

CPSC 304 Project Cover Page

Milestone #: 2

Date: Feb. 27

Group Number: 57

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Arya Subramanyam	76650738	c6i2b	arya2306@student.ubc.ca
Daniel Ng	70897350	n1f2b	dannsy@student.ubc.ca
Yeonguk Kim	71628911	j8u2b	golde0815@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

Project Summary:

Our project involves creating a job application portal where applicants can create a profile, find relevant postings for company roles and submit their applications. The domain of our application is recruiting for employers and job searching for general users. For their recruiting and employment activities, companies can add jobs and relevant postings for open positions. For job searching, users can add profiles, view jobs by their category, required skills and location.

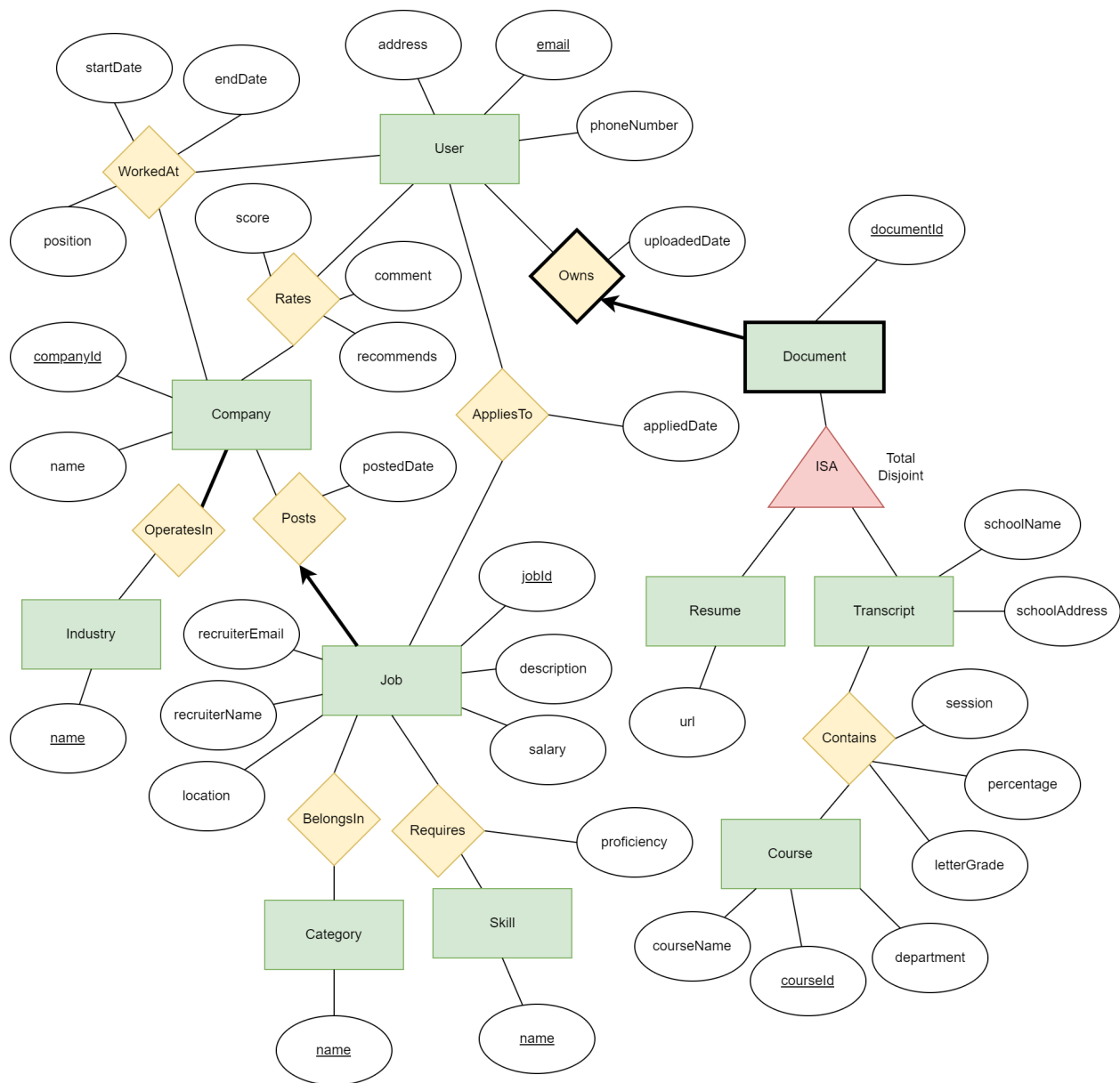
For the database, we will use MySQL, with the programming language of Java. Furthermore, our frontend will be built using ReactJS as the framework, with the programming languages of JavaScript and HTML/CSS.

