Project Report

COMP 5531
Files and Databases
Professor: Nagi Basha
TA: Amirali Ashraf

Group Name: zic55311

Nicholas LaMothe
ID: 40114705

Fady Attia
ID: 40032413

Reasonable Assumptions

We assume, based on the lack of security requirements, that there are no stringent security requirements for this application.

As such, important security practices, such as password hashing, have not been implemented. Passwords and security answers are stored in plaintext. For a hosted application with a real user base, this would be considered unacceptable, but for the purpose of this assignment it seems to be acceptable.

Additionally, we have not used prepared statements for data entered by users. Normally, using prepared statements is the standard method for completely ensuring the impossibility of SQL injections performed via user inputs.

However, mysqli_real_escape_string() has been used for each of the $_POST variables that contains user input. This is generally considered as a good safeguard against SQL injections, but it does not completely remove the risk in the way that prepared statements do.

We have not implemented .htaccess configuration files and Apache directives, which are normally used to control the display of or access to directories within the web server, among other things.

A good deal of validation has been implemented for relevant features, although it is surely incomplete. For the most important features, the user will be immediately notified if the query is successful or if it has failed.

Finally, and importantly, some of the database design does violate 3NF. Our design has at least one relation that is in 2NF. While unnecessary, this helped to simplify certain aspects of querying the database and thus made certain aspects of the back-end code simpler, more readable, and easier to refactor.

Overview of the Project and Functionality

We start the application at the **index.html** page within the root directory. Here, we have the simple decision of clicking one of two buttons, **Employer** or **Submit**.

Employer

Within the login screen for **Employer**, we can either login if we are of one of two user categories: **admin** or **employer**. Any given correct combination of an admin username and password will successfully redirect, as will any given combination of employer username and password. A correct combination of a <regular> user username and password will *not* successfully redirect.

Create New Account

This screen provides the option to create a new employer account. The end user (who is associated with an employer, e.g. a manager, in theory) can enter all of the relevant information and simply create an account, after which the user will be redirected to the login screen.

The value of the employer_ID is necessary to create an account. We assume, for the purpose of this project, that an ID is provided to the employer before they create separate accounts for those who are in charge of hiring.

Retrieve Your Password

From the **Employer** login screen, there is a link to retrieve one's password. As with the login screen, this will only work if the user is either admin or employer. A correct combination of an employer or admin username and security answer will retrieve the password—very insecurely—in plain text on the screen.

Employer Dashboard

This is where the most important functionality exists for an employer (or admin):

Need Help? Contact Us.

This button, located at the top-right-hand corner of the dashboard, will simply call an alert() function that displays a message telling the user to scroll down where they can read the relevant information.

User Maintenance

This first section is only usable by an admin. It allows an administrator to set a user's status to *active* or *inactive*, and this includes both regular user types (jobseekers) and employers. The 'Submit' button must be clicked to perform the action.

Employer Category

This will only function for an employer. It uses the $_SESSION['employer'] variable to change the employer's job_category to either 'Employer Prime' or 'Employer Gold' simply by clicking one of the relevant buttons. For each action, an alert() function is called, and then a message is displayed to the end user.

Note that $_SESSION[] variables are used throughout for authentication.

Post Job

Clearly one of the more important pieces of functionality, this is where an employer can post a job. It will only successfully post a job if all of the fields are filled in and the 'Post Job' button is clicked.

A primary key (job_ID) will automatically be created when a job is created. However, the employer must manually enter their employer_ID. Ideally, the employer should not have to enter this information.

Current Application Summary

The functionality of this section goes beyond the requirements, I believe.

If a user is signed in as an administrator (admin), then the panel will automatically—at a glance—display the most recently posted job out of *all* jobs that have been posted (by any employer).

If, however, an employer is signed in, then the panel will display *specifically* the most recently posted job *for that given employer*.

As such, the view for this functionality is properly restrictive.

Help and Contact

This section provides a brief overview of the functionality for the employer dashboard view, in addition to providing two administrator e-mail addresses in case the end user has any questions or concerns.

Update Job

In this case, having the job_ID is necessary, as an existing job is being updated. In any case, a user can update any number of attributes of a given job provided that the job_ID is entered and the 'Update Job' button is clicked.

Update Application

This feature is to update an application that has already been provided by a user (jobseeker) to an employer. It allows the employer to change the application status (e.g. 'accepted', 'rejected') and to provide a message for the applicant. All fields must be entered and the 'Update Application' button must be clicked.

Jobs Posted By Employer

This panel will output all of the jobs that have been posted by the employer currently in session.

Applications Received By Employer

This panel will output all of the applications that have been received by the employer currently in session.

Applications Accepted By Employer

This panel will output all of the current user's applications that have been accepted by an employer.

Return to Home

Upon redirecting back to the login page, several $_SESSION variables will be unset.

Payments

In the top-right-hand corner of the second employer dashboard panel, there is a button that can be clicked to navigate to the payments page.

Within this page, an employer can add a new payment option, edit a currently existing payment option, or remove a payment option.

Additionally, there is an 'Account Status' panel that tells the user whether their account is in good standing.

Importantly, if an account is frozen, then that user will be denied most of the useful functionality of the application, except for a few features that allow for the manipulation/changing of account settings.

Make a Payment

From the payments view,, there is a button in the top-right-hand corner of the dashboard. This will navigate to a 'Make a Payment' page.

In this page, the employer must enter their account ID, user category, and the total that they wish to pay.

Note that if the employer has insufficient funds in their payments account, then the query will be rejected.

User

Within the login screen for **User**, we can either login if we are of one of two user categories: **admin** or **user**. Any given correct combination of an admin username and password will successfully redirect, as will any given combination of user username and password. A correct combination of an employer username and password will *not* successfully redirect.

Create New Account

This screen provides the option to create a new account. The user can enter all of the relevant information and simply create an account, after which the user will be redirected to the login screen.

