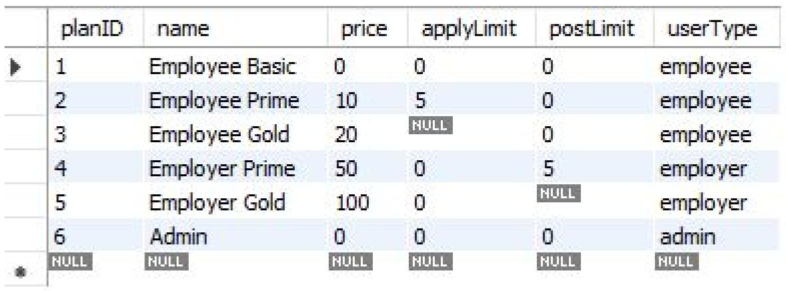
1. **Create/Delete/Edit/Display an Employer.**

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



	planID	name	price	applyLimit	postLimit	userType
▶	1	Employee Basic	0	0	0	employee
	2	Employee Prime	10	5	0	employee
	3	Employee Gold	20	NULL	0	employee
	4	Employer Prime	50	0	5	employer
	5	Employer Gold	100	0	NULL	employer
	6	Admin	0	0	0	admin
*	NULL	NULL	NULL	NULL	NULL	NULL

<u>CREATE</u>

```
INSERT INTO Users (userID,
             planID,
             email,
             password,
             dateCreated,
             isActive,
             balance,
             isAutomatic)
VALUES       (‘bob’,
              4,
              'bob@comp353.com',
              'bob',
              '2020-01-15',
              TRUE,
              0,
              TRUE),
```

<u>EDIT</u>

```
UPDATE Users
SET isActive = 0
WHERE userID =’bob’;
```

<u>DISPLAY</u>

```
SELECT * FROM Users
WHERE userID=’bob’;
```

<u>DELETE</u>

```
DELETE FROM Users
WHERE userID=’bob’
```

Note: This delete will cascade and delete rows from all the other tables where the userID ‘bob’ is a FK.

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

<u>Create</u>
```
INSERT INTO Employer_Categories (userID, categoryName)
VALUES ('alice', 'Software Engineer'),
            ('alice', 'Tech Lead');
```

<u>EDIT</u>
```
UPDATE Employer_Categories
SET categoryName = 'Project Manager'
WHERE categoryName = 'Tech Lead'
AND userID = ‘alice’
```

<u>DISPLAY</u>
```
SELECT * FROM Employer_Categories
WHERE categoryName = 'Project Manager'
AND userID = ‘alice’
```

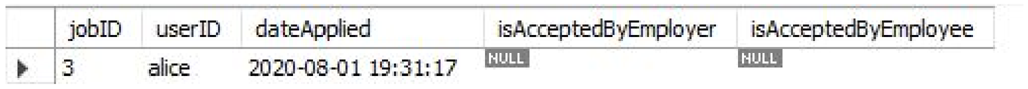
<u>DELETE</u>
```
DELETE FROM Employer_Categories
WHERE categoryName = 'Project Manager'
AND userID = 'alice'
```

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

```
INSERT INTO Jobs
(userID, locationID, title, salary, description, positionsAvailable, status)
VALUES ('bob', 1, 'React Developer', 25, 'Work on React Apps',10, ‘active’);
```

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

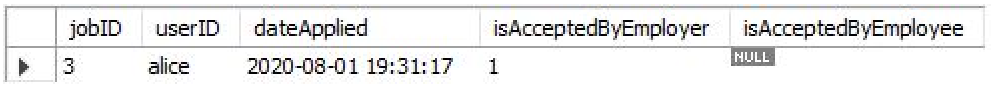
Applications table before (user has applied to a job):



	jobID	userID	dateApplied	isAcceptedByEmployer	isAcceptedByEmployee
▶	3	alice	2020-08-01 19:31:17	NULL	NULL

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

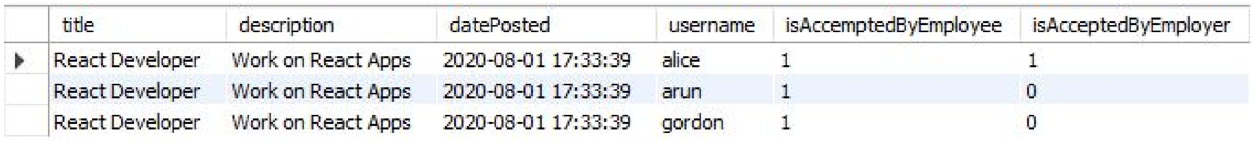
Application after query (employer has accepted the application and offered a job):



	jobID	userID	dateApplied	isAcceptedByEmployer	isAcceptedByEmployee
▶	3	alice	2020-08-01 19:31:17	1	NULL

