University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 4

Date: November 25, 2022

Group Number: 28

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Sandra Radic	23018088	w1f3b	sandra_radic@outlook.com
Sarah Wong	43440171	q9k0c	sarah.wong@shaw.ca
Carina Tze	77998987	s6y2b	carinatze0@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 Link

https://github.students.cs.ubc.ca/CPSC304-2022W-T1/project_q9k0c_s6y2b_w1f3b

Project Description

The **WiCS (Women in Computer Science) Mentorship Portal** is an online resource for UBC students to meet other students through our mentorship program and/or explore event opportunities with industry professionals from our sponsor companies. To use our application, users are able to register as mentors or mentees and login to see their saved data, edit their profiles and view their mentorship match.

Updates/Changes

Schema

There were no changes to the schema from the previous milestone. Relation 'Within' was renamed 'WithinField' to improve clarity but no changes to the schema or other relations were made.

Queries

During the implementation of our project, we noticed a few opportunities to reduce redundancy and improve our application. For instance, initially we wanted to have both a filter button and join button on our event-list page so that users would be able to select their filter event, and then project the event's details. However, we realized that this could all be done with just the filter button, and we were able to implement a more optimal projection query that is more efficient and gives our app a cleaner look. Additionally, instead of having a register button we decided to give users the opportunity to view who was already registered for an event, so that they could gain better insight into which students were interested in which events, and which were more popular (eg: we used aggregation with having, and nested aggregation with group by to implement the opportunity to see the lowest and highest attendances of events, as well as join to see who is attending certain events). We believe this change is more meaningful, because users have more information available to them when deciding which events to attend.

Copy of Schema and Data Present

Schema of each relation and data present in each relation after running init.sql. Screenshots of our full schema can be found at the bottom of this document in our **Appendix**.

Queries

*SQL Tables and data can be found in 'init.sql'

INSERT

Query can be found in 'profile-signup.php' at comment '// Query: INSERT' in handleInsertRequest()

```
INSERT INTO Person VALUES(
      $id,
      $email,
      $firstName,
      $lastName,
      $year,
      $gender,
      $genderPref,
      $degree
)
```

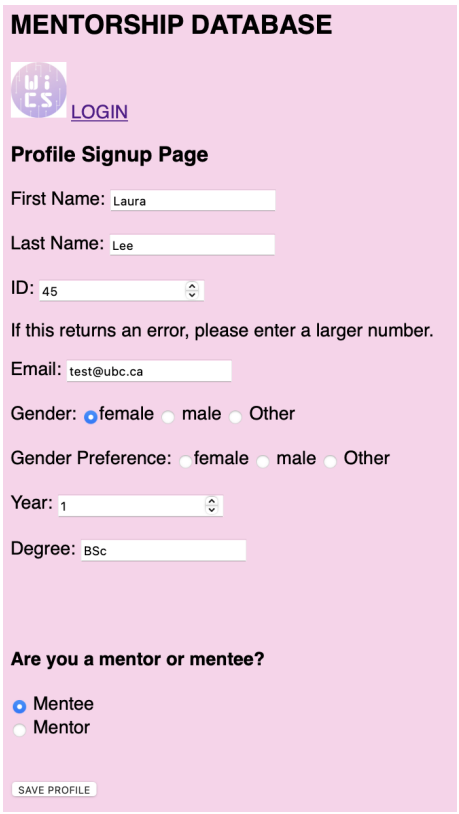
```php
//Getting the values from user and insert data into the table
$firstName = $_POST['firstName'];
$lastName = $_POST['lastName'];
$id = $_POST['pid'];
$email = $_POST['email'];
$gender = $_POST['gender'];
$genderPref = $_POST['genderPref'];
$year = $_POST['year'];
$degree = $_POST['degree'];
// echo "test";
// INSERT INTO Person VALUES (2, 'carlysmith@student.com','carly', 'smith', 2, 'female', 'male', 'BA');
$tuple = array (
    ":bind1" => $id,
    ":bind2" => $email,
    ":bind3" => $firstName,
    ":bind4" => $lastName,
    ":bind5" => $year,
    ":bind6" => $gender,
    ":bind7" => $genderPref,
    ":bind8" => $degree
);

$alltuples = array (
    $tuple
);

// Query: INSERT
executeBoundSQL("insert into Person values (:bind1, :bind2, :bind3, :bind4, :bind5, :bind6, :bind7, :bind8)", $alltuples);
OCICommit($db_conn);
```

Demonstration:

Before: there is no person with pid = 45

```
SQL> SELECT * FROM Person WHERE pid=45;

no rows selected
```

After:

Fill in information on profile-signup.php page	
Registering as a Mentor or Mentee is required, and a reminder will pop up if a user tries to register without making a selection.	

Press ‘SAVE PROFILE’ button and it will run the insert query and display new inserted profile information at the bottom of the page	**MENTORSHIP DATABASE** LOGIN **Profile Signup Page** First Name: Last Name: ID: If this returns an error, please enter a larger number. Email: Gender: female male Other Gender Preference: female male Other Year: Degree: **Are you a mentor or mentee?** Mentee Mentor SAVE PROFILE **Profile UserID: 45** Account Info: First Name: Laura Last Name: Lee Email: test@ubc.ca Year: 1 Gender: Gender Preference: female Degree: BSc Mentor or Mentee: mentee

New person is now inserted	`SQL> SELECT * FROM Person WHERE pid=45;`
	
	`PID EMAIL FIRSTNAME`
	`LASTNAME YEAR GENDERPREF GENDER DEGREE`
	`45 test@ubc.ca Laura`
	`Lee 1 female female BSc`

