University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 4

Date: 29th November 2023

Group Number: 47

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Abhinav Kansal	24054660	r8u5o@ugrad.cs.ubc.ca	abhinavkansal08@gmail.com
Kashish Joshipura	27745629	t6p2x@ugrad.cs.ubc.ca	kashishjoshipura@gmail.com
Divyansh Sharma	49509540	f9h9g@ugrad.cs.ubc.ca	divyansh.abbott9712@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.

University of British Columbia, Vancouver
Department of Computer Science

---

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

The domain for our project will be career, employment, job seeking, and recruiting. Mainly we are going to focus on the job seeking side i.e., from the job seeker's perspective including the functions like applying for jobs, scheduling interviews, shortlisting jobs, and gathering information about the company i.e., where the company is located, and whether the mode of working is remote, hybrid, or in-person. Another side of the project is the ability of the recruiter to directly communicate with job seekers being directly linked with employers and pass any necessary information from employer to employee. The service is that of a mediator.

b. A description of how your final schema differed from the schema you turned in.

Our final schema is almost the same as the schema we used in the previous Milestones. In this schema, we used the same tables, tuples, and attributes as we used previously. The only change we made is that we are using ON DELETE CASCADE only and not ON UPDATE CASCADE as Oracle does not support ON UPDATE CASCADE.

List of Table definitions:

Primary Key is represented as underlined

*Foreign Key is represented as **bold***

myUser(UserID:integer, UserName:varchar(255), UserPassword:varchar(255), Education:varchar(255))

Recruiter(**UserID:integer**, RecruiterName:varchar(255), **CompanyID:integer**)

JobSeeker(**UserID:integer**, JobSeekerName:varchar(255))

Company(CompanyID:integer, CompanyName:varchar(255))

CompanyLocation(City:varchar(255), Province:varchar(255), PostalCode:varchar(255), Country:varchar(255))

Interviews(InterviewID:integer, InterviewMode:varchar(255), **UserID:integer**, **CompanyID:integer**)

JobListing(JobID:integer, Requirements:varchar(255), Position:varchar(255), **UserID:integer**, **CompanyID:integer**)

JobApplications(ApplicationID:integer, **JobID:integer**, CoverLetterUploaded:char(1), ResumeUploaded:char(1), ApplicationDate:Date, ApplicantName: varchar(255), ApplicationStatus: char(8))

SavedJobs(SavedJobNumber:integer, SavedDate:Date, **UserID:integer**)

Mode(ModeName:varchar(255), WorkingHours:integer)

Attains(**UserID:integer**, **ApplicationID:integer**, **JobID:integer**)

AssociatedWith(**InterviewID:integer**, **JobID:integer**)

FilterWith(**SavedJobNumber:integer**, **JobID:integer**)

Specifying the primary key (PK), candidate key, (CK) foreign keys (FK), and other constraints (e.g., not null, unique, etc.) that the table must maintain:

	PRIMARY KEY	**CANDIDATE KEY**	**FOREIGN KEY**
User	UserID	UserID	-
Recruiter	UserID	UserID	UserID, CompanyID NOT NULL
JobSeeker	UserID	UserID	UserID NOT NULL
Company	CompanyID	CompanyID	-
Location	City, Province, PostalCode	City, Province, PostalCode	-
Interviews	InterviewID	InterviewID	UserID, CompanyID NOT NULL
JobListing	JobID	JobID	UserID, CompanyID NOT NULL
JobApplications	ApplicationID, JobID	ApplicationID, JobID	JobID NOT NULL
SavedJobs	SavedJob#	SavedJob#	UserID NOT NULL
Mode	ModeName	ModeName	-
Attains	UserID, ApplicationID, JobID	UserID, ApplicationID, JobId	UserID, ApplicationID, JobID

AssociatedWith	InterviewID, JobID	InterviewID, JobID	InterviewID, JobID
FilterWith	SaveID, JobID	SaveID, JobID	Interview, JobID