Retrieve Your Password

From the **User** login screen, there is a link to retrieve one's password. As with the login screen, this will only work if the user is either admin or user. A correct combination of a user or admin username and security answer will retrieve the password—very insecurely—in plain text on the screen.

User Dashboard

This is where the most important functionality exists for a user (or admin):

Search All Jobs

This yellow button—in the top-left-hand corner—upon being clicked, successfully outputs all of the job postings to the 'Job Data' panel view.

Need Help? Contact Us.

This button, located at the top-right-hand corner of the dashboard, will simply call an alert() function that displays a message telling the user to scroll down where they can read the relevant information.

Search By Category

Upon entering an existing job category (e.g. IT, Engineering, Food Preparation) and clicking the 'Search' button, jobs that are only of this type will be successfully displayed on the 'Job Data' panel view.

Search By Name

This is an additional piece of functionality that is not in the requirements.

Upon entering an existing job name (e.g. Web Developer, System Administrator, Barista) and clicking the 'Search' button, jobs that are only of this title will be successfully displayed on the 'Job Data' panel view

Job Data

This view, as previously stated, displays relevant job data. The logic is functional such that the end user can easily switch between the various search options with no errors or bugs.

Apply for a Job

This piece of functionality allows the end user to apply for a job posted by any employer. A job_ID is required, as is the submission text for the application itself. Everything is else is done by the back end. The end user must click the 'Submit Application' button.

Maintain Status

This feature allows a currently signed-in user to change the status of an application that they have submitted (e.g. 'active', 'inactive'). The end user must provide the ID and the status of the application, and the click the 'Update Status' button.

This feature will *correctly* only make such changes for applications that have been submitted by the user who is currently signed in.

Withdraw Application

This feature allows a currently signed-in user to withdraw an application that they have submitted. The end user must provide the ID, and then click the 'Withdraw' button.

This feature will *correctly* only make such changes for applications that have been submitted by the user who is currently signed in.

Note that this feature does not delete the tuple, but rather nullifies all relevant information.

Help and Contact

This section provides a brief overview of the functionality for the user dashboard view, in addition to providing two administrator e-mail addresses in case the end user has any questions or concerns.

User Category

This will only function for a user. It uses the $_SESSION['user'] variable to change the user's job_category to either 'User Basic', 'User Prime', or 'User Gold' simply by clicking one of the relevant buttons. For each action, an alert() function is called, and then a message is displayed to the end user.

Update User Profile

In this feature, a signed-in user can modify any number of attributes in their current profile/account. It uses the $_SESSION['user'] variable and as such will only make modifications for the user currently signed in. The end user must lcick the 'Update Profile' button to complete the action.

Delete User Account

A longwinded and sardonic message is provied to the end user, asking for the user's certainty in the decision that they are about to go through with.

In order to "delete" a user account, the user must enter 'Yes' into the input box, in addition to clicking the 'Delete Account' button. The $_SESSION['user'] variable is once again used so that only the signed-in user's account can be deleted.

Note that this option does not actually delete the user account tuple, but rather it nullifies all relevant information... except for the username, which must remain for other reasons.

Applications Sent By User

This panel will display all of the job applications that have been sent by the user currently in session.

Applications Accepted

This panel will display all of the current user's job applications that were **accepted** by the employer. (In other words, they got the job... or several!)

This panel will display all of the current user's job applications that were **rejected** by the employer.

Return to User Login

Upon redirecting back to the login page, several $_SESSION variables will be unset.

Payments

In the top-right-hand corner of the second user dashboard panel, there is a button that can be clicked to navigate to the payments page.

Within this page, a user can add a new payment option, edit a currently existing payment option, or remove a payment option.

Additionally, there is an 'Account Status' panel that tells the user whether their account is in good standing.

Importantly, if an account is frozen, then that employer will be denied most of the useful functionality of the application, except for a few features that allow for the manipulation/changing of account settings.

Make a Payment

From the payments view, there is a button in the top-right-hand corner of the dashboard. This will navigate to a 'Make a Payment' page.