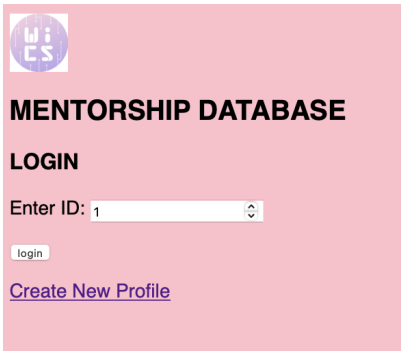
DELETE

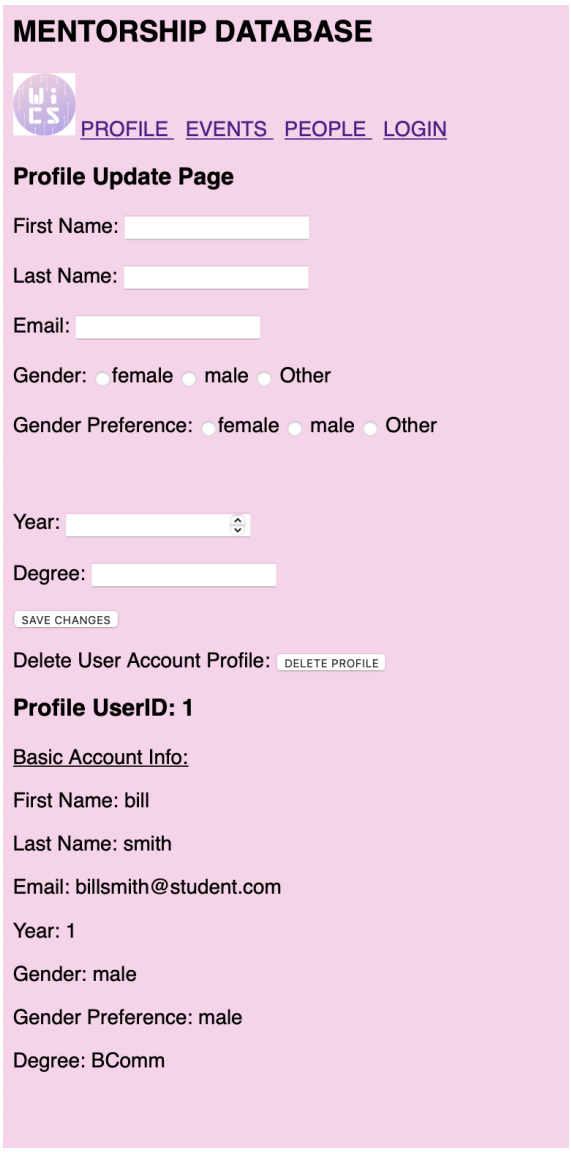
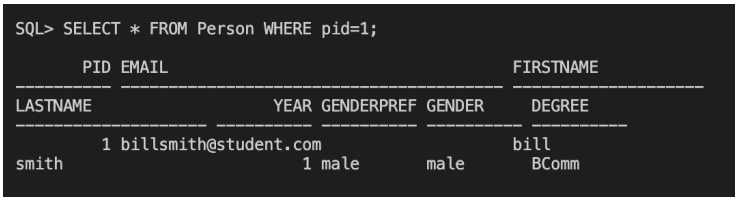
Query can be found in 'profile-update.php' at comment '// Query: DELETE' in handleDeleteRequest()

```
DELETE FROM Person WHERE pid = $pid;
```

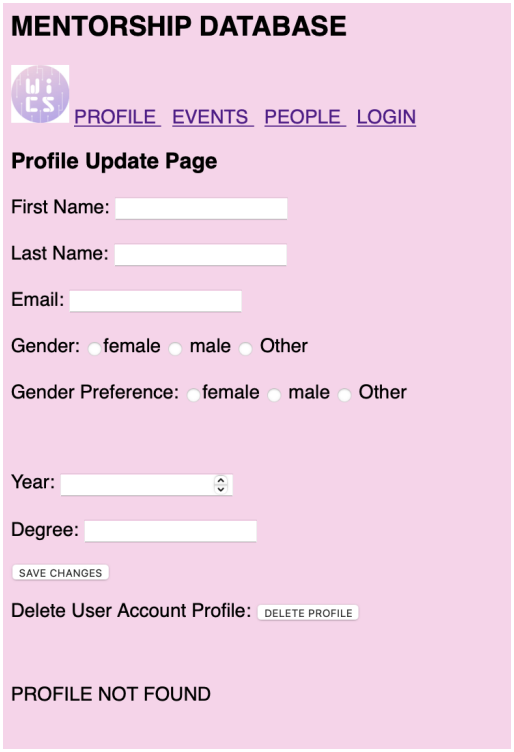
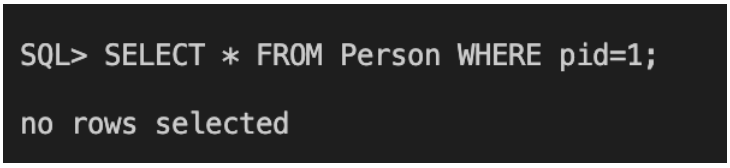
```php
function handleDeleteRequest() {
    global $db_conn;

    session_start();
    $pid = $_SESSION['pid'];

    // Query: DELETE
    executePlainSQL("DELETE FROM Person WHERE pid = '" . $pid . "'");

    OCICommit($db_conn);
}
```

Before:

On login.php page, you can login with ID being 1. Press the login button after entering ID.	

It will take you to profile-update.php page and it will display person information at the bottom.	
Can see the current tuple for pid=1 in Person	

After:

Press the 'DELETE PROFILE' button to run the delete query on the profile-update.php page. After pressing the button, it will delete the person and show 'PROFILE NOT FOUND'.	
Person tuple with pid=1 is now deleted.	

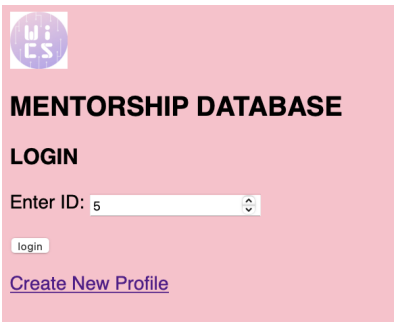
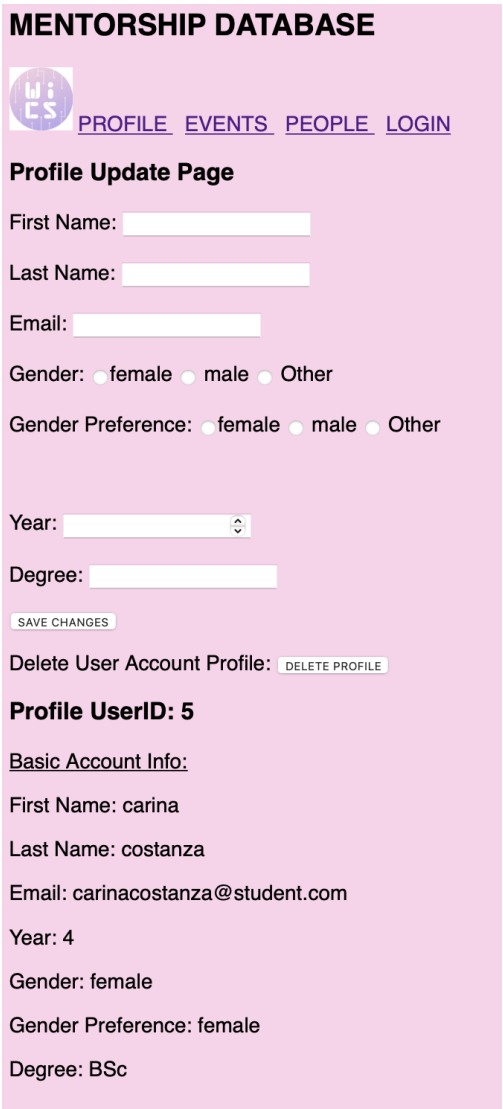
UPDATE