ER Diagram:

We have made multiple changes to our ER Diagram based on the TA's suggestions. We have decided to incorporate all the suggestions. These updates include:

- Ensuring the participation constraint of the ISA relationship is mentioned (total and disjoint)
- Ensuring that the Documents entity is correctly represented as a weak entity by emboldening the Owns relationship
- Adding a total participation constraints where
 - Companies must be in Industry
 - Jobs must have a Company
- Removing any trivial entities such as the Location Entity, and making it an attribute of the Jobs Entity instead
- Ensuring that the Transcript Entity is correctly represented by containing many Courses and their grades. Instead of having a Course and Grade attribute for the Transcript Entity, we now have a new Contains relationship (many-to-many) for Transcript with a new Courses Entity. The percentage, letterGrade, and session are attributes of the relationship.
- Removed employmentHistory attribute from User Entity because this attribute would have been a list (not conform to 1NF). Instead created a new WorkedAt relationship (many-to-many). The startDate, endDate, and position are attributes of the relationship.
- Removed education attribute from User Entity because this attribute would have been a list (not conform to 1NF). Instead this information will be present in the Transcript Entity which has 2 new attributes schoolName and schoolAddress.
- Added recommends attribute to User rates Company Relationship for non PK/CK FD
- Added recruiterName and recruiterEmail attributes to Job Entity for non PK/CK FD

University of British Columbia, Vancouver
Department of Computer Science



University of British Columbia, Vancouver

Department of Computer Science

Schema:

Note: PK = Primary Key, FK = Foreign Key, CK = Candidate Key. The naming of some schemas, for instance the relationship BelongsIn b/t Job and Category, has been renamed JobBelongsIn for better clarity.

- User(email (PK): CHAR(20), address: CHAR(30), phoneNumber (CK - UNIQUE): INTEGER)
- Company(companyId (PK): INTEGER, name: CHAR(50))
- Rates(email (PK, FK - ON DELETE CASCADE, ON UPDATE CASCADE): CHAR(20), companyId (PK, FK - ON DELETE CASCADE, ON UPDATE CASCADE): INTEGER, score (NOT NULL): INTEGER, recommends (NOT NULL): BOOLEAN, comment: CHAR(200))
- WorkedAt(email (PK, FK - ON DELETE CASCADE, ON UPDATE CASCADE): CHAR(20), companyId (PK, FK - ON DELETE CASCADE, ON UPDATE CASCADE): INTEGER, position: CHAR(20), startDate (NOT NULL): DATE, endDate (NOT NULL): DATE)
- Industry(name (PK): CHAR(50))
- OperatesIn(industryName (PK, FK - ON DELETE CASCADE, ON UPDATE CASCADE): CHAR(50), companyId (PK, FK - ON DELETE CASCADE, ON UPDATE CASCADE): INTEGER)
- PostedJob(jobId (PK): INTEGER, companyId (FK, NOT NULL, ON DELETE CASCADE, ON UPDATE CASCADE): INTEGER, postedDate: DATE, location (NOT NULL): CHAR(30), description (NOT NULL): CHAR(200), salary: INTEGER, recruiterName: CHAR(20), recruiterEmail: CHAR(20))
- Category(name (PK): CHAR(20))
- JobBelongsIn(categoryName (PK, FK - ON DELETE CASCADE, ON UPDATE CASCADE): CHAR(20), jobId (PK, FK - ON DELETE CASCADE, ON UPDATE CASCADE): INTEGER)
- Skill(name (PK): CHAR(20))
- JobRequires(skillName (PK, FK - ON DELETE CASCADE, ON UPDATE CASCADE): CHAR(20), jobId (PK, FK - ON DELETE CASCADE, ON UPDATE CASCADE): INTEGER, proficiency: INTEGER)
- UserAppliesTo(email (PK, FK - ON DELETE CASCADE, ON UPDATE CASCADE): CHAR(20), jobId (PK, FK - ON DELETE CASCADE, ON UPDATE CASCADE): INTEGER, appliedDate: DATE)
- Resume(email (PK, FK - ON DELETE CASCADE, ON UPDATE CASCADE): CHAR(20), documentId (PartialK): INTEGER, url (CK - UNIQUE): CHAR(50))
- Transcript(email (PK, FK - ON DELETE CASCADE, ON UPDATE CASCADE): CHAR(20), documentId (PartialK): INTEGER, schoolName (NOT NULL): CHAR(30), schoolAddress: CHAR(30))
- Course(courseId (PK): INTEGER, courseName (CK - UNIQUE): CHAR(20), department: CHAR(20))

