University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 2

Date: February 28, 2023

Group Number: 73

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Devin Chikhlia	80666621	o2x2a	devdevin007@gmail.com
Lily Zhang	86004801	l3a6l	zhangyuzhelily@gmail.com
Jessie Sheng	39641774	q8c8z	jessieshengyj@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

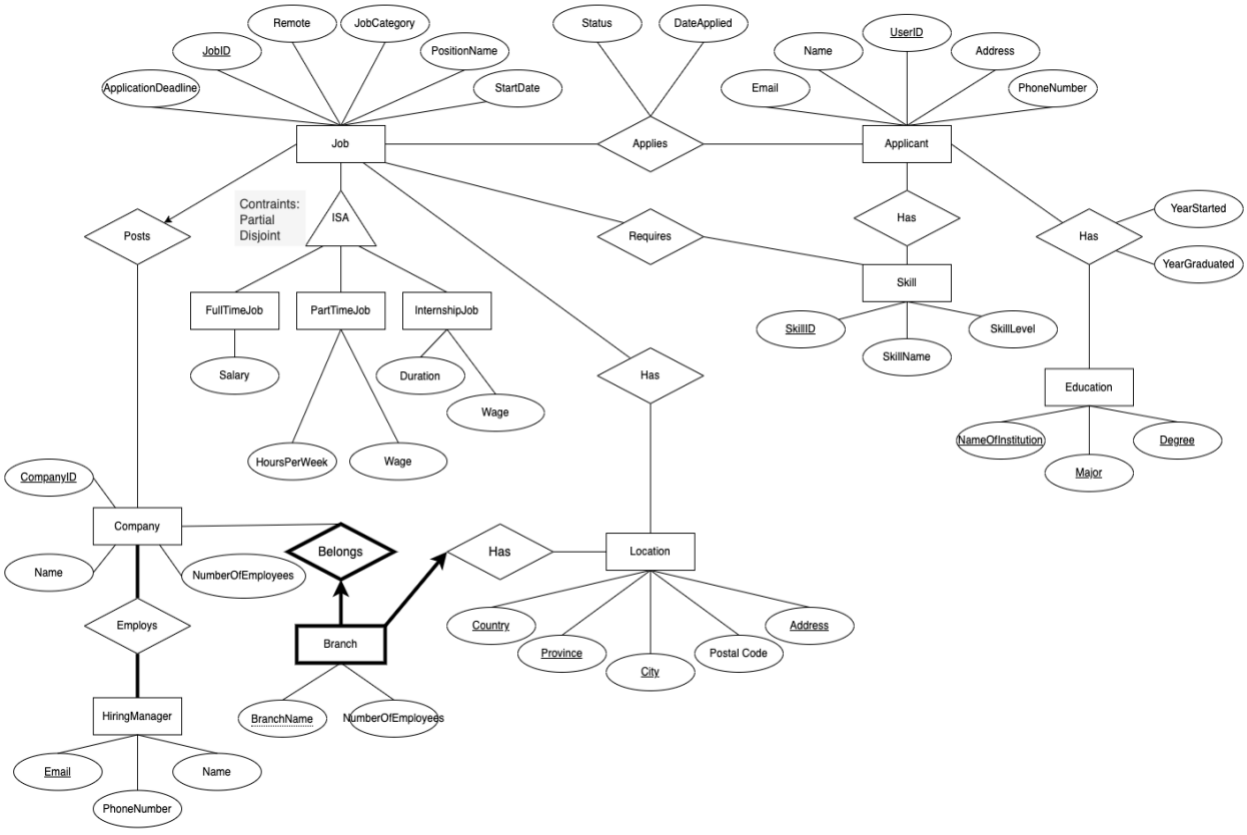
University of British Columbia, Vancouver
Department of Computer Science

Job Portal Database

Project Summary

The job postings system is the main domain of the application, which includes companies posting and finding applicants and job seekers searching and applying for jobs. The system models relevant information such as contact details, education, and skills for employers to evaluate. The applies relationship between jobs and applicants is the main function of the database.