5. **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).**

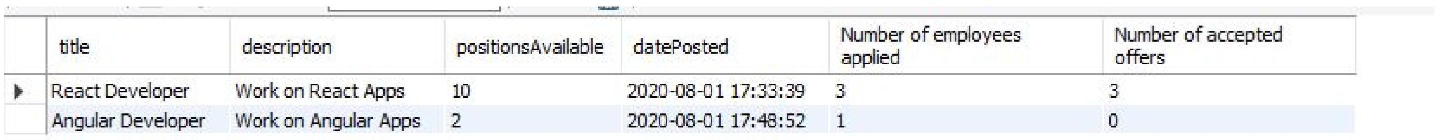
```
SELECT j.title, j.description, j.datePosted, IFNULL(a.userID, 'No Applications') AS 'username',
IFNULL(a.isAcceptedByEmployee, 'N/A') AS 'isAccemptedByEmployee',
IFNULL(a.isAcceptedByEmployer, 'N/A') AS 'isAcceptedByEmployer'
FROM Jobs AS j
LEFT JOIN Applications AS a ON j.jobID = a.jobID
WHERE j.jobID = [input.jobID]
```



title	description	datePosted	username	isAccemptedByEmployee	isAcceptedByEmployer
React Developer	Work on React Apps	2020-08-01 17:33:39	alice	1	1
React Developer	Work on React Apps	2020-08-01 17:33:39	arun	1	0
React Developer	Work on React Apps	2020-08-01 17:33:39	gordon	1	0

6. **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).**

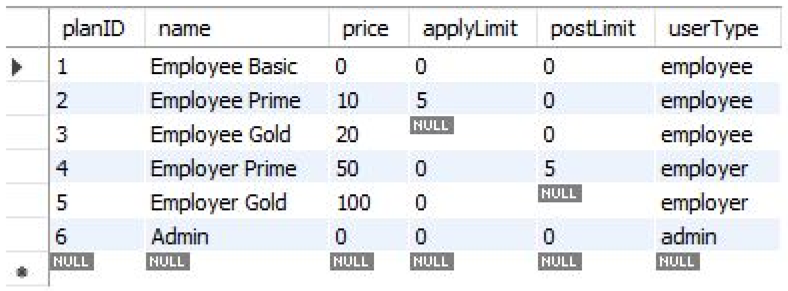
```
SELECT j.title, SUBSTRING(j.description, 1, 50) AS 'description', j.positionsAvailable,
j.datePosted, COUNT(a.jobID) AS 'Number of employees applied',
SUM(IFNULL(a.isAcceptedByEmployee, 0)) AS 'Number of accepted offers'
FROM Jobs AS j
LEFT JOIN Applications AS a ON j.jobID = a.jobID
WHERE j.userID = [input_userID] AND
j.datePosted BETWEEN 'YYYY-MM-DD' AND 'YYYY-MM-DD'
GROUP BY j.jobID;
```



title	description	positionsAvailable	datePosted	Number of employees applied	Number of accepted offers
React Developer	Work on React Apps	10	2020-08-01 17:33:39	3	3
Angular Developer	Work on Angular Apps	2	2020-08-01 17:48:52	1	0

7. Create/Delete/Edit/Display an Employee.

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



planID	name	price	applyLimit	postLimit	userType
1	Employee Basic	0	0	0	employee
2	Employee Prime	10	5	0	employee
3	Employee Gold	20	NULL	0	employee
4	Employer Prime	50	0	5	employer
5	Employer Gold	100	0	NULL	employer
6	Admin	0	0	0	admin
NULL	NULL	NULL	NULL	NULL	NULL

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';
```

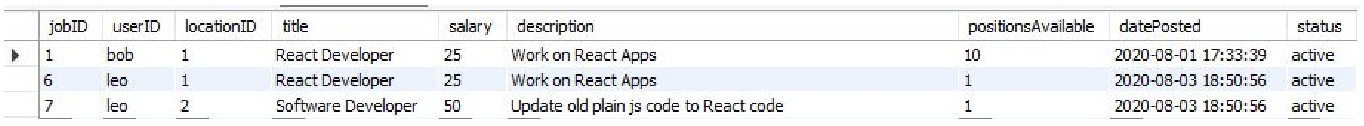
DELETE

