

## Milestone 3

March 30, 2022

Project Group Number on Canvas: n/a; group named “group 0”

Name	Student ID	CS Alias	Preferred Email Address
Steven Li	95102885	i0z2b	listeven39@gmail.com
Ethan Lin	56009681	n4h3b	ethan1688.lin@gmail.com
Anton Chen	75858795	y5y2b	antonzychen@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

## 2 Application code

Application code can be found here (<https://github.com/chenanton/cpsc-304-project/tree/master/milestone3/code/>)

### 3 Create Tables and Populate Data

The following scripts are also available in the Git repository linked above, in the `code/sql/init.sql`.

```
1  -- Clear database before running.
2  -- Source: https://stackoverflow.com/questions/31890032/how-to-d
   ↪  delete-all-data-in-oracle-database-with-sql
3
4  BEGIN
5      FOR cur_rec IN (SELECT object_name, object_type
6                      FROM user_objects
7                      WHERE object_type IN
8                          ('TABLE',
9                           'VIEW',
10                          'MATERIALIZED VIEW',
11                          'PACKAGE',
12                          'PROCEDURE',
13                          'FUNCTION',
14                          'SEQUENCE',
15                          'SYNONYM',
16                          'PACKAGE BODY'
17                      ))
18  LOOP
19      BEGIN
20          IF cur_rec.object_type = 'TABLE'
21          THEN
22              EXECUTE IMMEDIATE 'DROP '
23                              || cur_rec.object_type
24                              || ' "'
25                              || cur_rec.object_name
26                              || '" CASCADE CONSTRAINTS';
27          ELSE
28              EXECUTE IMMEDIATE 'DROP '
29                              || cur_rec.object_type
30                              || ' "'
31                              || cur_rec.object_name
32                              || '"';
33          END IF;
34      EXCEPTION
35      WHEN OTHERS
36      THEN
37          DBMS_OUTPUT.put_line ('FAILED: DROP '
38                              || cur_rec.object_type
39                              || ' "'
40                              || cur_rec.object_name
```

```

41         || ' '
42     );
43     END;
44     END LOOP;
45     FOR cur_rec IN (SELECT *
46                     FROM all_synonyms
47                     WHERE table_owner IN (SELECT USER FROM dual))
48     LOOP
49         BEGIN
50             EXECUTE IMMEDIATE 'DROP PUBLIC SYNONYM ' ||
51                 ↪ cur_rec.synonym_name;
52         END;
53     END LOOP;
54 END;
55 /
56 -- Create table statements
57
58 CREATE TABLE Applicant (
59     ApplicantID      int,
60     FirstName        varchar(50),
61     LastName         varchar(50),
62     CONSTRAINT applicant_pk PRIMARY KEY (ApplicantID)
63 );
64
65 CREATE TABLE JobApplication (
66     JobApplicationID int,
67     CoverLetterLink  varchar(100),
68     ResumeLink       varchar(100),
69     Decision          varchar(20),
70     ApplicantID      int,
71     PostingID        int,
72     RecruiterID      int,
73     CompanyName      varchar(100) NOT NULL,
74     CONSTRAINT application_pk PRIMARY KEY (ApplicantID, PostingID)
75 );
76
77 CREATE TABLE Posting (
78     PostingID        int,
79     PostingType       varchar(20),
80     Salary           int,
81     StartDate        date,
82     JobDescription    varchar(1000),
83     PostingLocation   varchar(100),
84     CompanyName      varchar(100) NOT NULL,
85     CONSTRAINT posting_pk PRIMARY KEY (PostingID)