ER Diagram



University of British Columbia, Vancouver
Department of Computer Science

ER Diagram Modifications & Naming Changes

Branch has company -> Branch belongs to company
- Changed to make the name better describe the relationship

Update no spaces on ER diagram
- Since spaces are not allowed for table names, to be consistent we removed any spaces in the attribute names

Got rid of benefits in Full time job
- Removed because we can't have lists or arrays to represent a list of job benefits (so it would not be feasible in this format) and to avoid straying too far from the original ER diagram by creating a new relationship for a minor detail

YearStarted and YearGraduated moved into ApplicantHasEducation
- Moved because we felt that the information would be better represented in the relationship since those 2 attributes are specific to the applicant getting the degree

Full-time and Part-time changed to FullTime and PartTime
- Changed because Full was a reserved term in SQL

ISA constraint changed from Total Disjoint to Partial Disjoint
- Changed because there may be other job types and people should not be restricted to those three jobs

Schemas

Entities:

FullTimeJob (**JobID:** integer, Salary: integer)
JobID is the PK and only CK
JobID is FK referencing Job

PartTimeJob (**JobID:** integer, HoursPerWeek: integer, Wage: real)
JobID is the PK and only CK
JobID is the FK referencing Job

InternshipJob (**JobID:** integer, Duration: integer, Wage: real)
JobID is the PK and only CK
JobID is FK referencing Job

Company (CompanyID: integer, Name: string, NumberOfEmployees: integer)
CompanyID is the PK and only CK

HiringManager (Email: string, PhoneNumber: string, Name: string)
Email is the PK and only CK

Location (Country: string, Province: string, City: string, Address: string, PostalCode: string)
Country, Province, City, and Address are the PK and only CK

Applicant (UserID: integer, Email: string, Name: string, Address: string, PhoneNumber: string)
UserID is the PK
Email is a CK and should be UNIQUE and NOT NULL

Skill (SkillID: integer, SkillName: string, SkillLevel: integer)
SkillID is the PK and CK
SkillLevel and SkillName are a CK and should be NOT NULL

Education (NameOfInstitution: string, Major: string, Degree: string)
NameOfInstitution, Major, and Degree are the PK and only CK

Relations:

Job (JobID: integer, ApplicationDeadline: date, Remote: integer, JobCategory: string, PositionName: string, StartDate: date, **CompanyID:** integer)
JobID is the PK and the only CK
CompanyID is the FK referencing Company

Employs (**CompanyID:** integer, **Email:** string)
CompanyID and Email are PK and only CK
Email is FK referencing HiringManager
Company ID is FK referencing Company

Branch (BranchName: string, NumberOfEmployees: integer, **CompanyID:** integer**, Country:** string, **Province:** string, **City:** string, **Address**: string)
Country, Province, Address, and City are NOT NULL
BranchName and CompanyID are the PK and the only CK
CompanyID is a FK referencing Company
Country, Province, Address, and City are FK referencing Location

Applies (Status: string, DateApplied: date, **UserID:** integer, **JobID:** integer)
UserId and JobId are the PK and only CK
UserID is FK referencing Applicant
JobID is FK referencing Job

Requires (**JobID:** integer, **SkillID:** integer)
JobID and the SkillID are the PK and the only CK
JobID and the SkillID are FK referencing Job

Job_Has_Location (**JobID:** integer, **Country:** string, **Province:** string, **City:** string, **Address:** string)
JobID, Country, Province, City and Address is the PK and is the only CK
JobID is the FK referencing Job
Country, Province, City, and Address are the FK referencing Location

Applicant_Has_Skill (**UserID:** integer, **SkillID:** integer)
UserID and SkillID are the PK and is the only CK
UserID is the FK referencing Applicant
SkillID is the FK referencing Skill

Applicant_Has_Education (**UserID:** integer, **NameOfInstitution:** string, **Major:** string, **Degree:** string, YearStarted: integer, YearGraduated: integer)
UserID, NameOfInstitution, Major, and Degree is the PK and is the only CK
UserID is the FK referencing Applicant
NameOfInstitution, Major, Degree is the FK referencing Education

University of British Columbia, Vancouver
Department of Computer Science

Functional Dependencies (FDs)

FullTimeJob (**JobID:** integer, Salary: integer)
JobID → Salary

PartTimeJob (**JobID:** integer, HoursPerWeek: integer, Wage: real)
JobID → HoursPerWeek, Wage

InternshipJob (**JobID:** integer, Duration: integer, Wage: real)
JobID → Duration, Wage

Company (CompanyID: integer, Name: string, NumberOfEmployees: integer)
CompanyID → Name, NumberOfEmployees

HiringManager (Email: string, PhoneNumber: string, Name: string)
Email → PhoneNumber, Name

Location (Country: string, Province: string, City: string, Address: string, PostalCode: string)
Country, Province, City, Address → PostalCode
Country, PostalCode → Province, City

Applicant (UserID: integer, Email: string, Name: string, Address: string, PhoneNumber: string)
UserID → Email, Name, Address, PhoneNumber
Email → UserID, Name, Address, PhoneNumber

Skill (SkillID: integer, SkillName: string, SkillLevel: integer)
SkillID → SkillName, SkillLevel
SkillName, SkillLevel → SkillID