```
DELETE FROM Users
WHERE userID = ‘gordon’;
```

Note: This delete will cascade and delete rows from all the other tables where the userID 'gordon' is a FK.

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

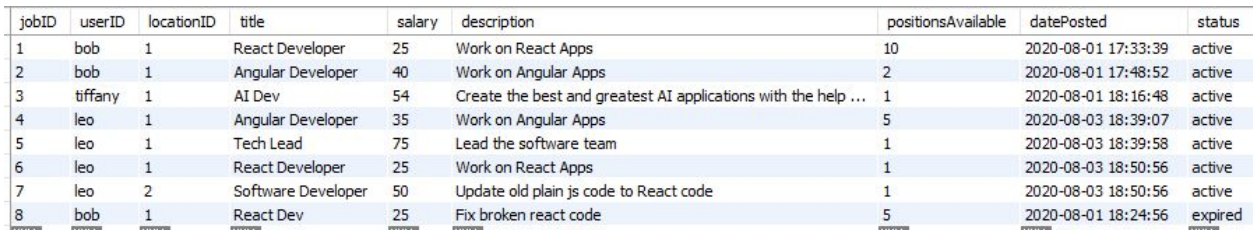
```
SELECT * FROM Jobs
WHERE status = 'active' AND (title LIKE '%React%' OR description LIKE '%React%');
```



jobID	userID	locationID	title	salary	description	positionsAvailable	datePosted	status
1	bob	1	React Developer	25	Work on React Apps	10	2020-08-01 17:33:39	active
6	leo	1	React Developer	25	Work on React Apps	1	2020-08-03 18:50:56	active
7	leo	2	Software Developer	50	Update old plain js code to React code	1	2020-08-03 18:50:56	active

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

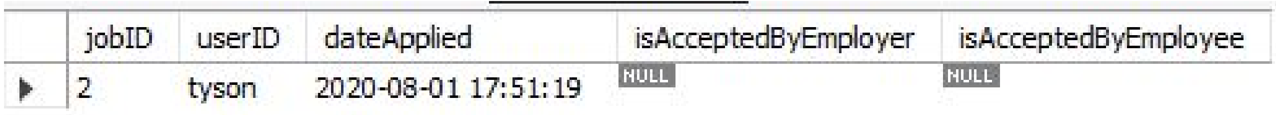
Existing jobs in the Jobs table:



jobID	userID	locationID	title	salary	description	positionsAvailable	datePosted	status
1	bob	1	React Developer	25	Work on React Apps	10	2020-08-01 17:33:39	active
2	bob	1	Angular Developer	40	Work on Angular Apps	2	2020-08-01 17:48:52	active
3	tiffany	1	AI Dev	54	Create the best and greatest AI applications with the help ...	1	2020-08-01 18:16:48	active
4	leo	1	Angular Developer	35	Work on Angular Apps	5	2020-08-03 18:39:07	active
5	leo	1	Tech Lead	75	Lead the software team	1	2020-08-03 18:39:58	active
6	leo	1	React Developer	25	Work on React Apps	1	2020-08-03 18:50:56	active
7	leo	2	Software Developer	50	Update old plain js code to React code	1	2020-08-03 18:50:56	active
8	bob	1	React Dev	25	Fix broken react code	5	2020-08-01 18:24:56	expired

Query:
INSERT INTO Applications (jobID, userID) VALUES (2, 'tyson')

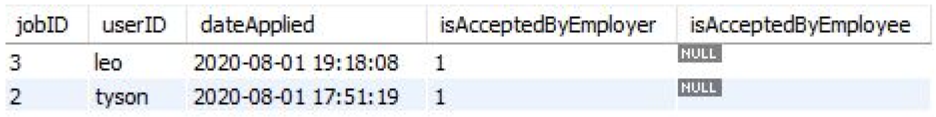
New Row in Applications table:



jobID	userID	dateApplied	isAcceptedByEmployer	isAcceptedByEmployee
2	tyson	2020-08-01 17:51:19	NULL	NULL

10. Accept/Deny a job offer by an employee

Existing applications in the Applications table

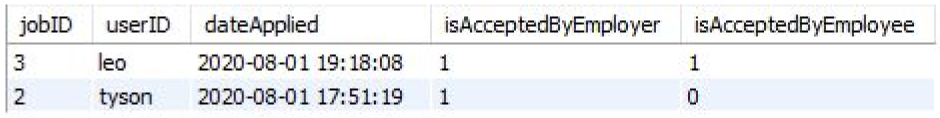


jobID	userID	dateApplied	isAcceptedByEmployer	isAcceptedByEmployee
3	leo	2020-08-01 19:18:08	1	NULL
2	tyson	2020-08-01 17:51:19	1	NULL

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



jobID	userID	dateApplied	isAcceptedByEmployer	isAcceptedByEmployee
3	leo	2020-08-01 19:18:08	1	1
2	tyson	2020-08-01 17:51:19	1	0

11. Withdraw from an applied job by an employee.

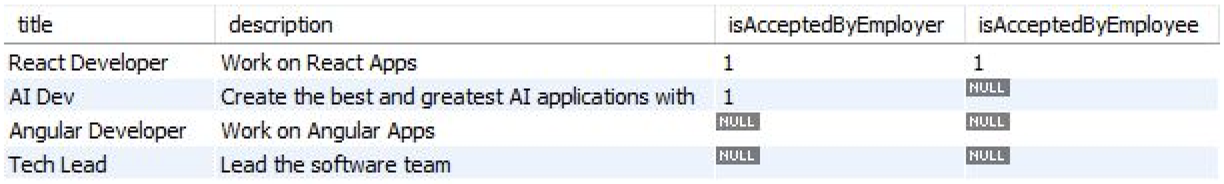
DELETE FROM Applications
WHERE jobID = 1 and userID = 'leo';

12. Delete a profile by an employee.

DELETE FROM Profiles
WHERE userID = 'gordon';

13. **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 = [input.userID] AND j.jobID = a.jobID
AND a.dateApplied BETWEEN 'YYYY-MM-DD' AND 'YYYY-MM-DD'
GROUP BY j.jobID;
```



