CPSC 304 Project Cover Page

Milestone #: 2

Date: July 27, 2022

Group Number: 14

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Patrick Conner	13781075	w3y6	pconner@student.ubc.ca
James Reinhardt	96645619	i5u3b	james.reinhardt222@gmail.com
Akriti Sharma	79884136	r0s7c	9akriti@gmail.com

ERD Changes

1. “Variant” weak entity added to allow for variations of preloaded questions.
2. “platform” attribute moved from “Recruiter” to “facilitated by”, as it is contingent on the relationship.
3. “streetAddress” attribute moved from “Company” to “Position” to better handle situations where a company changes address.
4. “StreetAddress” changed from attribute to entity to better describe its properties.
5. “text” & “title” attributes moved from “Coding”/”ShortAnswer” to “Question” as relevant to all subtypes.
6. “Question” PK changed to ID and title made unique. This is to allow easier reference between tables.
7. “Solution” PK changed to ID to allow for easier reference.
8. “Resources” PK changed to ID to allow for easier reference.
9. “Feedback” PK changed to ID to allow for easier reference.
10. “Solution” relationship now with “Coding” instead of “Question” as Question table not required due to Total/Disjoint quality of IsA relationship.
11. “Interview” relationship now with “Coding” & “Behavioral” instead of “Question” as Question table not required due to Total/Disjoint quality of IsA relationship.
12. Deleted relationship between “Interview” & “Position” as this relationship already exists indirectly.

Please see the last page for the updated ERD.

Schema, FDs & Normalization

Feedback(**userID**:int, **<u>appID</u>**:int, <u>feedID</u>:int, title:string, text:string, date:date) *(userID not null)*
FDs:
(PK, CK) feedID, appIP -> userID, title, text, date
appID->userID
Not in BCNF: Requires normalization since appID is not a superkey
Normalization Steps:

1. Split on the FD appID->userID
   Feedback1(<u>feedID</u>, **<u>appID</u>**, title, text, date)
   Feedback2(**<u>appID</u>**, **userID**)

```
Create Table Feedback1(
        feedID int(10),
        appID int(10),
        title varchar(200),
        text varchar(max),
        date  DATE,
        PRIMARY KEY (feedID, appID)
        FOREIGN KEY (appID) REFERENCES Application ON DELETE CASCADE
        )
INSERT INTO Feedback1 VALUES (01, 01, “Good job”, “Good job”, 7/25/2022);
INSERT INTO Feedback1 VALUES (02, 01, “Some advice...”, “Advice here”, 7/26/2022);
INSERT INTO Feedback1 VALUES (03, 02, “I like”, “:)”, 7/27/2022);
INSERT INTO Feedback1 VALUES (04, 03, “Pretty Bad”, “Try harder”, 7/27/2022);
INSERT INTO Feedback1 VALUES (05, 03, “What”, “what”, 7/27/2022);

Create Table Feedback2(
        appID int(10),
        userID int(10) not null,
        PRIMARY KEY (appID)
        FOREIGN KEY (userID) REFERENCES User ON DELETE SET DEFAULT
        FOREIGN KEY (appID) REFERENCES Feedback1 ON DELETE CASCADE
        );
INSERT INTO Feedback2  VALUES (01, 01);
INSERT INTO Feedback2  VALUES (01, 01);
INSERT INTO Feedback2  VALUES (02, 02);
INSERT INTO Feedback2  VALUES (03, 03);
INSERT INTO Feedback2  VALUES (03, 04);
```

User(<u>userID</u>:int, name:string, age:int, experience:int, education:string, about:string)
FD: (PK, CK) userID -> name, age, experience, education, about
In BCNF

```
Create Table User(
        userID int(10) DEFAULT 0,
        name varchar(100),
        age int(4),
        experience int(4),
        education varchar(max),
        about  varchar(max),
        PRIMARY KEY (userID)
        );
INSERT INTO User VALUES (01, “Patrick”, null, 0, “UBC”, null);
INSERT INTO User VALUES (02, “James”, null, 2, “UBC”, null);
INSERT INTO User VALUES (03, “Akriti”, null, 1, “UBC”, null);
INSERT INTO User VALUES (04, “Erika”, 21, 0, “UBC”, “Hello”);
INSERT INTO User VALUES (05, “Fred”, 22, 0, “UBC”, “I am Fred”);
```

Application(appID:int, **posID**:int, **userID**:int, resume:string, coverLetter:string, success?:boolean, date:date) *(posID, userID not null)*

FD: appID -> posID, userID, resume, coverLetter, success?, date

In BCNF

```
Create Table Application(
        appID int(10),
        posID  int(10) not null,
        userID int(10) not null,
        resume varchar(max),
        coverLetter varchar(max),
        success?  BOOLEAN,
        date DATE,
        PRIMARY KEY (appID)
        FOREIGN KEY (posID) REFERENCES Position
        FOREIGN KEY (userID) REFERENCES User ON DELETE SET DEFAULT
        );
INSERT INTO Application VALUES (01, 01, 01, “Resume...”, “Cover letter...”, False,
7/20/2022);
INSERT INTO Application VALUES (02, 01, 02, “Resume...”, “Cover letter...”, False,
7/22/2022);
INSERT INTO Application VALUES (03, 02, 01, “Resume...”, “Cover letter...”, True,
6/24/2022);
INSERT INTO Application VALUES (04, 04, 03, “Resume...”, “Cover letter...”, False,
3/06/2022);
INSERT INTO Application VALUES (05, 04, 03, “Resume...”, “Cover letter...”, False,
7/26/2022);
```