In this page, the user must enter their account ID, user category, and the total that they wish to pay.
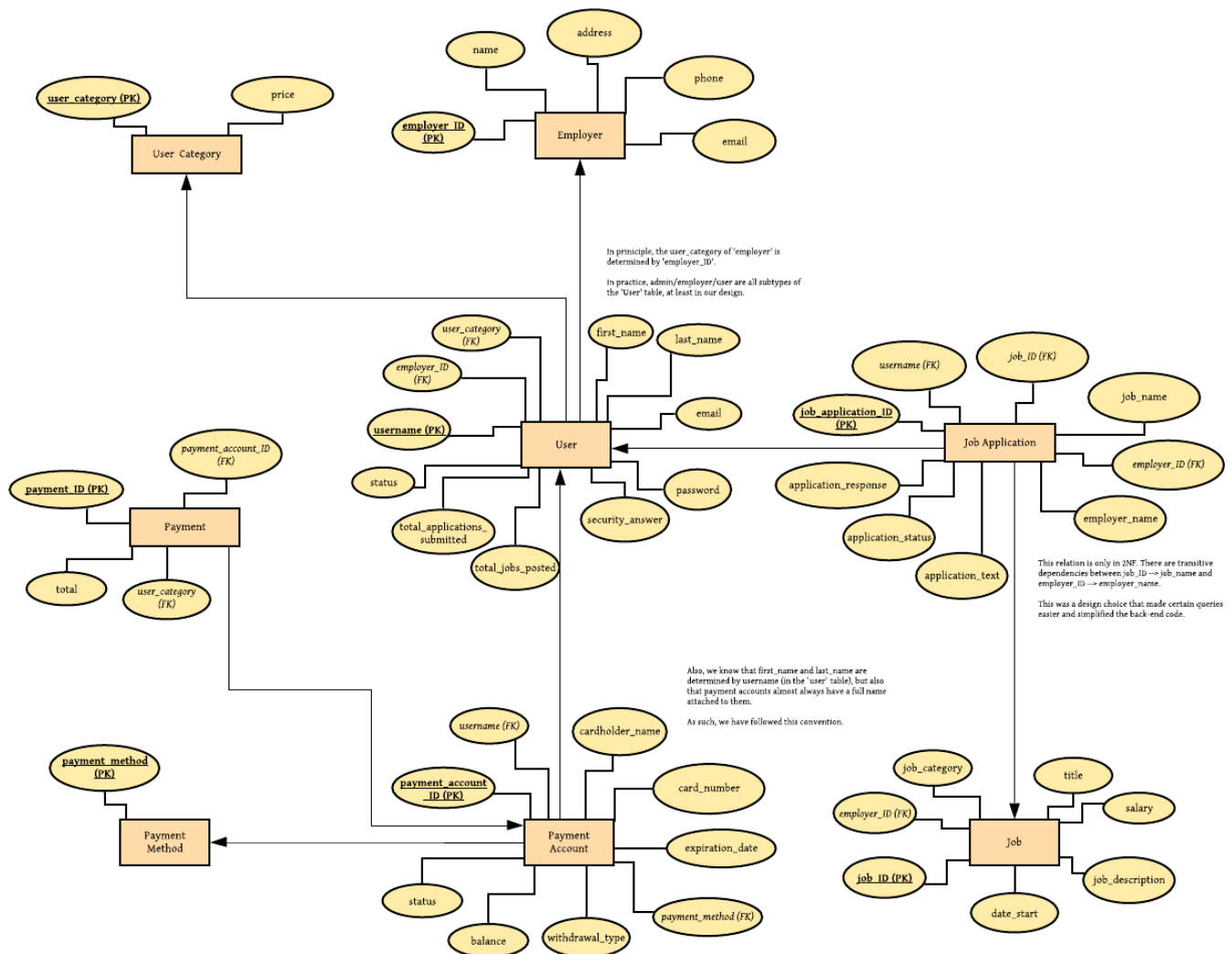
Note that if the user has insufficient funds in their payments account, then the query will be rejected.

Web Career Portal - ER Diagram

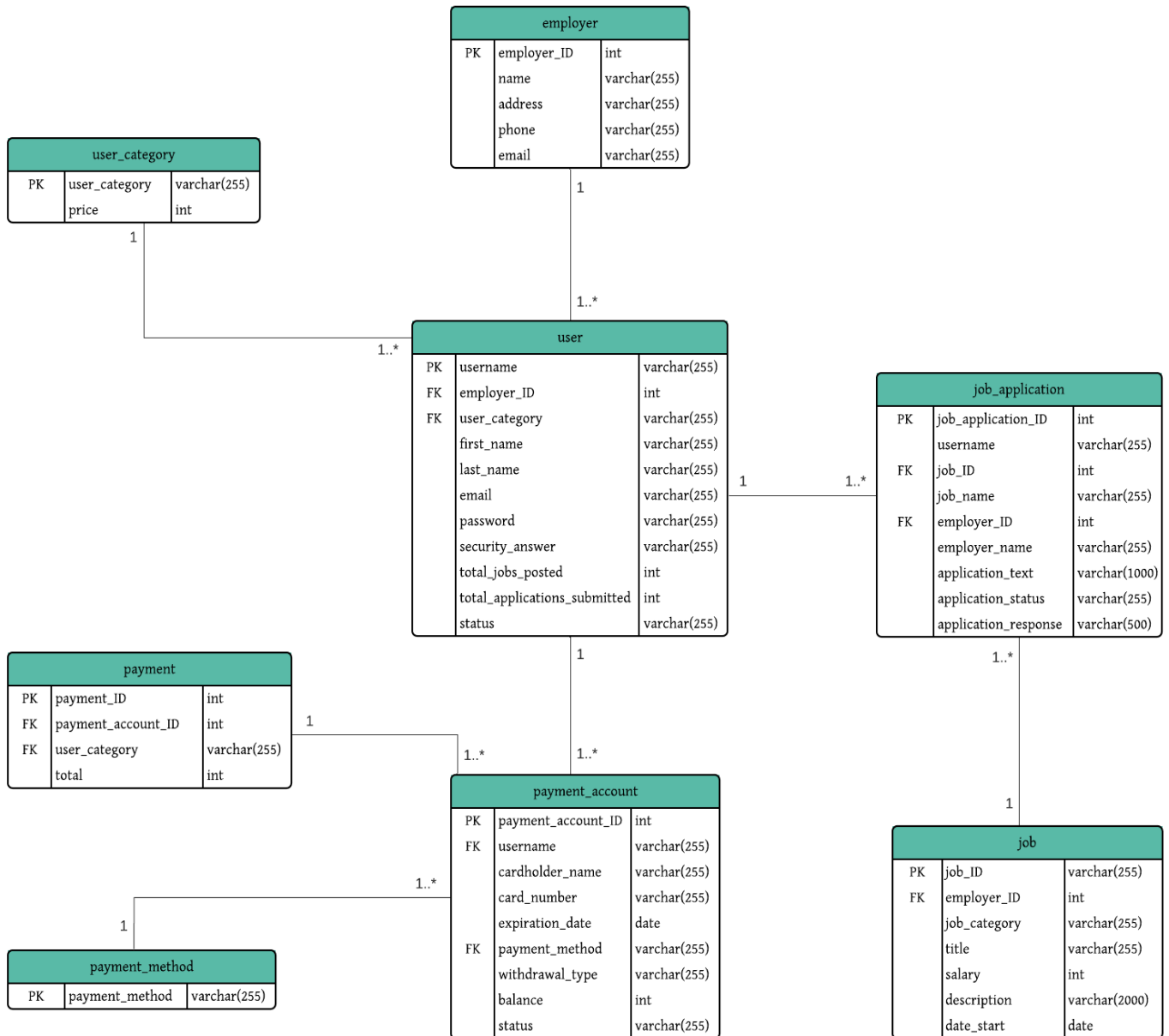Nicholas LaMothe / Fady Attia -- August 15, 2021

Quick overview:

- Primary keys are notated: **underlined_and_bold (PK)**.