```

```

86 );
87
88 CREATE TABLE Company (
89     CompanyName      varchar(100),
90     StreetName        varchar(100),
91     City              varchar(100),
92     StateProvince     varchar(100),
93     Country           varchar(100),
94     PostalCode        varchar(50),
95     CONSTRAINT company_pk PRIMARY KEY (CompanyName)
96 );
97
98 CREATE TABLE Interviewer (
99     InterviewerID     int,
100    FirstName          varchar(50),
101    LastName           varchar(50),
102    Position           varchar(50),
103    CompanyName        varchar(100) NOT NULL,
104    CONSTRAINT interviewer_pk PRIMARY KEY (InterviewerID)
105 );
106
107 CREATE TABLE Host (
108     InterviewID       int,
109     InterviewerID     int DEFAULT NULL,
110     CONSTRAINT host_pk PRIMARY KEY (InterviewID, InterviewerID)
111 );
112
113 CREATE TABLE Interview (
114     InterviewID       int,
115     CONSTRAINT interview_pk PRIMARY KEY (InterviewID)
116 );
117
118 CREATE TABLE OnlineAssessment (
119     InterviewID       int,
120     PositionType      varchar(50),
121     Duration          int,
122     Difficulty        varchar(10),
123     NumberOfQuestions int,
124     CutoffScore       int,
125     StartDateTime     date,
126     EndDateTime       date,
127     ApplicantID       int NOT NULL,
128     PostingID         int NOT NULL,
129     CONSTRAINT online_assessment_pk PRIMARY KEY (InterviewID)
130 );
131

```

```

132 CREATE TABLE PhoneScreen (
133     InterviewID      int,
134     StartDateTime    date,
135     EndDateTime       date,
136     ApplicantID      int NOT NULL,
137     PostingID        int NOT NULL,
138     CONSTRAINT phone_screen_pk PRIMARY KEY (InterviewID)
139 );
140
141 CREATE TABLE OnsiteInterview (
142     InterviewID      int,
143     StartDateTime    date,
144     EndDateTime       date,
145     ApplicantID      int NOT NULL,
146     PostingID        int NOT NULL,
147     CONSTRAINT onsite_interview_pk PRIMARY KEY (InterviewID)
148 );
149
150 CREATE TABLE TeamMatching (
151     InterviewID      int,
152     StartDateTime    date,
153     EndDateTime       date,
154     ApplicantID      int NOT NULL,
155     PostingID        int NOT NULL,
156     CONSTRAINT team_matching_pk PRIMARY KEY (InterviewID)
157 );
158
159 CREATE TABLE Recruiter (
160     RecruiterID      int,
161     FirstName        varchar(50),
162     LastName         varchar(50),
163     CompanyName      varchar(100) NOT NULL,
164     CONSTRAINT recruiter_pk PRIMARY KEY (RecruiterID)
165 );
166
167 CREATE TABLE InformationSession (
168     SessionID        int,
169     SessionLocation   varchar(100),
170     SessionDate       date,
171     CompanyName      varchar(100) NOT NULL,
172     CONSTRAINT information_session_pk PRIMARY KEY (SessionID)
173 );
174
175 CREATE TABLE Participate (
176     ApplicantID      int,
177     PostingID        int,

```

```

178     SessionID          int,
179     CONSTRAINT particpate_pk PRIMARY KEY (ApplicantID, PostingID,
      ↪ SessionID)
180 );
181
182 -- Add foreign keys
183
184 ALTER TABLE JobApplication ADD (
185     CONSTRAINT application_fk_applicant
186     FOREIGN KEY (ApplicantID)
187     REFERENCES Applicant (ApplicantID)
188     ON DELETE CASCADE,
189     CONSTRAINT application_fk_posting
190     FOREIGN KEY (PostingID)
191     REFERENCES Posting (PostingID)
192     ON DELETE CASCADE,
193     CONSTRAINT application_fk_recruiter
194     FOREIGN KEY (RecruiterID)
195     REFERENCES Recruiter (RecruiterID)
196     ON DELETE CASCADE,
197     CONSTRAINT application_fk_company
198     FOREIGN KEY (CompanyName)
199     REFERENCES Company (CompanyName)
200     ON DELETE CASCADE
201 );
202
203 ALTER TABLE Posting ADD (
204     CONSTRAINT posting_fk_company
205     FOREIGN KEY (CompanyName)
206     REFERENCES Company (CompanyName)
207     ON DELETE CASCADE
208 );
209
210 ALTER TABLE Interviewer ADD (
211     CONSTRAINT interviewer_fk_company
212     FOREIGN KEY (CompanyName)
213     REFERENCES Company (CompanyName)
214     ON DELETE CASCADE
215 );
216
217 ALTER TABLE Host ADD (
218     CONSTRAINT host_fk_interview
219     FOREIGN KEY (InterviewID)
220     REFERENCES Interview (InterviewID)
221     ON DELETE CASCADE,
222     CONSTRAINT host_fk_interviewer