Interview(<u>intID</u>:int, **appID**:int, date:date, length:int, notes:string,
inPerson?:boolean, numInterviewers:int) *(appID not null)*
FD:
(PK) intID -> appID, date, length, notes, inPerson?, numInterviewers, receivedOffer
In BCNF

```
Create Table Interview1(
        intID int(10),
        appID int(10) not null,
        date DATE,
        length int(4),
        notes varchar(max),
        inPerson? BOOLEAN,
        numInterviewers int(4),
        PRIMARY KEY (intID)
        FOREIGN KEY (appID) REFERENCES Application ON DELETE SET DEFAULT
        );
INSERT INTO Interview1 VALUES (01, 01, 07/21/2022, 2, null, True, 1);
INSERT INTO Interview1 VALUES (02, 01,  07/21/2022, 2, null, True, 1);
INSERT INTO Interview1 VALUES (03, 02,  06/14/2022, 2, null, False, 2);
INSERT INTO Interview1 VALUES (04, 04,  07/27/2022, 2, null, False, 1);
INSERT INTO Interview1 VALUES (05, 05,  07/27/2022, 2, null, False, 1);
```

Position(<u>posID</u>:int, **comID**:int, salary:int, requirements:string, title:string,**houseNum:**
int,**street:** string, **postalZipCode**: string)
FD: (PK, CK) posID -> comID, salary, requirements, title, houseNum, street,
postalZipCode
In BCNF

```
Create Table Position(
        posID int(10),
        comID int(10) not null,
        salary int(10),
        requirements varchar(10),
        title varchar(10),
        houseNum int(10) not null,
        street varchar(max) not null,
        postalZipCode varchar(max) not null,
        PRIMARY KEY(posID)
        FOREIGN KEY(comID) REFERENCES Company
        FOREIGN KEY(houseNum, street, postalZipCode) REFERENCES
```

```
                    Address(houseNum,street, postalZipCode)
          );
INSERT INTO Position VALUES (01, 01, 70000, null, “Junior Software Developer”, 1234,
“East St”, “V1L 2U3”);
INSERT INTO Position VALUES (02, 01, 75000, null, “Software Developer”, 1234, “East St”,
“V1L 2U3”);
INSERT INTO Position VALUES (03, 03, 100000, null, “Software Developer”, 4321, “West St”,
“V1L 2U5”);
INSERT INTO Position VALUES (04, 03, 120000, null, “Senior Software Developer”, 4321,
“West St”, “V1L 2U5”);
INSERT INTO Position VALUES (05, 01, 80000, null, “Software Developer”, 1234, “East St”,
“V1L 2U3”);
```

Address(street:string, houseNum:int, postalZipCode:string, city:string, provinceState:string)
Candidate key: {Street, houseNum, postalZipCode}

FD:
(PK, CK) Street, house#, postalZipCode -> city, provinceState
postalZipCode -> city, provinceState

Not in BCNF: Requires Normalization since postalZipCode is not a superkey

Normalization Steps:

1) Split on the FD postalZipCode -> city, provinceState
    - Address1(street, houseNum, postalZipCode)
    - Address2(**postalZipCode**, city, provinceState)

```
Create Table Address1(
         street varchar(max),
         houseNum int(10),
         postalZipCode varchar(max),
         PRIMARY KEY (street, houseNum, postalZipCode)
);
INSERT INTO Address1(1234, “East St”, “V1L2U3”);
INSERT INTO Address1(4321, “West St”, “V1L2U5”);
INSERT INTO Address1(9876, “Other St”, “V5K2U3”);
INSERT INTO Address1(6789, “Some St”, “V1L9P8”);
INSERT INTO Address1(1234, “Fake St”, “V9L6T5”);

Create Table Address2(
        postalZipCode varchar(max),
        city varchar(max),
        provinceState varchar(max),
        PRIMARY KEY (postalZipCode)
        FOREIGN KEY (postalZipCode) references Address1
);
```

```
INSERT INTO Address2(“V1L2U3”, “Vancouver”, “BC”);
INSERT INTO Address2(“V1L2U5”, “Vancouver”, “BC”);
INSERT INTO Address2(“V5K2U3”, “Toronto”, “ON”);
INSERT INTO Address2(“V1L9P8”, “Toronto”, “ON”);
INSERT INTO Address2(“V9L6T5”, “Calgary”, “AB”);
```

Company(comID: int, name:string)
FD: (PK, CK) comID -> name
In BCNF

```
Create Table Company(
        comID int(10),
        name varchar(max),
        PRIMARY KEY(comID)
        );
INSERT INTO Company(comID, name);
VALUES (01, “Google”);
INSERT INTO Company(comID, name);
VALUES (02, “Best Buy”);
INSERT INTO Company(comID, name);
VALUES (03, “Microsoft”);
INSERT INTO Company(comID, name);
VALUES (04, “Amazon”);
INSERT INTO Company(comID, name)
VALUES (05, “Tesla”)
```