- If a foreign key forms part of the primary key, it will be shown alongside the **primary_key (PK, FK)**.

- Non-primary key foreign keys are *italicized (FK)*..

- Two arrowheads indicates a one-to-one relationship.

- An arrowhead and a straight line is one-to-many.

- Simply a straight line will indicate that it is many-to-many

User Category
- user_category (PK)
- price

Employer
- name
- address
- phone
- employer_ID (PK)
- email

In priniciple, the user_category of 'employer' is determined by 'employer_ID'.

In practice, admin/employer/user are all subtypes of the 'User' table, at least in our design.

User
- user_category (FK)
- first_name
- last_name
- employer_ID (FK)
- email
- username (PK)
- status
- password
- total_applications_submitted
- security_answer
- total_jobs_posted

Job Application
- username (FK)
- job_ID (FK)
- job_name
- job_application_ID (PK)
- employer_ID (FK)
- application_response
- employer_name
- application_status
- application_text

This relation is only in 2NF. There are transitive dependencies between job_ID --> job_name and employer_ID --> employer_name.

This was a design choice that made certain queries easier and simplified the back-end code.

Payment
- payment_account_ID (FK)
- payment_ID (PK)
- total
- user_category (FK)

Also, we know that first_name and last_name are determined by username (in the 'user' table), but also that payment accounts almost always have a full name attached to them.

As such, we have followed this convention.

Payment Method
- payment_method (PK)

Payment Account
- username (FK)
- cardholder_name
- payment_account_ID (PK)
- card_number
- expiration_date
- status
- payment_method (FK)
- balance
- withdrawal_type

Job
- job_category
- title
- employer_ID (FK)
- salary
- job_ID (PK)
- job_description
- date_start

Web Career Portal - Conceptual Schema

Nicholas LaMothe / Fady Attia -- August 15, 2021

employer

PK	employer_ID	int
	name	varchar(255)
	address	varchar(255)
	phone	varchar(255)
	email	varchar(255)

user_category

PK	user_category	varchar(255)
	price	int

user

PK	username	varchar(255)
FK	employer_ID	int
FK	user_category	varchar(255)
	first_name	varchar(255)
	last_name	varchar(255)
	email	varchar(255)
	password	varchar(255)
	security_answer	varchar(255)
	total_jobs_posted	int
	total_applications_submitted	int
	status	varchar(255)

job_application

PK	job_application_ID	int
	username	varchar(255)
FK	job_ID	int
	job_name	varchar(255)
FK	employer_ID	int
	employer_name	varchar(255)
	application_text	varchar(1000)
	application_status	varchar(255)
	application_response	varchar(500)

payment

PK	payment_ID	int
FK	payment_account_ID	int
FK	user_category	varchar(255)
	total	int

payment_account

PK	payment_account_ID	int
FK	username	varchar(255)
	cardholder_name	varchar(255)
	card_number	varchar(255)
	expiration_date	date
FK	payment_method	varchar(255)
	withdrawal_type	varchar(255)
	balance	int
	status	varchar(255)

payment_method

PK	payment_method	varchar(255)

job

PK	job_ID	varchar(255)
FK	employer_ID	int
	job_category	varchar(255)
	title	varchar(255)
	salary	int
	description	varchar(2000)
	date_start	date

Database Schema

Relation:`employer`

```
`employer_ID` INT AUTO_INCREMENT,
`name` VARCHAR(255),
`address` VARCHAR(255),
`phone` VARCHAR(255),
`email` VARCHAR(255),
 PRIMARY KEY (employer_ID)
```

Relation:`user_category`

```
`user_category` VARCHAR(255),
`price` INT,
 PRIMARY KEY (user_category)
```

Relation: `user`

```
`username` VARCHAR(255),
`employer_ID` INT NULL,
`user_category` VARCHAR(255),
`first_name` VARCHAR(255),
`last_name` VARCHAR(255),
`email` VARCHAR(255),
`password` VARCHAR(255),
`security_answer` VARCHAR(255),
`total_jobs_posted` INT,
`total_applications_submitted` INT,
`status` VARCHAR(255), -- active, inactive
 PRIMARY KEY (username),
 FOREIGN KEY (employer_ID) REFERENCES employer (employer_ID),
 FOREIGN KEY (user_category) REFERENCES user_category (user_category)
```

Relation: `payment_method`

```
`payment_method` VARCHAR(255), -- Chequing, Credit
 PRIMARY KEY (payment_method)
```

Relation: `payment_account`

`payment_account_ID` INT AUTO_INCREMENT,
`username` VARCHAR(255),
`cardholder_name` VARCHAR(255),
`card_number` VARCHAR(255),
`expiration_date` DATE,
`payment_method` VARCHAR(255), -- Chequing, Credit
`withdrawal_type` VARCHAR(255), -- Manual, Automatic
`balance` INT,
`status` VARCHAR(255), -- Settled, Frozen
PRIMARY KEY (payment_account_ID),
FOREIGN KEY (username) REFERENCES user (username),
FOREIGN KEY (payment_method) REFERENCES payment_method
(payment_method)

Relation: `payment`

`payment_ID` INT AUTO_INCREMENT,
`payment_account_ID` INT,
`user_category` VARCHAR(255),
`total` INT,
PRIMARY KEY (payment_ID),
FOREIGN KEY (payment_account_ID) REFERENCES payment_account
(payment_account_ID),
FOREIGN KEY (user_category) REFERENCES user_category (user_category)

Relation: `job`

`job_ID` INT AUTO_INCREMENT,
`employer_ID` INT,
`job_category` VARCHAR(255),
`title` VARCHAR(255),
`salary` INT,
`description` VARCHAR(2000), -- much larger text
`date_start` DATE,
PRIMARY KEY (job_ID),
FOREIGN KEY (employer_ID) REFERENCES employer (employer_ID)

