Question 2

An updated ER diagram is attached. Changes include:

- Adding “partial” and “disjoint” constraints to our ISA relationship.
- Adding “jobTitle” and “baseSalary” attributes to our “JobPost” entity.
- Adding “province”, “city”, and “postalCode” attributes to our Company entity.
- Prefixing attributes with identical names to make each attribute unique (e.g. User.name and Company.name are now User.uname and Company.cname, respectively).

Question 3

Entities

Notes:

- Underline means primary keys, bold means foreign keys.
- For each relation, there are no other candidate keys other than their primary key.

User(uname: string, age: int, uid: int, iid: int) - note: iid must be unique
Recruiter(**uid**: int, passEndDate: date)
Message(mid: int, mbody: string, date: date, **uid**: int) - note: uid cannot be null
WorkExperience(etitle: string, **uid**: int, startDate: date, endDate: date)
Company(cname: string, province: string, city: string, postal code: string)
Hashtag(hname: string)
JobPost(pid: int, pbody: string, jobTitle: string, baseSalary: int, **uid**: int)

Combined HasInbox and Inbox with User
Combined SentBy into Message
Combined Publish with JobPost
Combined HasExperience with Work Experience

Relationships

Contains(**iid**: int, **mid**: int)
HasHashtag(**pid**: int, **hname**: string)
HiringCompany(**pid**: int, **cname**: string) - note: need assertion for participation constraint on Job Post
WorkedAt(**cname**: string, **startDate**: date, **uid**: int) - note: need assertion for participation constraint on Work Experience
Apply(**uid**: int, **pid**: int)
ReceivedBy(**mid**: int, **uid**: int)

Question 4

uid -> age
uid -> uname
uid -> passEndDate
mid -> mbody
mid -> date
uid, startDate -> etitle
uid, startDate -> endDate
cname -> province
cname -> city
cname -> postalCode
postalCode -> province
postalCode -> city
pid -> pbody
pid -> jobTitle
pid -> baseSalary
jobTitle -> baseSalary

Company(cname, province, city, postalCode)
postalCode -> city
postalCode -> province
cname -> province, city, postalCode

cname+ = {cname, province, city, postalCode}
postalCode+ = {postalCode, city, province}

“postalCode -> city” violates BCNF since postalCode is not a superkey.

Decompose Company on “postalCode -> City”
R1(postalCode, city), R2(cname, postalCode, province)

R1 is good because it is a 2-attribute relation. R2 still violates BCNF because “postalCode -> province” holds but postalCode is not a superkey for R2.

Decompose R2 on “postalCode -> province”
R3(postalCode, province), R4(postalCode, cname)

Both of the above relations are good because they’re both 2-attribute entities. Therefore, our final decomposition of Company is:

R1(postalCode, city), R3(postalCode, province), R4(postalCode, cname)

JobPost(pid: int, pbody: string, **uid**: int, jobTitle: char, baseSalary: int)
jobTitle -> baseSalary
pid -> pbody, jobTitle, baseSalary

jobTitle+ = {jobTitle, baseSalary}
pid+ = {pid, pbody, jobTitle, baseSalary, **uid**}

JobPost violates BCNF because the FD “jobTitle -> baseSalary” holds, but jobTitle is not a superkey.

Decompose JobPost on “jobTitle -> baseSalary”
R1(jobTitle, baseSalary), R2(pid, pbody, jobTitle, **uid**)

R1 is good because it is a 2-attribute entity. R2 is also good because the only functional dependency that holds involves pid and it is a superkey. Therefore, our final decomposition of JobPost is:

R1(jobTitle, baseSalary), R2(pid, pbody, jobTitle, **uid**)

Question 5

Entities

Notes:

- Underline means primary keys, bold means foreign keys.
- For each relation, there are no other candidate keys other than their primary key.

User(uname: string, age: int, uid: int, iid: int) - note: iid must be unique
Recruiter(**uid**: int, passEndDate: date)
Message(mid: int, mbody: string, date: date, **uid**: int) - note: uid cannot be null
WorkExperience(etitle: string, **uid**: int, startDate: date, endDate: date)
Company(cname: string, postalCode: string)
Municipality(postalCode: string, city: string)
District(postalCode: string, province: string)
Hashtag(hname: string)
JobPost(pid: int, pbody: string, jobTitle: string, **uid**: int)
JobDesc(jobTitle: string, baseSalary: int)

Combined HasInbox and Inbox with User
Combined SentBy into Message

Combined Publish with JobPost
HasExperience combined with Work Experience

Relationships

Contains(**iid**: int, **mid**: int)
HasHashtag(**pid**: int, **hname**: string)
HiringCompany(**pid**: int, **cname**: string) - note: need assertion for participation constraint on Job Post
WorkedAt(**cname**: string, **startDate**: date, **uid**: int) - note: need assertion for participation constraint on Work Experience
Apply(**uid**: int, **pid**: int)
ReceivedBy(**mid**: int, **uid**: int)