University of British Columbia, Vancouver

Department of Computer Science

- TranscriptContains(courseId (FK, PK - ON DELETE CASCADE, ON UPDATE CASCADE): INTEGER, documentId (FK, PK - ON DELETE CASCADE, ON UPDATE CASCADE): email (FK, PK - ON DELETE CASCADE, ON UPDATE CASCADE), INTEGER, percentage (NOT NULL): INTEGER, letterGrade (NOT NULL): CHAR(1), session: CHAR(10))

University of British Columbia, Vancouver

Department of Computer Science

Functional Dependencies (FDs):

- User relation
 - email \rightarrow email, address, phoneNumber
 - phoneNumber \rightarrow email, address, phoneNumber
- Company relation
 - companyId \rightarrow companyId, name
- Rates relation
 - email, companyId \rightarrow email, companyId, score, recommends, comment
 - score \rightarrow score, recommends
- WorkedAt relation
 - email, companyId \rightarrow email, companyId, position, startDate, endDate
- Industry relation (only has 1 attribute so FD is trivial)
 - name \rightarrow name
- OperatesIn relation (FD is trivial)
 - industryName, companyId \rightarrow industryName, companyId
- PostedJob relation
 - jobId \rightarrow jobId, companyId, postedDate, location, description, salary, recruiterName, recruiterEmail
 - recruiterEmail \rightarrow recruiterEmail, recruiterName
- Category relation (FD is trivial)
 - name \rightarrow name
- JobBelongsIn relation (FD is trivial)
 - categoryName, jobId \rightarrow categoryName, jobId
- Skill relation (FD is trivial)
 - name \rightarrow name
- JobRequires relation
 - skillName, jobId \rightarrow skillName, jobId, proficiency
- UserAppliesTo relation
 - email, jobId \rightarrow email, jobId, appliedDate
- Resume relation
 - email, documentId \rightarrow email, documentId, url
 - url \rightarrow email, documentId, url
- Transcript relation
 - email, documentId \rightarrow email, documentId, schoolName, schoolAddress
 - schoolName \rightarrow schoolName, schoolAddress
- Course relation
 - courseId \rightarrow courseId, courseName, department
 - courseName \rightarrow courseId courseName, department
- TranscriptContains relation
 - courseId, email, documentId \rightarrow courseId, email, documentId, percentage, letterGrade, session
 - percentage \rightarrow percentage, letterGrade

University of British Columbia, Vancouver

Department of Computer Science

Normalization:

- User relation
 - All FDs have minimal keys as determinant, therefore already in BCNF
 - Final table:
 - User(email (PK): CHAR(20), address: CHAR(30), phoneNumber (CK - UNIQUE): INTEGER)
- Company relation
 - Only 1 FD with minimal key as determinant, therefore already in BCNF
 - Final table:
 - Company(companyId (PK): INTEGER, name: CHAR(50))
- Rates relation
 - FD score -> score, recommends violates BCNF because score is not a key
 - Decompose score and recommends to new relation Score
 - Final tables:
 - Score(value (PK): INTEGER, recommends (NOT NULL): BOOLEAN)
 - Rates(email (PK, FK - ON DELETE CASCADE, ON UPDATE CASCADE): CHAR(20), companyId (PK, FK - ON DELETE CASCADE, ON UPDATE CASCADE): INTEGER, score (FK, NOT NULL - ON DELETE CASCADE, ON UPDATE CASCADE): INTEGER, comment: CHAR(200))
- WorkedAt relation
 - Only 1 FD with minimal key as determinant, therefore already in BCNF
 - Final table:
 - WorkedAt(email (PK, FK - ON DELETE CASCADE, ON UPDATE CASCADE): CHAR(20), companyId (PK, FK - ON DELETE CASCADE, ON UPDATE CASCADE): INTEGER, position: CHAR(20), startDate (NOT NULL): DATE, endDate (NOT NULL): DATE)
- Industry relation
 - Only 1 FD with minimal key as determinant, therefore already in BCNF
 - Final table:
 - Industry(name (PK): CHAR(50))
- OperatesIn relation
 - Only 1 FD with minimal key as determinant, therefore already in BCNF
 - Final table:
 - OperatesIn(industryName (PK, FK - ON DELETE CASCADE, ON UPDATE CASCADE): CHAR(50), companyId (PK, FK - ON DELETE CASCADE, ON UPDATE CASCADE): INTEGER)
- PostedJob relation
 - FD recruiterEmail -> recruiterEmail, recruiterName violates BCNF because recruiterEmail is not a key
 - Decompose recruiterEmail and recruiterName to new relation Recruiter
 - Final tables:
 - Recruiter(email (PK): CHAR(20), name (NOT NULL): CHAR(20))