Recruiter(recID:int, name:string, email:string, phoneNum:string)
FD:
(PK, CK) recID -> name, email, phone#
(CK) email -> name, email, recID
(CK) phoneNum -> name, recID , phone#
In BCNF

```
Create Table Recruiter(
        recID int(10),
        name varchar(max),
        email varchar(max),
        phoneNum string(20)
        PRIMARY KEY(recID)
        );
INSERT INTO Recruiter(recID, name, email, phoneNum);
VALUES (01, “Bill”, “bill@gmail.com”, “456-543-4321”);
INSERT INTO Recruiter(recID, name, email, phoneNum);
```

```
VALUES (02, “Bob”, “bob@gmail.com”, “456-543-4322”);
INSERT INTO Recruiter(recID, name, email, phoneNum);
VALUES (03, “Billy”, “billy@gmail.com”, “456-543-4323”);
INSERT INTO Recruiter(recID, name, email, phoneNum);
VALUES (04, “Phill”, “phill@gmail.com”, “456-543-4351”);
INSERT INTO Recruiter(recID, name, email, phoneNum);
VALUES (05, “Philly”, “philyl@gmail.com”, “456-533-4321”);
```

EmployedBy(**<u>recID</u>**:int, **<u>comID</u>**:int)
FD: (PK, CK) recID,comID - > recID, comID
In BCNF

```
Create Table EmployedBy(
        recID int(10),
        comID int(10),
        FOREIGN KEY(recID) REFERENCES Recruiter,
        FOREIGN KEY(comID) REFERENCES Company
        );
INSERT INTO EmployedBy(recID, comID);
VALUES(01, 01);
INSERT INTO EmployedBy(recID, comID);
VALUES(02, 02);
INSERT INTO EmployedBy(recID, comID);
VALUES(03, 03);
INSERT INTO EmployedBy(recID, comID);
VALUES(04, 04);
INSERT INTO EmployedBy(recID, comID);
VALUES(05, 05);
```

FacilitatedBy(**<u>intID</u>**:int, **<u>recID</u>**:int, platform:string)
FD: (PK, CK) intID, recID -> platform
In BCNF

```
Create Table FacilitatedBy(
        intID int(10),
        recID int(10),
        platform varchar(max),
        PRIMARY KEY(intID, recID)
        FOREIGN KEY(intID) REFERENCES Interview ON DELETE SET DEFAULT,
        FOREIGN KEY(recID) REFERENCES Recruiter,
        );
INSERT INTO FacilitatedBy(intID, recID, platform);
VALUES(01, 01, “LinkedIn);
```

```
INSERT INTO FacilitatedBy(intID, recID, platform);
VALUES(02, 02, “LinkedIn);
INSERT INTO FacilitatedBy(intID, recID, platform);
VALUES(03, 03, “LinkedIn);
INSERT INTO FacilitatedBy(intID, recID, platform);
VALUES(04, 04, “LinkedIn);
INSERT INTO FacilitatedBy(intID, recID, platform);
VALUES(05, 05, “LinkedIn);
```

Coding(<u>codingQID</u>**:**int, qTitle:string, type:string, difficulty:string, text:string, approved:bool)
(qTitle is unique)
FD:
(PK, CK) qID -> qTitle, type, difficulty, text, approvedByAdmin
(CK) qTitle-> codingQID, type, difficulty, text, approvedByAdmin
In BCNF

```
Create Table Coding(
        codingQID int(10),
        qTitle varchar(max) UNIQUE NOT NULL,
        type varchar(max),
        difficulty varchar(max),
        text varchar(max),
        approvedByAdmin boolean,
        PRIMARY KEY (codingQID)
        );

INSERT INTO Coding VALUES(01, “sort a list”,”sorting”,”easy”,null,TRUE);
INSERT INTO Coding VALUES(02, “find the lowest common ancestor”,”binary
search”,”medium”,null,TRUE);
INSERT INTO Coding VALUES(03, “rotting oranges”,”bfs”,”medium”,null,TRUE);
INSERT INTO Coding VALUES(04, “reverse a linked list”,”linked list”,”easy”,null,TRUE);
INSERT INTO Coding VALUES(05, “Maximum Subarray”,”dynamic
programming”,”easy”,null,TRUE);
```

Behavioral(<u>behaviourQID</u>**:**int, qTitle:string, text:string, approved:bool*) (qTitle is unique)*
FD:
(PK, CK) qID -> qTitle, text, approved
(CK) qTitle-> behaviourQID, text, approved
In BCNF

```
Create Table Behavioural(
        behavioralQID int(10),
        qTitle varchar(max) UNIQUE NOT NULL,
        text varchar(max),
        approvedByAdmin boolean,
        PRIMARY KEY (behaviourQID)
        );

INSERT INTO Behavioral VALUES (01,”tell us about yourself”, null, false);
INSERT INTO Behavioral VALUES (02,”what was your most challenging project”, null, false);
INSERT INTO Behavioral VALUES (03,”tell me about a time you made a mistake”, null, false);
INSERT INTO Behavioral VALUES (04,”what are your strengths”, null, false);
INSERT INTO Behavioral VALUES (05,”what is something you would like to improve on in this
project”, null, false);
```

InterviewCoding(**<u>intID</u>:** int**,** **<u>codingQID</u>**: int)
FD
(PK, CK) intID, codingQID -> intID, codingID
In BCNF

```
Create Table InterviewCoding(
        intID:int,
        codingQID: int
        PRIMARY KEY (intID, codingQID)
        FOREIGN KEY(intID) references Interview,
        FOREIGN KEY(codingQID) references Coding
        );

INSERT INTO InterviewCoding VALUES (01,02);
INSERT INTO InterviewCoding VALUES (01,03);
INSERT INTO InterviewCoding VALUES (02,03);
INSERT INTO InterviewCoding VALUES (03,04);
INSERT INTO InterviewCoding VALUES (03,05);
```