Education (NameOfInstitution: string, Major: string, Degree: string)
No FDs

Job (JobID: integer, ApplicationDeadline: date, Remote: integer, JobCategory: string, PositionName: string, StartDate: date, **CompanyID:** integer)
JobID → ApplicationDeadline, Remote, JobCategory, PositionName, StartDate, CompanyID
PositionName → JobCategory

Employs (**CompanyID:** integer, **Email:** string)
No FDs

Branch (BranchName: string, NumberOfEmployees: integer, **CompanyID:** integer**, Country:** string, **Province:** string, **City:** string, **Address**: string)
BranchName, CompanyID → NumberOfEmployees, Country, Province, City, Address

Applies (Status: string, DateApplied: date, **UserID:** integer, **JobID:** integer)
UserID, JobID → Status, DateApplied

Requires (**JobID:** integer, **SkillID:** integer)
No FDs

Job_Has_Location (**JobID:** integer, **Country:** string, **Province:** string, **City:** string, **Address:** string)
No FDs

Applicant_Has_Skill (**UserID:** integer, **SkillID:** integer)
No FDs

Applicant_Has_Education (**UserID:** integer, **NameOfInstitution:** string, **Major:** string, **Degree:** string, YearStarted: integer, YearGraduated: integer)
UserID, NameOfInstitution, Major, Degree → YearStarted, YearGraduated

Normalization into BCNF

Note:
If already in BCNF, indicated by "Already in BCNF" and all PK and CK remain the same.
If not already in BCNF, indicated by "Not in BCNF" and the normalization process is shown below.

FullTimeJob (**JobID:** integer, Salary: integer)
JobID → Salary
Already in BCNF

PartTimeJob (**JobID:** integer, HoursPerWeek: integer, Wage: real)
JobID → HoursPerWeek, Wage
Already in BCNF

InternshipJob (**JobID:** integer, Duration: integer, Wage: real)
JobID → Duration, Wage
Already in BCNF

Company (CompanyID: integer, Name: string, NumberOfEmployees: integer)
CompanyID → Name, NumberOfEmployees
Already in BCNF

HiringManager (Email: string, PhoneNumber: string, Name: string)
Email → PhoneNumber, Name
Already in BCNF

Location (Country: string, Province: string, City: string, Address: string, PostalCode: string)
Country, Province, City, Address → PostalCode
Country, PostalCode → Province, City
Not in BCNF

Normalization Process:

Putting it into 1 attribute on the RHS
Country, PostalCode → Province
Country, PostalCode → City
Country, Province, City, Address → PostalCode then we still have this from before

Closures:
Country, Province, City, Address$^+$ = {Country, Province, City, Address, PostalCode}
Country, PostalCode$^+$ = {Country, PostalCode, Province, City}

Since Country, PostalCode → Province, City violates BCNF as Country and PostalCode is not a superkey, we decompose on it.

Step 1: Decompose on Country, PostalCode → Province
$Location_1$(Country, PostalCode, Province), $Location_3$(Country, PostalCode, City), $Location_4$(Country, PostalCode, Address)

Step 2: Decompose on Country, PostalCode → City
$Location_1$ is in BCNF but $Location_2$ still violates BCNF since country and postal code is not a superkey in $Location_2$ so we decompose again.
$Location_3$(Country, PostalCode, City), $Location_4$(Country, PostalCode, Address)

Now we have no more FDs that violate BCNF, thus it is in BCNF.

The **FINAL ANSWER** is $Location_1$(Country, PostalCode, Province), $Location_3$(Country, PostalCode, City), $Location_4$(Country, PostalCode, Address)
Country, PostalCode are PK and the only CK for $Location_1$, Country and PostalCode are PK and the only CK for $Location_3$, Country, PostalCode and Address are PK and the only CK for $Location_4$.

Applicant (UserID: integer, Email: string, Name: string, Address: string, PhoneNumber: string)
UserID → Email, Name, Address, PhoneNumber
Email → UserID, Name, Address, PhoneNumber
Already in BCNF

Skill (SkillID: integer, SkillName: string, SkillLevel: integer)
SkillID → SkillName, SkillLevel
SkillName, SkillLevel → SkillID
Already in BCNF

Education (NameOfInstitution: string, Major: string, Degree: string)
No FDs
Already in BCNF

Job (JobID: integer, ApplicationDeadline: date, Remote: integer, JobCategory: string, PositionName: string, StartDate: date, **CompanyID:** integer)
JobID → ApplicationDeadline, Remote, JobCategory, PositionName, StartDate, CompanyID
PositionName → JobCategory
Not in BCNF

Normalization Process:

Putting into 1 attribute on RHS
JobID → ApplicationDeadline
JobID → CompanyID
JobID → Remote
JobID → StartDate
JobID → JobCategory, PositionName
JobID → PositionName
PositionName → JobCategory then we still have this from before

Closures:
$JobID^+$ = {ApplicationDeadline, Remote, JobCategory, PositionName, StartDate, CompanyID}
$PositionName^+$ = {PositionName, JobCategory}

We see that PositionName → JobCategory violates BCNF since PositionName is not a superkey, so we decompose on this.