University of British Columbia, Vancouver

Department of Computer Science

- PostedJob(jobId (PK): INTEGER, companyId (FK, NOT NULL, ON DELETE CASCADE, ON UPDATE CASCADE): INTEGER, postedDate: DATE, location (NOT NULL): CHAR(30), description (NOT NULL): CHAR(200), salary: INTEGER, recruiterEmail (FK): CHAR(20))
- Category relation
 - Only 1 FD with minimal key as determinant, therefore already in BCNF
 - Final table:
 - Category(name (PK): CHAR(20))
- JobBelongsIn relation
 - Only 1 FD with minimal key as determinant, therefore already in BCNF
 - Final table:
 - JobBelongsIn(categoryName (PK, FK - ON DELETE CASCADE, ON UPDATE CASCADE): CHAR(20), jobId (PK, FK - ON DELETE CASCADE, ON UPDATE CASCADE): INTEGER)
- Skill relation
 - Only 1 FD with minimal key as determinant, therefore already in BCNF
 - Final table:
 - Skill(name (PK): CHAR(20))
- JobRequires relation
 - Only 1 FD with minimal key as determinant, therefore already in BCNF
 - Final table:
 - JobRequires(skillName (PK, FK - ON DELETE CASCADE, ON UPDATE CASCADE): CHAR(20), jobId (PK, FK - ON DELETE CASCADE, ON UPDATE CASCADE): INTEGER, proficiency: INTEGER)
- UserAppliesTo relation
 - Only 1 FD with minimal key as determinant, therefore already in BCNF
 - Final table:
 - UserAppliesTo(email (PK, FK - ON DELETE CASCADE, ON UPDATE CASCADE): CHAR(20), jobId (PK, FK - ON DELETE CASCADE, ON UPDATE CASCADE): INTEGER, appliedDate: DATE)
- Resume relation
 - All FDs have minimal keys as determinant, therefore already in BCNF
 - Final table:
 - Resume(email (PK, FK - ON DELETE CASCADE, ON UPDATE CASCADE): CHAR(20), documentId (PartialK): INTEGER, url (CK - UNIQUE): CHAR(50))
- Transcript relation
 - FD schoolName -> schoolName, schoolAddress violates BCNF because schoolName is not a key
 - Decompose schoolName and schoolAddress to new relation School
 - Final tables:
 - School(name (PK): CHAR(30), address: CHAR(30))

University of British Columbia, Vancouver

Department of Computer Science

- Transcript(email (PK, FK - ON DELETE CASCADE, ON UPDATE CASCADE): CHAR(20), documentId (PartialK): INTEGER, schoolName (FK, NOT NULL): CHAR(30))
- Course relation
 - All FDs have minimal keys as determinant, therefore already in BCNF
 - Final table:
 - Course(courseId (PK): INTEGER, courseName (CK - UNIQUE): CHAR(20), department: CHAR(20))
- TranscriptContains relation
 - FD percentage -> percentage, letterGrade violates BCNF because percentage is not a key
 - Decompose percentage and letterGrade to new relation Grade
 - Final tables:
 - Grade(percentage (PK): INTEGER, letterGrade (NOT NULL): CHAR(1))
 - TranscriptContains(courseId (FK, PK - ON DELETE CASCADE, ON UPDATE CASCADE): INTEGER, email (FK, PK - ON DELETE CASCADE, ON UPDATE CASCADE), documentId (FK, PK - ON DELETE CASCADE, ON UPDATE CASCADE): INTEGER, percentage (FK, NOT NULL): INTEGER, session: CHAR(10))

University of British Columbia, Vancouver
Department of Computer Science

SQL DDL and Insertion:

```
CREATE TABLE User (email CHAR(20),  
                    address CHAR(30),  
                    phoneNumber INTEGER,  
                    PRIMARY KEY (email),  
                    UNIQUE (phoneNumber));
```

```
INSERT INTO User(email,address, phoneNumber)
```

```
Values ("golde0815@gmail.com", "V5G1C8", 7783987);
```

```
INSERT INTO User(email, address, phoneNumber)
```

```
Values ("my123@gmail.com", "1U72J8", 7747854);
```

```
INSERT INTO User(email, address, phoneNumber)
```

```
Values ("valid@gmail.com", "4Q71H4", 0187);
```

```
INSERT INTO User(email, address, phoneNumber)
```

```
Values ("ygim1509@gmail.com", "5Y72C8", 0107854);
```

```
INSERT INTO User(email, address, phoneNumber)
```

```
Values ("manred@gmail.com", "7Q18B3", 66464);
```

```
CREATE TABLE Company (companyID INTEGER,  
                        name CHAR(50),  
                        PRIMARY KEY (companyID));
```

```
INSERT INTO Company(companyID, name) Values (1234,"Daimler");
```

```
INSERT INTO Company(companyID, name) Values (12,"PCL");
```

```
INSERT INTO Company(companyID, name) Values (34,"Outback");
```

```
INSERT INTO Company(companyID, name) Values (14,"Bayer");
```

```
INSERT INTO Company(companyID, name) Values (177,"Novavax");
```