Query can be found in 'profile-update.php' at comment '// Query: UPDATE' in handleUpdateRequest()

```
UPDATE Person SET
      firstName = $firstName,
      lastName = $lastName,
      email = $email,
      gender = $gender,
      genderPreference = $genderPref,
      year = $year,
      degree = $degree,
      WHERE pid = $lastName;
```

```php
function handleUpdateRequest() {
    global $db_conn;

    //Getting the values from user and insert data into the table
    $firstName = $_POST['firstName'];
    $lastName = $_POST['lastName'];
    $email = $_POST['email'];
    $gender = $_POST['gender'];
    $genderPref = $_POST['genderPref'];
    $year = $_POST['year'];
    $degree = $_POST['degree'];

    session_start();
    $pid = $_SESSION['pid'];

    // Query: UPDATE
    executePlainSQL("UPDATE Person SET firstName='" . $firstName . "',
        lastName='" . $lastName . "',
        email='" . $email . "',
        gender='" . $gender . "',
        genderPreference='" . $genderPref . "',
        year='" . $year . "',
        degree='" . $degree . "'
        WHERE pid= '" . $pid ."'");

    OCICommit($db_conn);
}
```

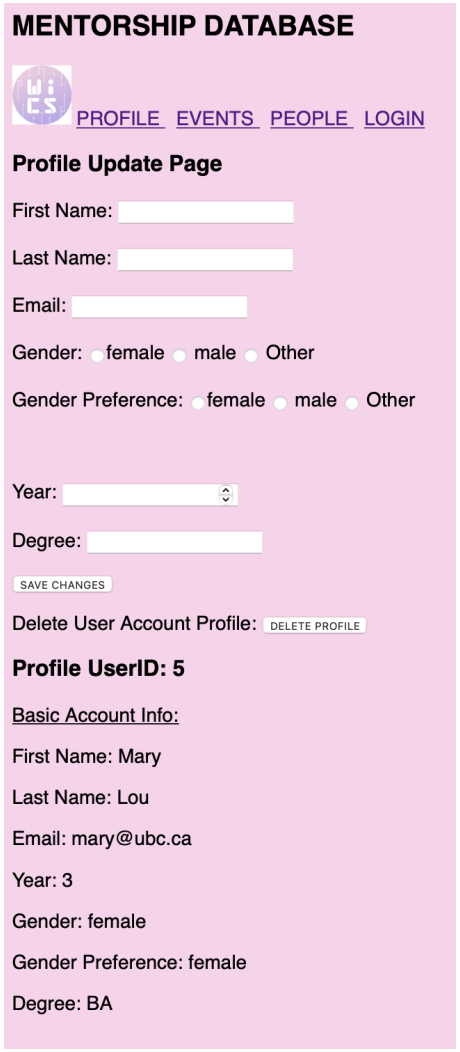
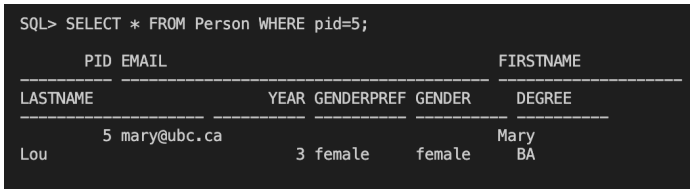
Before:

On login.php page, you can login with ID being 5. Press the login button after entering ID.	
It will take you to profile-update.php page and it will display current person information at the bottom.	

Can see current Person tuple for person with pid=5.	SQL> SELECT * FROM Person WHERE pid=5; PID EMAIL FIRSTNAME LASTNAME YEAR GENDERPREF GENDER DEGREE 5 carinacostanza@student.com carina costanza 4 female female BSc

After:

Can update information for person pid=5 by entering the new information into the fields on the profile-update.php page. To run the update query, press the 'SAVE CHANGES' button.	**MENTORSHIP DATABASE** PROFILE EVENTS PEOPLE LOGIN **Profile Update Page** First Name: Mary Last Name: Lou Email: mary@ubc.ca Gender: female male Other Gender Preference: female male Other Year: 3 Degree: BA SAVE CHANGES Delete User Account Profile: DELETE PROFILE **Profile UserID: 5** Basic Account Info: First Name: carina Last Name: costanza Email: carinacostanza@student.com Year: 4 Gender: female Gender Preference: female Degree: BSc

After pressing the 'SAVE CHANGES' button, the query will be run and the updated profile information will be displayed at the bottom of the page.	
Can see the updated tuple for person with pid=5.	

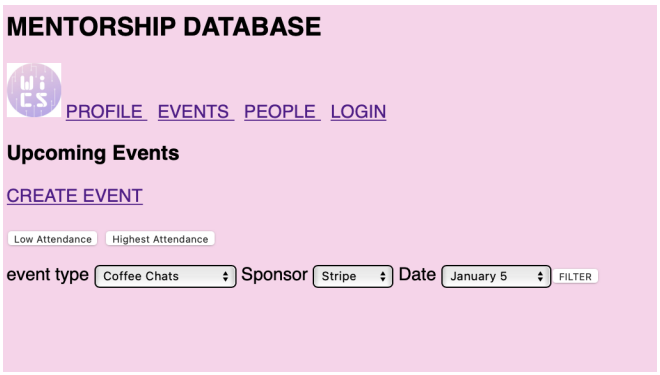
SELECTION

Query can be found in 'event-list.php' at comment '// Query: SELECTION' in displayEventsList()

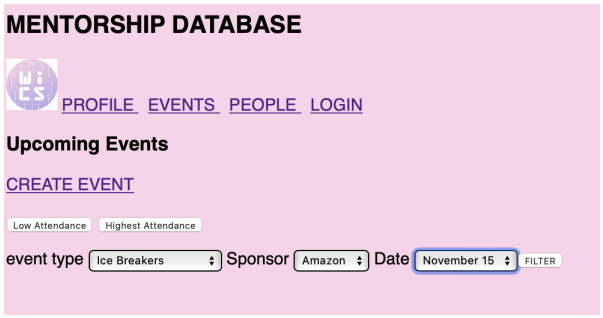
```
SELECT * FROM SponsoredEvent
      WHERE eventName = $eventType
      AND eventDate = $eventDate
      AND sponsorName = $sponsorName;
```

```
function displayEventsList($eventType, $eventDate, $sponsorName) {
    // Query: SELECTION
    $result = executePlainSQL("SELECT * FROM SponsoredEvent
                               WHERE eventName = '" . $eventType . "' AND
                                  eventDate='" . $eventDate . "'
                                  AND sponsorName='" . $sponsorName . "'");
```

Before:

On event-list.php page, there are no events displayed when there are no fields of events selected.	

After:

Can update information for event type = 'Ice Breakers', sponsorName = 'Amazon', and eventDate = 'November 15' into the fields on the event-list.php page. To run the update query, press the 'FILTER' button.	

After pressing the 'FILTER' button, the query will be run and the event that matches the fields will display the event details and attendee names that are attending that event on the page.	
Can see the correct event with the fields are displayed on the page.	