Step 1: Decompose on PositionName → JobCategory
Job_1(JobID, ApplicationDeadline, Remote, PositionName, StartDate, **CompanyID**), Job_2(PositionName, JobCategory)

Here we see that there are no more FDs that violate BCNF, thus it is in BCNF.

The **FINAL ANSWER** is Job_1(JobID: integer, ApplicationDeadline: date, Remote: integer, PositionName: string, StartDate: date, **CompanyID:** integer) and Job_2(PositionName: string, JobCategory: string)
For Job_1 the PK and only CK is JobID
For Job_2 the PK and only CK is PositionName

Employs (**CompanyID:** integer, **Email:** string)
No FDs
Already in BCNF

Branch (BranchName: string, NumberOfEmployees: integer, **CompanyID:** integer**, Country:** string, **Province:** string, **City:** string, **Address**: string)
BranchName, CompanyID → NumberOfEmployees, Country, Province, City, Address
Already in BCNF

Applies (Status: string, DateApplied: date, **UserID:** integer, **JobID:** integer)
UserID, JobID → Status, DateApplied
Already in BCNF

Requires (**JobID:** integer, **SkillID:** integer)
No FDs
Already in BCNF

Job_Has_Location (**JobID:** integer, **Country:** string, **Province:** string, **City:** string, **Address:** string)
No FDs
Already in BCNF

Applicant_Has_Skill (**UserID:** integer, **SkillID:** integer)
No FDs
Already in BCNF

Applicant_Has_Education (**UserID:** integer, **NameOfInstitution:** string, **Major:** string, **Degree:** string, YearStarted: integer, YearGraduated: integer)
UserID, NameOfInstitution, Major, Degree → YearStarted, YearGraduated
Already in BCNF

SQL DDL Statements

FullTimeJob (**JobID:** integer, Salary: integer)

```
CREATE TABLE FullTimeJob (
        JobID           INTEGER     PRIMARY KEY,
        Salary          INTEGER,
        FOREIGN KEY (JobID) REFERENCES Job
)
```

PartTimeJob (**JobID:** integer, HoursPerWeek: integer, Wage: real)

```
CREATE TABLE PartTimeJob (
        JobID               INTEGER     PRIMARY KEY,
        HoursPerWeek        INTEGER,
        Wage                INTEGER,
        FOREIGN KEY (JobID) REFERENCES Job
)
```

InternshipJob (**JobID:** integer, Duration: integer, Wage: real)

```
CREATE TABLE InternshipJob (
        JobID           INTEGER     PRIMARY KEY,
        Duration        INTEGER,
        Wage            INTEGER,
        FOREIGN KEY (JobID) REFERENCES Job
)
```

Company (CompanyID: integer, Name: string, NumberOfEmployees: integer)

```
CREATE TABLE Company (
        CompanyID               INTEGER     PRIMARY KEY,
        Name                    VARCHAR(80),
        NumberOfEmployees       INTEGER
)
```

HiringManager (Email: string, PhoneNumber: string, Name: string)

```
CREATE TABLE HiringManager (
        Email           VARCHAR(80)     PRIMARY KEY,
        PhoneNumber     VARCHAR(80),
        Name            VARCHAR(80)
)
```

Location$_1$(Country: string, PostalCode: string, Province: string),

```
CREATE TABLE Location1 (
        Country         VARCHAR(40),
        PostalCode      VARCHAR(20),
        Province        VARCHAR(80),
        PRIMARY KEY (Country, PostalCode)
)
```

Location$_3$(Country: string, PostalCode: string, City: string)

```
CREATE TABLE Location3 (
        Country         VARCHAR(40),
        PostalCode      VARCHAR(20),
        City            VARCHAR(80),
        PRIMARY KEY (Country, PostalCode)
)
```

Location$_4$(Country: string, PostalCode: string, Address: string)

```
CREATE TABLE Location4 (
        Country         VARCHAR(40),
        PostalCode      VARCHAR(20),
        Address         VARCHAR(80),
        PRIMARY KEY (Country, PostalCode, Address)
)
```

Applicant (UserID: integer, Email: string, Name: string, Address: string, PhoneNumber: string)

```
CREATE TABLE Applicant (
        UserID          INTEGER         PRIMARY KEY,
        Email           VARCHAR(80)     NOT NULL,
        Name            VARCHAR(80),
        Address         VARCHAR(80),
        PhoneNumber     VARCHAR(30),
        UNIQUE (Email)
)
```

