University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 2

Date: July 27, 2022

Group Number: 9

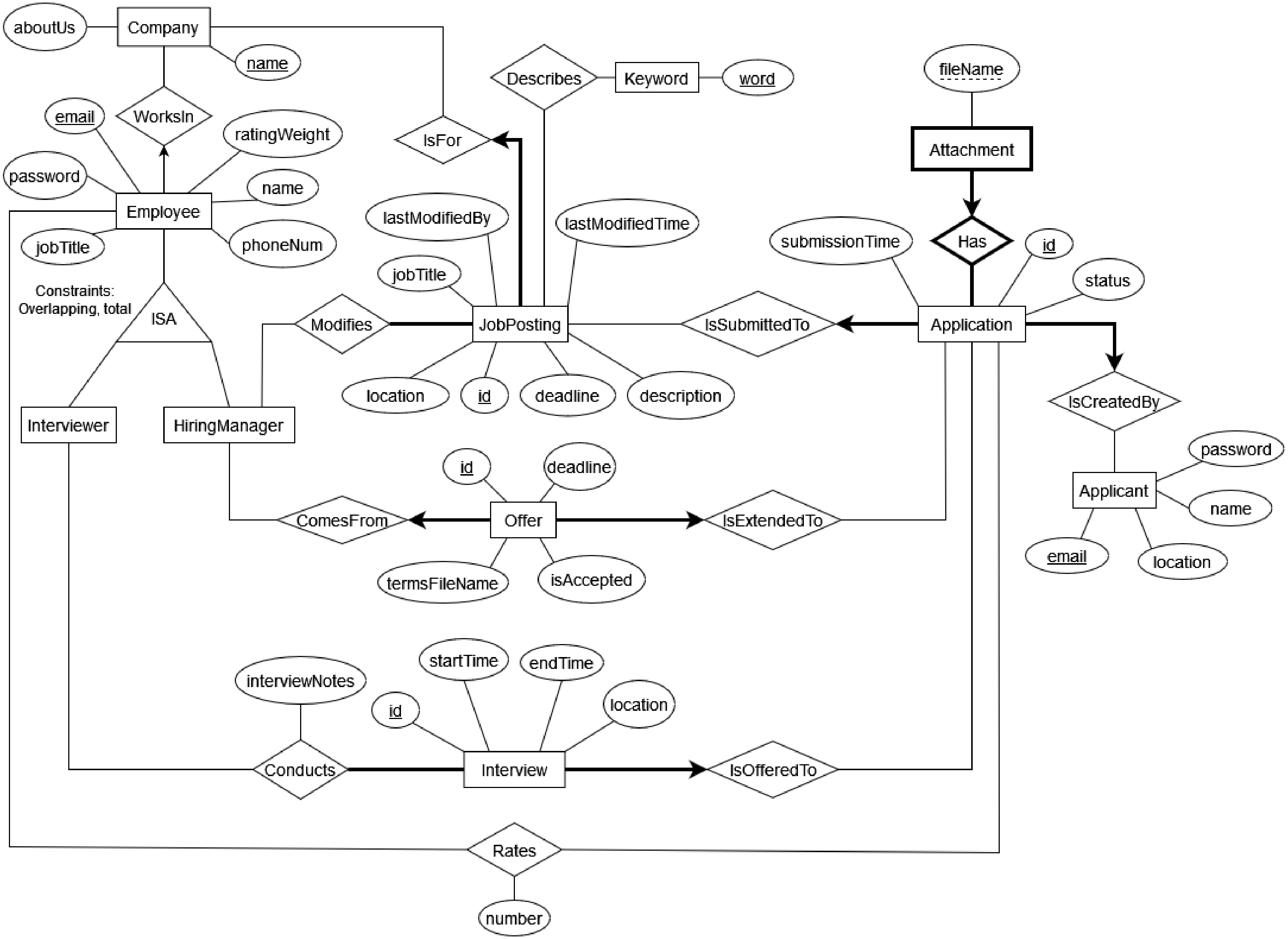
Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Eugene Chau	32451429	g2w2b	eugene.chau01@gmail.com
Martin Lou	35042712	l7r3p	martinlou2002@gmail.com
Jacqueline (Wai Ting) Chan	89652028	u6w2b	jacquelinechan09@gmail.com

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

The ER Diagram

The ER diagram we are using for milestone 2 is shown below.



The ER diagram has changed from milestone 1 as follows:

1. Multiple relationship names have been renamed for clarity. The changes are shown in the table below.

Entity sets involved in relationship set	**Old relationship name**	**New relationship name**
HiringManager, Offer	Extends	ComesFrom
Offer, Application	Receives	IsExtendedTo
Application, Applicant	Creates	IsCreatedBy
Interview, Application	IsSelectedFor	IsOfferedTo

2. The following changes to attributes have been made:

Entity set	**Changes**	**Reason/Details**
JobPosting	Added “location”, “lastModifiedBy”, and “lastModifiedTime” attributes	Applicants should know the location of a job. Hiring managers may find it useful to know who most recently edited a job posting and when.
Employee	Added “ratingWeight” attribute	Some employees’ ratings (e.g. those of senior employees) should be treated as more important. Rating weights are assigned by hiring managers, and the default is 1. Every employee with the same job title in the same company has the same weight.
Company	Added “aboutUs” attribute	This is a text field where a company can describe what they do. Applicants would find it helpful to see this information.
Applicant	Changed “postalCode” to “location”	It is more flexible for an applicant to input a general location rather than a postal code.
Application	Changed “submissionDate” to “submissionTime”	Recording the timestamp gives better precision.
Attachment	Removed “fileContent” attribute	Working with files in database columns is fairly awkward. We therefore plan to store each attachment in the local file system, at a path which is determined by the application ID and the attachment name.

