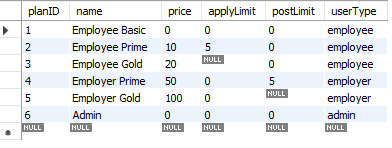
1. **Create/Delete/Edit/Display an Employer.**

From the Plans table, we see that the plan with ID 4 has userType ‘employer’

****

CREATE

INSERT INTO Users (userID,

planID,

email,

password,

dateCreated,

isActive,

balance,

isAutomatic)

VALUES (‘bob’,

**4,**

'bob@comp353.com',

'bob',

'2020-01-15',

TRUE,

0,

TRUE),

EDIT

UPDATE Users

SET isActive = 0

WHERE userID =’bob’;

DISPLAY

SELECT \* FROM Users

WHERE userID=’bob’;

DELETE

1. **Create/Delete/Edit/Display a category by an Employer.**

Create

INSERT INTO Employer\_Categories (userID, categoryName)

VALUES ('alice', 'Software Engineer'),

('alice', 'Tech Lead');

EDIT

UPDATE Employer\_Categories

SET categoryName = 'Project Manager'

WHERE categoryName = 'Tech Lead'

AND userID = ‘alice’

DISPLAY

SELECT \* FROM Employer\_Categories

WHERE categoryName = 'Project Manager'

AND userID = ‘alice’

DELETE

DELETE FROM Employer\_Categories

WHERE categoryName = 'Project Manager'

AND userID = 'alice'

1. **Post a new job by an employer.**

INSERT INTO Jobs

(userID, locationID, title, salary, description, companyName, positionsAvailable, status)

VALUES ('bob', 1, 'React Developer', 25, 'Work on React Apps', 'Google', 10, 'Open');

1. **Provide a job offer for an employee by an employer.**

Applications table before (user has applied to a job):



Query:

UPDATE Applications SET isAcceptedByEmployer = 1 WHERE userID = 'alice' AND jobID = 3;

Application after query (employer has accepted the application and offered a job):

****

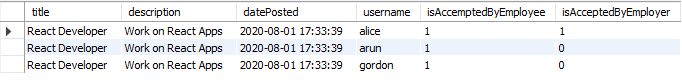
1. **Report of a posted job by an employer (Job title and description, date posted, list of employees applied to the job and status of each application).**

SELECT j.title, j.description, j.datePosted, IFNULL(a.userID, 'No Applications') AS 'username', IFNULL(a.isAcceptedByEmployee, 'N/A') AS 'isAccemptedByEmployee', IFNULL(a.isAcceptedByEmployer, 'N/A') AS 'isAcceptedByEmployer'

FROM Jobs AS j

LEFT JOIN Applications AS a ON j.jobID = a.jobID

WHERE j.jobID = 3



1. **Report of posted jobs by an employer during a specific period of time (Job title, date posted, short description of the job up to 50 characters, number of needed employees to the post, number of applied jobs to the post, number of accepted offers).**

SELECT j.title, SUBSTRING(j.description, 1, 50) AS 'description', j.positionsAvailable, j.datePosted, COUNT(a.jobID) AS 'Number of employees applied', SUM(IFNULL(a.isAcceptedByEmployee, 0)) AS 'Number of accepted offers'

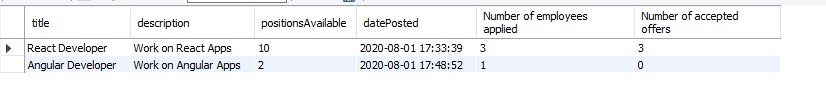
FROM Jobs AS j

LEFT JOIN Applications AS a ON j.jobID = a.jobID

WHERE j.userID = [input\_userID] AND

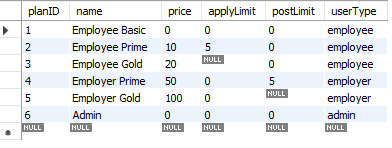
j.datePosted BETWEEN 'YYYY-MM-DD' AND 'YYYY-MM-DD'

GROUP BY j.jobID;



1. **Create/Delete/Edit/Display an Employee.**

From the Plans table, we see that the plan with ID 1 has userType ‘employee’

****

CREATE

INSERT INTO Users (userID,

planID,

email,

password,

dateCreated,

isActive,

balance,

isAutomatic)

VALUES ('gordon',

1,

'gordon@comp353.com',

'gordon',

'2020-01-15',

TRUE,

0,

TRUE),

EDIT

UPDATE Users

SET isActive = 0

WHERE userID = 'gordon';

DISPLAY

SELECT \* FROM Users

WHERE userID= 'gordon';

DELETE

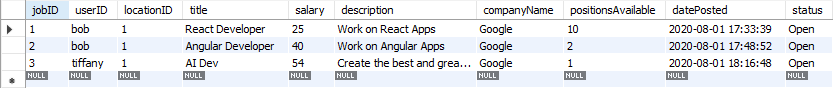
1. **Search for a job by an employee**

SELECT \* FROM Jobs

WHERE userID = 'bob' AND title LIKE 'React%';

1. **Apply for a job by an employee.**

Existing jobs in the Jobs table:

****

Query:

INSERT INTO Applications (jobID, userID) VALUES (2, ‘tyson’)

New Row in Applications table:

****

1. **Accept/Deny a job offer by an employee**

Existing applications in the Applications table



Deny job offer:

UPDATE Applications

SET isAcceptedByEmployee = 0

WHERE jobID = 2 AND userID = 'tyson';

Accept job offer:

UPDATE Applications

SET isAcceptedByEmployee = 1

WHERE jobID = 3 AND userID = 'leo';

Applications after user ‘leo’ accepted and user ‘tyson’ denies



1. **Withdraw from an applied job by an employee.**

DELETE FROM Applications

WHERE jobID = 1 and userID = 'leo';

1. **Delete a profile by an employee.**

DELETE FROM Profiles

WHERE userID = 'gordon';

1. **Report of applied jobs by an employee during a specific period of time (Job title, date applied, short description of the job up to 50 characters, status of the application).**

SELECT j.title, SUBSTRING(j.description, 1, 50) AS 'description', a.isAcceptedByEmployer, a.isAcceptedByEmployee

FROM Applications AS a, Jobs as j

WHERE a.userID = 'alice' AND j.jobID = a.jobID

AND a.dateApplied BETWEEN 'YYYY-MM-DD' AND 'YYYY-MM-DD'

GROUP BY j.jobID;

1. **Add/Delete/Edit a method of payment by a user.**

CREATE

INSERT INTO Payment\_Methods (paymentMethodID,

userID,

isPreSelected,

cardNumber,

paymentType)

VALUES (1,

'gordon',

1,

1111111111,

'checking')

DELETE

DELETE FROM Payment\_Methods

WHERE userID = 'gordon'

EDIT

UPDATE Payment\_Methods

SET isPreSelected = 0

WHERE userID = 'gordon'

1. **Add/Delete/Edit an automatic payment by a user.**
2. **Make a manual payment by a user.**
3. **Report of all users by the administrator for employers or employees (Name, email, category, status, balance.**

SELECT p.firstName as 'First Name', p.lastName as 'Last Name', u.email as 'Email', pl.name as 'Plan Name', pl.userType as 'User Type', u.isActive, u.balance as 'Balance'

FROM Users as u, Profiles as p, Plans as pl

WHERE u.planID = pl.planID AND u.userID = p.userID

AND pl.userType <> 'admin'

1. **Report of all outstanding balance accounts (User name, email, balance, since when the account is suffering).**

SELECT userID, email, balance, startSufferingDate

FROM Users

WHERE balance < 0;