Skill (SkillID: integer, SkillName: string, SkillLevel: integer)

```
CREATE TABLE Skill (
        SkillID         INTEGER         PRIMARY KEY,
        SkillName       VARCHAR(80)     NOT NULL,
        SkillLevel      INTEGER         NOT NULL
)
```

Education (NameOfInstitution: string, Major: string, Degree: string)

```
CREATE TABLE Education (
        NameOfInstitution       VARCHAR(80),
        Major                   VARCHAR(40),
        Degree                  VARCHAR(40),
        PRIMARY KEY (NameOfInstitution, Major, Degree)
)
```

Job$_1$(JobID: integer, ApplicationDeadline: date, Remote: integer, PositionName: string, StartDate: date, **CompanyID:** integer)

```
CREATE TABLE Job1(
        JobID               INTEGER     PRIMARY KEY,
        ApplicationDeadline DATE,
        Remote              INTEGER,
        PositionName        VARCHAR(40),
        StartDate           DATE,
        CompanyID           INTEGER,
        FOREIGN KEY (CompanyID) REFERENCES Company
                            ON DELETE SET NULL
                            ON UPDATE CASCADE
)
```

Job$_2$(PositionName: string, JobCategory: string)

```
CREATE TABLE Job2(
        PositionName        VARCHAR(40)     PRIMARY KEY,
        JobCategory         VARCHAR(40)
)
```

Employs (**CompanyID:** integer, **Email:** string)

```
CREATE TABLE Employs (
        CompanyID           VARCHAR(40),
        Email               VARCHAR(80),
        PRIMARY KEY (CompanyID, Email),
        FOREIGN KEY (CompanyID) REFERENCES Company,
        FOREIGN KEY (Email) REFERENCES HiringManager
)
```

Branch (BranchName: string, NumberOfEmployees: integer, **CompanyID:** integer**, Country:** string, **Province:** string, **City:** string, **Address**: string)

```
CREATE TABLE Branch (
        BranchName          VARCHAR(80),
        NumberOfEmployees   INTEGER,
        CompanyID           INTEGER,
        Country             VARCHAR(40),
        Province            VARCHAR(20),
        City                VARCHAR(20),
        Address             VARCHAR(80),
        PRIMARY KEY (BranchName, CompanyID),
        FOREIGN KEY (CompanyID) REFERENCES Company
                            ON DELETE SET NULL
                            ON UPDATE CASCADE)
        FOREIGN KEY (Country, Province, City, Address) REFERENCES Location
)
```

Applies (Status: string, DateApplied: date, **UserID:** integer, **JobID:** integer)

```
CREATE TABLE Applies (
        Status          VARCHAR(80),
        DateApplied     DATE,
        UserID          INTEGER,
        JobID           INTEGER,
        PRIMARY KEY (UserID, JobID),
        FOREIGN KEY (UserID) REFERENCES Applicant,
        FOREIGN KEY (JobID) REFERENCES Job
)
```

Requires (**JobID:** integer, **SkillID:** integer)

```
CREATE TABLE Requires (
        JobID           INTEGER,
        SkillID         INTEGER,
        PRIMARY KEY (JobID, SkillID),
        FOREIGN KEY (JobID) REFERENCES Job,
        FOREIGN KEY (SkillID) REFERENCES Skill
)
```

Job_Has_Location (**JobID:** integer, **Country:** string, **Province:** string, **City:** string, **Address:** string)

```
CREATE TABLE Job_Has_Location (
        JobID           INTEGER,
        Country         VARCHAR(80),
        Province        VARCHAR(80),
        City            VARCHAR(80),
        Address         VARCHAR(80),
        PRIMARY KEY (JobID, Country, Province, City, Address),
        FOREIGN KEY (Country, Province, City, Address) REFERENCES Location
)
```

Applicant_Has_Skill (**UserID:** integer, **SkillID:** integer)

```
CREATE TABLE Applicant_Has_Skill (
        UserID          INTEGER,
        SkillID         INTEGER,
        PRIMARY KEY (UserID, SkillID),
        FOREIGN KEY (SkillID) REFERENCES Skill,
        FOREIGN KEY (UserID) REFERENCES Applicant
)
```

Applicant_Has_Education (**UserID:** integer, **NameOfInstitution:** string, **Major:** string, **Degree:** string, YearStarted: integer, YearGraduated: integer)

```
CREATE TABLE Applicant_Has_Education (
        UserID                  INTEGER,
        NameOfInstitution       VARCHAR(80)     NOT NULL,
        Major                   VARCHAR(80)     NOT NULL,
        Degree                  VARCHAR(80)     NOT NULL,
        YearStarted             INTEGER         NOT NULL,
```

```
        YearGraduated           INTEGER,
        PRIMARY KEY (UserID, NameOfInstitution, Major, Degree),
        FOREIGN KEY (UserID) REFERENCES Applicant,
        FOREIGN KEY (NameOfInstitution, Major, Degree) REFERENCES Education
)
```

