AS Avg_Material_Cost
    FROM Project P
    JOIN OwnerEntity OE ON P.Owner_ID = OE.Owner_ID
    JOIN Budget B ON P.Budget_ID = B.Budget_ID
    GROUP BY OE.Owner_Type
";
```

The implementation for the aggregation with GROUP BY query can be found in lines 472-478 of supervisorpage.php file located at src/supervisorpage.php

Link:
https://github.students.cs.ubc.ca/CPSC304-2024W-T1/project_i2c5g_p3c3i_x4n6s/blob/deec483a3c4d35530d5f37edc2f174abcf15d9ca/src/supervisorpage.php#L472

8. **Aggregation with HAVING**

Our Aggregation with HAVING query displays supervisors who manage more than a specified number of projects by the user. It joins the Supervisor and Project tables, groups the results by Supervisor ID and Supervisor Name, and counts the number of projects each supervisor manages. The HAVING clause filters the results to include only those supervisors who manage more projects than the specified number by the user.

```
$query = "
    SELECT
        s.Supervisor_ID,
        s.Supervisor_Name,
        COUNT(p.Project_ID) AS Number_Of_Projects
    FROM
        Supervisor s
    JOIN
        Project p ON s.Supervisor_ID = p.Supervisor_ID
    GROUP BY
        s.Supervisor_ID, s.Supervisor_Name
    HAVING
        COUNT(p.Project_ID) > :threshold
";
```

The implementation for the aggregation with HAVING query can be found in lines 328-341 of budgetpage.php file located at src/budgetpage.php

Link:
https://github.students.cs.ubc.ca/CPSC304-2024W-T1/project_i2c5g_p3c3i_x4n6s/blob/deec483a3c4d35530d5f37edc2f174abcf15d9ca/src/budgetpage.php#L328

9. **Nested Aggregation with GROUP BY**

Our Nested Aggregation with GROUP BY query identifies supervisors who are managing projects with an average total cost higher than the overall average total cost of all projects. It uses a nested query with GROUP BY and HAVING to calculate the average total cost of projects supervised by each supervisor and compares it to the average total cost across all projects in the Project table. The outer query filters and retrieves the details of such supervisors.

```
$nested_query = "
 SELECT
    S.Supervisor_ID,
    S.Supervisor_Name
 FROM
    Supervisor S
 WHERE EXISTS (
    SELECT 1
    FROM Project P
    JOIN Budget B ON P.Budget_ID = B.Budget_ID
    WHERE P.Supervisor_ID = S.Supervisor_ID
    GROUP BY P.Supervisor_ID
    HAVING AVG(B.Budget_Total_Cost) > (
        SELECT AVG(B2.Budget_Total_Cost)
        FROM Budget B2
    )
 )
 ";
```

The implementation for the Nested aggregation with GROUP BY query can be found in lines 493-510 of supervisorpage.php file located at src/supervisorpage.php

Link: https://github.students.cs.ubc.ca/CPSC304-2024W-T1/project_i2c5g_p3c3i_x4n6s/blob/deec483a3c4d35530d5f37edc2f174abcf15d9ca/src/supervisorpage.php#L493

10. DIVISION

Our DIVISION query identifies supervisors who manage projects across all owner types. It uses division by ensuring there is no owner type in the OwnerEntity table that is not associated with a project managed by the supervisor. The outer query retrieves the details of such supervisors who cover all property types.

```
$division_sql = "
SELECT S.Supervisor_ID, S.Supervisor_Name
FROM Supervisor S
WHERE NOT EXISTS (
    SELECT OE.Owner_Type
    FROM OwnerEntity OE
    WHERE OE.Owner_Type NOT IN (
        SELECT DISTINCT OE2.Owner_Type
        FROM Project P
        JOIN OwnerEntity OE2 ON P.Owner_ID = OE2.Owner_ID
        WHERE P.Supervisor_ID = S.Supervisor_ID
    )
)";
```

The implementation for the Nested aggregation with GROUP BY query can be found in lines 530-541 of supervisorpage.php file located at src/supervisorpage.php

Link:

https://github.students.cs.ubc.ca/CPSC304-2024W-T1/project_i2c5g_p3c3i_x4n6s/blob/deec483a3c4d35530d5f37edc2f174abcf15d9ca/src/supervisorpage.php#L530