Relation: `job_application`

    `job_application_ID` INT AUTO_INCREMENT,
    `username` VARCHAR(255),
    `job_ID` INT,
    `job_name` VARCHAR(255),
    `employer_ID` INT,
    `employer_name` VARCHAR(255),
    `application_text` VARCHAR(1000),
    `application_status` VARCHAR(255), -- active, inactive, accepted, rejected
    `application_response` VARCHAR(500),
    PRIMARY KEY (job_application_ID),
    FOREIGN KEY (username) REFERENCES user (username),
    FOREIGN KEY (job_ID) REFERENCES job (job_ID),
    FOREIGN KEY (employer_ID) REFERENCES employer (employer_ID)

SQL Queries

```
-- i.
-- Create/delete/edit/display an employer
INSERT INTO `employer` VALUES (DEFAULT, 'Business Name', '555 Falsehood Boulevard', '555-515-2314', 'businessname@notreal.fake');
UPDATE `employer` SET employer.name = 'Different Business Name' WHERE employer.name = 'Business Name';
SELECT * FROM `employer` WHERE employer.name = 'Different Business Name';
DELETE FROM `employer` WHERE employer.name = 'Different Business Name';
```

✓	71	16:22:29	INSERT INTO `employer` VALUES (DEFAULT, 'Business Name', '555 Falsehood Boulevard', '555-515-2314', 'businessname@notreal.fake')	1 row(s) affected
✓	72	16:22:29	UPDATE `employer` SET employer.name = 'Different Business Name' WHERE employer.name = 'Business Name'	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0
✓	73	16:22:29	SELECT * FROM `employer` WHERE employer.name = 'Different Business Name' LIMIT 0, 1000	1 row(s) returned
✓	74	16:22:29	DELETE FROM `employer` WHERE employer.name = 'Different Business Name'	1 row(s) affected

```
-- ii.
-- Create/delete/edit/display a category by an employer
INSERT INTO `employer` VALUES (DEFAULT, 'Different Business Name', '555 Falsehood Boulevard', '555-515-2314', 'businessname@notreal.fake');
UPDATE `employer` SET employer.email = 'differentbusiness@fake.notreal' WHERE employer.name = 'Different Business Name';
SELECT * FROM `employer` WHERE employer.email = 'differentbusiness@fake.notreal';
DELETE FROM `employer` WHERE employer.email = 'differentbusiness@fake.notreal';
```

✓	75	16:25:03	INSERT INTO `employer` VALUES (DEFAULT, 'Different Business Name', '555 Falsehood Boulevard', '555-515-2314', 'businessname@notreal....	1 row(s) affected
✓	76	16:25:03	UPDATE `employer` SET employer.email = 'differentbusiness@fake.notreal' WHERE employer.name = 'Different Business Name'	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0
✓	77	16:25:03	SELECT * FROM `employer` WHERE employer.email = 'differentbusiness@fake.notreal' LIMIT 0, 1000	1 row(s) returned
✓	78	16:25:03	DELETE FROM `employer` WHERE employer.email = 'differentbusiness@fake.notreal'	1 row(s) affected
✓	80	16:27:39	INSERT INTO `job` VALUES (DEFAULT, 1, 'IT', 'Network Analyst', 110000, 'Here at Alpha Computing, we are on the lookout for a network ana...	1 row(s) affected

```
-- iii.
-- Post a new job by an employer
INSERT INTO `job` VALUES (DEFAULT, 1, 'IT', 'Network Analyst', 110000, 'Here at Alpha Computing, we are on the lookout for a network analyst who posesses at least
10 years of experience in networking solutiosn. A MSc in Computer Science, Information Technology, or Cybersecurity is required.', '2021-09-07');
```

|				
✓	80	16:27:39	INSERT INTO `job` VALUES (DEFAULT, 1, 'IT', 'Network Analyst', 110000, 'Here at Alpha Computing, we are on the lookout for a network ana...	1 row(s) affected

```
-- iv.
-- Provide a job offer for an employee by an employerr
UPDATE `job_application` SET application_status='accepted',
application_response='It is with pleasure that you contacting you to provide a job offer. Based on your credentials, experience,
and your performance on the tests, we believe that you have the requisite qualifications for this positions. We are so certain of
this that we intend to offer you $120000 per year for this position. Please provide us with your response within a week\'s time'
WHERE (job_application_ID = 7 AND username = 'gord' AND job_ID = 3);
```

✓	79	16:26:19	UPDATE `job_application` SET application_status='accepted', application_response='It is with pleasure that you contacting you to provide a jo...	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0

```sql
-- v.
-- Report of a posted job by an employer (job title, job description, date posted, list of employees applied to job, status of each application)
SELECT job.title, job.description, job.date_start, job_application.username, job_application.application_status
FROM `job`, `job_application`
WHERE job.job_ID = job_application.job_ID
AND job.title = 'System Administrator';
```

title	description	date_start	username	application_status
System Administrator	This role requires knowledge of the system admi...	2021-08-27	zeba	inactive
System Administrator	This role requires knowledge of the system admi...	2021-08-27	gord	accepted

	81	16:28:39	SELECT job.title, job.description, job.date_start, job_application.username, job_application.application_status FROM `job`, `job_application` WH...	2 row(s) returned

```sql
-- vi.
-- Report of posted jobs by an employer during a specific period of time (job title, date posted, short description, number of needed employees, number of accepted offers)
SELECT DISTINCT job.title, job.date_start, job.description, job_application.application_status
FROM job, job_application
WHERE date_start BETWEEN '2021-06-01' AND '2021-08-31'
AND job_application.application_status = 'accepted';
```

title	date_start	description	application_status
System Administrator	2021-08-27	This role requires knowledge of the system admi...	accepted
Electrical Engineer	2021-08-28	This role requires knowledge of circuit design an...	accepted
Finishing Carpenter	2021-08-16	We pride ourselves on our careful, classic finishi...	accepted
Technical Support	2021-08-26	This is a level-2 support position that requires a ...	accepted
Technical Support	2021-08-28	At Jimba Microprocessors, we truly value our rel...	accepted