Insert Statements

FullTimeJob (**JobID:** integer, Salary: integer)
INSERT INTO FullTimeJob VALUES (1, 50000);
INSERT INTO FullTimeJob VALUES (2, 70000);
INSERT INTO FullTimeJob VALUES (3, 50000);
INSERT INTO FullTimeJob VALUES (4, 30000);
INSERT INTO FullTimeJob VALUES (5, 36000);

PartTimeJob (**JobID:** integer, HoursPerWeek: integer, Wage: real)
INSERT INTO PartTimeJob VALUES (6, 15, 16.50);
INSERT INTO PartTimeJob VALUES (7, 20, 30);
INSERT INTO PartTimeJob VALUES (8, 20, 35);
INSERT INTO PartTimeJob VALUES (9, 7, 22.75);
INSERT INTO PartTimeJob VALUES (10, 10,19);

InternshipJob (**JobID:** integer, Duration: integer, Wage: real)
INSERT INTO InternshipJob VALUES (11, 8, 17);
INSERT INTO InternshipJob VALUES (12, 12, 20.5);
INSERT INTO InternshipJob VALUES (13, 8, 28);
INSERT INTO InternshipJob VALUES (14, 9, 18.5);
INSERT INTO InternshipJob VALUES (15, 16, 32);

Company (CompanyID: integer, Name: string, NumberOfEmployees: integer)
INSERT INTO Company VALUES (1, 'Amazon', 1541000);
INSERT INTO Company VALUES (2, 'Aritzia', 6569);
INSERT INTO Company VALUES (3, 'PepsiCo', 315000);
INSERT INTO Company VALUES (4, 'Apple', 164000);
INSERT INTO Company VALUES (5, 'SAP', 111961);

HiringManager (Email: string, PhoneNumber: string, Name: string)
INSERT INTO HiringManager VALUES (‘smith@ubc.ca’, 7781234567, John Smith’’);
INSERT INTO HiringManager VALUES (‘sharon@outlook.com’, 1234567890, ‘Sharon Lee’);
INSERT INTO HiringManager VALUES (‘oliver@gmail.com’, 6874706123, ‘Oliver Jones’);
INSERT INTO HiringManager VALUES (‘jon@hotmail.com’, 42810274208, ‘Jon Williams’);
INSERT INTO HiringManager VALUES (‘emma@yahoo.com’, 0138297382, ‘Emma Brown’);

Applicant (UserID: integer, Email: string, Name: string, Address: string, PhoneNumber: string)
INSERT INTO Applicant VALUES (1, 'johnsmiths@hotmail.com', 'John Smiths', '4621 Flinderation Road', '(569) 898-4344');

INSERT INTO Applicant VALUES (2, 'bob@gmail.com', 'Bob Echo', '1772 New York Avenue', '(677) 442-6726');
INSERT INTO Applicant VALUES (3, 'govind@hotmail.com', 'Govind Nuur', '996 Perine Street', '(837) 763-6568');
INSERT INTO Applicant VALUES (4, 'lilavati@gmail.com', 'Lilavati Rajani', '3282 Union Street', '(707) 207-4888');
INSERT INTO Applicant VALUES (5, 'neoneela@hotmail.com', 'Neo Neela', '4621 Flinderation Road', '(676) 952-0954');

Skill (SkillID: integer, SkillName: string, SkillLevel: integer)
INSERT INTO Skill VALUES (1, 'Teamwork', 5);
INSERT INTO Skill VALUES (2, 'Communication', 4);
INSERT INTO Skill VALUES (3, 'Problem Solving', 5);
INSERT INTO Skill VALUES (4, 'Leadership', 3);
INSERT INTO Skill VALUES (5, 'Time Management', 4);

Education (NameOfInstitution: string, Major: string, Degree: string)
INSERT INTO Education VALUES (‘University of British Columbia’, ‘Marketing’, ‘Bachelor’);
INSERT INTO Education VALUES (‘Washington State University’, ‘Chemical Engineering’, ‘Master’);
INSERT INTO Education VALUES (‘University of Toronto’, ‘Cognitive Science’, ‘Bachelor’);
INSERT INTO Education VALUES (‘Stanford University’, Computer Science, ‘PHD’);
INSERT INTO Education VALUES (‘University College London’, ‘Psychology’, ‘Doctorate’);