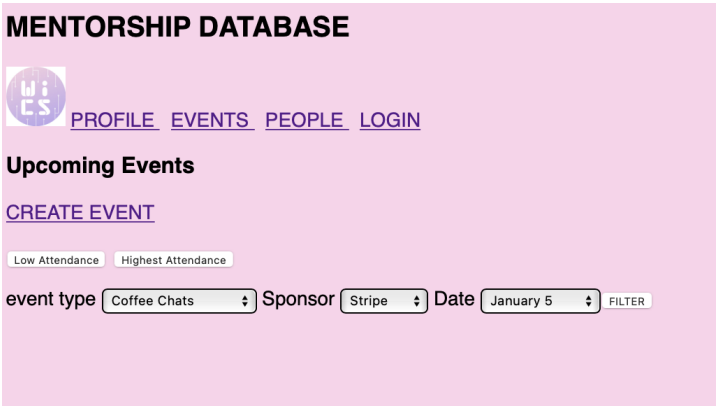
PROJECTION

Query can be found in 'event-list.php' at comment '// Query: PROJECTION' in displayOptions()

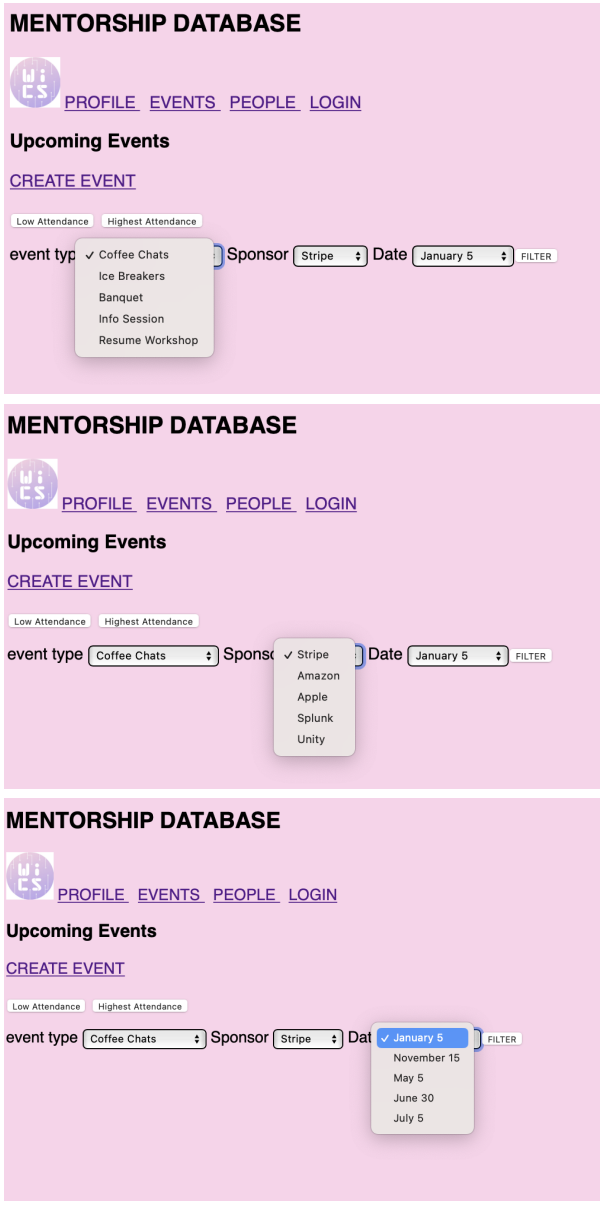
```
SELECT eventName,
eventDate, SponsorName
FROM SponsoredEvent;
```

```
function displayOptions() {
    // Query: PROJECTION
    //EVENT NAME
    $result = executePlainSQL("SELECT eventName, eventDate, sponsorName FROM SponsoredEvent");
```

Before:

On event-list.php page, there is only one value displayed for each field for the dropdowns.	

After:

Can click and will display a full list of all available event types, sponsorName, and eventDates in the event-list.php.	

Can see the correct list of all eventNames, sponsorNames, and eventDates on the page.	

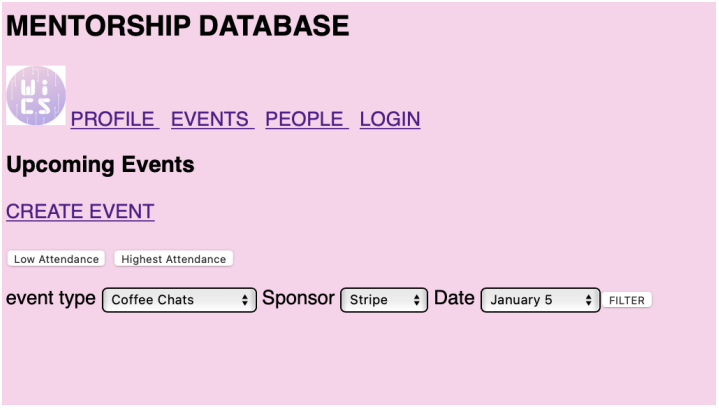
JOIN

Query can be found in 'event-list.php' at comment '// Query: JOIN' in displayEventsList()

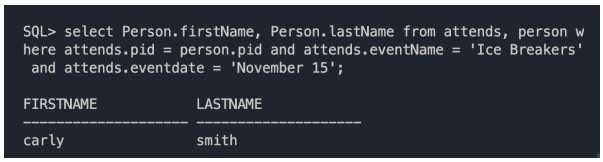
```
function displayEventsList($eventType, $eventDate, $sponsorName) {
    // Query: SELECTION
    $result = executePlainSQL("SELECT * FROM SponsoredEvent
                               WHERE eventName = '" . $eventType . "' AND
                                   eventDate='" . $eventDate . "'
                                   AND sponsorName='" . $sponsorName . "'");

    // Query: JOIN
    $numResult = executePlainSQL("SELECT Person.firstName, Person.lastName
                                  FROM Attends, Person
                                  WHERE Attends.pid = Person.pid
                                  AND Attends.eventName = '" . $eventType . "'
                                  AND Attends.eventDate = '" . $eventDate . "'");
```

```
SELECT Person.firstName,
Person.lastName
FROM Attends,Person,
WHERE Attends.pid =
Person.pid
AND Attends.eventName =
$eventType
AND Attends.eventDate = $eventDate;
```

Before:

On event-list.php page, there are no events displayed when the user that is logged in where no fields are selected.	

After:

After pressing the 'FILTER' button, the query will be run and the event that matches the fields will display the event details and attendee names that are attending that event on the page.	
Can see the correct list of all attendees that are attending the matched eventName, sponsor, and eventDate on the page.	