```

```

223     FOREIGN KEY (InterviewerID)
224     REFERENCES Interviewer (InterviewerID)
225     ON DELETE CASCADE
226 );
227
228 ALTER TABLE OnlineAssessment ADD (
229     CONSTRAINT online_assessment_fk_application
230     FOREIGN KEY (ApplicantID, PostingID)
231     REFERENCES JobApplication (ApplicantID, PostingID)
232     ON DELETE CASCADE,
233     CONSTRAINT online_assessment_fk_interview
234     FOREIGN KEY (InterviewID)
235     REFERENCES Interview (InterviewID)
236     ON DELETE CASCADE
237 );
238
239 ALTER TABLE PhoneScreen ADD (
240     CONSTRAINT phone_screen_fk_application
241     FOREIGN KEY (ApplicantID, PostingID)
242     REFERENCES JobApplication (ApplicantID, PostingID)
243     ON DELETE CASCADE,
244     CONSTRAINT phone_screen_fk_interview
245     FOREIGN KEY (InterviewID)
246     REFERENCES Interview (InterviewID)
247     ON DELETE CASCADE
248 );
249
250 ALTER TABLE OnsiteInterview ADD (
251     CONSTRAINT onsite_interview_fk_application
252     FOREIGN KEY (ApplicantID, PostingID)
253     REFERENCES JobApplication (ApplicantID, PostingID)
254     ON DELETE CASCADE,
255     CONSTRAINT onsite_interview_fk_interview
256     FOREIGN KEY (InterviewID)
257     REFERENCES Interview (InterviewID)
258     ON DELETE CASCADE
259 );
260
261 ALTER TABLE TeamMatching ADD (
262     CONSTRAINT team_matching_fk_application
263     FOREIGN KEY (ApplicantID, PostingID)
264     REFERENCES JobApplication (ApplicantID, PostingID)
265     ON DELETE CASCADE,
266     CONSTRAINT team_matching_fk_interview
267     FOREIGN KEY (InterviewID)
268     REFERENCES Interview (InterviewID)

```



```

269         ON DELETE CASCADE
270     );
271
272     ALTER TABLE Recruiter ADD (
273         CONSTRAINT recruiter_fk_company
274         FOREIGN KEY (CompanyName)
275         REFERENCES Company (CompanyName)
276         ON DELETE CASCADE
277     );
278
279     ALTER TABLE InformationSession ADD (
280         CONSTRAINT information_session_fk_company
281         FOREIGN KEY (CompanyName)
282         REFERENCES Company (CompanyName)
283         ON DELETE CASCADE
284     );
285
286     ALTER TABLE Participate ADD (
287         CONSTRAINT participates_fk_application
288         FOREIGN KEY (ApplicantID, PostingID)
289         REFERENCES JobApplication (ApplicantID, PostingID)
290         ON DELETE CASCADE,
291         CONSTRAINT participates_fk_information_session
292         FOREIGN KEY (SessionID)
293         REFERENCES InformationSession(SessionID)
294         ON DELETE CASCADE
295     );
296
297     -- Populate tables
298     INSERT ALL
299     INTO Applicant VALUES (1, 'Steven', 'Li')
300     INTO Applicant VALUES (2, 'Ethan', 'Lin')
301     INTO Applicant VALUES (3, 'Anton', 'Chen')
302     INTO Applicant VALUES (4, 'Raymond', 'Ng')
303     INTO Applicant VALUES (5, 'Kanye', 'West')
304     INTO Company VALUES ('Amazon', 'Vine', 'Vancouver', 'BC',
305         ⇨ 'Canada', 'V2A3A6')
306     INTO Company VALUES ('Google', 'Heather', 'Vancouver', 'BC',
307         ⇨ 'Canada', 'V6H7A6')
308     INTO Company VALUES ('Asana', 'Victoria', 'Vancouver', 'BC',
309         ⇨ 'Canada', 'V2H7U6')
310     INTO Company VALUES ('Citadel', 'Queen', 'Vancouver', 'BC',
311         ⇨ 'Canada', 'V1N4B6')
312     INTO Company VALUES ('Rippling', 'Oak', 'Vancouver', 'BC',
313         ⇨ 'Canada', 'V7A5A6')