University of British Columbia, Vancouver
Department of Computer Science

```
CREATE TABLE Score (value INTEGER,  
                      recommends BOOLEAN NOT NULL,  
                      PRIMARY KEY (value));  
  
INSERT INTO Score(value, recommends) Values(90, True);  
INSERT INTO Score(value, recommends) Values(80, True);  
INSERT INTO Score(value, recommends) Values(70, True);  
INSERT INTO Score(value, recommends) Values(40, False);  
INSERT INTO Score(value, recommends) Values(20, False);
```

```
CREATE TABLE Rates (email CHAR(20),  
                     companyID INTEGER,  
                     score INTEGER,  
                     comment CHAR(200),  
                     PRIMARY KEY(email, companyID),  
                     FOREIGN KEY (email) REFERENCES User (email)  
                     ON DELETE CASCADE ON UPDATE CASCADE,  
                     FOREIGN KEY (companyID) REFERENCES Company (companyID)  
                     ON DELETE CASCADE ON UPDATE CASCADE,  
                     FOREIGN KEY (score) REFERENCES Score (value)  
                     ON DELETE CASCADE ON UPDATE CASCADE);  
  
INSERT INTO Rates(email,companyID, score, comment)  
Values ("golde0815@gmail.com", 1234, 90, "very good");  
INSERT INTO Rates(email,companyID, score, comment)  
Values ("my123@gmail.com", 14, 80, "good");  
INSERT INTO Rates(email,companyID, score, comment)  
Values ("golde0815@gmail.com", 14, 70, "recommend");
```

University of British Columbia, Vancouver
Department of Computer Science

```
INSERT INTO Rates(email,companyID, score, comment)
```

```
Values ("ygim1509@gmail.com", 177, 40, "not good");
```

```
INSERT INTO Rates(email,companyID, score, comment)
```

```
Values ("manred@gmail.com", 12, 20, "bad");
```

```
CREATE TABLE Industry (name CHAR(50) Primary key)
```

```
;Insert INTO Industry(name) Values ("Manufacturing");
```

```
Insert INTO Industry(name) Values ("Construction");
```

```
Insert INTO Industry(name) Values ("Food Service");
```

```
Insert INTO Industry(name) Values ("Health Care");
```

```
Insert INTO Industry(name) Values ("Biotechnology");
```

```
CREATE TABLE OperatesIn (industryName CHAR(50),
```

```
    companyID INTEGER,
```

```
    PRIMARY KEY (industryName, companyID),
```

```
    FOREIGN KEY (industryName) REFERENCES Industry (name)
```

```
    ON DELETE CASCADE ON UPDATE CASCADE,
```

```
    FOREIGN KEY (companyID) REFERENCES Company (companyID)
```

```
    ON DELETE CASCADE ON UPDATE CASCADE);
```

```
Insert INTO OperatesIn(industryName, companyID) Values("Manufacturing",1234);
```

```
Insert INTO OperatesIn(industryName, companyID) Values("Construction",12);
```

```
Insert INTO OperatesIn(industryName, companyID) Values("Food Service",34);
```

```
Insert INTO OperatesIn(industryName, companyID) Values("Health Care",14);
```

```
Insert INTO OperatesIn(industryName, companyID) Values("Biotechnology",177);
```