Final Schema (job_search.sql)

```
DROP TABLE myUser CASCADE CONSTRAINTS;
DROP TABLE Company CASCADE CONSTRAINTS;
DROP TABLE Recruiter CASCADE CONSTRAINTS;
DROP TABLE JobSeeker CASCADE CONSTRAINTS;
DROP TABLE CompanyLocation CASCADE CONSTRAINTS;
DROP TABLE Interviews CASCADE CONSTRAINTS;
DROP TABLE JobListing CASCADE CONSTRAINTS;
DROP TABLE JobApplications CASCADE CONSTRAINTS;
DROP TABLE SavedJobs CASCADE CONSTRAINTS;
DROP TABLE JobMode CASCADE CONSTRAINTS;
DROP TABLE Attains CASCADE CONSTRAINTS;
DROP TABLE AssociatedWith CASCADE CONSTRAINTS;
DROP TABLE FilterWith CASCADE CONSTRAINTS;

CREATE TABLE myUser (
UserID INTEGER,
UserName      VARCHAR(255),
UserPassword VARCHAR(255),
Education      VARCHAR(255),
PRIMARY KEY(UserID)
);

CREATE TABLE Company (
CompanyID       INTEGER,
CompanyName VARCHAR(255),
PRIMARY KEY(CompanyID)
);

CREATE TABLE Recruiter (
RecruiterName       VARCHAR(255),
UserID          INTEGER,
CompanyID               INTEGER,
PRIMARY KEY(UserID),
FOREIGN KEY(UserID) REFERENCES myUser(UserID) ON DELETE CASCADE,
```

```
FOREIGN KEY(CompanyID) REFERENCES Company(CompanyID) ON DELETE CASCADE
);

CREATE TABLE JobSeeker (
JobSeekerName       VARCHAR(255),
UserID          INTEGER,
PRIMARY KEY (UserID),
FOREIGN KEY(UserID)
        REFERENCES myUser(UserID)
                ON DELETE CASCADE
);

CREATE TABLE CompanyLocation (
City            VARCHAR(255),
Province        VARCHAR(255),
PostalCode      VARCHAR(255),
Country         VARCHAR(255),
PRIMARY KEY (City, Province, PostalCode)
);

CREATE TABLE Interviews (
InterviewID             INTEGER,
InterviewModeVARCHAR(255),
UserID          INTEGER     NOT NULL,
CompanyID               INTEGER     NOT NULL,
PRIMARY KEY(InterviewID),
FOREIGN KEY(UserID) REFERENCES myUser(UserID) ON DELETE CASCADE,
FOREIGN KEY(CompanyID) REFERENCES Company(CompanyID) ON DELETE CASCADE
);

CREATE TABLE JobListing (
JobID       INTEGER,
Requirements VARCHAR(255),
Position                VARCHAR(255),
UserID          INTEGER     NOT NULL,
CompanyID               INTEGER     NOT NULL,
PRIMARY KEY(JobID),
FOREIGN KEY(UserID) REFERENCES myUser(UserID) ON DELETE CASCADE,
FOREIGN KEY(CompanyID) REFERENCES Company(CompanyID) ON DELETE CASCADE
);

CREATE TABLE JobApplications (
ApplicationID       INTEGER,
```

```
JobID                   INTEGER,
CoverLetterUploaded CHAR(1),
ResumeUploaded              CHAR(1),
ApplicationDate         DATE,
ApplicantName           VARCHAR(255),
ApplicationStatus       CHAR(8),
PRIMARY KEY (ApplicationID, JobID),
FOREIGN KEY(JobID) REFERENCES JobListing(JobID) ON DELETE CASCADE
);

CREATE TABLE SavedJobs(
SavedJobNumber              INTEGER,
SavedDate               DATE,