InterviewBehavioural(**<u>intID</u>: int, <u>behaviourQID</u>**: int)
FD
(PK, CK) intID,behaviourQID -> intID, behaviourQID
In BCNF

```
Create Table InterviewBehavioural(
        intID int(10),
        behaviorQID int(10),
        PRIMARY KEY (intID, behaviourID)
        FOREIGN KEY(intID) REFERENCES Interview,
        FOREIGN KEY(behaviourQID) references Behavioral
        );
```

INSERT INTO InterviewBehavioural(IntID, BehaviouralQID);
VALUES (01, 01);
INSERT INTO InterviewBehavioural(IntID, BehaviouralQID);
VALUES (02, 0 2);
INSERT INTO InterviewBehavioural(IntID, BehaviouralQID);
VALUES (03, 03);
INSERT INTO InterviewBehavioural(IntID, BehaviouralQID);
VALUES (04, 04);
INSERT INTO InterviewBehavioural(IntID, BehaviouralQID);
VALUES (05, 05);

Resources(resID:int, **codingQID,** link:string, title:string, description:string, website:string)
FD:
(PK, CK) resID -> codingQID, link, title, description, website
link -> website
Requires normalization since link is not a superkey
Normalization Steps:

1. Split on the FD link -> website to obtain the BCNF decomposition
   Resources1(resID, **codingQID**, link, title, description)
   Resources2(**link**, website)

```
Create Table Resources1(
        resID int (10),
        codingQID int (10),
        link varchar(max),
        title varchar(max),
        description varchar(max),
        PRIMARY KEY (resID),
        FOREIGN KEY (codingQID) references Coding
);

INSERT
INTO Resources1
VALUES (01,01,"https://leetcode.com/problems/merge-intervals/", "merge intervals", null);

INSERT
INTO Resources1
VALUES (02,02,"https://www.geeksforgeeks.org/check-if-a-number-is-palindrome/",
"palindrom", null);

INSERT
INTO Resources1
```

```
VALUES (03,04,“https://leetcode.com/problems/merge-intervals/”, “merge intervals”, null);

INSERT
INTO Resources1
VALUES (04,01,“https://leetcode.com/problems/merge-intervals/”, “merge intervals”, null);

INSERT
INTO Resources1
VALUES (05,02,”https://www.geeksforgeeks.org/check-if-a-number-is-palindrome/”,
“palindrom”, null);

Create Table Resources2(
        link varchar(max),
        website varchar(max),
        PRIMARY KEY (link)
        FOREIGN KEY (link) references Resources1
);

INSERT I
NTO Resources2
VALUES (“https://www.geeksforgeeks.org/reverse-a-linked-list/”, “www.geeksforgeeks.org”);

INSERT
INTO Resources2
VALUES  (“https://www.geeksforgeeks.org/check-if-a-number-is-palindrome/”,
“www.geeksforgeeks.org”);

INSERT
INTO Resources2
VALUES (“https://leetcode.com/problems/merge-intervals/”, “www.leetcode.com”);

INSERT
INTO Resources2
VALUES (“https://leetcode.com/problems/accounts-merge/”, “www.leetcode.com”);

INSERT
INTO Resources2
VALUES (“https://leetcode.com/problems/time-based-key-value-store/”,“www.leetcode.com”);
```

CodingResources(**resID**:int, **codingQID**:int)
FD:
(PK,CK) resID, codingQID - > resID, codingQID
In BCNF

```
CREATE TABLE CodingResources(
        resID int(10),
        codingQID int(10),
        PRIMARY KEY (resID, codingQID)
        FOREIGN KEY(resID) REFERENCES Resources1,
        FOREIGN KEY(codingQID) REFERENCES Coding
);

INSERT INTO codingResources VALUES (01, 01);
INSERT INTO codingResources VALUES (02, 01);
INSERT INTO codingResources VALUES (01, 03);
INSERT INTO codingResources VALUES (04, 05);
INSERT INTO codingResources VALUES (05, 05);
```

Variant(**<u>codingQID:</u>** int, <u>vTtitle</u>:string, question:string, solution:string)
FD:
(PK, CK) codingQID, vTitle -> question, solution
In BCNF

```
Create Table Variant(
        codingQID int(10)
        vTitle varchar(max),
        question varchar(max),
        solution varchar(max),
        PRIMARY KEY (codingQID, vTitle)
        FOREIGN KEY(codingQID) REFERENCES Coding
        );

INSERT INTO Variant VALUES(01, “variant on peak element”, “find the peak element in a
rotated array”,”used binary search n rotate array as follows- please leave feedback”);

INSERT INTO Variant VALUES(01, “find all peak elements”, “find all the peak elements in an
array”,”used linear search …..”);

INSERT INTO Variant VALUES(02, “ merge k sorted lists”, “given k sorted lists, merge into
one list”,”used merge sort variant…..”);

INSERT INTO Variant VALUES(02, “merge intervals”, “Given an array of intervals where
intervals[i] = [starti, endi], merge all overlapping intervals”,”public int[][] merge(int[][]
intervals),Arrays.sort(intervals, (a, b) -> a[0] - b[0]);”);
```

INSERT INTO Variant VALUES(02, “merge accounts”, “Given a list of accounts where each element accounts[i] is a list of strings, where the first element accounts[i][0] is a name……..”,”public List<List<String>> accountsMerge(List<List<String>> accounts).......”);

CodingSolution(solID: int, **codingQID**: int, title: string, text: string, rating: int)
FD
(PK, CK) solID -> codingQID, text, rating
In BCNF

```
Create Table codingSolution(
        solID int(10),
        codingQID int(10),
        title varchar(max)
        text varchar(max),
        rating int(4),
        PRIMARY KEY (solID, codingQID)
        FOREIGN KEY (codingQID) REFERENCES Coding ON DELETE CASCADE
) ;
```