	184	17:29:07	SELECT DISTINCT job.title, job.date_start, job.description, job_application.application_status FROM job, job_application WHERE date_start B...	5 row(s) returned

```sql
-- vii.
-- Create/delete/edit/display a category by an employer
INSERT INTO `user` VALUES ('new_user', NULL, 'User Basic', 'UserFirstName', 'UserLastName', 'UserEmail' ,'Password12!' ,'MySecurityAnswer', 0, 0,'active');
UPDATE `user` SET user.first_name = 'Updated first_name' WHERE user.first_name = 'UserFirstName';
SELECT * FROM `user` WHERE user.first_name = 'Updated first_name';
DELETE FROM `user` WHERE user.first_name = 'Updated first_name';
```

username	employer_ID	user_category	first_name	last_name	email	password	security_answer	total_jobs_posted	total_applications_submitted	status
new_user	NULL	User Basic	Updated first_name	UserLastName	UserEmail	Password12!	MySecurityAnswer	0	0	active
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

|	82	16:30:20	INSERT INTO `user` VALUES ('new_user', NULL, 'User Basic', 'UserFirstName', 'UserLastName', 'UserEmail' ,'Password12!' ,'MySecurityAnswe...	1 row(s) affected
	83	16:30:20	UPDATE `user` SET user.first_name = 'Updated first_name' WHERE user.first_name = 'UserFirstName'	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0
	84	16:30:20	SELECT * FROM `user` WHERE user.first_name = 'Updated first_name' LIMIT 0, 1000	1 row(s) returned
	85	16:30:20	DELETE FROM `user` WHERE user.first_name = 'Updated first_name'	1 row(s) affected

```sql
-- viii.
-- Search for a job by an employee
SELECT *
FROM `job`
WHERE job.title = 'Technical Support';
```

job_ID	employer_ID	job_category	title	salary	description	date_start
11	1	IT	Technical Support	45000	This is a level-2 support position that requires a ...	2021-08-26
12	3	IT	Technical Support	55000	At Jimba Microprocessors, we truly value our rel...	2021-08-28
NULL	NULL	NULL	NULL	NULL	NULL	NULL

	86	16:33:19	SELECT * FROM `job` WHERE job.title = 'Technical Support' LIMIT 0, 1000	2 row(s) returned

```
-- ix.
-- Apply for a job by an employee
INSERT INTO `job_application` VALUES (DEFAULT, 'zeba', 4, 'Barista', 4, 'Kettle Coffee',
'Barista has always been my passion. The art of creating something form basic ingredients is a wow for me.', 'active', NULL);
```

| | 87 | 16:34:28 | INSERT INTO `job_application` VALUES (DEFAULT, 'zeba', 4, 'Barista', 4, 'Kettle Coffee', 'Barista has always been my passion. The art of cre... | 1 row(s) affected |

```
-- x.
-- Accept/deny a job offer by an employee
UPDATE `job_application` SET application_status='accepted' WHERE (job_application.job_application_ID = 4 AND job_application.username = 'damo');
UPDATE `job_application` SET application_status='rejected' WHERE (job_application.job_application_ID = 4 AND job_application.username = 'damo');
```

| | 105 | 16:40:02 | UPDATE `job_application` SET application_status='accepted' WHERE (job_application.job_application_ID = 4 AND job_application.username ... | 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0 |
| | 106 | 16:40:02 | UPDATE `job_application` SET application_status='rejected' WHERE (job_application.job_application_ID = 4 AND job_application.username =... | 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0 |

```
-- xi.
-- Withdraw from an applied job by an employee
UPDATE `job_application` SET username=NULL, job_ID=NULL, job_name=NULL, employer_ID=NULL, employer_name=NULL, application_text=NULL, application_status=NULL,
application_response=NULL WHERE (job_application.job_application_ID = 8 AND job_application.username = 'zeba');
```

| | 115 | 16:41:57 | UPDATE `job_application` SET username=NULL, job_ID=NULL, job_name=NULL, employer_ID=NULL, employer_name=NULL, application_te... | 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0 |

```
-- xii.
-- Delete a profile by an employee
UPDATE `user` SET employer_ID=NULL, user_category=NULL, first_name=NULL, last_name=NULL, email=NULL, password=NULL, security_answer=NULL,
total_jobs_posted=NULL, total_applications_submitted=NULL, status='inactive' WHERE user.username='j_randal';
```

| | 124 | 16:43:26 | UPDATE `user` SET employer_ID=NULL, user_category=NULL, first_name=NULL, last_name=NULL, email=NULL, password=NULL, security_... | 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0 |

```
-- xiii
-- Report of applied jobs by an employee during a specific period of time (job title, date applied, short description, status)
SELECT job_application.job_name, job_application.application_text, job_application.application_status
FROM job_application, job
WHERE job_application.job_ID = job.job_ID
AND job.date_start BETWEEN '2021-06-01' AND '2021-08-31'
AND job_application.username = 'gord';
```