UserID          INTEGER     NOT NULL,
PRIMARY KEY(SavedJobNumber),
FOREIGN KEY(UserID) REFERENCES myUser(UserID) ON DELETE CASCADE
);

CREATE TABLE JobMode (
   ModeName VARCHAR(255),
   WorkingHours INTEGER,
   PRIMARY KEY(ModeName)
);

CREATE TABLE Attains (
   UserID INTEGER,
   ApplicationID INTEGER,
   JobID INTEGER,
   PRIMARY KEY (UserID, ApplicationID, JobID),
   FOREIGN KEY (UserID) REFERENCES myUser(UserID) ON DELETE CASCADE,
   FOREIGN KEY (ApplicationID, JobID) REFERENCES JobApplications(ApplicationID, JobID)
ON DELETE CASCADE
);

CREATE TABLE AssociatedWith (
InterviewID             INTEGER,
JobID               INTEGER,
PRIMARY KEY (InterviewID, JobID),
FOREIGN KEY(InterviewID) REFERENCES Interviews(InterviewID) ON DELETE CASCADE,
FOREIGN KEY(JobID) REFERENCES JobListing(JobID) ON DELETE CASCADE
);

CREATE TABLE FilterWith (
```

```
SavedJobNumber          INTEGER,
JobID                   INTEGER,
PRIMARY KEY (SavedJobNumber, JobID),
FOREIGN KEY(SavedJobNumber) REFERENCES SavedJobs(SavedJobNumber) ON DELETE
CASCADE,
FOREIGN KEY(JobID) REFERENCES JobListing(JobID) ON DELETE CASCADE
);


INSERT INTO myUser
VALUES(12345, 'Adam Troy', 'adam123', 'BSc. in Computer Science');
INSERT INTO myUser
VALUES(10201, 'Max Tan', 'maxxi', 'BSc. in Statistics');
INSERT INTO myUser
VALUES(20190, 'Kate Menon', 'km567', 'BSc. in Mathematics');
INSERT INTO myUser
VALUES(11111, 'Nathon Karl', 'kl1234', 'BSc. in Biology');
INSERT INTO myUser
VALUES(90123, 'John Miller', 'john312', 'BSc. in Computer Science');

INSERT INTO Company
VALUES(90410, 'Apple');
INSERT INTO Company
VALUES(32130, 'Tesla');
INSERT INTO Company
VALUES(80801, 'Microsoft');
INSERT INTO Company
VALUES(41029, 'IBM');
INSERT INTO Company
VALUES(50192, 'Facebook');

INSERT INTO Recruiter
VALUES('Edward Kim', 12345, 90410);
INSERT INTO Recruiter
VALUES('Dr. Saint Lauraine', 10201, 32130);
INSERT INTO Recruiter
VALUES('Jason Seo', 20190, 80801);
INSERT INTO Recruiter
VALUES('Katy Leo', 11111, 41029);
INSERT INTO Recruiter
VALUES('Pamela Ellis', 90123, 50192);

INSERT INTO JobSeeker
```

```
VALUES('Adam Troy', 12345);
INSERT INTO JobSeeker
VALUES('Max Tan', 10201);
INSERT INTO JobSeeker
VALUES('Kate Menon', 20190);
INSERT INTO JobSeeker
VALUES('Nathon Karl', 11111);
INSERT INTO JobSeeker
VALUES('John Miller', 90123);

INSERT INTO CompanyLocation
VALUES('Toronto', 'Ontario',  'M5J 0A8', 'Canada');
INSERT INTO CompanyLocation
VALUES('Richmond', 'British Columbia', 'V6X 1S1', 'Canada');
INSERT INTO CompanyLocation
VALUES('Vancouver', 'British Columbia', 'V6B 1C1', 'Canada');
INSERT INTO CompanyLocation
VALUES('Toronto', 'Ontario', 'M5J 0E7', 'Canada');
INSERT INTO CompanyLocation