Question 6

```
CREATE TABLE User(
    uname CHAR(40),
    age INT,
    uid INT,
    iid INT,
    PRIMARY KEY (uid),
    UNIQUE (iid)
)

CREATE TABLE Recruiter(
    uid INT,
    passEndDate DATE,
    PRIMARY KEY(uid),
    FOREIGN KEY(uid) REFERENCES User
        ON DELETE CASCADE
        ON UPDATE CASCADE
)


CREATE TABLE Message(
    mid INT,
    mbody CHAR(2000),
    date DATE,
    uid INT NOT NULL,
    PRIMARY KEY(mid),
    FOREIGN KEY(uid) REFERENCES User
        ON DELETE NO ACTION
```

```
                ON UPDATE CASCADE
)

CREATE TABLE WorkExperience(
        etitle CHAR(200),
        uid INT,
        startDate DATE,
        endDate DATE,
        PRIMARY KEY(uid, startDate),
        FOREIGN KEY(uid) REFERENCES User
                ON DELETE CASCADE
                ON UPDATE CASCADE
)

CREATE TABLE Company(
        cname CHAR(200),
        postalCode CHAR(20),
        PRIMARY KEY(cname)
)


CREATE TABLE Municipality(
        postalCode CHAR(20),
        city CHAR(60),
        PRIMARY KEY(postalCode)
)

CREATE TABLE District(
        postalCode CHAR(20),
        province CHAR(60),
        PRIMARY KEY(postalCode)
)

CREATE TABLE Hashtag(
        hname CHAR(40),
        PRIMARY KEY(hname)
)

CREATE TABLE JobPost(
        pid INT,
        pbody CHAR(400),
        jobTitle CHAR(60),
        uid INT,
        PRIMARY KEY(pid),
    FOREIGN KEY(uid) REFERENCES User
        ON DELETE CASCADE
        ON UPDATE CASCADE
)

CREATE TABLE JobDesc(
    jobTitle CHAR(60),
    baseSalary INT,
    PRIMARY KEY(jobTitle)
)

CREATE TABLE Contains(
    iid: INT,
    mid: INT,
    PRIMARY KEY (iid, mid),
    FOREIGN KEY (iid) REFERENCES User
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (mid) REFERENCES Message
        ON DELETE CASCADE
        ON UPDATE CASCADE
)

CREATE TABLE HasHashtag(
    pid: INT,
    hname: CHAR(20),
    PRIMARY KEY (pid, hname),
    FOREIGN KEY (pid) REFERENCES JobPost
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (hname) REFERENCES Hashtag
        ON DELETE CASCADE
        ON UPDATE CASCADE
)

CREATE TABLE HiringCompany(
    pid: INT,
    cname: CHAR(200),
    PRIMARY KEY (pid, cname),
    FOREIGN KEY (pid) REFERENCES JobPost
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (cname) REFERENCES Company
        ON DELETE CASCADE
        ON UPDATE CASCADE
)

CREATE TABLE WorkedAt(
    cname: CHAR(200),
    startDate: DATE,
    uid: INT,
    PRIMARY KEY (cname, startDate, uid),
    FOREIGN KEY (cname) REFERENCES Company
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (uid, startDate) REFERENCES WorkExperience(uid, startDate)
        ON DELETE CASCADE
        ON UPDATE CASCADE
)

CREATE TABLE Apply(
    uid: INT,
    pid: INT,
    PRIMARY KEY (uid, pid),
    FOREIGN KEY (uid) REFERENCES User
        ON DELETE CASCADE
        ON UPDATE CASCADE
    FOREIGN KEY (pid) REFERENCES JobPost
        ON DELETE CASCADE
        ON UPDATE CASCADE
)

CREATE TABLE ReceivedBy(
    mid: INT,
    uid: INT,
    PRIMARY KEY (mid, uid),
    FOREIGN KEY (mid) REFERENCES JobPost
        ON DELETE CASCADE
        ON UPDATE CASCADE
    FOREIGN KEY (uid) REFERENCES User
        ON DELETE CASCADE
        ON UPDATE CASCADE
)
```

Question 7

User:

uname	uid	age	iid
Colton Quan	1	20	1
Kevin Nguyen	2	21	2
Amy Yung	3	22	3
John Doe	4	23	4
Aubrey Graham	5	24	5
Bruce Wayne	6	25	6
Eveline Saiful	7	26	7
Sebastian Vettel	8	27	8
Lebron James	9	28	9
Christian Pulisic	10	29	10

Recruiter:

uid	passEndDate
6	April 30, 2022
7	April 30, 2022
8	April 30, 2022
9	April 30, 2022
10	April 30, 2022

Message:

mid	mbody	date	uid
001	“Lmao hilarious”	Oct 5th, 2021	4
041	“Please give me a job I’m begging you”	Oct 21st, 2021	3