INSERT INTO CodingSolution VALUES (01, 02,”bfs approach”,”used a bfs approach as shown in code below….” ,3);
INSERT INTO CodingSolution VALUES (02,03, ”dfs approach”,“used a dfs approach as shown in code below….”, 5);
INSERT INTO CodingSolution VALUES (03, 07,”reversing a linked list”, “public void reverseLL()...........”, 1);
INSERT INTO CodingSolution VALUES (04, 08, ”isPalindrome”,“public boolean isPalindrome().......”, 3);
INSERT INTO CodingSolution VALUES (05,02, ”binary search for peak element, “performed binary search as shown in code below….”, 0, 2);

SQL FILE CONTENT

```
Create Table Address1(
        street varchar(max),
        houseNum int(10),
        postalZipCode varchar(max),
        PRIMARY KEY (street, houseNum, postalZipCode)
);

INSERT INTO Address1(1234, “East St”, “V1L2U3”);
INSERT INTO Address1(4321, “West St”, “V1L2U5”);
INSERT INTO Address1(9876, “Other St”, “V5K2U3”);
INSERT INTO Address1(6789, “Some St”, “V1L9P8”);
INSERT INTO Address1(1234, “Fake St”, “V9L6T5”);



Create Table Address2(
        postalZipCode varchar(max),
        city varchar(max),
        provinceState varchar(max),
        PRIMARY KEY (postalZipCode)
        FOREIGN KEY (postalZipCode) references Address1
);

INSERT INTO Address2(“V1L2U3”, “Vancouver”, “BC”);
INSERT INTO Address2(“V1L2U5”, “Vancouver”, “BC”);
INSERT INTO Address2(“V5K2U3”, “Toronto”, “ON”);
INSERT INTO Address2(“V1L9P8”, “Toronto”, “ON”);
INSERT INTO Address2(“V9L6T5”, “Calgary”, “AB”);



Create Table User(
        userID int(10) DEFAULT 0,
        name varchar(100),
        age int(4),
        experience int(4),
        education varchar(max),
        about  varchar(max),
        PRIMARY KEY (userID)
        );
```

```
INSERT INTO User VALUES (01, “Patrick”, null, 0, “UBC”, null);
INSERT INTO User VALUES (02, “James”, null, 2, “UBC”, null);
INSERT INTO User VALUES (03, “Akriti”, null, 1, “UBC”, null);
INSERT INTO User VALUES (04, “Erika”, 21, 0, “UBC”, “Hello”);
INSERT INTO User VALUES (05, “Fred”, 22, 0, “UBC”, “I am Fred”);



Create Table Company(
        comID int(10),
        name varchar(max),
        PRIMARY KEY(comID)
        );

INSERT INTO Company(comID, name);
VALUES (01, “Google”);
INSERT INTO Company(comID, name);
VALUES (02, “Best Buy”);
INSERT INTO Company(comID, name);
VALUES (03, “Microsoft”);
INSERT INTO Company(comID, name);
VALUES (04, “Amazon”);
INSERT INTO Company(comID, name)
VALUES (05, “Tesla”)



Create Table Position(
        posID int(10),
        comID int(10) not null,
        salary int(10),
        requirements varchar(10),
        title varchar(10),
        houseNum int(10) not null,
        street varchar(max) not null,
        postalZipCode varchar(max) not null,
        PRIMARY KEY(posID)
        FOREIGN KEY(comID) REFERENCES Company
        FOREIGN KEY(houseNum, street, postalZipCode) REFERENCES
                Address(houseNum,street, postalZipCode)
        );
```

```
INSERT INTO Position VALUES (01, 01, 70000, null, “Junior Software Developer”, 1234, “East
St”, “V1L 2U3”);
INSERT INTO Position VALUES (02, 01, 75000, null, “Software Developer”, 1234, “East St”,
“V1L 2U3”);
INSERT INTO Position VALUES (03, 03, 100000, null, “Software Developer”, 4321, “West St”,
“V1L 2U5”);
INSERT INTO Position VALUES (04, 03, 120000, null, “Senior Software Developer”, 4321,
“West St”, “V1L 2U5”);
INSERT INTO Position VALUES (05, 01, 80000, null, “Software Developer”, 1234, “East St”,
“V1L 2U3”);
```

```
Create Table Application(
        appID int(10),
        posID   int(10) not null,
        userID int(10) not null,
        resume varchar(max),
        coverLetter varchar(max),
        success?  BOOLEAN,
        date DATE,
        PRIMARY KEY (appID)
        FOREIGN KEY (posID) REFERENCES Position
        FOREIGN KEY (userID) REFERENCES User ON DELETE SET DEFAULT
        );
```

```
INSERT INTO Application VALUES (01, 01, 01, “Resume...”, “Cover letter...”, False, 7/20/2022);
INSERT INTO Application VALUES (02, 01, 02, “Resume...”, “Cover letter...”, False, 7/22/2022);
INSERT INTO Application VALUES (03, 02, 01, “Resume...”, “Cover letter...”, True, 6/24/2022);
INSERT INTO Application VALUES (04, 04, 03, “Resume...”, “Cover letter...”, False, 3/06/2022);
INSERT INTO Application VALUES (05, 04, 03, “Resume...”, “Cover letter...”, False, 7/26/2022);
```