```

```

309 INTO Posting VALUES (1, 'Internship', 10, TO_DATE('10/22/2022',
    ↪ 'MM/DD/YYYY'), 'janitor', 'company bathroom', 'Asana')
310 INTO Posting VALUES (2, 'Internship', 50, TO_DATE('5/1/2022',
    ↪ 'MM/DD/YYYY'), 'secretary', 'office', 'Google')
311 INTO Posting VALUES (3, 'FullTime', 100, TO_DATE('12/4/2022',
    ↪ 'MM/DD/YYYY'), 'sales', 'office', 'Amazon')
312 INTO Posting VALUES (4, 'FullTime', 500, TO_DATE('10/30/2022',
    ↪ 'MM/DD/YYYY'), 'worker', 'warehouse', 'Amazon')
313 INTO Posting VALUES (5, 'Internship', 30, TO_DATE('2/28/2023',
    ↪ 'MM/DD/YYYY'), 'sorting documents', 'office', 'Google')
314 INTO Recruiter VALUES (1, 'Kevin', 'Durant', 'Asana')
315 INTO Recruiter VALUES (2, 'Kyrie', 'Irving', 'Google')
316 INTO Recruiter VALUES (3, 'Jeff', 'Bezos', 'Amazon')
317 INTO Recruiter VALUES (4, 'Lil', 'Pump', 'Rippling')
318 INTO Recruiter VALUES (5, 'Kanye', 'West', 'Google')
319 INTO Interviewer VALUES (1, 'James', 'Harden', 'Senior
    ↪ Engineer', 'Asana')
320 INTO Interviewer VALUES (2, 'LeBron', 'James', 'Junior
    ↪ Engineer', 'Google')
321 INTO Interviewer VALUES (3, 'Michael', 'Jordan', 'Junior
    ↪ Engineer', 'Amazon')
322 INTO Interviewer VALUES (4, 'Stephen', 'Curry', 'Project
    ↪ Manager', 'Rippling')
323 INTO Interviewer VALUES (5, 'Chris', 'Paus', 'Principal
    ↪ Engineer', 'Google')
324 INTO JobApplication VALUES (1, 'stevenli.com/coverletter.pdf',
    ↪ 'stevenli.com/resume.pdf', 'Offer', 1, 1, 1, 'Asana')
325 INTO JobApplication VALUES (2, 'ethanlin.com/coverletter.pdf',
    ↪ 'ethanlin.com/resume.pdf', 'Accepted', 2, 2, 2, 'Google')
326 INTO JobApplication VALUES (3, 'antonchen.com/coverletter.pdf',
    ↪ 'antonchen.com/resume.pdf', 'Rejected', 3, 3, 3, 'Amazon')
327 INTO JobApplication VALUES (4, 'antonchen.com/coverletter.pdf',
    ↪ 'antonchen.com/resume.pdf', 'Rejected', 3, 2, 2, 'Google')
328 INTO JobApplication VALUES (5, 'antonchen.com/coverletter.pdf',
    ↪ 'antonchen.com/resume.pdf', 'Rejected', 3, 1, 1, 'Asana')
329 INTO InformationSession VALUES (1, 'zoom', TO_DATE('5/1/2022',
    ↪ 'MM/DD/YYYY'), 'Asana')
330 INTO InformationSession VALUES (2, 'zoom', TO_DATE('5/10/2022',
    ↪ 'MM/DD/YYYY'), 'Google')
331 INTO InformationSession VALUES (3, 'zoom', TO_DATE('10/10/2022',
    ↪ 'MM/DD/YYYY'), 'Amazon')
332 INTO InformationSession VALUES (4, 'zoom', TO_DATE('11/11/2022',
    ↪ 'MM/DD/YYYY'), 'Citadel')
333 INTO InformationSession VALUES (5, 'zoom', TO_DATE('12/12/2022',
    ↪ 'MM/DD/YYYY'), 'Rippling')
334 INTO Interview VALUES (1)