Offer	Replaced “termsFile” with “termsFileName”	Same reason as the change to the “Attachment” entity

3. We add an “IsFor” relationship between JobPosting and Company, so that we can keep track of what company each job posting is for. This will allow us to enforce that only people in the same company can modify a job posting or view the applications submitted to it.
4. We remove the key constraint on Application in the “Offer IsExtendedTo Application” relationship. One application could receive multiple offers if negotiations happen.
5. We remove the total participation constraint on Employee in the “Employee WorksFor Company” relationship. It is possible that an employee leaves a company.

Translation to the Relational Model

The translation of our ER diagram to the relational model is shown below. Primary keys are underlined, and foreign keys are bolded. **Note**: the names of foreign keys are formed by concatenating the name of the relation being referenced and the name of the attribute being referenced.

We do not have any candidate keys other than the primary keys.

Applicant(email:CHAR(80), password:CHAR(80), name:CHAR(80), location:CHAR(80))

- All attributes must be non-null.

Company(name:CHAR(80), aboutUs:CHAR(4000))

- All attributes must be non-null. The aboutUs attribute defaults to an empty string.

Employee_WorksIn(email:CHAR(80), password:CHAR(80), name:CHAR(80), ratingWeight:REAL, phoneNum:CHAR(20), **companyName**:CHAR(80), jobTitle:CHAR(80))

- All attributes except for phoneNum and companyName must be non-null.

HiringManager(**email**:CHAR(80))

Interviewer(**email**:CHAR(80))

JobPosting_IsFor(id:INTEGER, deadline:TIMESTAMP, description:CHAR(4000), location:CHAR(80), jobTitle:CHAR(80), lastModifiedBy:CHAR(170), lastModifiedTime:TIMESTAMP, **companyName**:CHAR(80))

- All attributes must be non-null. The description, if unspecified, defaults to an empty string.

Keyword(word:CHAR(80))

Application_IsCreatedBy_IsSubmittedTo(id:INTEGER, submissionTime:TIMESTAMP, status:CHAR(80), **applicantEmail**:CHAR(80), **jobPostingId**:INTEGER)

- All attributes must be non-null. Status defaults to “Submitted”.

Attachment(fileName:CHAR(80), **applicationId**:INTEGER)

Offer_ComesFrom_IsExtendedTo(id:INTEGER, deadline:TIMESTAMP, termsFileName:CHAR(80), isAccepted:BOOLEAN, **hiringManagerEmail**:CHAR(80), **applicationId**:INTEGER)

- All attributes must be non-null, except for hiringManagerEmail. Normally, hiringManagerEmail is not null either, but in case of deletion, we set the email to NULL by default.

Interview_IsOfferedTo(id:INTEGER**,** startTime:TIMESTAMP, endTime:TIMESTAMP, location:CHAR(80), **applicationId**:INTEGER)

- All attributes must be non-null.

Conducts(**interviewerEmail**:CHAR(80), **interviewId**:INTEGER, interviewNotes:CHAR(500))

- All attributes must be non-null. The interviewNotes defaults to an empty string.

Describes(**keywordWord**:CHAR(80), **id**:INTEGER)

Modifies(**hiringManagerEmail**:CHAR(80), **jobPostingId**:INTEGER)

Rates(**employeeEmail**:CHAR(80), **applicationId**:INTEGER, number:INTEGER)

- All attributes must be non-null.

Functional Dependencies

Some relations have no non-trivial functional dependencies; in this case, we simply write “N/A”.

Company:

- name -> aboutUs

Employee_WorksIn:

- email -> password, name, ratingWeight, phoneNum, jobTitle, companyName
- companyName, jobTitle -> ratingWeight

Interviewer: N/A

HiringManager: N/A

Conducts:

- hiringManagerEmail, interviewId -> interviewNotes

Interview_IsOfferedTo:

- id -> startTime, endTime, location, applicationId

Rates:

- hiringManagerEmail, applicationId -> number

Application_IsCreatedBy_IsSubmittedTo:

- id -> status, submissionTime, applicantEmail, jobPostingId

Attachment: N/A

JobPosting_IsFor:

- id -> lastModifiedBy, lastModifiedTime, companyName, jobTitle, location, deadline, description
- lastModifiedBy -> companyName

Modifies: N/A

Describes: N/A

Keyword: N/A

Offer_ComesFrom_IsExtendedTo:

- id -> deadline, termsFileName, isAccepted, hiringManagerEmail, applicationId

Applicant:

- email -> password, name, location

Normalization

We now proceed to normalization. We have two relations that require normalization: JobPosting_IsFor and Employee_WorksIn.