VALUES('Toronto', 'Ontario', 'M4P 1E4', 'Canada');

INSERT INTO Interviews
VALUES(10203, ' Online', 12345, 90410);
INSERT INTO Interviews
VALUES(10902, ' Online', 10201, 32130);
INSERT INTO Interviews
VALUES(20304, ' In-Person', 20190, 80801);
INSERT INTO Interviews
VALUES(40506, ' In-Person', 11111, 41029);
INSERT INTO Interviews
VALUES(60060, ' Online', 90123, 50192);

INSERT INTO JobListing
VALUES(55555, 'Bsc. in Computer Science', 'Software Developer',
12345, 90410);
INSERT INTO JobListing
VALUES(54545, 'Bsc. in Mathematics', 'Mathematician', 10201,
 32130);
INSERT INTO JobListing
VALUES(67676, 'Bsc. in Computer Science', 'Front End Developer',
20190, 80801);
INSERT INTO JobListing
VALUES(78651, 'Bsc. in Statistics', 'Data Analyst', 11111, 41029);
```

```
INSERT INTO JobListing
VALUES(30456, 'Bsc. in Biology', 'Research Assistant', 90123, 50192);

INSERT INTO JobApplications
VALUES(10921, 55555, 'T', 'T', TO_DATE('2023-09-10', 'YYYY-MM-DD'), 'Adam Troy',
'Accepted');
INSERT INTO JobApplications
VALUES(20891, 54545, 'F', 'T', TO_DATE('2023-09-25', 'YYYY-MM-DD'), 'Max Tan', 'Rejected');
INSERT INTO JobApplications
VALUES(80801, 67676, 'T', 'T', TO_DATE('2023-10-01', 'YYYY-MM-DD'), 'Kate Menon',
'Accepted');
INSERT INTO JobApplications
VALUES(30121, 78651, 'F', 'F', TO_DATE('2023-10-08', 'YYYY-MM-DD'), 'Nathon Karl',
'Pending');
INSERT INTO JobApplications
VALUES(16080, 30456, 'T', 'T', TO_DATE('2023-10-10', 'YYYY-MM-DD'), 'John Miller',
'Accepted');

INSERT INTO SavedJobs
VALUES(1,  TO_DATE('2023-09-01', 'YYYY-MM-DD'), 12345);
INSERT INTO SavedJobs
VALUES(2, TO_DATE('2023-09-03', 'YYYY-MM-DD'), 10201);
INSERT INTO SavedJobs
VALUES(3, TO_DATE('2023-09-05', 'YYYY-MM-DD'), 20190);
INSERT INTO SavedJobs
VALUES(4, TO_DATE('2023-10-05', 'YYYY-MM-DD'), 11111);
INSERT INTO SavedJobs
VALUES(5, TO_DATE('2023-10-10', 'YYYY-MM-DD'), 90123);

INSERT INTO JobMode
VALUES('Hybrid', 35);
INSERT INTO JobMode
VALUES('In-Person', 35);
INSERT INTO JobMode
VALUES('Online', 40);

INSERT INTO Attains
VALUES(12345, 10921, 55555);
INSERT INTO Attains
VALUES(10201, 20891, 54545);
INSERT INTO Attains
VALUES(20190, 80801, 67676);
INSERT INTO Attains
```

```
VALUES(11111, 30121, 78651);
INSERT INTO Attains
VALUES(90123, 16080, 30456);

INSERT INTO AssociatedWith
VALUES(10203, 55555);
INSERT INTO AssociatedWith
VALUES(10902, 54545);
INSERT INTO AssociatedWith
VALUES(20304, 67676);
INSERT INTO AssociatedWith
VALUES(40506, 78651);
INSERT INTO AssociatedWith
VALUES(60060, 30456);

INSERT INTO FilterWith
VALUES(1, 55555);
INSERT INTO FilterWith
VALUES(2, 54545);
INSERT INTO FilterWith
VALUES(3, 67676);
INSERT INTO FilterWith
VALUES(4, 78651);
INSERT INTO FilterWith
VALUES(5, 30456);
```