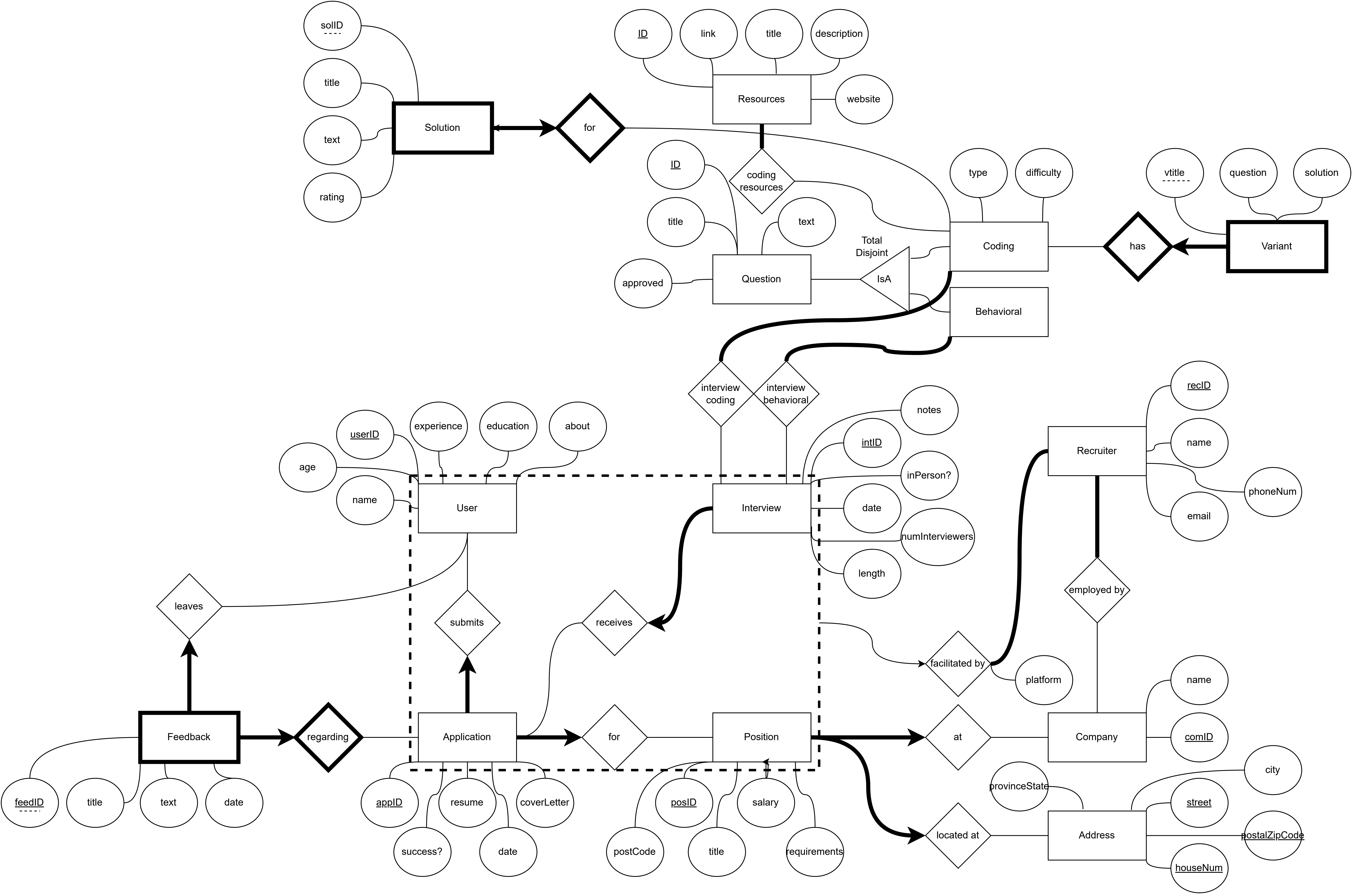
```
Create Table Interview1(
        intID int(10),
        appID int(10) not null,
        date DATE,
        length int(4),
        notes varchar(max),
        inPerson? BOOLEAN,
        numInterviewers int(4),
        PRIMARY KEY (intID)
```

```
	FOREIGN KEY (appID) REFERENCES Application ON DELETE SET DEFAULT
	);
INSERT INTO Interview1 VALUES (01, 01, 07/21/2022, 2, null, True, 1);
INSERT INTO Interview1 VALUES (02, 01,  07/21/2022, 2, null, True, 1);
INSERT INTO Interview1 VALUES (03, 02,  06/14/2022, 2, null, False, 2);
INSERT INTO Interview1 VALUES (04, 04,  07/27/2022, 2, null, False, 1);
INSERT INTO Interview1 VALUES (05, 05,  07/27/2022, 2, null, False, 1);



Create Table Recruiter(
	recID int(10),
	name varchar(max),
	email varchar(max),
	phoneNum string(20)
	PRIMARY KEY(recID)
	);

INSERT INTO Recruiter(recID, name, email, phoneNum);
VALUES (01, "Bill", "bill@gmail.com", "456-543-4321");
INSERT INTO Recruiter(recID, name, email, phoneNum);
VALUES (02, "Bob", "bob@gmail.com", "456-543-4322");
INSERT INTO Recruiter(recID, name, email, phoneNum);
VALUES (03, "Billy", "billy@gmail.com", "456-543-4323");
INSERT INTO Recruiter(recID, name, email, phoneNum);
VALUES (04, "Phill", "phill@gmail.com", "456-543-4351");
INSERT INTO Recruiter(recID, name, email, phoneNum);
VALUES (05, "Philly", "philyl@gmail.com", "456-533-4321");



Create Table Feedback1(
	feedID int(10),
	appID int(10),
	title varchar(200),
	text varchar(max),
	date  DATE,
	PRIMARY KEY (feedID, appID)
	FOREIGN KEY (appID) REFERENCES Application ON DELETE CASCADE
	)

INSERT INTO Feedback1 VALUES (01, 01, "Good job", "Good job", 7/25/2022);
INSERT INTO Feedback1 VALUES (02, 01, "Some advice...", "Advice here", 7/26/2022);
INSERT INTO Feedback1 VALUES (03, 02, “I like”, “:)”, 7/27/2022);
INSERT INTO Feedback1 VALUES (04, 03, “Pretty Bad”, “Try harder”, 7/27/2022);
INSERT INTO Feedback1 VALUES (05, 03, “What”, “what”, 7/27/2022);



Create Table Feedback2(
        appID int(10),
        userID int(10) not null,
        PRIMARY KEY (appID)
        FOREIGN KEY (userID) REFERENCES User ON DELETE SET DEFAULT
        FOREIGN KEY (appID) REFERENCES Feedback1 ON DELETE CASCADE
        );

INSERT INTO Feedback2  VALUES (01, 01);
INSERT INTO Feedback2  VALUES (01, 01);
INSERT INTO Feedback2  VALUES (02, 02);
INSERT INTO Feedback2  VALUES (03, 03);
INSERT INTO Feedback2  VALUES (03, 04);



Create Table EmployedBy(
        recID int(10),
        comID int(10),
        FOREIGN KEY(recID) REFERENCES Recruiter,
        FOREIGN KEY(comID) REFERENCES Company
        );

INSERT INTO EmployedBy(recID, comID);
VALUES(01, 01);
INSERT INTO EmployedBy(recID, comID);
VALUES(02, 02);
INSERT INTO EmployedBy(recID, comID);
VALUES(03, 03);
INSERT INTO EmployedBy(recID, comID);
VALUES(04, 04);
INSERT INTO EmployedBy(recID, comID);
VALUES(05, 05);



Create Table FacilitatedBy(
        intID int(10),
        recID int(10),
        platform varchar(max),
        PRIMARY KEY(intID, recID)
        FOREIGN KEY(intID) REFERENCES Interview ON DELETE SET DEFAULT,
        FOREIGN KEY(recID) REFERENCES Recruiter,
        );

INSERT INTO FacilitatedBy(intID, recID, platform);
VALUES(01, 01, “LinkedIn);
INSERT INTO FacilitatedBy(intID, recID, platform);
VALUES(02, 02, “LinkedIn);
INSERT INTO FacilitatedBy(intID, recID, platform);
VALUES(03, 03, “LinkedIn);
INSERT INTO FacilitatedBy(intID, recID, platform);
VALUES(04, 04, “LinkedIn);
INSERT INTO FacilitatedBy(intID, recID, platform);
VALUES(05, 05, “LinkedIn);



Create Table Coding(