Observe that JobPosting_IsFor has the non-trivial FD lastModifiedBy -> companyName, where lastModifiedBy is not a superkey, and companyName is not part of any key. Hence, JobPosting_IsFor is not in 3NF, and we decompose the relation as follows:

JobPostingModifier(<u>lastModifiedBy</u>:CHAR(170), **companyName**:CHAR(80))

JobPosting(<u>id</u>:INTEGER, deadline:TIMESTAMP, description:CHAR(4000), location:CHAR(80), jobTitle:CHAR(80), **lastModifiedBy**:CHAR(80), lastModifiedTime:TIMESTAMP)

Now, both JobPostingModifier and JobPosting are in BCNF.

We turn to Employee_WorksIn. The FD companyName, jobTitle -> ratingWeight violates 3NF because {companyName, jobTitle} is not a superkey, and ratingWeight is not part of a key. Hence, Employee_WorksIn is not in 3NF. We decompose the table into:

RatingWeights(**companyName**:CHAR(80), jobTitle:CHAR(80), ratingWeight:REAL)

Employee(email:CHAR(80), password:CHAR(80), name:CHAR(80), phoneNum:CHAR(20), **jobTitle**:CHAR(80), **companyName**:CHAR(80))

Notice that RatingWeights and Employee are both in BCNF now. This completes the normalization process.

To summarize, here are all of our tables, in the same format as step 3. There are no candidate keys other than primary keys.

Applicant(email:CHAR(80), password:CHAR(80), name:CHAR(80), location:CHAR(80))

- All attributes must be non-null.

Company(name:CHAR(80), aboutUs:CHAR(4000))

- All attributes must be non-null. The aboutUs attribute defaults to an empty string.

RatingWeights(**companyName**:CHAR(80), jobTitle:CHAR(80), ratingWeight:REAL)

- All attributes must be non-null.

Employee(email:CHAR(80), password:CHAR(80), name:CHAR(80), phoneNum:CHAR(20), **jobTitle**:CHAR(80), **companyName**:CHAR(80))

- All attributes except for phoneNum and companyName must be non-null.

HiringManager(**email**:CHAR(80))

Interviewer(**email**:CHAR(80))

JobPostingModifier(lastModifiedBy:CHAR(170), **companyName**:CHAR(80))

- All attributes must be non-null.

JobPosting(id:INTEGER, deadline:TIMESTAMP, description:CHAR(4000), location:CHAR(80), jobTitle:CHAR(80), **lastModifiedBy**:CHAR(80), lastModifiedTime:TIMESTAMP)

- All attributes must be non-null. The description, if unspecified, defaults to an empty string.

Keyword(word:CHAR(80))

Application_IsCreatedBy_IsSubmittedTo(id:INTEGER, submissionTime:TIMESTAMP, status:CHAR(80), **applicantEmail**:CHAR(80), **jobPostingId**:INTEGER)

- All attributes must be non-null. Status defaults to “Submitted”.

Attachment(fileName:CHAR(80), **applicationId**:INTEGER)

Offer_ComesFrom_IsExtendedTo(id:INTEGER, deadline:TIMESTAMP, termsFileName:CHAR(80), isAccepted:BOOLEAN, **hiringManagerEmail**:CHAR(80), **applicationId**:INTEGER)

- All attributes must be non-null, except for hiringManagerEmail. Normally, hiringManagerEmail is not null either, but in case of deletion, we set the email to NULL by default.

Interview_IsOfferedTo(id:INTEGER**,** startTime:TIMESTAMP, endTime:TIMESTAMP, location:CHAR(80), **applicationId**:INTEGER)

- All attributes must be non-null.

Conducts(**interviewerEmail**:CHAR(80), **interviewId**:INTEGER, interviewNotes:CHAR(500))

- All attributes must be non-null. The interviewNotes defaults to an empty string.

Describes(**keywordWord**:CHAR(80), **id**:INTEGER)

Modifies(**hiringManagerEmail**:CHAR(80), **jobPostingId**:INTEGER)

Rates(**employeeEmail**:CHAR(80), **applicationId**:INTEGER, number:INTEGER)

- All attributes must be non-null.

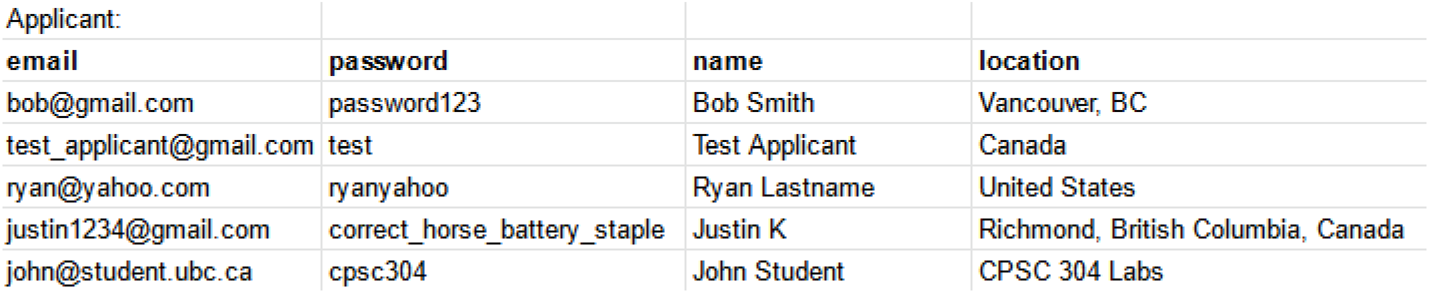
SQL DDL

```
CREATE TABLE Applicant(
   email CHAR(80) PRIMARY KEY,
   password CHAR(80) NOT NULL,
   name CHAR(80) NOT NULL,
   location CHAR(80) NOT NULL
);

CREATE TABLE Company(
   name CHAR(80) PRIMARY KEY,
   aboutUs CHAR(4000) NOT NULL DEFAULT ''
);
```