SQL Queries Used

(Insert)
```
INSERT INTO JobListing VALUES(:bind1, :bind2, :bind3, :bind4, :bind5)
```

(DELETE)
```
DELETE FROM SAVEDJOBS WHERE SavedJobNumber ='" . $savedJobID . "'
```

(UPDATE)
```
UPDATE Interviews SET InterviewMode='" . $new_mode . "' WHERE InterviewID='" .
$interviewID . "'
UPDATE Interviews SET UserID='" . $new_userID . "' WHERE InterviewID='" . $interviewID . "'
UPDATE Interviews SET CompanyID='" . $new_CID . "' WHERE InterviewID='" . $interviewID .
"'
```

(Selection)
```
SELECT " . $select . " FROM " . $from . " WHERE " . $where
```

(Projection)
```
$select . " FROM JobListing
```

(Join)
```
SELECT * FROM FilterWith f, SavedJobs s WHERE s.SavedJobNumber = f.SavedJobNumber
```

(Aggregation with Group By)
```
SELECT COUNT(userID) AS Applicants,CompanyID FROM jobListing GROUP BY companyID
```

(Aggregation with Having)
```
SELECT COUNT(userID) AS Applicants,CompanyID FROM jobListing GROUP BY companyID
HAVING COUNT(userID) > 1
```

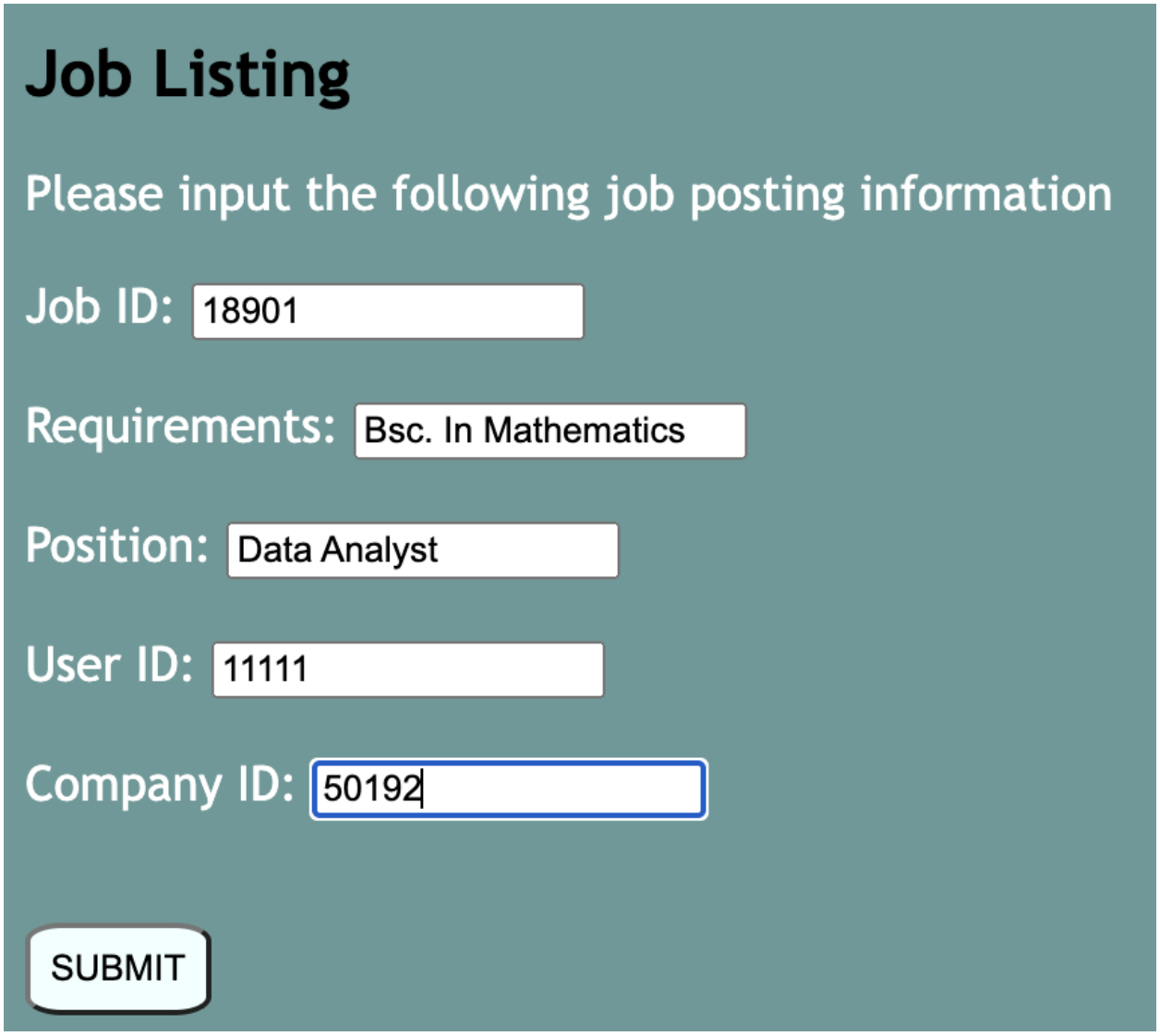
(Nested Aggregation with Group By)
```
SELECT AVG(ListedJobs) AS AverageListedJobs FROM (SELECT CompanyID, COUNT(*) AS
ListedJobs FROM JobListing GROUP BY CompanyID)
```