        codingQID int(10),
        qTitle varchar(max) UNIQUE NOT NULL,
        type varchar(max),
        difficulty varchar(max),
        text varchar(max),
        approvedByAdmin boolean,
        PRIMARY KEY (codingQID)
        );

INSERT INTO Coding VALUES(01, “sort a list”,”sorting”,”easy”,null,TRUE);
INSERT INTO Coding VALUES(02, “find the lowest common ancestor”,”binary
search”,”medium”,null,TRUE);
INSERT INTO Coding VALUES(03, “rotting oranges”,”bfs”,”medium”,null,TRUE);
INSERT INTO Coding VALUES(04, “reverse a linked list”,”linked list”,”easy”,null,TRUE);
INSERT INTO Coding VALUES(05, “Maximum Subarray”,”dynamic
programming”,”easy”,null,TRUE);



Create Table Behavioural(
        behavioralQID int(10),
        qTitle varchar(max) UNIQUE NOT NULL,
        text varchar(max),
        approvedByAdmin boolean,
        PRIMARY KEY (behaviourQID)
        );

INSERT INTO Behavioral VALUES (01,"tell us about yourself", null, false);
INSERT INTO Behavioral VALUES (02,"what was your most challenging project", null, false);
INSERT INTO Behavioral VALUES (03,"tell me about a time you made a mistake", null, false);
INSERT INTO Behavioral VALUES (04,"what are your strengths", null, false);
INSERT INTO Behavioral VALUES (05,"what is something you would like to improve on in this
project", null, false);



Create Table InterviewCoding(
        intID:int,
        codingQID: int
        PRIMARY KEY (intID, codingQID)
        FOREIGN KEY(intID) references Interview,
        FOREIGN KEY(codingQID) references Coding
        );

INSERT INTO InterviewCoding VALUES (01,02);
INSERT INTO InterviewCoding VALUES (01,03);
INSERT INTO InterviewCoding VALUES (02,03);
INSERT INTO InterviewCoding VALUES (03,04);
INSERT INTO InterviewCoding VALUES (03,05);



Create Table InterviewBehavioural(
        intID int(10),
        behaviorQID int(10),
        PRIMARY KEY (intID, behaviourID)
        FOREIGN KEY(intID) REFERENCES Interview,
        FOREIGN KEY(behaviourQID) references Behavioral
        );

INSERT INTO InterviewBehavioural(IntID, BehaviouralQID);
VALUES (01, 01);
INSERT INTO InterviewBehavioural(IntID, BehaviouralQID);
VALUES (02,  02);
```

```
INSERT INTO InterviewBehavioural(IntID, BehaviouralQID);
VALUES (03, 03);
INSERT INTO InterviewBehavioural(IntID, BehaviouralQID);
VALUES (04,  04);
INSERT INTO InterviewBehavioural(IntID, BehaviouralQID);
VALUES (05,  05);


Create Table Variant(
        codingQID int(10)
        vTitle varchar(max),
        question varchar(max),
        solution varchar(max),
        PRIMARY KEY (codingQID, vTitle)
        FOREIGN KEY(codingQID) REFERENCES Coding
        );

INSERT INTO Variant VALUES(01, “variant on peak element”, “find the peak element in a
rotated array”,”used binary search n rotate array as follows- please leave feedback”);

INSERT INTO Variant VALUES(01, “find all peak elements”, “find all the peak elements in an
array”,”used linear search …..”);

INSERT INTO Variant VALUES(02, “ merge k sorted lists”, “given k sorted lists, merge into one
list”,”used merge sort variant…..”);

INSERT INTO Variant VALUES(02, “merge intervals”, “Given an array of intervals where
intervals[i] = [starti, endi], merge all overlapping intervals”,”public int[][] merge(int[][]
intervals),Arrays.sort(intervals, (a, b) -> a[0] - b[0]);”);

INSERT INTO Variant VALUES(02, “merge accounts”, “Given a list of accounts where each
element accounts[i] is a list of strings, where the first element accounts[i][0] is a name……..”,”public
List<List<String>> accountsMerge(List<List<String>> accounts).......”);