University of British Columbia, Vancouver

Department of Computer Science

```
CREATE TABLE WorkedAt(email CHAR(20),  
  
    companyID INTEGER,  
  
    position CHAR(20),  
  
    startDate DATE NOT NULL,  
  
    endDate DATE NOT NULL,  
  
    PRIMARY KEY (email, companyID),  
  
    FOREIGN KEY (email) REFERENCES User (email)  
  
    ON DELETE CASCADE ON UPDATE CASCADE,  
  
    FOREIGN KEY (companyID) REFERENCES Company (companyID)  
  
    ON DELETE CASCADE ON UPDATE CASCADE);
```

```
Insert INTO WorkedAt(email,companyID, position, startDate, endDate)
```

```
Values ("golde0815@gmail.com", 12, "Constructor", '2008-01-01', '2012-12-31');
```

```
Insert INTO WorkedAt(email,companyID, position, startDate, endDate)
```

```
Values ("valid@gmail.com", 1234, "Manufacturer", '2022-11-11', '2022-11-12');
```

```
Insert INTO WorkedAt(email,companyID, position, startDate, endDate)
```

```
Values ("golde0815@gmail.com", 34, "Chef", '2018-11-11', '2020-12-31');
```

```
Insert INTO WorkedAt(email,companyID, position, startDate, endDate)
```

```
Values ("ygim1509@gmail.com", 12, "Constructor", '2008-11-11', '2022-12-31');
```

```
Insert INTO WorkedAt(email,companyID, position, startDate, endDate)
```

```
Values ("manred@gmail.com", 12, "Constructor", '2008-11-11', '2022-12-31');
```

```
CREATE TABLE Recruiter(email CHAR(20) PRIMARY KEY,
```

```
    name CHAR(20) NOT NULL);
```

```
INSERT INTO Recruiter(email, name) Values ("mr10@gmail.com", "Marcus Rashford");
```

```
INSERT INTO Recruiter(email, name) Values ("portugal@gmail.com", "Bruno Fernandes");
```

```
INSERT INTO Recruiter(email, name) Values ("njsdfja@gmail.com", "Lisandro Martinez");
```

University of British Columbia, Vancouver

Department of Computer Science

```
INSERT INTO Recruiter(email, name) Values ("kilometer@gmail.com", "Luke Shaw");
```

```
INSERT INTO Recruiter(email, name) Values ("f78moref@gmail.com", "Rafael Varane");
```

```
CREATE TABLE PostedJob(jobId INTEGER,  
    companyId INTEGER NOT NULL,  
    postedDate DATE,  
    location CHAR(30) NOT NULL,  
    description CHAR(200) NOT NULL,  
    salary INTEGER,  
    recruiterEmail CHAR(20),  
    Primary Key (jobID),  
    FOREIGN KEY (companyId) REFERENCES Company (companyId)  
    ON DELETE CASCADE ON UPDATE CASCADE,  
    FOREIGN KEY (recruiterEmail) REFERENCES Recruiter (email)  
    ON DELETE CASCADE ON UPDATE CASCADE);
```

```
INSERT INTO PostedJob(jobID,companyId,postedDate, location, description, salary,recruiterEmail)  
Values (15571601,1234,'2023-01-01',"Burnaby", "manufacturing job", 2000, "mr10@gmail.com");
```

```
INSERT INTO PostedJob(jobID,companyId,postedDate, location, description, salary,recruiterEmail)  
Values (12345678,12,'2023-02-01',"Burnaby", "construction job", 2200, "portugal@gmail.com");
```

```
INSERT INTO PostedJob(jobID,companyId,postedDate, location, description, salary,recruiterEmail)  
Values (101,34,'2023-01-02',"Coquitlam", "food service job", 5000, "njsdfja@gmail.com");
```

```
INSERT INTO PostedJob(jobID,companyId,postedDate, location, description, salary,recruiterEmail)  
Values (15501,14,'2023-01-15',"Richmond", "health care job", 1000, "kilometer@gmail.com");
```

```
INSERT INTO PostedJob(jobID,companyId,postedDate, location, description, salary,recruiterEmail)  
Values (171601,177,'2023-01-30',"Richmond", "biotech job", 400, "f78moref@gmail.com");
```

University of British Columbia, Vancouver

Department of Computer Science

```
CREATE TABLE Category(name CHAR(20) PRIMARY KEY);
```

```
INSERT INTO Category(name) Values ("Salesman");
```

```
INSERT INTO Category(name) Values ("Computer Programmer");
```

```
INSERT INTO Category(name) Values ("Accountant");
```

```
INSERT INTO Category(name) Values ("Web Developer");
```

```
INSERT INTO Category(name) Values ("Software Engineerer");
```

```
CREATE TABLE JobBelongsIn(categoryName CHAR(20),
```

```
    jobId INTEGER,
```

```
    PRIMARY KEY (categoryName, jobId),
```

```
    FOREIGN KEY (categoryName) REFERENCES Category (name)
```

```
    ON DELETE CASCADE ON UPDATE CASCADE,
```

```
    FOREIGN KEY (jobId) REFERENCES PostedJob (jobId)
```

```
    ON DELETE CASCADE ON UPDATE CASCADE);
```

```
INSERT INTO JobBelongsIn(categoryName, jobId) VALUES ("Salesman", 15571601);
```

```
INSERT INTO JobBelongsIn(categoryName, jobId) VALUES ("Computer Programmer", 12345678);
```

```
INSERT INTO JobBelongsIn(categoryName, jobId) VALUES ("Accountant", 101);
```

```
INSERT INTO JobBelongsIn(categoryName, jobId) VALUES ("Web Developer", 15501);
```

```
INSERT INTO JobBelongsIn(categoryName, jobId) VALUES ("Software Engineerer", 171601);
```

```
CREATE TABLE Skill(name CHAR(20) PRIMARY KEY);
```

```
INSERT INTO Skill(name) VALUES("MySQL");
```

```
INSERT INTO Skill(name) VALUES("JavaScript");
```

```
INSERT INTO Skill(name) VALUES("C#");
```

```
INSERT INTO Skill(name) VALUES("Python");
```

```
INSERT INTO Skill(name) VALUES("Java");
```