Filter Rows: | Export: | Wrap Cell Content: |

job_name	application_text	application_status
System Administrator	20+ years experience in computer architecture, includin...	accepted
Electrical Engineer	20+ years experience in computer architecture, includin...	accepted

| | 214 | 17:41:03 | SELECT job_application.job_name, job_application.application_text, job_application.application_status FROM job_application, job WHERE job... | 2 row(s) returned |

```
-- xiv.
-- Add/delete/edit a method of payment by a user.
INSERT INTO `payment_account` VALUES (DEFAULT, 'alpha', 'Ali Grandich', '4839818028408122', '2021-09-01', 'Chequing', 'Automatic', 8400, 'Settled');
UPDATE `payment_account` SET payment_account.payment_method ='Credit' WHERE payment_account.payment_method = 'Chequing' AND payment_account.username = 'alpha';
SELECT * FROM `payment_account` WHERE payment_account.payment_method = 'Credit' AND payment_account.username = 'alpha';
```

payment_account_ID	username	cardholder_name	card_number	expiration_date	payment_method	withdrawal_type	balance	status
2	alpha	Ali Grandich	4759201938233423	2021-08-28	Credit	Manual	2000	Settled
3	alpha	Ali Grandich	4757920390238432	2022-02-24	Credit	Automatic	4000	Settled
5	alpha	Ali Grandich	4839818028408123	2021-09-01	Credit	Automatic	9402	Settled
13	alpha	Ali Grandich	4839818028408122	2021-09-01	Credit	Automatic	8400	Settled
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
UPDATE `payment_account` SET username = NULL, cardholder_name = NULL, card_number = NULL, expiration_date = NULL, payment_method = NULL, withdrawal_type = NULL,
balance = NULL, status = NULL WHERE payment_account_ID = 13;
```

✓	127 16:45:26	INSERT INTO `payment_account` VALUES (DEFAULT, 'alpha', 'Ali Grandich', '48398180284081', '2021-09-01', 'Chequing', 'Automatic', 8400, '...	1 row(s) affected
✓	128 16:45:26	UPDATE `payment_account` SET payment_account.payment_method ='Credit' WHERE payment_account.payment_method = 'Chequing' AN...	3 row(s) affected Rows matched: 3 Changed: 3 Warnings: 0
✓	129 16:45:26	SELECT * FROM `payment_account` WHERE payment_account.payment_method = 'Credit' AND payment_account.username = 'alpha' LIMIT ...	4 row(s) returned
✓	131 16:51:01	UPDATE `payment_account` SET username = NULL, cardholder_name = NULL, card_number = NULL, expiration_date = NULL, payment_met...	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0

```
-- xv.
-- Add/delete/edit an automatic payment by a user
INSERT INTO `payment_account` VALUES (DEFAULT, 'alpha', 'Ali Grandich', '48398180284081', '2021-09-01', 'Chequing', 'Automatic', 2200, 'Settled');
UPDATE `payment_account` SET payment_account.withdrawal_type ='Manual' WHERE payment_account.withdrawal_type = 'Automatic' AND payment_account.username = 'alpha';
SELECT * FROM `payment_account` WHERE payment_account.withdrawal_type = 'Manual' AND payment_account.username = 'alpha';
UPDATE `payment_account` SET username = NULL, cardholder_name = NULL, card_number = NULL, expiration_date = NULL, payment_method = NULL, withdrawal_type = NULL,
balance = NULL, status = NULL WHERE payment_account_ID = 14 AND payment_account.withdrawal_type = 'Automatic';
```

payment_account_ID	username	cardholder_name	card_number	expiration_date	payment_method	withdrawal_type	balance	status
2	alpha	Ali Grandich	4759201938233423	2021-08-28	Credit	Manual	2000	Settled
3	alpha	Ali Grandich	4757920390238432	2022-02-24	Credit	Manual	4000	Settled
5	alpha	Ali Grandich	4839818028408123	2021-09-01	Credit	Manual	9402	Settled
14	alpha	Ali Grandich	48398180284081	2021-09-01	Chequing	Manual	2200	Settled
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

✓	133 16:54:42	INSERT INTO `payment_account` VALUES (DEFAULT, 'alpha', 'Ali Grandich', '48398180284081', '2021-09-01', 'Chequing', 'Automatic', 2200, '...	1 row(s) affected
✓	134 16:54:42	UPDATE `payment_account` SET payment_account.withdrawal_type ='Manual' WHERE payment_account.withdrawal_type = 'Automatic' AN...	3 row(s) affected Rows matched: 3 Changed: 3 Warnings: 0
✓	135 16:54:42	SELECT * FROM `payment_account` WHERE payment_account.withdrawal_type = 'Manual' AND payment_account.username = 'alpha' LIMIT ...	4 row(s) returned
✓	136 16:54:42	UPDATE `payment_account` SET username = NULL, cardholder_name = NULL, card_number = NULL, expiration_date = NULL, payment_met...	0 row(s) affected Rows matched: 0 Changed: 0 Warnings: 0

```
-- xvi.
-- Make a manual payment by a user.
INSERT INTO `payment` VALUES (DEFAULT, 1, 'User Gold', 20);
```

✓	145 16:56:59	INSERT INTO `payment` VALUES (DEFAULT, 1, 'User Gold', 20)	1 row(s) affected

```
-- xvii.
-- Report of all users by the administrator for employers or employees (Name, email, category, status, balance)
SELECT user.first_name, user.last_name, user.email, user.user_category, user.status, payment_account.balance
FROM `user`, `payment_account`
WHERE user.username = payment_account.username;
```

Filter Rows: [   ]   Export: ▦ | Wrap Cell Content: I̲A

first_name	last_name	email	user_category	status	balance
Jim	Jameson	jimjameson@stillfake.notreal	User Gold	active	4500
Ali	Grandich	aligrandy@alpha.org	Employer Prime	active	2000
Ali	Grandich	aligrandy@alpha.org	Employer Prime	active	4000
Damo	Suzuki	damo@mysticalvoice.org	User Basic	active	-6000
Ali	Grandich	aligrandy@alpha.org	Employer Prime	active	9402
Darryl	Randal	darryl@fakeman.com	Employer Prime	active	20391
Gord	Willard	gordwillard@fake.net	User Gold	active	4820
Gord	Willard	gordwillard@fake.net	User Gold	active	603
Stanley	Silverman	stanley@fakest.ca	Employer Gold	active	302
Stanley	Silverman	stanley@fakest.ca	Employer Gold	active	48920
NULL	NULL	NULL	NULL	inactive	294
Sarah	Wilkinson	sandra@kettlecoffeeisgood...	Employer Gold	active	2201
Ali	Grandich	aligrandy@alpha.org	Employer Prime	active	2200

✅ 148 16:57:49 SELECT user.first_name, user.last_name, user.email, user.user_category, user.status, payment_account.balance FROM `user`, `payment_acco...   13 row(s) returned