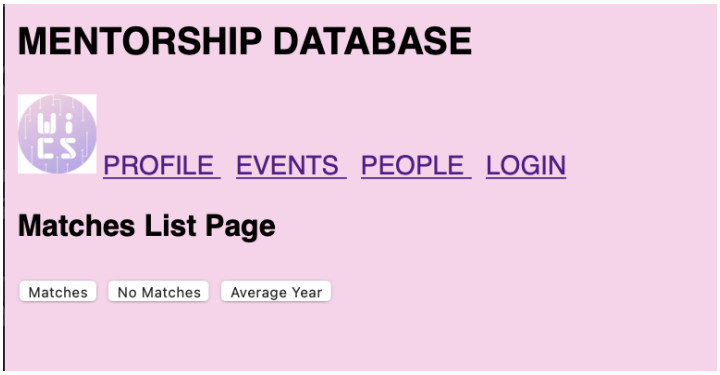
AGGREGATION WITH GROUP BY

Query can be found in 'matches.php' at comment '// Query: AGGREGATION WITH GROUP BY' in showAverageAge()

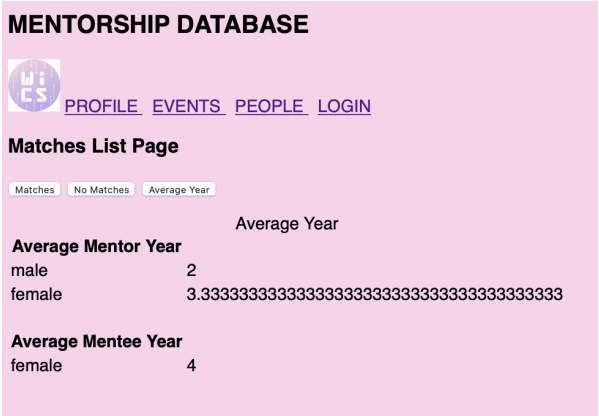
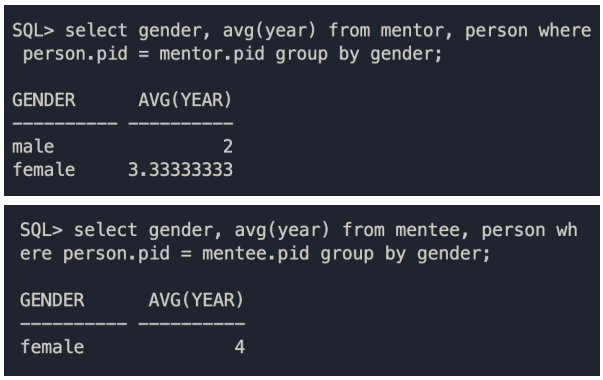
```
SELECT gender, AVG(year)
FROM Mentor,Person
WHERE Person.pid = Mentor.pid
GROUP BY gender;

SELECT gender, AVG(year)
FROM Mentee,Person
WHERE Person.pid = Mentee.pid
GROUP BY gender;
```

```
function showAverageAge() {
    // Query: AGGREGATION WITH GROUP BY
    $mentorAge = executePlainSQL("SELECT gender, AVG(year)
                FROM person, mentor
                WHERE person.pid = mentor.pid
                GROUP BY gender");
    $menteeAge = executePlainSQL("SELECT gender, AVG(year)
                FROM person, mentee
                WHERE person.pid = mentee.pid
                GROUP BY gender");
```

Before:

On matches.php page, there is no data displaying the average age of mentors and mentees.	

After:

After pressing the 'Average Year' button, the query will be run and the average year of the mentors and mentees of each gender are displayed on the page.	
Can see the correct average age of all the female and male mentor and mentees on the page.	

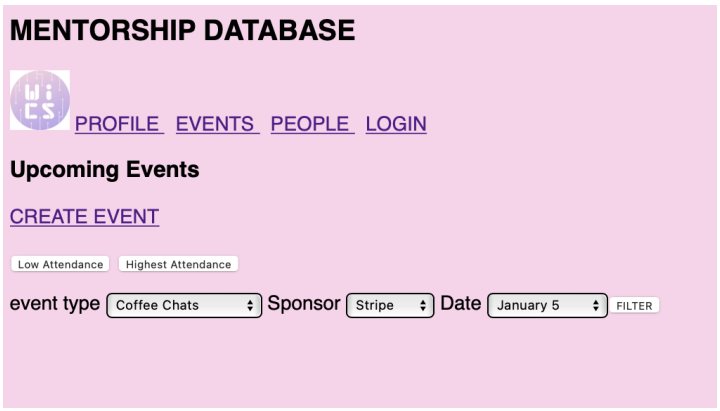
AGGREGATION WITH HAVING

Query can be found in `event-list.php' at comment '// Query: AGGREGATION WITH HAVING' in displayLowAttendance()

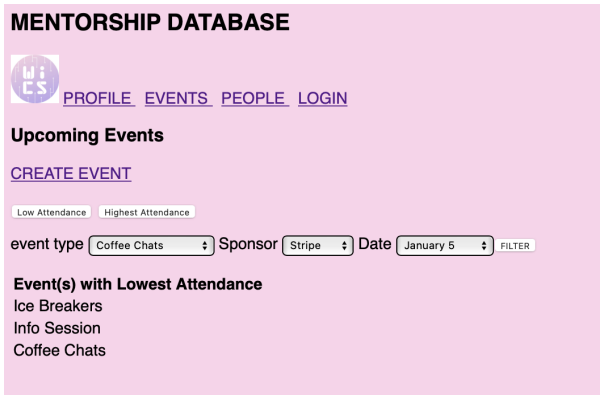
```
SELECT eventName,
COUNT(eventName)
FROM Attends
GROUP BY eventName
HAVING COUNT(*) < 2;
```

```
function displayLowAttendance() {
    // Query: AGGREGATION WITH HAVING
    $result = executePlainSQL("SELECT eventName, COUNT(eventName)
                               FROM Attends
                               GROUP BY eventName
                               HAVING COUNT(*) < 2");
```

Before:

On events-list.php page, there is no data displaying the event(s) with the lowest attendance count on the page.	

After:

After pressing the 'Low Attendance' button, the query will be run and the eventNames of the events with attendance count less than 2 are displayed on the page.	

