University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 4

Date: Aug 9, 2022

Group Number: 11

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Aaron Chan	78154739	j0o6f	chan.aaron73@gmail.com
Alvin Zhou	26767608	t8g0z	alvinzen9@gmail.com
Sam Zhao	20992368	g7j6w	samzhao273@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

Repository: https://github.students.cs.ubc.ca/CPSC304-2022S-T2/project_g7j6w_j0o6f_t8g0z

Our final project is a dating application database in which the user can interact with to insert and query data in an easy and efficient manner. It is able to query through the database to insert, update, project and delete profiles, select certain attributes from different tables, filter out names based on certain filters, and be able to show different aggregation relationships, such as count the number of profiles within each gender group (as well as strictly male and female), find the tallest profiles for each gender, and find profiles who have matched with everyone. It accomplished a multitude of tasks such as being able to analyze data for a dating application company and is useful to these companies as they can see how effective their applications are for each profile.

Our final schema had one change for ProfileCreate_3, as it previously had cyclic behavior between foreign keys, which prevented the SQL script from running correctly. As a result, we deleted the uid attribute that referenced UserDeactivate_2. Furthermore, the email attribute for UserDeactivate_1 and UserDeactivate_2 previously used CHAR(20). However, we realized that the length of 20 was too short, thus we changed it to CHAR(30) to account for longer emails.

These are the final tables that have been edited:

ProfileCreate$_3$(id: integer, name: char(20), height: integer, gender: char(6), dob: date)

UserDeactivate$_1$(email: char(30), password: char(50))

UserDeactivate$_2$(uid: integer, email: char(30), **id: integer**)

SQL Queries:

Insert (found in handleInsertRequest):

```
INSERT INTO ProfileCreate_1(dob, age) VALUES (TO_DATE('2022-08-07',
'YYYY-MM-DD'), 0);
INSERT INTO ProfileCreate_2(dob, horoscope) VALUES(TO_DATE('2022-08-07',
'YYYY-MM-DD'), 'leo');
INSERT INTO ProfileCreate_3(id, name, height, gender, dob) VALUES(5, 'Sam', 100, 'male',
TO_DATE('2022-08-07', 'YYYY-MM-DD'));
```

Delete (found in handleDeleteRequest):

```
DELETE FROM ProfileCreate_3
WHERE id=5;
```

Update (found in handleUpdateRequest):

```
UPDATE ProfileCreate_3
SET gender = 'female'
WHERE id=1;

UPDATE ProfileCreate_3
SET dob = TO_DATE('2022-08-22', 'YYYY-MM-DD')
WHERE id=1;
```

Select (found in handleSelectRequest):

```
SELECT id, name
FROM ProfileCreate_3
WHERE height > 100;
```

Projection (found in handleDeleteRequest):

```
SELECT id, name, dob
FROM ProfileCreate_3;
```

Join (found in handleJoinRequest):

```
SELECT p.name
FROM ProfileCreate_3 p
JOIN PremiumSet ps
ON p.id = ps.id
JOIN Preference pr
ON ps.id = pr.id
WHERE pr.distance < 50;
```

Aggregation with Group By (found in handleAggregationGroupByRequest):

```
SELECT COUNT(*) AS Count
FROM ProfileCreate_3
GROUP BY gender;
```

Aggregation with Having (found in handleAggregationHavingRequest):

```
SELECT COUNT(*) AS Count
FROM ProfileCreate_3
GROUP BY gender
HAVING gender <> ‘other’;
```

Nested Aggregation with Group By (found in handleNestedAggregationGroupByRequest):

```
SELECT p.name, p.id, p.height, p.gender
FROM ProfileCreate_3 p
WHERE height IN (SELECT MAX(height) FROM ProfileCreate_3 GROUP BY gender);
```

Division (found in handleDivisionRequest):

```
SELECT p.id, p.name
FROM ProfileCreate_3 p
WHERE NOT EXISTS
        ((SELECT m.mid
         FROM Match m)
         MINUS
         (SELECT r.mid
         FROM Receive r
         WHERE r.id = p.id))";
```

Sample Outputs:

Insert (Before and After):

Dating Application Database

Sam Zhao, Aaron Chan, Alvin Zhou

Insert New Profile

Profile ID: 10

Name: Joanne Jiwoo

Height (cm): 50

Gender: female

Date of Birth: 2002-09-29

Age: 19

Horoscope: Libra

Insert

Retrieved data from table:

ID	Name
1	Sam Zhao
2	Alvin Chao
3	Justin Zhao
4	Aaron Zhou
5	John Smith

Retrieved data from table:

ID	Name
10	Joanne Jiwoo
1	Sam Zhao
2	Alvin Chao
3	Justin Zhao
4	Aaron Zhou
5	John Smith

Selection:

Select Attributes from Any Table

Table to select from: ProfileCreate_3

Attributes (seperate with comma): id, name,height

Filters (Field): height > 179

Submit

Attributes Selected: id, name,height

Retrieved data from table:

Attribute 1	Attribute 2	Attribute 3	Attribute 4	Attribute 5
1	Sam Zhao	182		
2	Alvin Chao	181		
3	Justin Zhao	180		

Delete and Projection (Before and After):

Delete Profile

Profile ID: 1

Delete

ProfileCreate:

Profile ID	Name	DoB
1	Sam Zhao	25-OCT-02
2	Alvin Chao	20-OCT-02
3	Justin Zhao	26-SEP-02
4	Aaron Zhou	01-JAN-01
5	John Smith	07-JUL-22

UserDeactivate:

User ID	Email	Profile ID
1	samzhao273@gmail.com	1
2	alvinchao273@hotmail.com	2
3	justinzhao273@outlook.com	3
4	calzone@gmail.com	4
5	johnsmith@gmail.com	5

Profile with ID 1 deleted

ProfileCreate:

Profile ID	Name	DoB
2	Alvin Chao	20-OCT-02
3	Justin Zhao	26-SEP-02
4	Aaron Zhou	01-JAN-01
5	John Smith	07-JUL-22

UserDeactivate:

User ID	Email	Profile ID
2	alvinchao273@hotmail.com	2
3	justinzhao273@outlook.com	3
4	calzone@gmail.com	4
5	johnsmith@gmail.com	5

Update (Before and After):

Update Profile Settings

The values are case sensitive and if you enter in the wrong case, the update statement will not do anything.

Profile ID : 2

Attribute to Change: gender

Values of Attribute: other

Update dob (click checkbox) yyyy-mm-dd

Update

id	name	height	gender	dob
2	Alvin Chao	181	female	20-OCT-02
3	Justin Zhao	180	male	26-SEP-02
4	Aaron Zhou	179	female	01-JAN-01
5	John Smith	20	other	07-JUL-22

Updated Attributes in Profile

id	name	height	gender	dob
2	Alvin Chao	181	other	20-OCT-02
3	Justin Zhao	180	male	26-SEP-02
4	Aaron Zhou	179	female	01-JAN-01
5	John Smith	20	other	07-JUL-22

Update (dob):

Update Profile Settings

The values are case sensitive and if you enter in the wrong case, the update statement will not do anything.

Profile ID : 2

Attribute to Change:

Values of Attribute:

Update dob (click checkbox) 2002-10-30

Update

id	name	height	gender	dob
2	Alvin Chao	181	female	20-OCT-02
3	Justin Zhao	180	male	26-SEP-02
4	Aaron Zhou	179	female	01-JAN-01
5	John Smith	20	other	07-JUL-22

Updated Attributes in Profile

id	name	height	gender	dob
2	Alvin Chao	181	other	30-OCT-02
3	Justin Zhao	180	male	26-SEP-02
4	Aaron Zhou	179	female	01-JAN-01
5	John Smith	20	other	07-JUL-22

Join:

Distance Filter (Premium Feature)

Find the names of people closest to your location!

Must be less than (km) : 50

Submit

Retrieved names from Dating Application:

ID	Name
2	Alvin Chao
3	Justin Zhao
5	John Smith

Aggregation Group By:

Aggregation Group By: Count the Number of Profiles for Each Gender

Submit

Count by Gender

Gender	Count
male	1
other	2
female	1

Aggregation Having:

Aggregation Having: Count The Number of Males and Females

Submit

Count By Male and Female

Gender	Count
male	1
female	1

Nested Aggregation:

Nested Aggregation: Find Information from the Tallest People for Each Gender

Submit

Tallest Height by Gender

Name	ID	Height	Gender
Alvin Chao	2	181	other
Justin Zhao	3	180	male
Aaron Zhou	4	179	female

Division:

Division: Find Profile Who Have Received Every Single Match

Submit

Name of Profiles with Every Single Match

Profile ID	Name
2	Alvin Chao