```
-- xviii.
-- Report of all outstanding balance accounts (e.g. where money is owing, where there is debt)
SELECT * FROM payment_account
WHERE balance < 0;
```

Filter Rows: [   ]   Edit: 🖉 🖩 🖩 | Export/Import: 🖩 🖩 | Wrap Cell Content: I̲A

payment_account_ID	username	cardholder_name	card_number	expiration_date	payment_method	withdrawal_type	balance	status
4	damo	Damo Suzuki	4759204992409222	2022-03-16	Credit	Automatic	-6000	Frozen
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

✅ 149 16:58:40 SELECT * FROM payment_account WHERE balance < 0 LIMIT 0, 1000   1 row(s) returned

Normalization

Employer – 3NF

employer_ID	name	address	phone	email
1	Alpha Computing	515 Alpha Way	555-515-5245	alphacomputing@alpha.org
2	Darryl Electronics	654 Simba Way	555-515-6216	darrylelectro@darrylelectro.com
3	Jimba Microprocessors	721 Jimba Ave	555-425-5215	jimbamicropro@jimba.net
4	Kettle Coffee	421 Kettle Boulevard	555-745-2910	kettle@kettlecoffeeisgood.ca
5	Stan\'s Bagels	425 Bagel Road	521-542-4919	stanley@stansbagels.com
6	Tony Pizzeria	234 Tony Street	512-492-1928	tony@tonypizzanow.org
7	Denton Photography	555 Alpha Way	555-234-8239	denton@dentonphoto.ca
8	Fine Refinishing	7820 Wood Way	514-849-2938	refinish@finewood.now
9	James Richardson Brandy	514 Avenue Decadent	512-481-2381	jrichardson@bestbrandies.org
10	Smoke Show	290 Smoker Boulevard	555-512-1932	smokeshow@smokeshowsmokeables.org

Primary Key

user

username	employer_ID	user_category	first_name	last_name	email	password	security_answer	total_jobs_posted	total_applications_submitted	status
n_lamo	NULL	Admin	Nicholas	LaMothe	n_lamo@encs.concordia.ca	steppenwolf	Vertigo	0	0	active
f_attia	NULL	Admin	Fady	Attia	f_attia@encs.concordia.ca	password	Unknown	0	0	active
zeba	NULL	User Basic	Jim	James	jimmyjames@fake.org	jimmy	Unforgiven	0	2	active
damo	NULL	User Prime	Damo	Suzuki	damo@mysticalvoice.org	damo	Rashomon	0	2	active
gord	NULL	User Gold	Gord	Willard	gordwillard@fake.net	gordo	Munich	0	3	active
alpha	1	Employer Prime	Ali	Grandich	aligrandy@alpha.org	alpha	Slackers	1	0	active
darryl	2	Employer Prime	Darryl	Randal	darryl@fakeman.com	darryl	Clerks	1	0	active
jimba	3	Employer Gold	Jim	Brando	jimbrando@faker.net	jimba	Akira	1	0	active
kettle	4	Employer Gold	Sarah	Wilkinson	sandra@kettlecoffeeisgood.ca	sandra	Sleepless in Sea	1	0	active
stanley	5	Employer Gold	Stanley	Silverman	stanley@fakest.ca	stanley	The Conversation	1	0	active

Primary Key

Foreign Key

user_category

user_category	price
Admin	0
Employer Prime	50
Employer Gold	100
User Basic	0
User Prime	10
User Gold	20

This relationship is Many to one as 2 unique user can choose the same category eg User Basic

job

job_ID	employer_ID	job_category	title	salary	descritpion	date_start
1	1	IT	System Administrator	80000		2021-08-30
2	2	Engineering	Electrical Engineer	92000		2021-09-02
3	3	Engineering	Computer Architect	120000		2021-10-01
4	4	Food Preparat	Barista	26000		2021-09-28
5	5	IT	Web Developer	50000		2021-09-01
6	6	Food Preparat	Line Cook	32000		2021-09-15
7	7	Photography	Wedding Photographer	44000		2021-08-16
8	8	Carpentry	Finishing Carpenter	65000		2021-09-20
9	9	Service	Front of House	45000		2021-09-15
10	10	Service	Retail Clerk	27500		

job_application

job_application_ID	username	job_ID	job_name	employer_ID	employer_name	application_text	application_stat	application_response
1	zeba	1	System Administrator	1	Alpha Computing		active	NULL
2	zeba	5	Web Developer	5	Stan\s Bagels		active	NULL
3	damo	4	Barista	4	Kettle Coffee		active	NULL
4	damo	6	Line Cook	6	Tony Pizzeria		active	NULL
5	gord	1	System Administrator	1	Alpha Computing		active	NULL
6	gord	2	Electrical Engineer	2	Darryl Electronics		active	NULL
7	gord	3	Computer Architect	3	Jimba Microprocessors		active	NULL

job_application

job_application_ID	username	job_ID	application_status	application_response
1	zeba	1	active	NULL
2	zeba	5	active	NULL
3	damo	4	active	NULL
4	damo	6	active	NULL
5	gord	1	active	NULL
6	gord	2	active	NULL
7	gord	3	active	NULL

Foreign Key

Contributions

Nicholas:
- HTML
- CSS
- PHP
- SQL / SQL Queries
- ER Diagram
- Conceptual Schema
- Database Design
- Documentation
- Project Report

Fady:
- SQL Queries
- Normalization
- Project Report
- Functional Testing
- Minor Features / Touch-ups