Job_1(JobID: integer, ApplicationDeadline: date, Remote: integer, PositionName: string, StartDate: date, **CompanyID:** integer)
INSERT INTO Job1 VALUES (1, '2023-05-23', 0, 'Salesmen', '2023-09-01', 3);
INSERT INTO Job1 VALUES (2, '2023-06-15', 1, 'Software Engineer', '2023-09-01', 1);
INSERT INTO Job1 VALUES (3, '2023-05-31', 1, 'Marketing Manager', '2023-08-15', 2);
INSERT INTO Job1 VALUES (4, '2023-06-30', 0, 'Graphic Designer', '2023-09-01', 4);
INSERT INTO Job1 VALUES (5, '2023-06-30', 1, 'Customer Support Representative', '2023-08-15', 5);
INSERT INTO Job1 VALUES (6, '2023-06-30', 0, 'Accountant', '2023-08-01', 1);
INSERT INTO Job1 VALUES (7, '2023-07-15', 1, 'Web Developer', '2023-10-01', 2);
INSERT INTO Job1 VALUES (8, '2023-07-31', 0, 'Human Resources Manager', '2023-09-15', 3);
INSERT INTO Job1 VALUES (9, '2023-08-15', 1, 'Social Media Specialist', '2023-11-01', 4);
INSERT INTO Job1 VALUES (10, '2023-09-01', 0, 'Office Manager', '2023-11-15', 5);
INSERT INTO Job1 VALUES (11, '2023-09-30', 1, 'Data Analyst', '2024-01-01', 1);
INSERT INTO Job1 VALUES (12, '2023-10-15', 0, 'Legal Assistant', '2023-12-01', 2);
INSERT INTO Job1 VALUES (13, '2023-10-31', 1, 'Product Manager', '2024-01-15', 3);
INSERT INTO Job1 VALUES (14, '2023-11-15', 0, 'Sales Representative', '2024-02-01', 4);
INSERT INTO Job1 VALUES (15, '2023-12-01', 1, 'Software Developer', '2024-03-01', 5);

Job_2(PositionName: string, JobCategory: string)
INSERT INTO Job2 VALUES ('Salesmen', 'Sales');
INSERT INTO Job2 VALUES ('Software Engineer', 'Information Technology');
INSERT INTO Job2 VALUES ('Marketing Manager', 'Marketing');
INSERT INTO Job2 VALUES ('Graphic Designer', 'Arts and Design');
INSERT INTO Job2 VALUES ('Customer Support Representative', 'Customer Service');
INSERT INTO Job2 VALUES ('Accountant', 'Finance');

INSERT INTO Job2 VALUES ('Web Developer', 'Information Technology');
INSERT INTO Job2 VALUES ('Human Resources Manager', 'Human Resources');
INSERT INTO Job2 VALUES ('Social Media Specialist', 'Marketing');
INSERT INTO Job2 VALUES ('Office Manager', 'Administration');
INSERT INTO Job2 VALUES ('Data Analyst', 'Information Technology');
INSERT INTO Job2 VALUES ('Legal Assistant', 'Legal');
INSERT INTO Job2 VALUES ('Product Manager', 'Management');
INSERT INTO Job2 VALUES ('Sales Representative', 'Sales');
INSERT INTO Job2 VALUES ('Software Developer', 'Information Technology');

Employs (**CompanyID:** integer, **Email:** string)
INSERT INTO Employs VALUES (‘1’, ‘smith@ubc.ca’);
INSERT INTO Employs VALUES (‘2’, ‘sharon@outlook.com’);
INSERT INTO Employs VALUES (‘3’, ‘oliver@gmail.com’);
INSERT INTO Employs VALUES (‘4’, ‘jon@hotmail.com’);
INSERT INTO Employs VALUES (‘5’, ‘emma@yahoo.com’);

Branch (BranchName: string, NumberOfEmployees: integer, **CompanyID:** integer**, Country:** string, **Province:** string, **City:** string, **Address**: string)
INSERT INTO Branch VALUES (‘Vancouver Branch’, 50, 3, ‘Canada', 'British Columbia', 'Vancouver', '291 Burrard Street');
INSERT INTO Branch VALUES (‘Burnaby Branch’, 35, 1, 'Canada', 'British Columbia', 'Burnaby', '8098 11th Ave');
INSERT INTO Branch VALUES (LA Branch’, 46, 2, ‘United States of America’, 'California', ‘Los Angeles’, '6908 S Central Ave’);
INSERT INTO Branch VALUES (‘Calgary Branch’, 28, 4, 'Canada', 'Alberta', ‘Calgary’, '1030 10 Ave SW');
INSERT INTO Branch VALUES (‘London Branch’, 62, 5, 'Canada', ‘Ontario’, ‘London’, '1151 Richmond St');