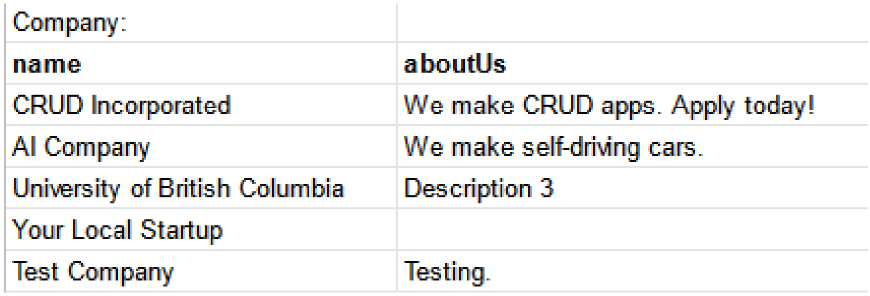
```
CREATE TABLE RatingWeights(
    companyName CHAR(80) NOT NULL,
    jobTitle CHAR(80) NOT NULL,
    ratingWeight REAL NOT NULL DEFAULT 1,
    PRIMARY KEY (companyName, jobTitle),
    FOREIGN KEY (companyName) REFERENCES Company(name)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

CREATE TABLE Employee(
    email CHAR(80) PRIMARY KEY,
    password CHAR(80) NOT NULL,
    name CHAR(80) NOT NULL,
    phoneNum CHAR(20),
    companyName CHAR(80),
    jobTitle CHAR(80) NOT NULL,
    FOREIGN KEY (companyName) REFERENCES Company(name)
        ON DELETE SET NULL
        ON UPDATE CASCADE,
    FOREIGN KEY (companyName, jobTitle) REFERENCES RatingWeights(companyName,
jobTitle)
        ON UPDATE CASCADE
);
CREATE TABLE HiringManager(
    email CHAR(80) PRIMARY KEY,
    FOREIGN KEY (email) REFERENCES Employee(email)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

CREATE TABLE Interviewer(
    email CHAR(80) PRIMARY KEY,
    FOREIGN KEY (email) REFERENCES Employee(email)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

CREATE TABLE JobPostingModifier(
    lastModifiedBy CHAR(170) NOT NULL,
    companyName CHAR(80) NOT NULL,
    PRIMARY KEY (lastModifiedBy),
    FOREIGN KEY (companyName) REFERENCES Company(name)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

CREATE TABLE JobPosting(
    id INTEGER PRIMARY KEY,
    deadline TIMESTAMP NOT NULL,
    description CHAR(4000) NOT NULL DEFAULT '',
    location CHAR(80) NOT NULL,
    jobTitle CHAR(80) NOT NULL,
    lastModifiedBy CHAR(170) NOT NULL,
    lastModifiedTime TIMESTAMP NOT NULL,
    FOREIGN KEY (lastModifiedBy) REFERENCES JobPostingModifier(lastModifiedBy)
        ON UPDATE CASCADE
);

CREATE TABLE Keyword(
    word CHAR(30) PRIMARY KEY
);

CREATE TABLE Application_IsCreatedBy_IsSubmittedTo(
    id INTEGER PRIMARY KEY,
    submissionTime TIMESTAMP NOT NULL,
    status CHAR(80) NOT NULL DEFAULT 'Submitted',
    applicantEmail CHAR(80) NOT NULL,
    jobPostingId INTEGER NOT NULL,
    FOREIGN KEY (applicantEmail) REFERENCES Applicant(email)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (jobPostingId) REFERENCES JobPosting(id)
        ON DELETE CASCADE
);

CREATE TABLE Attachment(
    fileName CHAR(80) NOT NULL,
    applicationId INTEGER NOT NULL,
    PRIMARY KEY (fileName, applicationId),
    FOREIGN KEY (applicationId) REFERENCES Application_IsCreatedBy_IsSubmittedTo(id)
        ON DELETE CASCADE
);

CREATE TABLE Offer_ComesFrom_IsExtendedTo(
    id INTEGER PRIMARY KEY,
    deadline TIMESTAMP NOT NULL,
    termsFileName CHAR(80) NOT NULL,
    isAccepted BOOLEAN NOT NULL DEFAULT FALSE,
    hiringManagerEmail CHAR(80),
    applicationId INTEGER NOT NULL,
    FOREIGN KEY (hiringManagerEmail) REFERENCES HiringManager(email)
        ON DELETE SET NULL
        ON UPDATE CASCADE,
    FOREIGN KEY (applicationId) REFERENCES Application_IsCreatedBy_IsSubmittedTo(id)
        ON DELETE CASCADE
);

CREATE TABLE Interview_IsOfferedTo(
    id INTEGER PRIMARY KEY,
    startTime TIMESTAMP NOT NULL,
    endTime TIMESTAMP NOT NULL,
    location CHAR(80) NOT NULL,
    applicationId INTEGER NOT NULL,
    FOREIGN KEY (applicationId) REFERENCES Application_IsCreatedBy_IsSubmittedTo(id)
        ON DELETE CASCADE
);

CREATE TABLE Conducts(
    interviewerEmail CHAR(80) NOT NULL,
    interviewId INTEGER NOT NULL,
    interviewNotes CHAR(500) NOT NULL DEFAULT '',
    PRIMARY KEY (interviewerEmail, interviewId),
    FOREIGN KEY (interviewerEmail) REFERENCES Interviewer(email)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (interviewId) REFERENCES Interview_IsOfferedTo(id)
        ON DELETE CASCADE
);

CREATE TABLE Describes(
    keywordWord CHAR(80) NOT NULL,
    jobPostingId INTEGER NOT NULL,
    PRIMARY KEY (keywordWord, jobPostingId),
    FOREIGN KEY (keywordWord) REFERENCES Keyword(word)
        ON DELETE CASCADE,
    FOREIGN KEY (jobPostingId) REFERENCES JobPosting(id)
        ON DELETE CASCADE
);

CREATE TABLE Modifies(
    hiringManagerEmail CHAR(80) NOT NULL,
    jobPostingId INTEGER NOT NULL,
    PRIMARY KEY (hiringManagerEmail, jobPostingId),
    FOREIGN KEY (hiringManagerEmail) REFERENCES HiringManager(email)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (jobPostingId) REFERENCES JobPosting(id)
        ON DELETE CASCADE
);

CREATE TABLE Rates(
    employeeEmail CHAR(80) NOT NULL,
    applicationId INTEGER NOT NULL,
    number INTEGER NOT NULL,
    PRIMARY KEY (employeeEmail, applicationId),
    FOREIGN KEY (employeeEmail) REFERENCES Employee(email)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (applicationId) REFERENCES Application_IsCreatedBy_IsSubmittedTo(id)
        ON DELETE CASCADE
);
```