Can see the correct eventNames with count of less than 2 are displayed on the page.	SQL> select eventname, count(eventname) from attends group by eventname having count(*) < 2; EVENTNAME COUNT(EVENTNAME) ---------- ---------- Ice Breakers 1 Info Session 1 Coffee Chats 1

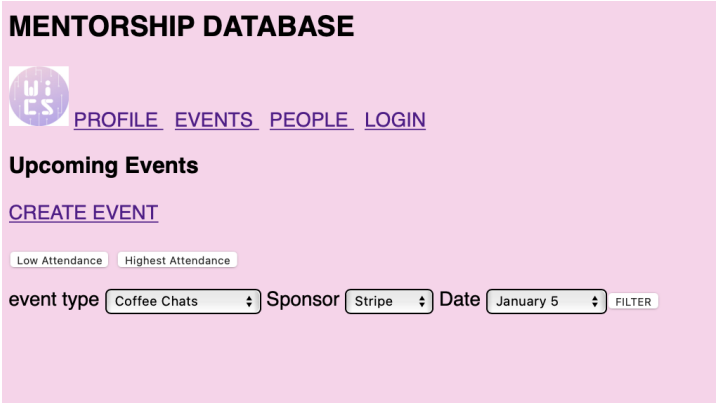
NESTED AGGREGATION WITH GROUP BY

Query can be found in `event-list.php' at comment '// Query: NESTED AGGREGATION WITH GROUP BY' in displayGreatestAttendance()

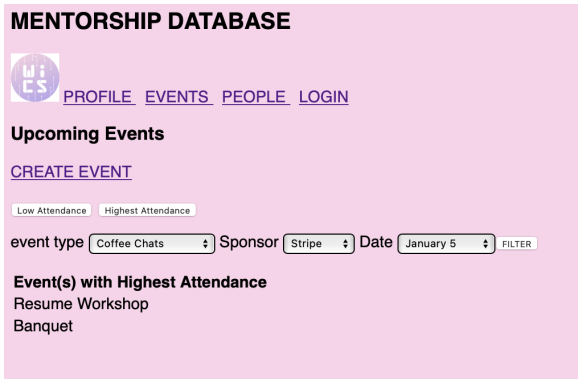
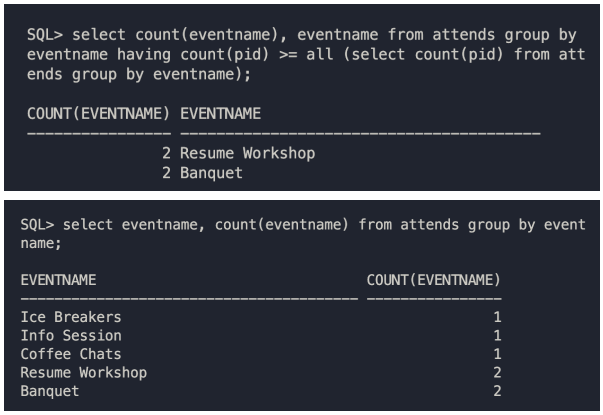
```
SELECT eventName
FROM Attends
GROUP BY eventName
HAVING COUNT(pid) >=
all(SELECT count(pid)
      FROM Attends
      GROUP BY eventName);
```

```
function displayGreatestAttendance() {
    // Query: NESTED AGGREGATION WITH GROUP BY
    $result = executePlainSQL("SELECT eventName
                               FROM attends
                               GROUP BY eventName
                               HAVING count(pid) >=
                                   all(SELECT count(pid)
                                       FROM attends
                                       GROUP BY eventName)");
```

Before:

On events-list.php page, there is no data displaying the event(s) with the highest attendance count on the page.	

After:

After pressing the 'Highest Attendance' button, the query will be run and the eventNames of the events with the greatest attendance are displayed on the page.	
Can see the correct eventNames with the highest count out of all the events are displayed on the page.	

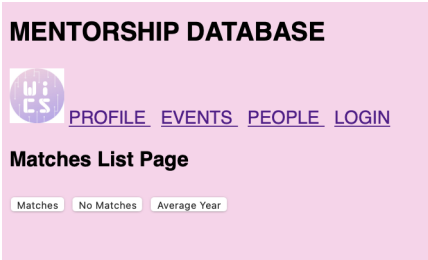
DIVISION

Query can be found in `matches.php' at comment '// Query: DIVISION' in displayNoMatchList()

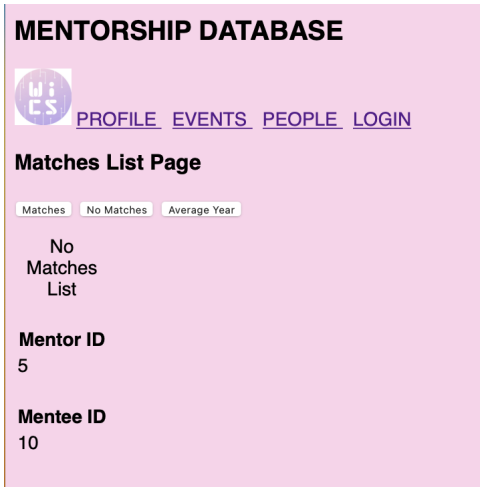
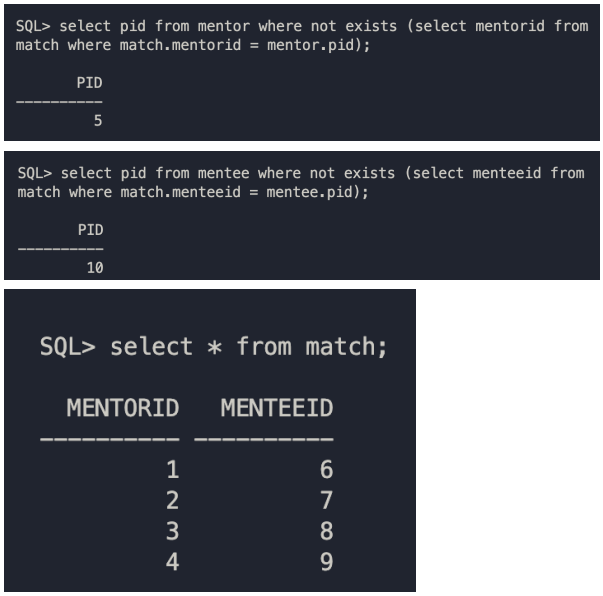
```
SELECT pid
FROM Mentor m
WHERE NOT EXISTS
(SELECT mentorID
 FROM Match ma
 WHERE ma.mentorID = m.pid);

SELECT pid
FROM Mentee m
WHERE NOT EXISTS
(SELECT menteeID
 FROM Match ma
 WHERE ma.menteeID = m.pid);
```

```
function displayNoMatchList() {
    // Query: DIVISION
    $resultMentor = executePlainSQL("SELECT pid
                                     FROM Mentor m
                                     WHERE NOT EXISTS
                                         (SELECT mentorID
                                         FROM Match ma
                                         WHERE ma.mentorID = m.pid)");
    $resultMentee = executePlainSQL("SELECT pid
                                     FROM Mentee m
                                      WHERE NOT EXISTS
                                          (SELECT menteeID
                                          FROM Match ma
                                          WHERE ma.menteeID = m.pid)");
```

Before:

On matches.php page, there is no data displaying the mentees or mentors without a match.	

After:

After pressing the 'No Matches' button, the query will be run and the ids of the mentees and mentors that do not belong to a match are displayed on the page.	
Can see the correct pid of the mentor and mentee that do not belong to a match are displayed on the page.	