title	description	isAcceptedByEmployer	isAcceptedByEmployee
React Developer	Work on React Apps	1	1
AI Dev	Create the best and greatest AI applications with	1	NULL
Angular Developer	Work on Angular Apps	NULL	NULL
Tech Lead	Lead the software team	NULL	NULL

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

CREATE

```
INSERT INTO Payment_Methods (paymentMethodID,
                userID,
                isPreSelected,
                cardNumber,
                paymentType)
VALUES     (1,
                'gordon',
                0,
                1111111111,
                'checking')
```

DELETE

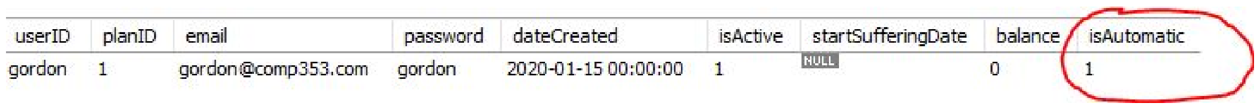
```
DELETE FROM Payment_Methods
WHERE userID = 'gordon' AND paymentMethodID = 1
```

EDIT

```
UPDATE Payment_Methods
SET isPreSelected = 1
WHERE userID = 'gordon' AND paymentMethodID = 1
```

15. Add/Delete/Edit an automatic payment by a user.

From the Users table, we see that this user uses automatic payments



userID	planID	email	password	dateCreated	isActive	startSufferingDate	balance	isAutomatic
gordon	1	gordon@comp353.com	gordon	2020-01-15 00:00:00	1	NULL	0	1

Since the user wants to use automatic payments, they will be required to add at least one payment method where isPreSelected = 1. This pre-selected payment method will be the one used by the system to charge the user on the first of the month.

<u>CREATE</u>

```
INSERT INTO Payment_Methods (paymentMethodID,
            userID,
            isPreSelected,
            cardNumber,
            paymentType)
VALUES      (1,
             'gordon',
             1,
             1111111111,
              'checking')
```

<u>DELETE</u>

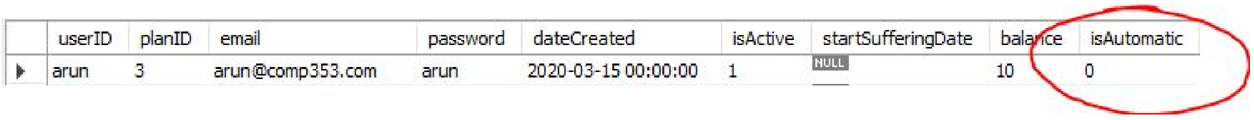
```
DELETE FROM Payment_Methods
WHERE userID = 'gordon' AND paymentMethodID = 1
```

<u>EDIT</u>

```
UPDATE Payment_Methods
SET isPreSelected = 0
WHERE userID = 'gordon' AND paymentMethodID = 1
```

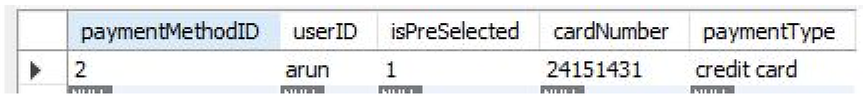
16. Make a manual payment by a user.