University of British Columbia, Vancouver
Department of Computer Science

```
CREATE TABLE JobRequires (skillName CHAR(20),  
    jobId INTEGER,  
    proficiency INTEGER,  
    PRIMARY KEY (skillName, jobId),  
    FOREIGN KEY (skillName) REFERENCES Skill (name)  
    ON DELETE CASCADE ON UPDATE CASCADE,  
    FOREIGN KEY (jobId) REFERENCES PostedJob (jobId)  
    ON DELETE CASCADE ON UPDATE CASCADE);  
  
INSERT INTO JobRequires(skillName,jobId,proficiency) VALUES("MySQL",171601,100);  
INSERT INTO JobRequires(skillName,jobId,proficiency) VALUES("JavaScript",15501,90);  
INSERT INTO JobRequires(skillName,jobId,proficiency) VALUES("C#",15501,70);  
INSERT INTO JobRequires(skillName,jobId,proficiency) VALUES("Python",12345678,100);  
INSERT INTO JobRequires(skillName,jobId,proficiency) VALUES("Java",12345678,80);
```

```
CREATE TABLE UserAppliesTo(email CHAR(20),  
    jobId INTEGER,  
    appliedDate DATE,  
    PRIMARY KEY (email, jobId),  
    FOREIGN KEY (email) REFERENCES User (email)  
    ON DELETE CASCADE ON UPDATE CASCADE,  
    FOREIGN KEY (jobId) REFERENCES PostedJob (jobId)  
    ON DELETE CASCADE ON UPDATE CASCADE);  
  
INSERT INTO UserAppliesTo(email, jobId, appliedDate)  
VALUES ("ygim1509@gmail.com",171601,'2023-02-01');
```


University of British Columbia, Vancouver

Department of Computer Science

```
INSERT INTO UserAppliesTo(email, jobId, appliedDate)
VALUES ("golde0815@gmail.com",171601,'2023-02-01');

INSERT INTO UserAppliesTo(email, jobId, appliedDate)
VALUES ("my123@gmail.com",12345678,'2023-02-07');

INSERT INTO UserAppliesTo(email, jobId, appliedDate)
VALUES ("ygim1509@gmail.com",15501,'2023-01-22');

INSERT INTO UserAppliesTo(email, jobId, appliedDate)
VALUES ("valid@gmail.com",15571601,'2023-01-10');
```

```
CREATE TABLE Resume(email CHAR(20),
    documentId INTEGER,
    url CHAR(50),
    PRIMARY KEY (email, documentId),
    UNIQUE (url),
    FOREIGN KEY (email) REFERENCES User (email)
    ON DELETE CASCADE ON UPDATE CASCADE);

INSERT INTO Resume (email, documentId, url)
VALUES ("golde0815@gmail.com",11000,"https://www.example.org/");

INSERT INTO Resume (email, documentId, url)
VALUES ("my123@gmail.com",12000,"http://www.example.edu/attraction");

INSERT INTO Resume (email, documentId, url)
VALUES ("ygim1509@gmail.com",13000,"https://www.example.com/");

INSERT INTO Resume (email, documentId, url)
```