Appendix:

Copy of Schema and Data Population

Person

Schema

```
SQL> describe Person
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PID                                       NOT NULL NUMBER(38)
 EMAIL                                              CHAR(40)
 FIRSTNAME                                          CHAR(20)
 LASTNAME                                           CHAR(20)
 YEAR                                               NUMBER(38)
 GENDERPREFERENCE                                   CHAR(10)
 GENDER                                             CHAR(10)
 DEGREE                                             CHAR(10)
```

Data

```
SQL> SELECT * FROM Person;

       PID EMAIL                                     FIRSTNAME
---------- ----------------------------------------- --------------------
LASTNAME                  YEAR GENDERPREF GENDER     DEGREE
-------------------- ---------- ---------- ---------- ----------
         1 billsmith@student.com                     bill
smith                        1 male       male       BComm

         2 carlysmith@student.com                    carly
smith                        2 female     male       BA

         3 sandrabullock@student.com                 sandra
bullock                      3 female     female     BSc


       PID EMAIL                                     FIRSTNAME
---------- ----------------------------------------- --------------------
LASTNAME                  YEAR GENDERPREF GENDER     DEGREE
-------------------- ---------- ---------- ---------- ----------
         4 sarahpaulson@student.com                  sarah
paulson                      3 female     female     BSc

         5 carinacostanza@student.com                carina
costanza                     4 female     female     BSc

         6 temp@student.com                          stella
costanza                     4 female     female     BSc


       PID EMAIL                                     FIRSTNAME
---------- ----------------------------------------- --------------------
LASTNAME                  YEAR GENDERPREF GENDER     DEGREE
-------------------- ---------- ---------- ---------- ----------
         7 anon@student.com                          bella
costanza                     4 female     female     BSc

         8 athenacostanza@student.com                athena
costanza                     4 female     female     BSc

         9 violetcostanza@student.com                violet
costanza                     4 female     female     BSc


       PID EMAIL                                     FIRSTNAME
---------- ----------------------------------------- --------------------
LASTNAME                  YEAR GENDERPREF GENDER     DEGREE
-------------------- ---------- ---------- ---------- ----------
        10 olacostanza@student.com                   violet
costanza                     4 female     female     BSc


10 rows selected.
```

Mentor	Schema
	`SQL> describe mentor`
	`Name Null? Type`
	`PID NOT NULL NUMBER(38)`
	`MAJOR CHAR(20)`
	Data
	`SQL> SELECT * FROM Mentor;`
	`PID MAJOR`
	`1 BUCS`
	`2 Cognitive Sciences`
	`3 Computer Science`
	`4 Computer Science`
	`5 Computer Science`

Mentee	Schema
	`SQL> describe mentee`
	`Name Null? Type`
	`PID NOT NULL NUMBER(38)`
	Data
	`SQL> SELECT * FROM Mentee;`
	`PID`
	`6`
	`7`
	`8`
	`9`
	`10`

WorkExperienceDuration	Schema
	`SQL> describe WorkExperienceDuration`
	`Name Null? Type`
	`PID NOT NULL NUMBER(38)`
	`COMPANY NOT NULL CHAR(40)`
	`DURATION NUMBER(38)`
	Data
	`SQL> SELECT * FROM WorkExperienceDuration;`
	`PID COMPANY DURATION`
	`1 SAP 8`
	`2 Stripe 10`
	`3 Teck 4`
	`4 Amazon 16`
	`5 Google 8`

WorkPay

Schema

```
SQL> describe WorkPay;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PID                                       NOT NULL NUMBER(38)
 COMPANY                                   NOT NULL CHAR(40)
 JOBTITLE                                  NOT NULL CHAR(40)
 SALARY                                             NUMBER(38)
```

Data

```
SQL> SELECT * FROM WorkPay;

       PID COMPANY
---------- ----------------------------------------
JOBTITLE                                   SALARY
---------------------------------------- ----------
         1 SAP
Business Analyst                           100000

         2 Stripe
Recruiter                                   98000

         3 Teck
Data Analyst                               270000


       PID COMPANY
---------- ----------------------------------------
JOBTITLE                                   SALARY
---------------------------------------- ----------
         4 Amazon
Project Manager                            345000

         5 Google
Software Engineer                          400000
```

Match

Schema

```
SQL> describe match;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 MENTORID                                  NOT NULL NUMBER(38)
 MENTEEID                                  NOT NULL NUMBER(38)
```

Data

```
SQL> SELECT * FROM Match;

  MENTORID   MENTEEID
---------- ----------
         1          6
         2          7
         3          8
         4          9
         5         10
```

Major

Schema

```
SQL> describe major
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 FACULTY                                   NOT NULL CHAR(40)
 MAJORNAME                                 NOT NULL CHAR(40)
```

Data

```
SQL> SELECT * FROM Major;

FACULTY
------------------------------------
MAJORNAME
------------------------------------
Arts
Computer Science

Business
Computer Science

Science
Computer Science


FACULTY
------------------------------------
MAJORNAME
------------------------------------
Science
Math

Science
Statistics
```

Country

Schema

```
SQL> describe country
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 COUNTRY                                            CHAR(30)
 CITY                                      NOT NULL CHAR(30)
 PROVINCE                                  NOT NULL CHAR(30)
```

Data

```
SQL> SELECT * FROM Country;

COUNTRY                        CITY
------------------------------ ------------------------------
PROVINCE
------------------------------
Canada                         British Columbia
Vancouver

Canada                         British Columbia
Victoria

Canada                         Alberta
Edmonton


COUNTRY                        CITY
------------------------------ ------------------------------
PROVINCE
------------------------------
USA                            Washington
Seattle

USA                            California
Los Angeles
```

PostalCode

Schema

```
SQL> describe postalcode
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 POSTALCODE                                NOT NULL CHAR(6)
 CITY                                               CHAR(30)
 PROVINCE                                           CHAR(30)
```

	Data
	`SQL> SELECT * FROM PostalCode;`
	
	`POSTAL CITY PROVINCE`
	`------ ------------------------------ ------------------------------`
	`V5H6U3 Vancouver British Columbia`
	`V4K8K9 Vancouver British Columbia`
	`V8H9I0 Vancouver British Columbia`
	`C95H63 Edmonton Alberta`
	`B7H8J9 Victoria British Columbia`

WorkPlace	Schema
	`SQL> describe workplace`
	`Name Null? Type`
	`----------------------------------------- -------- ----------------`
	`POSTALCODE NOT NULL CHAR(6)`
	`PLATFORM CHAR(30)`
	
	Data
	`SQL> SELECT * FROM WorkPlace`
	`2 ;`
	
	`POSTAL PLATFORM`
	`------ ------------------------------`
	`V5H6U3 online`
	`V4K8K9 in person`
	`V8H9I0 in person`
	`V5H8U3 in person`
	`C95H63 online`

