Milestone #: 4

Date: Mar. 28, 2025

Group Number: 98

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Alex Yang	82158916	g1q1p	alexemail67@gmail.com
Gordon Zhou	60448990	e0g8g	gordonzhou223@gmail.com
Andy Xie	17324963	q4p5y	andyxiehi@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

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

The domain of the application is disaster crisis response/logistics. The application is meant to address the challenges faced during natural disasters, where organizing volunteers, missions, and aid distribution can be overwhelming. This app will help make that process more efficient by ensuring that those in need get the help they need as quickly as possible.

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

Our final schema did not change from the schema that we turned in in the previous milestones, other than making a few minor syntactical changes to allow the sql file to successfully run.

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)).

1. Insert
   project_e0g8g_g1q1p_q4p5y/backend/rc_queries.php
   Lines 63-66

```
"INSERT INTO Mission VALUES ({$missionID},
      '{$missionType}', SYSDATE, {$helpNeeded}, '{$name}',
      '{$disasterDate}', '{$location}',
      '{$rc_name}', '{$rc_location}', '{$priority}')"
```

2. Update
   project_e0g8g_g1q1p_q4p5y/backend/rc_queries.php
   Lines 171-172

```
"UPDATE Supplies SET quantity = quantity + {$sendAmount}
                     WHERE supplyID = {$existingSupplyID}"
```

3. Delete
   project_e0g8g_g1q1p_q4p5y/backend/rc_queries.php
   Line 259

```
"DELETE FROM Supplies WHERE supplyID='{$supplyID}'"
```

4. Selection - project_e0g8g_g1q1p_q4p5y/backend/main_page_queries.php
   Lines 12-40

```
    $query = "SELECT * FROM Disaster WHERE 1=1";
    $params = [];

    $filters = [
        "name" => "disasterName",
        "disasterDate" => "disasterDate",
```

```
        "location" => "disasterLocation",
        "damageCost" => "damageCost",
        "casualties" => "casualties",
        "severityLevel" => "severityLevel",
        "type" => "type"
    ];

    foreach ($filters as $column => $param) {
        if (!empty($_GET[$param])) {
            if (in_array($column, ['damageCost', 'casualties', 'severityLevel'])) {
                // use exact match for numbers
                $query .= " AND $column = :$param";
                $params[$param] = $_GET[$param];
            } elseif ($column == 'disasterDate') {
                $query .= " AND TO_CHAR($column, 'YYYY-MM-DD') = :$param";
                $params[$param] = $_GET[$param];
            } else {
                // use LIKE for text fields
                $query .= " AND LOWER($column) LIKE LOWER(:$param)";
                $params[$param] = "%" . $_GET[$param] . "%";
            }
        }
    }
```

5. Projection

```
$query = "SELECT DISTINCT";
    if (isset($_POST['checkboxes']) && is_array($_POST['checkboxes'])) {
        // Loop through all selected checkbox values
        foreach ($_POST['checkboxes'] as $checkbox) {
            if (in_array($checkbox, $validColumns)) {
            // adds to select query
            $query .= " " . $checkbox . ",";
            }
        }
        // gets rid of leading ","
        $query = substr($query,0,-1);
        $query .= " FROM Mission";
    }
```

project_e0g8g_g1q1p_q4p5y/backend/contributor_queries.php
Lines 84-96

6. Join - project_e0g8g_g1q1p_q4p5y/backend/main_page_queries.php
Lines 221-225

```
    $missionType = strtolower($_GET['missionType']);
```

```
    $query = "SELECT DISTINCT rc.name, rc.location, m.missionType, m.priority
              FROM Mission m
              JOIN ReliefCenter rc ON rc.name = m.rcName AND rc.location = m.rcL
              WHERE LOWER(m.missionType) LIKE :missionType";
```

7. Aggregation with GROUP BY

```
 $result = executePlainSQL("SELECT MissionType, AVG(Priority) AS AvgPriority,
 COUNT(*) AS MissionCount
FROM Mission GROUP BY MissionType");
```

project_e0g8g_g1q1p_q4p5y/backend/contributor_queries.php
Line 110

8. Aggregation with HAVING

```
$query = "SELECT disasterName, SUM(helpNeeded) AS totalHelp
FROM Mission GROUP BY disasterName HAVING SUM(helpNeeded) > {$value}
ORDER BY SUM(helpNeeded) DESC";
```

project_e0g8g_g1q1p_q4p5y/backend/contributor_queries.php
Line 60

9. Nested aggregation with GROUP BY - relief center
   project_e0g8g_g1q1p_q4p5y/backend/rc_queries.php
   Lines 29-39

```
"
  SELECT m.missionID, m.helpNeeded - COUNT(DISTINCT vf.name || vf.phoneNUmber)
    AS helpStillNeeded
  FROM VolunteersFor vf, Mission m
  WHERE m.missionID = vf.missionID
  GROUP BY m.missionID, m.helpNeeded
  HAVING m.helpNeeded > (
      SELECT COUNT(*)
      FROM VolunteersFor vf2
      WHERE vf2.missionID = m.missionID
  )
  "
```

10. Division

```
"SELECT DISTINCT D.name, D.disasterDate, D.location
                    FROM Disaster D
                    WHERE NOT EXISTS (
                        SELECT M.missionType
                        FROM Mission M
                        WHERE NOT EXISTS (
                            SELECT *
```

```
                            FROM Mission M2
                            WHERE M2.disasterName = D.name
                            AND M2.disasterDate = D.disasterDate
                            AND M2.disasterLocation = D.location
                            AND M2.missionType = M.missionType
                        )
                    )"
```

project_e0g8g_g1q1p_q4p5y/backend/contributor_queries.php
Line 27-40

For SQL queries 7 through 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

Aggregation with GROUP BY - Contributor

```
"SELECT MissionType, AVG(Priority) AS AvgPriority, COUNT(*) AS MissionCount
      FROM Mission GROUP BY MissionType"
```

Find Average Priority of Each Mission Type and the counts of every mission per type.

Aggregation with HAVING - Contributor

```
"SELECT disasterName, SUM(helpNeeded) AS totalHelp FROM Mission
GROUP BY disasterName HAVING SUM(helpNeeded) > {$value}
ORDER BY SUM(helpNeeded) DESC"
```

Returns a relation that sums all the helpNeeded grouped by disasterName having the total helpNeeded be greater than an amount $value specified by the user.

Nested aggregation with GROUP BY - relief center

```
"
    SELECT m.missionID, m.helpNeeded - COUNT(DISTINCT vf.name || vf.phoneNUmber)
      AS helpStillNeeded
    FROM VolunteersFor vf, Mission m
    WHERE m.missionID = vf.missionID
    GROUP BY m.missionID, m.helpNeeded
    HAVING m.helpNeeded > (
        SELECT COUNT(*)
```

```
        FROM VolunteersFor vf2
        WHERE vf2.missionID = m.missionID
    )
    "
```

Finds missions that have less volunteers than the amount of help needed and shows the difference.

Division

```
"SELECT DISTINCT D.name, D.disasterDate, D.location
                    FROM Disaster D
                    WHERE NOT EXISTS (
                        SELECT M.missionType
                        FROM Mission M
                        WHERE NOT EXISTS (
                            SELECT *
                            FROM Mission M2
                            WHERE M2.disasterName = D.name
                            AND M2.disasterDate = D.disasterDate
                            AND M2.disasterLocation = D.location
                            AND M2.missionType = M.missionType
                        )
                    )"
```

Finds the disaster with all mission types. Get the disasters that need all the types of help.