University of British Columbia, Vancouver

Department of Computer Science

```
VALUES ("valid@gmail.com",14000,"http://www.example.com/aftermath.aspx");
```

```
INSERT INTO Resume (email, documentId, url)
```

```
VALUES ("manred@gmail.com",15000,"https://bird.example.com/bag/basin.php");
```

```
CREATE TABLE School(name CHAR(30) PRIMARY KEY,
```

```
    address CHAR(30));
```

```
INSERT INTO School(name, address) VALUES("UBC","Vancouver, BC V6T 1Z4");
```

```
INSERT INTO School(name, address) VALUES("SFU","Burnaby, BC V5A 1S6");
```

```
INSERT INTO School(name, address) VALUES("Langara College","Vancouver, BC V5Y 2Z6");
```

```
INSERT INTO School(name, address) VALUES("BCIT","Burnaby, BC V5G 3H2");
```

```
INSERT INTO School(name, address) VALUES("UVic","Victoria, BC V8P 5C2");
```

```
CREATE TABLE Transcript(email CHAR(20),
```

```
    documentId INTEGER,
```

```
    schoolName CHAR(30) NOT NULL,
```

```
    PRIMARY KEY (email, documentId),
```

```
    FOREIGN KEY (email) REFERENCES User (email)
```

```
    ON DELETE CASCADE,
```

```
    FOREIGN KEY (schoolName) REFERENCES School (name)
```

```
    ON DELETE CASCADE ON UPDATE CASCADE);
```

```
INSERT INTO Transcript(email, documentId, schoolName)
```

```
VALUES("golde0815@gmail.com", 21000, "UBC");
```

```
INSERT INTO Transcript(email, documentId, schoolName)
```

University of British Columbia, Vancouver

Department of Computer Science

```
VALUES("manred@gmail.com", 22000, "SFU");
```

```
INSERT INTO Transcript(email, documentId, schoolName)
```

```
VALUES("valid@gmail.com", 23000, "UVic");
```

```
INSERT INTO Transcript(email, documentId, schoolName)
```

```
VALUES("ygim1509@gmail.com", 24000, "Langara College");
```

```
INSERT INTO Transcript(email, documentId, schoolName)
```

```
VALUES("my123@gmail.com", 25000, "BCIT");
```

```
CREATE TABLE Course(courseId INTEGER,
```

```
    courseName CHAR(20),
```

```
    department CHAR(20),
```

```
    PRIMARY KEY (courseId),
```

```
    UNIQUE (courseName));
```

```
INSERT INTO Course(courseId, courseName, department) VALUES (22200, "CHEM 200", "Chemistry");
```

```
INSERT INTO Course(courseId, courseName, department) VALUES (32300, "CHEM 300", "Chemistry");
```

```
INSERT INTO Course(courseId, courseName, department) VALUES (40000, "CPSC 404", "Computer  
Science");
```

```
INSERT INTO Course(courseId, courseName, department) VALUES (29910, "PHYS 210", "Physics");
```

```
INSERT INTO Course(courseId, courseName, department) VALUES (32200, "ITAL 301", "Italian");
```

```
CREATE TABLE Grade(percentage INTEGER PRIMARY KEY,
```

```
    letterGrade CHAR(1) NOT NULL);
```

```
INSERT INTO Grade(percentage, letterGrade) VALUES (89, "A");
```

```
INSERT INTO Grade(percentage, letterGrade) VALUES (75, "B");
```

University of British Columbia, Vancouver

Department of Computer Science

```
INSERT INTO Grade(percentage,letterGrade) VALUES (60, "C");
```

```
INSERT INTO Grade(percentage,letterGrade) VALUES (50, "D");
```

```
INSERT INTO Grade(percentage,letterGrade) VALUES (30, "F");
```

```
CREATE TABLE TranscriptContains(courseId INTEGER,
```

```
    email CHAR(20),
```

```
    documentId INTEGER,
```

```
    percentage INTEGER NOT NULL,
```

```
    session CHAR(10),
```

```
    PRIMARY KEY (courseId, email, documentId),
```

```
    FOREIGN KEY (courseId) REFERENCES Course (courseId)
```

```
    ON DELETE CASCADE ON UPDATE CASCADE,
```

```
    FOREIGN KEY (email,documentId) REFERENCES Transcript (email,documentId)
```

```
    ON DELETE CASCADE ON UPDATE CASCADE,
```

```
    FOREIGN KEY (percentage) REFERENCES Grade (percentage));
```

```
INSERT INTO TranscriptContains(courseId,email,documentId,percentage,session)
```

```
VALUES (22200, "golde0815@gmail.com", 21000,89,"2020W1");
```

```
INSERT INTO TranscriptContains(courseId,email,documentId,percentage,session)
```

```
VALUES (32300, "manred@gmail.com", 22000,89,"2020W2");
```

```
INSERT INTO TranscriptContains(courseId,email,documentId,percentage,session)
```

```
VALUES (40000, "valid@gmail.com", 23000,89,"2021W1");
```

```
INSERT INTO TranscriptContains(courseId,email,documentId,percentage,session)
```

University of British Columbia, Vancouver

Department of Computer Science

```
VALUES (29910, "ygim1509@gmail.com", 24000,89,"2021W2");
```

```
INSERT INTO TranscriptContains(courseId,email,documentId,percentage,session)
```

```
VALUES (32200, "my123@gmail.com", 25000,89,"2021W2");
```