(Division)
```
SELECT UserName FROM myUser U
                        WHERE NOT EXISTS (
                           (SELECT C.CompanyID FROM Company C)
                           MINUS
                           (SELECT J.CompanyID FROM JobListing J WHERE J.UserID =
U.UserID))
```

University of British Columbia, Vancouver
Department of Computer Science

Screenshots

(Insert)



Before:

Retrieved data from table JobListing:

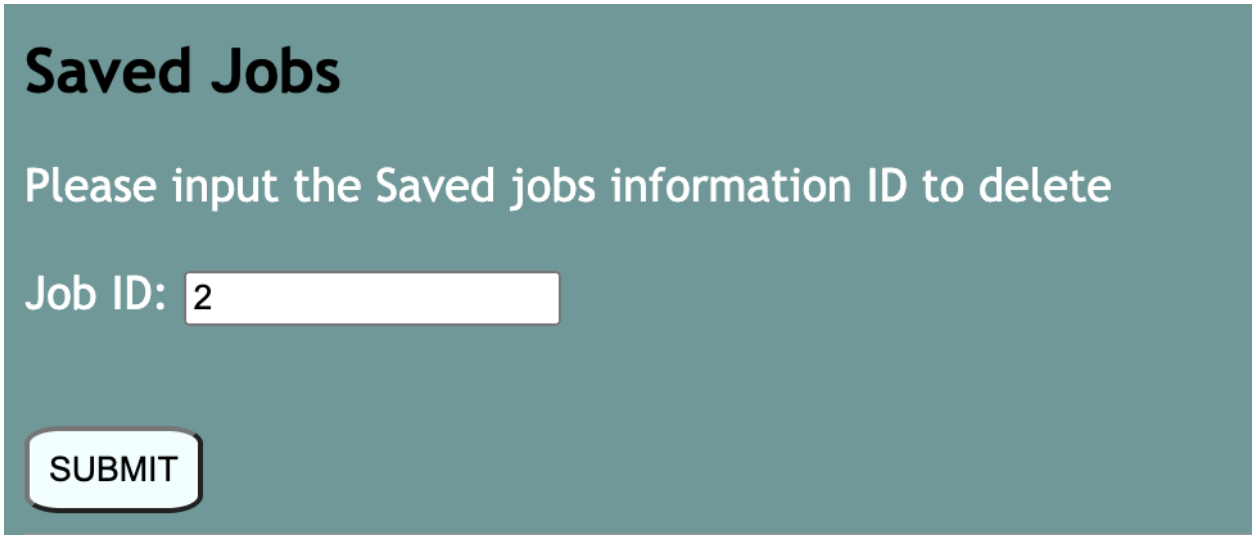
JobID	Req	Pos	Uid	Cid
55555	Bsc. in Computer Science	Software Developer	12345	90410
54545	Bsc. in Mathematics	Mathematician	10201	32130
67676	Bsc. in Computer Science	Front End Developer	20190	80801
78651	Bsc. in Statistics	Data Analyst	11111	41029
30456	Bsc. in Biology	Research Assistant	90123	50192

After:

Retrieved data from table JobListing:

JobID	Req	Pos	Uid	Cid
18901	Bsc. In Mathematics	Data Analyst	11111	50192
55555	Bsc. in Computer Science	Software Developer	12345	90410
54545	Bsc. in Mathematics	Mathematician	10201	32130
67676	Bsc. in Computer Science	Front End Developer	20190	80801
78651	Bsc. in Statistics	Data Analyst	11111	41029
30456	Bsc. in Biology	Research Assistant	90123	50192