Industry	Schema
	`SQL> describe industry;`
	`Name Null? Type`
	`----------------------------------------- -------- ----------------`
	`INDUSTRYNAME NOT NULL CHAR(40)`
	
	Data
	`SQL> SELECT * FROM Industry;`
	
	`INDUSTRYNAME`
	`----------------------------------------`
	`Data Science`
	`Engineering`
	`HR`
	`IT/Technology`
	`UX Design`

Room	Schema
	`SQL> describe room;`
	`Name Null? Type`
	`----------------------------------------- -------- ----------------`
	`ROOMNUMBER NOT NULL NUMBER(38)`
	`CAPACITY NUMBER(38)`
	`ADDRESSMAIN NOT NULL CHAR(40)`
	`FLOORNUMBER NUMBER(38)`
	
	Data

```
SQL> SELECT * FROM Room;

ROOMNUMBER   CAPACITY ADDRESSMAIN                                FLOORNUMBER
---------- ---------- ---------------------------------------- -----------
       236         30 1111 Main Mall                                     2
       304         50 1111 Main Mall                                     3
       144         25 75 Agronomy Road                                   1
       100         30 81 West Mall                                       1
       423         23 1111 Main Mall                                     4
```

Sponsor

Schema

```
SQL> describe sponsor
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
 SPONSORNAME                               NOT NULL CHAR(30)
 REPNAME                                            CHAR(30)
 REPEMAIL                                           CHAR(30)
```

Data

```
SQL> SELECT * FROM Sponsor;

SPONSORNAME                    REPNAME
------------------------------ ------------------------------
REPEMAIL
------------------------------
Stripe                         Mary Lamb
marylamb@stripe.com

Amazon                         Jeff Bezos
jeff@amazon.ca

Apple                          Violet James
violet@apple.com


SPONSORNAME                    REPNAME
------------------------------ ------------------------------
REPEMAIL
------------------------------
Splunk                         Boris Kovacevic
boris@splunk.ca

Unity                          Lola Evans
lola@unity.com
```

PotentialCareer

Schema

```
SQL> describe potentialcareer
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
 JOBTITLE                                  NOT NULL CHAR(20)
```

Data

```
SQL> SELECT * FROM PotentialCareer;

JOBTITLE
--------------------
Data Analyst
Data Engineer
Data Scientist
Software Engineer
Web Developer
```

InterestedIn	Schema SQL> describe interestedin Name Null? Type PID NOT NULL NUMBER(38) FACULTY NOT NULL CHAR(40) MAJORNAME NOT NULL CHAR(40) JOBTITLE CHAR(20) Data SQL> SELECT * FROM InterestedIn; PID FACULTY MAJORNAME JOBTITLE 10 Science Computer Science Web Developer 6 Science Statistics Data Scientist 7 Arts Computer Science Web Developer PID FACULTY MAJORNAME JOBTITLE 8 Science Math Data Engineer 9 Business Computer Science Data Analyst
WithinField	Schema SQL> describe withinfield; Name Null? Type INDUSTRYNAME NOT NULL CHAR(40) PID NOT NULL NUMBER(38) COMPANY NOT NULL CHAR(40) JOBTITLE NOT NULL CHAR(40) Data

```
SQL> SELECT * FROM WithinField;

INDUSTRYNAME                                        PID
---------------------------------------- ----------
COMPANY
----------------------------------------
JOBTITLE
----------------------------------------
Data Science                                          3
Teck
Data Analyst

Engineering                                           4
Amazon
Project Manager

INDUSTRYNAME                                        PID
---------------------------------------- ----------
COMPANY
----------------------------------------
JOBTITLE
----------------------------------------

Engineering                                           5
Google
Software Engineer

HR                                                    2
Stripe

INDUSTRYNAME                                        PID
---------------------------------------- ----------
COMPANY
----------------------------------------
JOBTITLE
----------------------------------------
Recruiter

IT/Technology                                         1
SAP
Business Analyst
```

SponsoredEvent

Schema

```
SQL> describe SponsoredEvent;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EVENTNAME                                 NOT NULL CHAR(40)
 EVENTDATE                                 NOT NULL CHAR(40)
 SPONSORNAME                                        CHAR(30)
 ADDRESSMAIN                                        CHAR(40)
 ROOMNUMBER                                         NUMBER(38)
```

Data

```
SQL> SELECT * FROM SponsoredEvent;

EVENTNAME
----------------------------------------
EVENTDATE                                SPONSORNAME
---------------------------------------- ------------------------------
ADDRESSMAIN                              ROOMNUMBER
---------------------------------------- ----------
Coffee Chats
January 5                                Stripe
1111 Main Mall                                  236

Ice Breakers
November 15                              Amazon
1111 Main Mall                                  304

EVENTNAME
----------------------------------------
EVENTDATE                                SPONSORNAME
---------------------------------------- ------------------------------
ADDRESSMAIN                              ROOMNUMBER
---------------------------------------- ----------

Banquet
May 5                                    Apple
75 Agronomy Road                                144

Info Session
June 30                                  Splunk

EVENTNAME
----------------------------------------
EVENTDATE                                SPONSORNAME
---------------------------------------- ------------------------------
ADDRESSMAIN                              ROOMNUMBER
---------------------------------------- ----------
81 West Mall                                    100

Resume Workshop
July 5                                   Unity
1111 Main Mall                                  423
```

Attends

Schema

```
SQL> describe attends
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
 PID                                       NOT NULL NUMBER(38)
 EVENTNAME                                 NOT NULL CHAR(40)
 EVENTDATE                                 NOT NULL CHAR(40)
```

Data

```
SQL> SELECT * FROM Attends;

       PID EVENTNAME
---------- ----------------------------------------
EVENTDATE
----------------------------------------
         1 Coffee Chats
January 5

         2 Ice Breakers
November 15

         3 Info Session
June 30


       PID EVENTNAME
---------- ----------------------------------------
EVENTDATE
----------------------------------------
         4 Resume Workshop
July 5

         5 Banquet
May 5
```

BasedIn

Schema

```
SQL> describe BasedIn
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
 POSTALCODE                                NOT NULL CHAR(6)
 PLATFORM                                  NOT NULL CHAR(10)
 PID                                       NOT NULL NUMBER(38)
 COMPANY                                   NOT NULL CHAR(40)
 JOBTITLE                                  NOT NULL CHAR(40)
```

Data

```
SQL> SELECT * FROM BasedIn;

POSTAL PLATFORM          PID COMPANY
------ ---------- ---------- ----------------------------------------
JOBTITLE
----------------------------------------
C95H63 online              5 Google
Software Engineer

V4K8K9 in person           2 Stripe
Recruiter

V5H6U3 in person           3 Teck
Data Analyst


POSTAL PLATFORM          PID COMPANY
------ ---------- ---------- ----------------------------------------
JOBTITLE
----------------------------------------
V5H8U3 online              1 SAP
Business Analyst

V8H9I0 in person           4 Amazon
Project Manager
```