Create Table Resources1(
       resID int (10),
       codingQID int (10),
       link varchar(max),
       title varchar(max),
       description varchar(max),
        PRIMARY KEY (resID),
        FOREIGN KEY (codingQID) references Coding
);

INSERT
INTO Resources1
VALUES (01,01,“https://leetcode.com/problems/merge-intervals/”, “merge intervals”, null);

INSERT
INTO Resources1
VALUES (02,02,”https://www.geeksforgeeks.org/check-if-a-number-is-palindrome/”, “palindrom”,
null);

INSERT
INTO Resources1
VALUES (03,04,“https://leetcode.com/problems/merge-intervals/”, “merge intervals”, null);

INSERT
INTO Resources1
VALUES (04,01,“https://leetcode.com/problems/merge-intervals/”, “merge intervals”, null);

INSERT
INTO Resources1
VALUES (05,02,”https://www.geeksforgeeks.org/check-if-a-number-is-palindrome/”, “palindrom”,
null);



Create Table Resources2(
        link varchar(max),
        website varchar(max),
        PRIMARY KEY (link)
        FOREIGN KEY (link) references Resources1
);

INSERT I
NTO Resources2
VALUES (“https://www.geeksforgeeks.org/reverse-a-linked-list/”, “www.geeksforgeeks.org”);

INSERT
INTO Resources2
VALUES  (“https://www.geeksforgeeks.org/check-if-a-number-is-palindrome/”,
“www.geeksforgeeks.org”);
```

```
INSERT
INTO Resources2
VALUES (“https://leetcode.com/problems/merge-intervals/”, “www.leetcode.com”);

INSERT
INTO Resources2
VALUES (“https://leetcode.com/problems/accounts-merge/”, “www.leetcode.com”);

INSERT
INTO Resources2
VALUES (“https://leetcode.com/problems/time-based-key-value-store/”,“www.leetcode.com”);



CREATE TABLE CodingResources(
        resID int(10),
        codingQID int(10),
        PRIMARY KEY (resID, codingQID)
        FOREIGN KEY(resID) REFERENCES Resources1,
        FOREIGN KEY(codingQID) REFERENCES Coding
);

INSERT INTO codingResources VALUES (01, 01);
INSERT INTO codingResources VALUES (02, 01);
INSERT INTO codingResources VALUES (01, 03);
INSERT INTO codingResources VALUES (04, 05);
INSERT INTO codingResources VALUES (05, 05);



Create Table codingSolution(
        solID int(10),
        codingQID int(10),
        title varchar(max)
        text varchar(max),
        rating int(4),
        PRIMARY KEY (solID, codingQID)
        FOREIGN KEY (codingQID) REFERENCES Coding ON DELETE CASCADE
) ;

INSERT INTO CodingSolution VALUES (01, 02,”bfs approach”,”used a bfs approach as shown
in code below….” ,3);
```

INSERT INTO CodingSolution VALUES (02,03, ”dfs approach”,“used a dfs approach as shown in code below….”, 5);
INSERT INTO CodingSolution VALUES (03, 07,”reversing a linked list”, “public void reverseLL()...........”, 1);
INSERT INTO CodingSolution VALUES (04, 08, ”isPalindrome”,“public boolean isPalindrome().......”, 3);
INSERT INTO CodingSolution VALUES (05,02, ”binary search for peak element, “performed binary search as shown in code below….”, 0, 2);


solID
title
text
rating
Solution
for
ID
link
title
description
Resources
website
coding resources
ID
title
text
approved
Question
Total Disjoint
IsA
type
difficulty
Coding
Behavioral
has
vtitle
question
solution
Variant
interview coding
interview behavioral
notes
intID
inPerson?
date
numInterviewers
length
Interview
userID
experience
education
about
age
name
User
leaves
submits
receives
facilitated by
platform
Recruiter
recID
name
phoneNum
email
employed by
Feedback
regarding
Application
for
Position
at
Company
name
comID
feedID
title
text
date
appID
resume
coverLetter
success?
date
posID
salary
postCode
title
requirements
located at
Address
provinceState
city
street
postalZipCode
houseNum