From the Users table, we see that this user does not use automatic payments



	userID	planID	email	password	dateCreated	isActive	startSufferingDate	balance	isAutomatic
▶	arun	3	arun@comp353.com	arun	2020-03-15 00:00:00	1	NULL	10	0

In the Payment_Methods table, user ‘arun’ has the following payment methods:

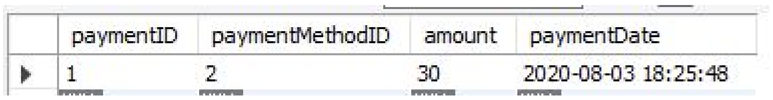


	paymentMethodID	userID	isPreSelected	cardNumber	paymentType
▶	2	arun	1	24151431	credit card

Now when making a payment, we use the following query:

```
INSERT INTO Payments (paymentMethodID, amount)
                VALUES (2, 30);
```

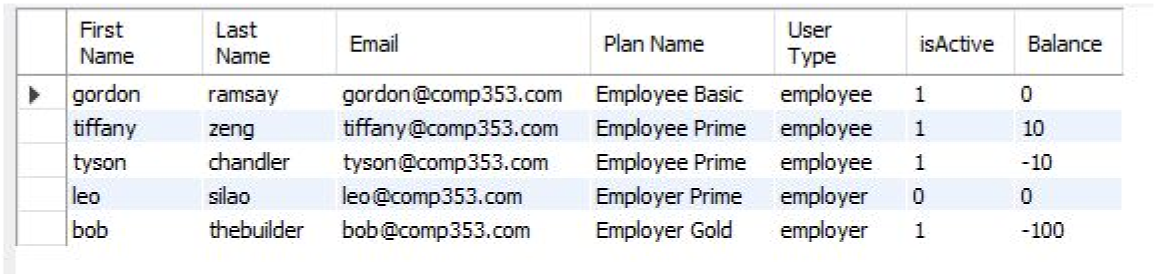
Content in the payments table:



	paymentID	paymentMethodID	amount	paymentDate
▶	1	2	30	2020-08-03 18:25:48

17. Report of all users by the administrator for employers or employees (Name, email, category, status, balance.

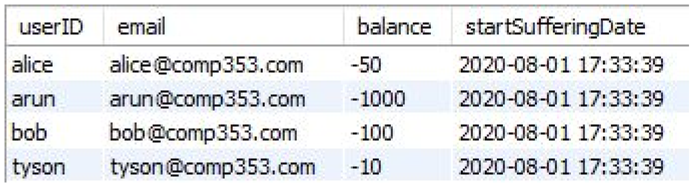
```
SELECT p.firstName as 'First Name', p.lastName as 'Last Name', u.email as 'Email', pl.name
as 'Plan Name', pl.userType as 'User Type', u.isActive, u.balance as 'Balance'
FROM Users as u, Profiles as p, Plans as pl
WHERE u.planID = pl.planID AND u.userID = p.userID
AND pl.userType <> 'admin'
```



	First Name	Last Name	Email	Plan Name	User Type	isActive	Balance
▶	gordon	ramsay	gordon@comp353.com	Employee Basic	employee	1	0
	tiffany	zeng	tiffany@comp353.com	Employee Prime	employee	1	10
	tyson	chandler	tyson@comp353.com	Employee Prime	employee	1	-10
	leo	silao	leo@comp353.com	Employer Prime	employer	0	0
	bob	thebuilder	bob@comp353.com	Employer Gold	employer	1	-100

18. **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;
```



userID	email	balance	startSufferingDate
alice	alice@comp353.com	-50	2020-08-01 17:33:39
arun	arun@comp353.com	-1000	2020-08-01 17:33:39
bob	bob@comp353.com	-100	2020-08-01 17:33:39
tyson	tyson@comp353.com	-10	2020-08-01 17:33:39