Populating the Tables

Below, we show screenshots of an instance of each table. If a cell's value is the empty string, we leave it blank. If a cell's value is null, we say "[this field is left null]".



Applicant:			
email	**password**	**name**	**location**
bob@gmail.com	password123	Bob Smith	Vancouver, BC
test_applicant@gmail.com	test	Test Applicant	Canada
ryan@yahoo.com	ryanyahoo	Ryan Lastname	United States
justin1234@gmail.com	correct_horse_battery_staple	Justin K	Richmond, British Columbia, Canada
john@student.ubc.ca	cpsc304	John Student	CPSC 304 Labs



Company:	
name	**aboutUs**
CRUD Incorporated	We make CRUD apps. Apply today!
AI Company	We make self-driving cars.
University of British Columbia	Description 3
Your Local Startup	
Test Company	Testing.

RatingWeights:

companyName	jobTitle	ratingWeight
AI Company	Junior SWE	1
AI Company	Senior SWE	2.5
AI Company	Senior Manager	4
CRUD Incorporated	Software Tester	1
CRUD Incorporated	Software Developer	1
CRUD Incorporated	Associate Manager	3.2
CRUD Incorporated	Senior Manager	5
University of British Columbia	Research Assistant	0.5
University of British Columbia	Instructor	1.5
University of British Columbia	Professor	3
University of British Columbia	President	100
Your Local Startup	CEO	10
Your Local Startup	Intern	0.5
Test Company	Test Job 1	1
Test Company	Test Job 2	2

Employee:

email	password	name	phoneNum	companyName	jobTitle
michael@ai.com	password	Michael	604-583-6812	AI Company	Junior SWE
emily@ai.com	password2	Emily Lee	778-390-2045	AI Company	Senior Manager
donald@ai.com	password3	Donald Tan	[this field is left null]	AI Company	Senior Manager
tester@crud.org	qa_is_my_middle_name	John Doe	+1 (604) 238 4829	CRUD Incorporated	Software Tester
manager@crud.org	123456	John Doe Manager	[this field is left null]	CRUD Incorporated	Senior Manager
john@student.ubc.ca	a_different_password	John Student	123-456-7890	University of British Columbia	Research Assistant
cheeren@cs.ubc.ca	cpsc221_rocks	Cinda Heeren	604-822-9880	University of British Columbia	Professor
jzahl@math.ubc.ca	password	Joshua Zahl	[this field is left null]	University of British Columbia	Professor
pres.admin@ubc.ca	test	Santa J. Ono	[this field is left null]	University of British Columbia	President
ceo@startup.domain	ceo	Brandon T	(098) 765-4321	Your Local Startup	CEO
test_employee@test.com	test	Test Employee	[this field is left null]	Test Company	Test Job 1

HiringManager:

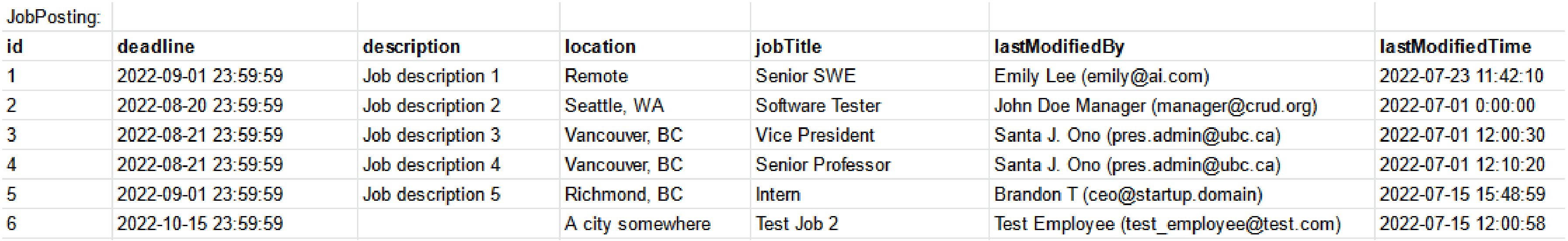
email
emily@ai.com
donald@ai.com
manager@crud.org
pres.admin@ubc.ca
ceo@startup.domain
test_employee@test.com

Interviewer:

email
michael@ai.com
emily@ai.com
tester@crud.org
manager@crud.org
john@student.ubc.ca
cheeren@cs.ubc.ca
jzahl@math.ubc.ca
ceo@startup.domain

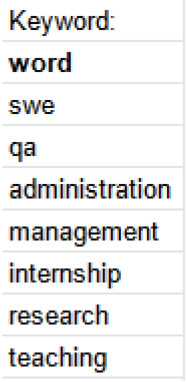
JobPostingModifier:

lastModifiedBy	companyName
Emily Lee (emily@ai.com)	AI Company
Donald Tan (donald@ai.com)	AI Company
John Doe Manager (manager@crud.org)	CRUD Incorporated
Santa J. Ono (pres.admin@ubc.ca)	University of British Columbia
Brandon T (ceo@startup.domain)	Your Local Startup
Test Employee (test_employee@test.com)	Test Company



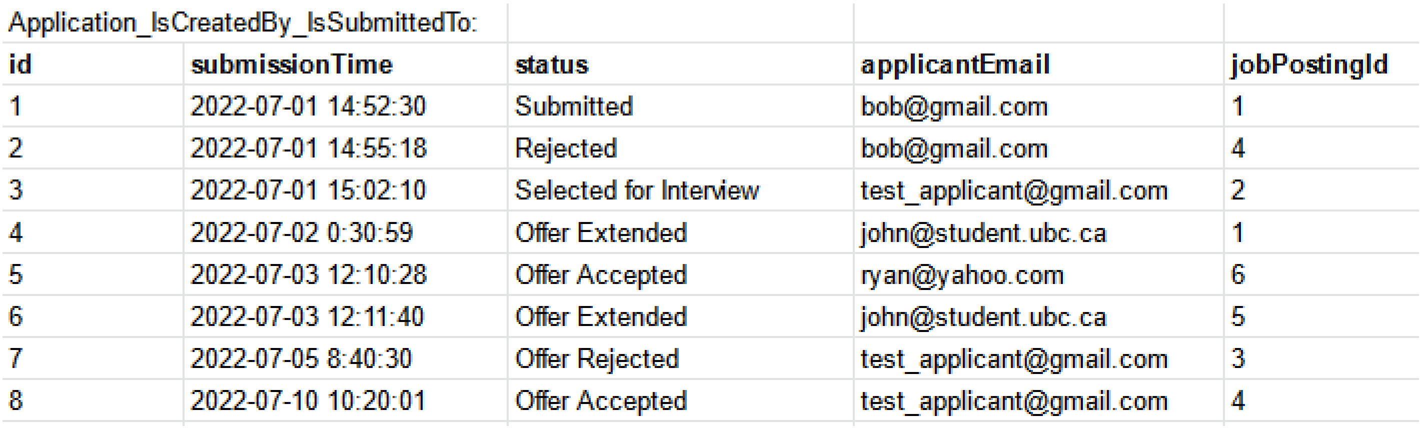
JobPosting:

id	deadline	description	location	jobTitle	lastModifiedBy	lastModifiedTime
1	2022-09-01 23:59:59	Job description 1	Remote	Senior SWE	Emily Lee (emily@ai.com)	2022-07-23 11:42:10
2	2022-08-20 23:59:59	Job description 2	Seattle, WA	Software Tester	John Doe Manager (manager@crud.org)	2022-07-01 0:00:00
3	2022-08-21 23:59:59	Job description 3	Vancouver, BC	Vice President	Santa J. Ono (pres.admin@ubc.ca)	2022-07-01 12:00:30
4	2022-08-21 23:59:59	Job description 4	Vancouver, BC	Senior Professor	Santa J. Ono (pres.admin@ubc.ca)	2022-07-01 12:10:20
5	2022-09-01 23:59:59	Job description 5	Richmond, BC	Intern	Brandon T (ceo@startup.domain)	2022-07-15 15:48:59
6	2022-10-15 23:59:59		A city somewhere	Test Job 2	Test Employee (test_employee@test.com)	2022-07-15 12:00:58