```

```

335 INTO Interview VALUES (2)
336 INTO Interview VALUES (3)
337 INTO Interview VALUES (4)
338 INTO Interview VALUES (5)
339 INTO Interview VALUES (6)
340 INTO Interview VALUES (7)
341 INTO Interview VALUES (8)
342 INTO Interview VALUES (9)
343 INTO Interview VALUES (10)
344 INTO Interview VALUES (11)
345 INTO Interview VALUES (12)
346 INTO Interview VALUES (13)
347 INTO Interview VALUES (14)
348 INTO Interview VALUES (15)
349 INTO Interview VALUES (16)
350 INTO Interview VALUES (17)
351 INTO Interview VALUES (18)
352 INTO Interview VALUES (19)
353 INTO Interview VALUES (20)
354 INTO OnlineAssessment VALUES
355 (1, 'Intern', 70, 'Easy', 4, 800, TO_DATE('5/2/2022',
    ↪ 'MM/DD/YYYY'), TO_DATE('5/9/2022', 'MM/DD/YYYY'), 1, 1)
356 INTO OnlineAssessment VALUES
357 (2, 'Intern', 80, 'Medium', 4, 800, TO_DATE('3/4/2022',
    ↪ 'MM/DD/YYYY'), TO_DATE('5/1/2022', 'MM/DD/YYYY'), 2, 2)
358 INTO OnlineAssessment VALUES
359 (3, 'Intern', 90, 'Hard', 4, 800, TO_DATE('5/6/2022',
    ↪ 'MM/DD/YYYY'), TO_DATE('5/19/2022', 'MM/DD/YYYY'), 3, 3)
360 INTO OnlineAssessment VALUES
361 (4, 'Intern', 70, 'Medium', 4, 800, TO_DATE('5/7/2022',
    ↪ 'MM/DD/YYYY'), TO_DATE('5/9/2022', 'MM/DD/YYYY'), 3, 2)
362 INTO OnlineAssessment VALUES
363 (5, 'Intern', 60, 'Easy', 4, 800, TO_DATE('5/8/2022',
    ↪ 'MM/DD/YYYY'), TO_DATE('6/9/2022', 'MM/DD/YYYY'), 3, 1)
364 INTO PhoneScreen VALUES
365 (6, TO_DATE('5/1/2022', 'MM/DD/YYYY'), TO_DATE('10/30/2022',
    ↪ 'MM/DD/YYYY'), 1, 1)
366 INTO PhoneScreen VALUES
367 (7, TO_DATE('6/2/2022', 'MM/DD/YYYY'), TO_DATE('11/20/2022',
    ↪ 'MM/DD/YYYY'), 2, 2)
368 INTO PhoneScreen VALUES
369 (8, TO_DATE('7/2/2022', 'MM/DD/YYYY'), TO_DATE('8/10/2022',
    ↪ 'MM/DD/YYYY'), 3, 3)
370 INTO PhoneScreen VALUES
371 (9, TO_DATE('8/2/2022', 'MM/DD/YYYY'), TO_DATE('8/10/2022',
    ↪ 'MM/DD/YYYY'), 3, 1)