(DELETE)



Before:

Retrieved data from table SavedJobs:

Saved Job #	Saved Date	User ID
1	01-SEP-23	12345
2	03-SEP-23	10201
3	05-SEP-23	20190
4	05-OCT-23	11111
5	10-OCT-23	90123

After:

Retrieved data from table SavedJobs:

Saved Job #	Saved Date	User ID
1	01-SEP-23	12345
3	05-SEP-23	20190
4	05-OCT-23	11111
5	10-OCT-23	90123

(UPDATE)


Interviews
Interview ID: 10902
New Interview Mode: In-Person
New User ID: 90123
New Company ID: 41029
SUBMIT

Before:

Retrieved data from table Interviews:

Interview ID	Interview Mode	User ID	Company ID
10203	Online	12345	90410
10902	Online	10201	32130
20304	In-Person	20190	80801
40506	In-Person	11111	41029
60060	Online	90123	50192

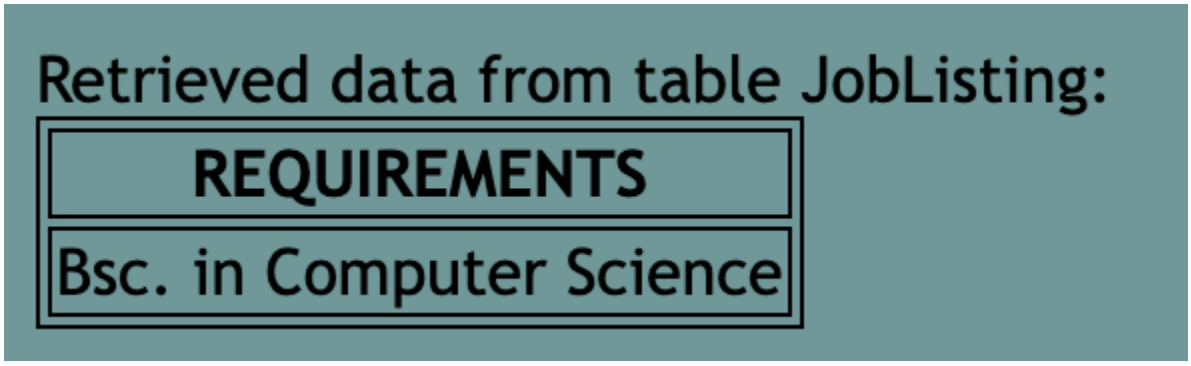
After:

Retrieved data from table Interviews:

Interview ID	Interview Mode	User ID	Company ID
10203	Online	12345	90410
10902	In-Person	90123	41029
20304	In-Person	20190	80801
40506	In-Person	11111	41029
60060	Online	90123	50192

(Selection)

Input:



Output:

University of British Columbia, Vancouver
Department of Computer Science

(Projection)
Input:

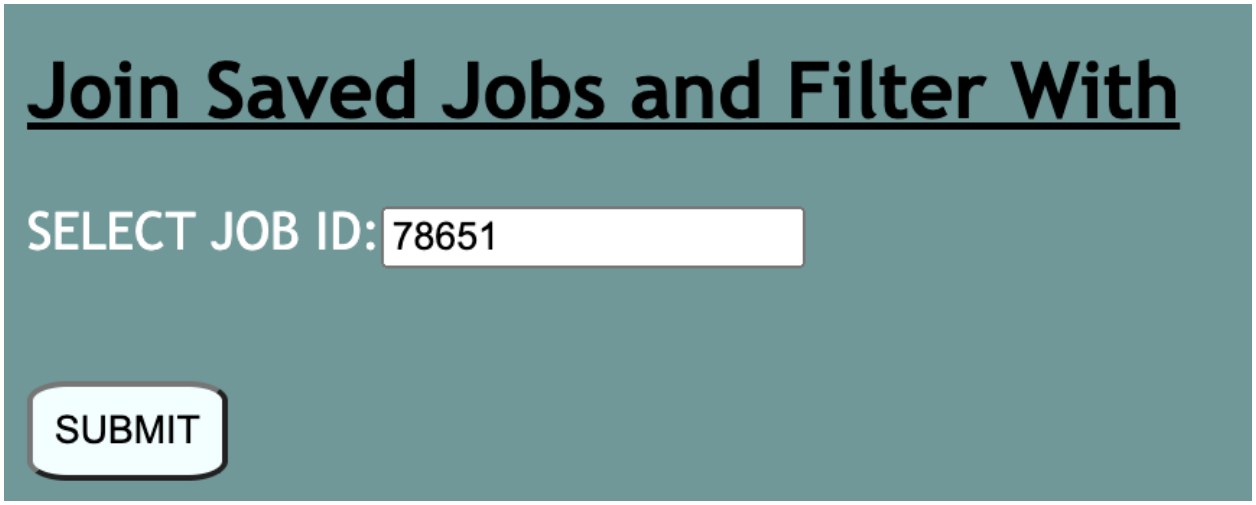
Output:

Retrieved data from table JobListing:

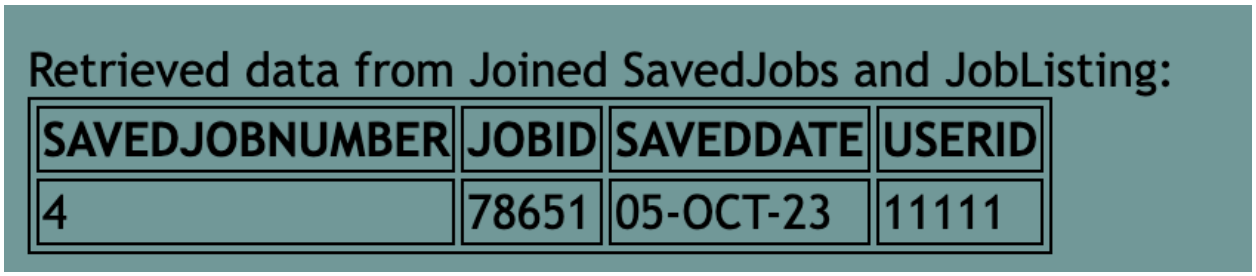
JOBID	REQUIREMENTS	POSITION	USERID	COMPANYID
18901	Bsc. In Mathematics	Data Analyst	11111	50192
55555	Bsc. in Computer Science	Software Developer	12345	90410
54545	Bsc. in Mathematics	Mathematician	10201	32130
67676	Bsc. in Computer Science	Front End Developer	20190	80801
78651	Bsc. in Statistics	Data Analyst	11111	41029
30456	Bsc. in Biology	Research Assistant	90123	50192

(Join)

Input:



Output:



University of British Columbia, Vancouver
Department of Computer Science

(Aggregation with Group By)

Find the number of applicants for each company (using GROUP BY)

Submit

Retrieved data from table JobListing:

Applicants	Cid
2	50192
1	80801
1	90410
3	32130
1	41029

(Aggregation with Having)

Find the number of applicants for each company having more than 1 applicants (using GROUP BY and HAVING)

Submit

Retrieved data from table JobListing:

Applicants	Cid
2	50192
3	32130

(Nested Aggregation with Group By)

Find the average number of jobs posted grouped by company

Submit

Retrieved data from myUser:

AVERAGELISTEDJOBS
1.8

(Division)

Find the users who have applied to a job in every company

Submit

Retrieved data from table JobListing:

JobID	Req	Pos	Uid	Cid
55557	BSc. in CS	Front-End Developer	12345	80801
55558	BSc. in CS	Data Analyst	12345	41029
55559	BSc. in CS	Research Assistant	12345	50192
55555	Bsc. in Computer Science	Software Developer	12345	90410
54545	Bsc. in Mathematics	Mathematician	10201	32130
67676	Bsc. in Computer Science	Front End Developer	20190	80801
78651	Bsc. in Statistics	Data Analyst	11111	41029
30456	Bsc. in Biology	Research Assistant	90123	50192
55556	Bsc. in Computer Science	Mathematician	12345	32130


Retrieved data from myUser:
USERNAME
Adam Troy