Keyword:

word
swe
qa
administration
management
internship
research
teaching



Application_IsCreatedBy_IsSubmittedTo:

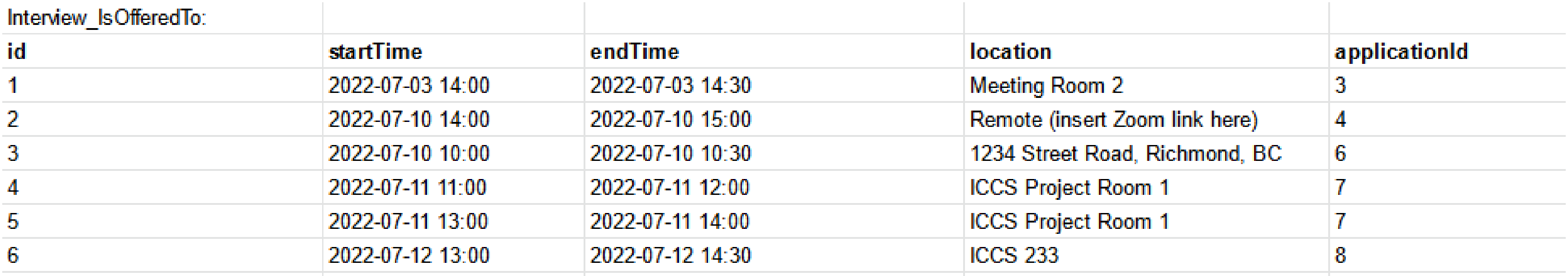
id	submissionTime	status	applicantEmail	jobPostingId
1	2022-07-01 14:52:30	Submitted	bob@gmail.com	1
2	2022-07-01 14:55:18	Rejected	bob@gmail.com	4
3	2022-07-01 15:02:10	Selected for Interview	test_applicant@gmail.com	2
4	2022-07-02 0:30:59	Offer Extended	john@student.ubc.ca	1
5	2022-07-03 12:10:28	Offer Accepted	ryan@yahoo.com	6
6	2022-07-03 12:11:40	Offer Extended	john@student.ubc.ca	5
7	2022-07-05 8:40:30	Offer Rejected	test_applicant@gmail.com	3
8	2022-07-10 10:20:01	Offer Accepted	test_applicant@gmail.com	4

Attachment:

fileName	applicationId
Bob_Resume.pdf	1
Bob_CoverLetter.pdf	1
Bob's Bad Resume.txt	2
Test_Resume.pdf	3
Test_CoverLetter.pdf	3
Test_References.pdf	3
John_Student_Resume.pdf	4
Ryan_CV.pdf	5
John_Student_Resume.pdf	6
John_Student_CoverLetter.pdf	6
Test_Resume.pdf	7
Test_Resume.pdf	8

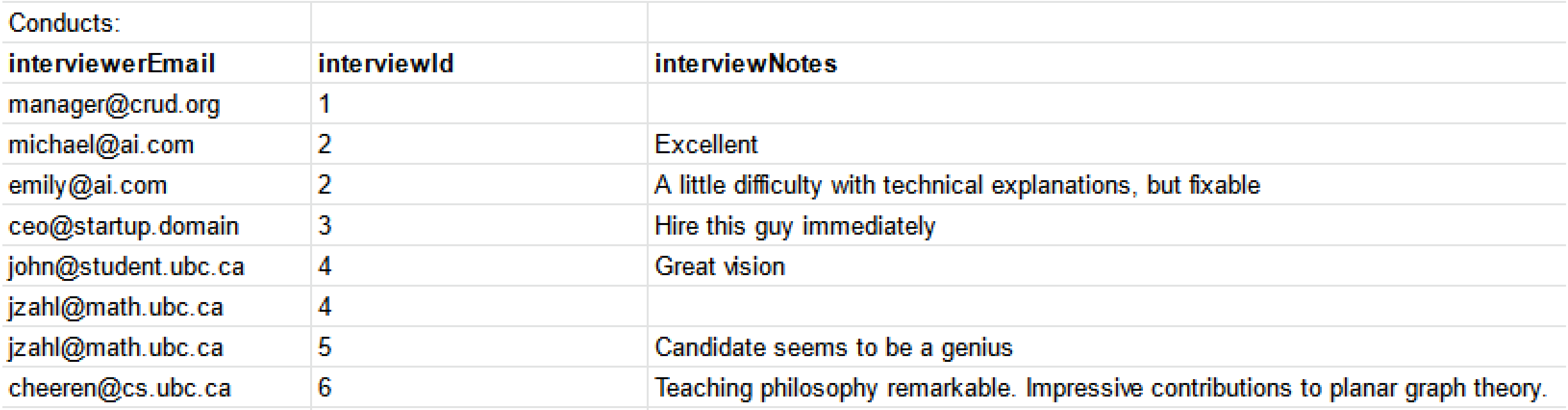
Offer_ComesFrom_IsExtendedTo:

id	deadline	termsFileName	isAccepted	hiringManagerEmail	applicationId
1	2022-07-15 23:59:59	UBC_Vice_President_Offer.pdf	FALSE	pres.admin@ubc.ca	7
2	2022-07-15 23:59:59	UBC_Senior_Professor_Offer.pdf	TRUE	pres.admin@ubc.ca	8
3	2022-07-15 23:59:59	AI_Company_Senior_SWE_Offer.pdf	FALSE	emily@ai.com	4
4	2022-09-01 23:59:59	AI_Company_Senior_SWE_Offer.pdf	FALSE	donald@ai.com	4
5	2022-09-01 23:59:59	Test_Job_2_Offer_Letter.pdf	TRUE	test_employee@test.com	5
6	2022-08-28 23:59:59	Startup_Internship_Offer.pdf	FALSE	ceo@startup.domain	6



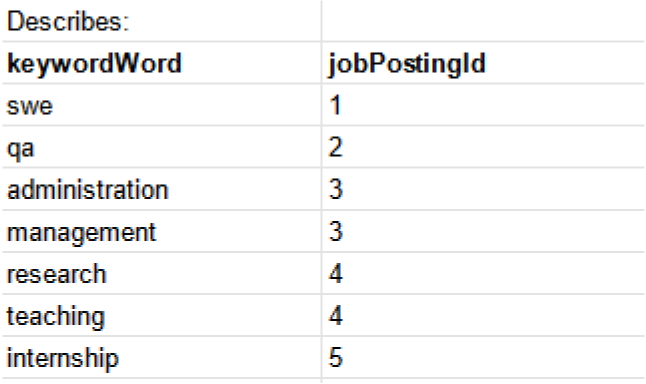
Interview_IsOfferedTo:

id	startTime	endTime	location	applicationId
1	2022-07-03 14:00	2022-07-03 14:30	Meeting Room 2	3
2	2022-07-10 14:00	2022-07-10 15:00	Remote (insert Zoom link here)	4
3	2022-07-10 10:00	2022-07-10 10:30	1234 Street Road, Richmond, BC	6
4	2022-07-11 11:00	2022-07-11 12:00	ICCS Project Room 1	7
5	2022-07-11 13:00	2022-07-11 14:00	ICCS Project Room 1	7
6	2022-07-12 13:00	2022-07-12 14:30	ICCS 233	8



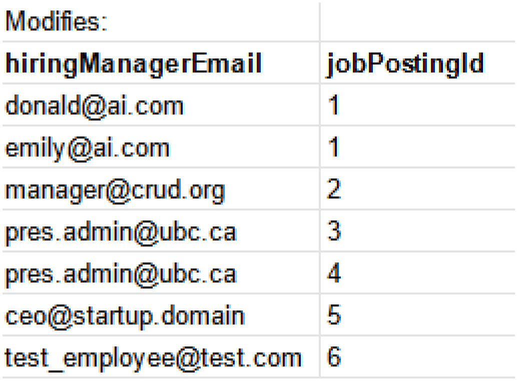
Conducts:

interviewerEmail	interviewId	interviewNotes
manager@crud.org	1	
michael@ai.com	2	Excellent
emily@ai.com	2	A little difficulty with technical explanations, but fixable
ceo@startup.domain	3	Hire this guy immediately
john@student.ubc.ca	4	Great vision
jzahl@math.ubc.ca	4	
jzahl@math.ubc.ca	5	Candidate seems to be a genius
cheeren@cs.ubc.ca	6	Teaching philosophy remarkable. Impressive contributions to planar graph theory.



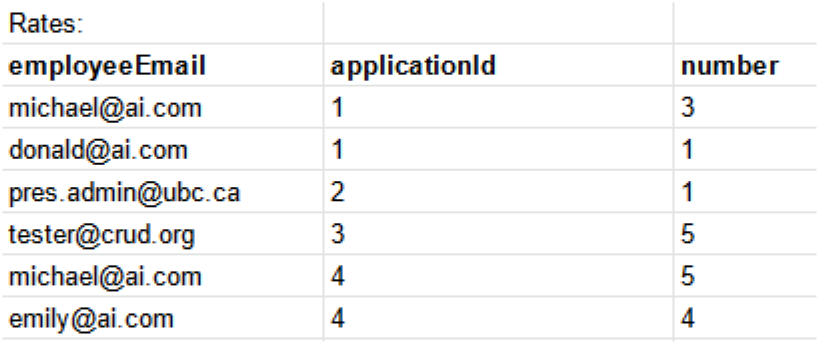
Describes:

keywordWord	jobPostingId
swe	1
qa	2
administration	3
management	3
research	4
teaching	4
internship	5



Modifies:

hiringManagerEmail	jobPostingId
donald@ai.com	1
emily@ai.com	1
manager@crud.org	2
pres.admin@ubc.ca	3
pres.admin@ubc.ca	4
ceo@startup.domain	5
test_employee@test.com	6



Rates:

employeeEmail	applicationId	number
michael@ai.com	1	3
donald@ai.com	1	1
pres.admin@ubc.ca	2	1
tester@crud.org	3	5
michael@ai.com	4	5
emily@ai.com	4	4