```

```

372 INTO PhoneScreen VALUES
373 (10, TO_DATE('6/2/2022', 'MM/DD/YYYY'), TO_DATE('6/1/2022',
    ↪ 'MM/DD/YYYY'), 3, 2)
374 INTO OnsiteInterview VALUES
375 (11, TO_DATE('5/1/2022', 'MM/DD/YYYY'), TO_DATE('10/30/2022',
    ↪ 'MM/DD/YYYY'), 1, 1)
376 INTO OnsiteInterview VALUES
377 (12, TO_DATE('6/2/2022', 'MM/DD/YYYY'), TO_DATE('11/20/2022',
    ↪ 'MM/DD/YYYY'), 2, 2)
378 INTO OnsiteInterview VALUES
379 (13, TO_DATE('7/2/2022', 'MM/DD/YYYY'), TO_DATE('8/10/2022',
    ↪ 'MM/DD/YYYY'), 3, 3)
380 INTO OnsiteInterview VALUES
381 (14, TO_DATE('8/2/2022', 'MM/DD/YYYY'), TO_DATE('8/10/2022',
    ↪ 'MM/DD/YYYY'), 3, 1)
382 INTO OnsiteInterview VALUES
383 (15, TO_DATE('6/2/2022', 'MM/DD/YYYY'), TO_DATE('6/1/2022',
    ↪ 'MM/DD/YYYY'), 3, 2)
384 INTO TeamMatching VALUES
385 (16, TO_DATE('5/1/2022', 'MM/DD/YYYY'), TO_DATE('10/30/2022',
    ↪ 'MM/DD/YYYY'), 1, 1)
386 INTO TeamMatching VALUES
387 (17, TO_DATE('6/2/2022', 'MM/DD/YYYY'), TO_DATE('11/20/2022',
    ↪ 'MM/DD/YYYY'), 2, 2)
388 INTO TeamMatching VALUES
389 (18, TO_DATE('7/2/2022', 'MM/DD/YYYY'), TO_DATE('8/10/2022',
    ↪ 'MM/DD/YYYY'), 3, 3)
390 INTO TeamMatching VALUES
391 (19, TO_DATE('8/2/2022', 'MM/DD/YYYY'), TO_DATE('8/10/2022',
    ↪ 'MM/DD/YYYY'), 3, 1)
392 INTO TeamMatching VALUES
393 (20, TO_DATE('6/2/2022', 'MM/DD/YYYY'), TO_DATE('6/1/2022',
    ↪ 'MM/DD/YYYY'), 3, 2)
394 INTO Participate VALUES (1, 1, 1)
395 INTO Participate VALUES (2, 2, 2)
396 INTO Participate VALUES (3, 3, 3)
397 INTO Participate VALUES (3, 1, 1)
398 INTO Participate VALUES (3, 2, 2)
399 INTO Host VALUES (1, 1)
400 INTO Host VALUES (2, 2)
401 INTO Host VALUES (3, 3)
402 INTO Host VALUES (4, 4)
403 INTO Host VALUES (5, 5)
404 SELECT 1 FROM DUAL;

```

## 4 Project Description

### 4.1 What is this project? What has this project accomplished?

Our final project is a web application which models a functioning job board, containing all the basic components such as applicants applying to job postings, companies posting jobs, scheduling interviews for applications, and finally decisions from both applicants and companies regarding the application. A job posting can either be an internship or a full time job. Interviews can be one of online assessment, phone screen, onsite interview, or team matching. We also included an information session functionality so applicants can get to know about their jobs and companies more.

To start, the project successfully allows for companies to add their own postings to the website, as demonstrated by the insertion query screenshotted in section 6. Additionally, the project allows for companies to make decisions on accepting or rejecting candidates, with the update query. Finally, statistics regarding the project can also be looked up, e.g. finding the number of recruiters per company.

### 4.2 Discrepancies between final and turned in schema

Internship and full time changed to postings. Added interview relation in order to implement ISA relationship. Renamed a few relations, but functionally the same. We change the implementation of the ISA relationship for `Posting` (internship or full time). We implement it by adding an attribute called `PostingType` to `Posting` so it's easier and more efficient to get data from the `JobApplication` and `Posting` relationship. The other minor changes are name change from `Application` to `JobApplication`, and `InformationSessionHosting` to `InformationSession` for clarity purposes.