021	“I’m so interested in the work you do - please connect with me I need validation”	Sept 10th, 2021	2
024	“Temp_x is a valid way to name your variables - I don’t care what you say”	Oct 23rd, 2021	2
049	“I am a naming master - trust me I worked at amazon	Oct 23rd, 2021	1

WorkExperience:

etitle	uid	startDate	endDate
Frontend developer	1	Jan 1, 2021	April 30, 2021
Backend developer	2	Feb 1, 2021	May 30, 2021
Fullstack developer	3	March 1, 2021	June 30, 2021
UI/UX designer	4	April 1, 2021	July 30, 2021
QA developer	5	May 1, 2021	Aug 30, 2021

Company:

cname	postalCode
Railtown AI	V6C 3E8
IBM	V5G 4X3
Amazon	V6B 0M3
Visier	V6B 1C1
Semios	V5T 4T5

Municipality:

postalCode	city

V6C 3E8	Vancouver
V5G 4X3	Burnaby
V6B 0M3	Vancouver
V6B 1C1	Vancouver
V5T 4T5	Vancouver

District:

postalCode	province
V6C 3E8	BC
V5G 4X3	BC
P1L 2R3	ON
T7V 8J4	AB
E7H 5P5	NB

Hashtag:

hname
#WashedKing
#DataScience
#motivation
#ImHired
#ApplyNow

JobPost:

pid	jobTitle	pbody
1	Data Analyst	“HIRING! URGENT! But we will ghost you”

2	Data Scientist	“Looking for Data Scientist Intern for Summer 2022 Term”
3	Social Media Intern	“Hiring Social Media Intern, no experience needed”
4	Backend Developer	“Are you looking for a career change in tech? Apply to this position now! No experience required.”
5	UI/UX Designer	“We are hiring for a UI/UX designer position. If you are a creative person who loves sharing your artistic vision, consider dropping your resume below.”

JobDesc:

jobTitle	baseSalary
Data Analyst	$80,000
Data Scientist	$80,000
Social Media Intern	$10,000
Backend Developer	$80,000
UI/UX Designer	$60,000

Contains:

iid	mid
1	001
2	041
3	021
4	024
5	049

HasHashtag:

pid	hname
1	#WashedKing
2	#DataScience
3	#motivation
4	#ImHired
5	#ApplyNow

HiringCompany:

pid	cname
1	Railtown AI
2	IBM
3	Amazon
4	Visier
5	Semios

WorkedAt:

cname	startDate	uid
Railtown AI	Jan 1, 2021	1
IBM	Feb 1, 2021	2
Amazon	March 1, 2021	3
Visier	April 1, 2021	4
Semios	May 1, 2021	5

Apply:

uid	pid
1	1
2	2
3	3
4	4
5	5

ReceivedBy:

mid	uid
001	5
041	4
021	3
024	2
049	1

Question 8

We have not covered: projection, join, aggregation with group by, aggregation with having, nested aggregation with group by and division, so we cannot list the queries for them.

Insertion:

- Add users to user table
- Add recruiters to recruiter table
- Add messages to message table
- Add work experiences to workExperience table
- Add companies to company table
- Add cities (and their associated postal code) to municipality table
- Add provinces (and their associated postal code) to district table
- Add hashtags to hashtag table
- Add job postings to jobPost table

- Add job descriptions to jobDesc table
- Add inbox ID and associated message ID to contains table
- Add hashtag and their associated posting ID to hasHashtag table
- Add job posting id and their associated company name to hiringCompany table
- Add company name and associated ID to workedAt table
- Add job posting id and user id to apply table
- Add message id and recipient user id to receivedBy table

Delete:

- Remove users from user table
- Remove recruiters from recruiter table
- Remove messages from message table
- Remove work experiences from workExperience table
- Remove companies from company table
- Remove cities (and their associated postal code) from municipality table
- Remove provinces (and their associated postal code) from district table
- Remove hashtags from hashtag table
- Remove job postings from jobPost table
- Remove job descriptions from jobDesc table
- Remove inbox ID and associated message ID from contains table on update and on delete
- Remove hashtag and their associated posting ID from hasHashtag table on update and on delete
- Remove job posting id and their associated company name from hiringCompany table on update and on delete
- Remove company name and associated ID from workedAt table on update and on delete
- Remove job posting id and user id from apply table on update and on delete
- Remove message id and recipient user id from receivedBy table on update and on delete

Update:

- Update a user's age
- Update a recruiter's pass end date (in case they decide to renew)
- Update a work experience's start and end date
- Update a company's name and postal code
- Update a job posting's title and body
- Update base salary in job description
- Update start date in worked at table

Select:

- Companies can select users who've worked at certain companies before in the past
- Users can select job postings from a certain company
- Users can select job postings by position title

- Users can select job postings by companies’ location (province and city)
- Companies/users/recruiters can select users based on name
- Users can select job postings by hashtags
- Companies can select users by age (have to be older than 19 to work at a bar for example)