Applies (Status: string, DateApplied: date, **UserID:** integer, **JobID:** integer)
INSERT INTO Applies VALUES ('Applied', '2023-02-28', 1, 2);
INSERT INTO Applies VALUES ('Applied', '2023-03-01', 2, 3);
INSERT INTO Applies VALUES ('Applied', '2023-02-28', 3, 5);
INSERT INTO Applies VALUES ('Applied', '2023-02-27', 4, 7);
INSERT INTO Applies VALUES ('Applied', '2023-03-01', 5, 12);

Requires (**JobID:** integer, **SkillID:** integer)
INSERT INTO Requires VALUES (1, 2);
INSERT INTO Requires VALUES (2, 3);
INSERT INTO Requires VALUES (3, 1);
INSERT INTO Requires VALUES (4, 5');
INSERT INTO Requires VALUES (5, 4);

Location1(Country, PostalCode, Province)
INSERT INTO Location1 VALUES ('Canada', ‘V6C 2G8’,'British Columbia');
INSERT INTO Location1 VALUES ('Canada', ‘V3N 2N7’,'British Columbia');
INSERT INTO Location1 VALUES (‘United States of America’, ‘90001’, 'California');

INSERT INTO Location1 VALUES ('Canada', 'T2R 1M4', 'Alberta');
INSERT INTO Location1 VALUES ('Canada', 'N6A 3K7', 'Ontario');

Location$_3$(Country, PostalCode, City)
INSERT INTO Location3 VALUES ('Canada', 'V6C 2G8', 'Vancouver');
INSERT INTO Location3 VALUES ('Canada', 'V3N 2N7', 'Burnaby');
INSERT INTO Location3 VALUES ('United States of America', '90001', 'Los Angeles');
INSERT INTO Location3 VALUES ('Canada', 'T2R 1M4', 'Calgary');
INSERT INTO Location3 VALUES ('Canada', 'N6A 3K7', 'London');

Location$_4$(Country, PostalCode, Address)
INSERT INTO Location4 VALUES ('Canada', 'V6C 2G8', '291 Burrard Street');
INSERT INTO Location4 VALUES ('Canada', 'V3N 2N7', '8098 11th Ave');
INSERT INTO Location4 VALUES ('United States of America', '90001', '6908 S Central Ave');
INSERT INTO Location4 VALUES ('Canada', 'T2R 1M4', '1030 10 Ave SW');
INSERT INTO Location4 VALUES ('Canada', 'N6A 3K7', '1151 Richmond St');

Job_Has_Location (**JobID:** integer, **Country:** string, **Province:** string, **City:** string, **Address:** string)
INSERT INTO Job_Has_Location VALUES (1, 'Canada', 'British Columbia', 'Vancouver', '291 Burrard Street');
INSERT INTO Job_Has_Location VALUES (2, 'Canada', 'British Columbia', 'Burnaby', '8098 11th Ave');
INSERT INTO Job_Has_Location VALUES (3, 'United States of America', 'California', 'Los Angeles', '6908 S Central Ave');
INSERT INTO Job_Has_Location VALUES (4, 'Canada', 'Alberta', 'Calgary', '1030 10 Ave SW');
INSERT INTO Job_Has_Location VALUES (5, 'Canada', 'Ontario', 'London', '1151 Richmond St');

Applicant_Has_Skill (**UserID:** integer, **SkillID:** integer)
INSERT INTO Applicant_Has_Skill VALUES (1, 1);
INSERT INTO Applicant_Has_Skill VALUES (1, 2);
INSERT INTO Applicant_Has_Skill VALUES (3, 1);
INSERT INTO Applicant_Has_Skill VALUES (4, 5);
INSERT INTO Applicant_Has_Skill VALUES (5, 3);

Applicant_Has_Education (**UserID:** integer, **NameOfInstitution:** string, **Major:** string, **Degree:** string, YearStarted: integer, YearGraduated: integer)
INSERT INTO Applicant_Has_Education VALUES (1, 'University of British Columbia', 'Marketing', 'Bachelor', 2001, 2005);
INSERT INTO Applicant_Has_Education VALUES (2, 'University of British Columbia', 'Marketing', 'Bachelor', 2004, 2008);
INSERT INTO Applicant_Has_Education VALUES (1, 'University of Toronto', 'Cognitive Science', 'Bachelor', 2016, 2021);
INSERT INTO Applicant_Has_Education VALUES (3, 'Stanford University', Computer Science, 'PHD', 2016, 2020);
INSERT INTO Applicant_Has_Education VALUES (3, 'University College London', 'Psychology', 'Doctorate', 2010, 2013);