## 5 Queries

All queries are written in `code/sql/queries.sql`.

```
1  -- INSERT
2  -- "Insert a new job posting."
3  INSERT
4  INTO Posting
5  VALUES (
6      :posting_id,
7      :posting_type,
8      :salary,
9      :start_date,
10     :job_description,
11     :location,
12     :company_name
13 );
14
15 -- DELETE
16 -- "Remove a company from the database."
17 DELETE
18 FROM Company C
19 WHERE C.CompanyName = :company_name;
20
21 -- UPDATE
22 -- "Reject an applicant from all applied positions at a company."
23 UPDATE JobApplication J
24 SET J.Decision = 'Rejected'
25 WHERE J.CompanyName = :company_name
26     AND J.ApplicantID IN (
27     SELECT A.ApplicantID
28     FROM Applicant A
29     WHERE A.FirstName = :first_name
30           AND A.LastName = :last_name
31 );
32
33 -- SELECTION
34 -- "Find all the information sessions at specified location."
35 SELECT *
36 FROM InformationSession I
37 WHERE I.Location IS :location;
38
39 -- PROJECTION QUERY
40 -- "Find the names of all companies which have internships
41   ↪ available."
42 SELECT DISTINCT P.Posting
```

```

42 FROM Posting P;
43
44 -- JOIN QUERY
45 -- "Select all applicants who will join a position in a given
   ↳ location."
46 SELECT A.FirstName, A.LastName
47 FROM Applicant A, JobApplication J, Posting P
48 WHERE A.ApplicantID = J.ApplicantID
49       AND J.PostingID = P.PostingID
50       AND P.PostingLocation = :location
51       AND J.Decision = 'Accepted';
52
53 -- AGGREGATION
54 -- "Find the number of recruiters for each company."
55 SELECT R.CompanyName, COUNT(*) AS NumRecruiters
56 FROM Recruiter R
57 GROUP BY R.CompanyName
58 ORDER BY NumRecruiters DESC;
59
60 -- NESTED AGGREGATION
61 -- "Find the company who has given out the most offers."
62 WITH OfferCount AS (
63     SELECT J.CompanyName, COUNT(*) AS NumOffers
64     FROM JobApplication J, Posting P
65     WHERE J.PostingID = P.PostingID
66           AND (J.Decision = 'Offer' OR J.Decision = 'Accepted')
67     GROUP BY J.CompanyName
68 )
69 SELECT OC.CompanyName, OC.NumOffers
70 FROM OfferCount OC
71 WHERE OC.NumOffers = (
72     SELECT MAX(OC2.NumOffers)
73     FROM OfferCount OC2
74 );
75
76 -- DIVISION
77 -- "Find all the applicants who have applied to every company
   ↳ that has postings."
78 SELECT A.FirstName, A.LastName
79 FROM Applicant A
80 WHERE NOT EXISTS (
81     (
82         SELECT P.CompanyName
83         FROM Posting P
84     ) MINUS (
85         SELECT J.CompanyName

```

```
86      FROM JobApplication J
87      WHERE A.ApplicantID = J.ApplicantID
88  )
89  );
```



## 6 Query Screenshots

### 6.1 Insertion Query

The following input

#### Insertion

Insert a new posting

Posting ID:

Posting Type:

Salary:

Start Date (MM/DD/YYYY):

Job Description:

Posting Location :

Company Name:

yields the following output:

#### Result:

POSTING BEFORE INSERT:

Retrieved data from Posting table:

Posting ID	Posting Type	Salary	Start Date	Job Description	Posting Location	Company Name
1	Internship	10	22-OCT-22	janitor	company bathroom	Asana
2	Internship	50	01-MAY-22	secretary	office	Google
3	FullTime	100	04-DEC-22	sales	office	Amazon
4	FullTime	500	30-OCT-22	worker	warehouse	Amazon
5	Internship	30	28-FEB-23	sorting documents	office	Google

POSTING AFTER INSERT:

Retrieved data from Posting table:

Posting ID	Posting Type	Salary	Start Date	Job Description	Posting Location	Company Name
1	Internship	10	22-OCT-22	janitor	company bathroom	Asana
2	Internship	50	01-MAY-22	secretary	office	Google
3	FullTime	100	04-DEC-22	sales	office	Amazon
4	FullTime	500	30-OCT-22	worker	warehouse	Amazon
5	Internship	30	28-FEB-23	sorting documents	office	Google
101	Internship	10000	05-FEB-22	Software developer	Vancouver	Amazon

## 6.2 Deletion Query

The following input

### Deletion

Remove a company from the database

Company Name:

yields the following output:

### Result:

BEFORE DELETE REQUEST:

Retrieved data from Company table:

Company Name	Street Name	City	State/Province	Country	Postal Code
Amazon	Vine	Vancouver	BC	Canada	V2A3A6
Google	Heather	Vancouver	BC	Canada	V6H7A6
Asana	Victoria	Vancouver	BC	Canada	V2H7U6
Citadel	Queen	Vancouver	BC	Canada	V1N4B6
Rippling	Oak	Vancouver	BC	Canada	V7A5A6

AFTER DELETE REQUEST:

Retrieved data from Company table:

Company Name	Street Name	City	State/Province	Country	Postal Code
Amazon	Vine	Vancouver	BC	Canada	V2A3A6
Asana	Victoria	Vancouver	BC	Canada	V2H7U6
Citadel	Queen	Vancouver	BC	Canada	V1N4B6
Rippling	Oak	Vancouver	BC	Canada	V7A5A6

### 6.3 Update Query

The following input

#### Update

Reject an applicant from all applied positions at a company

First Name:

Last Name:

Company Name:

yields the following output:

## Result:

#### RESULT BEFORE UPDATE:

Retrieved data from JobApplication and Applicant tables:

First Name	Last Name	Company Name	Decision
Anton	Chen	Amazon	Rejected
Anton	Chen	Asana	Rejected
Steven	Li	Asana	Offer

#### RESULT AFTER UPDATE:

Retrieved data from JobApplication and Applicant tables:

First Name	Last Name	Company Name	Decision
Anton	Chen	Amazon	Rejected
Anton	Chen	Asana	Rejected
Steven	Li	Asana	Rejected

## 6.4 Selection Query

The following input

### Selection Query

Find all the information sessions at specified location

Location:

yields the following output:

## Result:

Retrieved data from InformationSession table:

Session ID	Location	Date	CompanyName
1	zoom	01-MAY-22	Asana
3	zoom	10-OCT-22	Amazon
4	zoom	11-NOV-22	Citadel
5	zoom	12-DEC-22	Rippling

## 6.5 Projection Query

The following input

### Projection Query

Find the names of all the companies which have positions available.

yields the following output:

# Result:

Retrieved data from Posting table:

**Company Name**

Amazon

Asana

## 6.6 Join Query

The following input

**Join Applicant, Application, Position Tables**

Select all applicants who will join a position in a given location.

Location:

yields the following output:

# Result:

Retrieved data from Join:

First Name	Last Name
------------	-----------

Ethan	Lin
-------	-----

## 6.7 Aggregation Query

The following input

### Aggregation with Count

Find the number of recruiters for each company

Submit Query

yields the following output:

## Result:

Retrieved aggregate data from Recruiters:

### Company Name Number of Recruiters

Google	2
Rippling	1
Asana	1
Amazon	1

## 6.8 Nested Aggregation Query

The following input

### **Nested Aggregation with Group By**

Find the company who has given out the most offers.

[Submit Query](#)

yields the following output:

## **Result:**

Retrieved nested aggregate data:

<b>Company Name</b>	<b>Number of Offers</b>
---------------------	-------------------------

Google	1
--------	---

Asana	1
-------	---



## 6.9 Division Query

The following input

### **Division**

Find all the applicants who have applied to every company that has open postings.

[Submit Query](#)

yields the following output:

# Result:

Retrieved division data:

First Name	Last Name
------------	-----------

Anton	Chen
-------	------