**Project Report**

|  |  |
| --- | --- |
| **Product Name** | Advanced Certificate in Web Development |
| **Qualification Name (NICF)** | NICF-Advanced Certificate in Infocomm Technology (Software & Applications) |
| **Product Name** | Database Design and Implementation |
| **Module Name (NICF)** | ITSF-Database Design and Implementation |

|  |  |  |  |
| --- | --- | --- | --- |
| **Student name** | | **Assessor name** | |
| Darren Farrell | |  | |
| **Date issued** | **Completion date** | | **Submitted on** |
|  |  | |  |
|  | | | |
| **Project title** | Design, Implement, Test & Document Community Portal Database. | | |

|  |
| --- |
| **Learner declaration** |
| I certify that the work submitted for this assignment is my own and research sources are fully acknowledged.  Student signature: Date: 26/01/2023 |

**Content**

1. Project background
2. Project Objectives
3. Database Requirement Specifications
4. Task 1
5. Task 2
6. Task 3
7. Task 4
8. Task 5
9. Task 6

**Project Background**

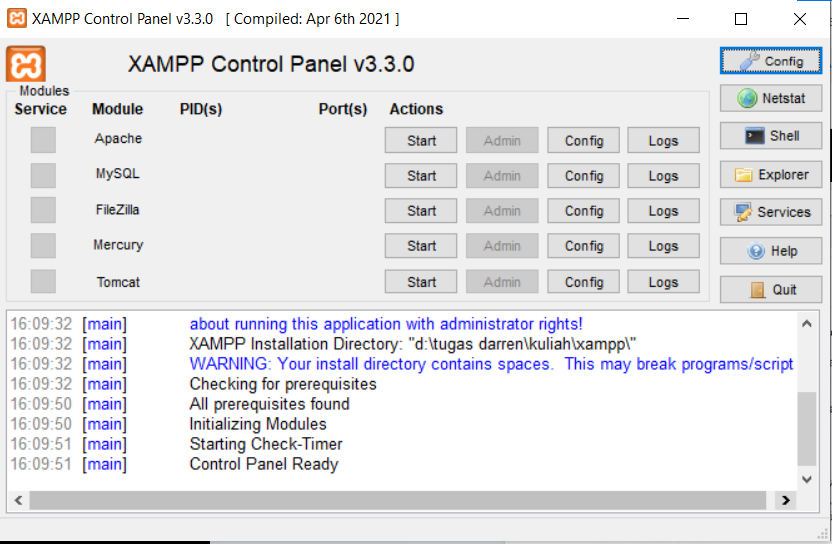
We have been contacted by ABC Jobs to develop a community portal for programmer. The community portal will be used by 2 kinds of users which is administrator and programmer. The administrator should be able to manage the users and also send bulk email to other people. The users should be able to login, register, request forgotten password and update their information at their will. The users can also search for other users using various parameters such as their name, country, job, experience to view their profile. They can also send each other messages and post and reply to thread, post job opportunities. In the previous module we have created the prototype for the community portal so in this module we will be creating the database to store all of our users data.

**Project Objectives**

The objective of this module is to create a functioning database management system for our community portal using MySQL workbench or PHPMyAdmin.

Tools and platform used:

* XAMPP, to use phpmyadmin



* Mockaroo, To generate sample data



* Excel, to do normalization on data



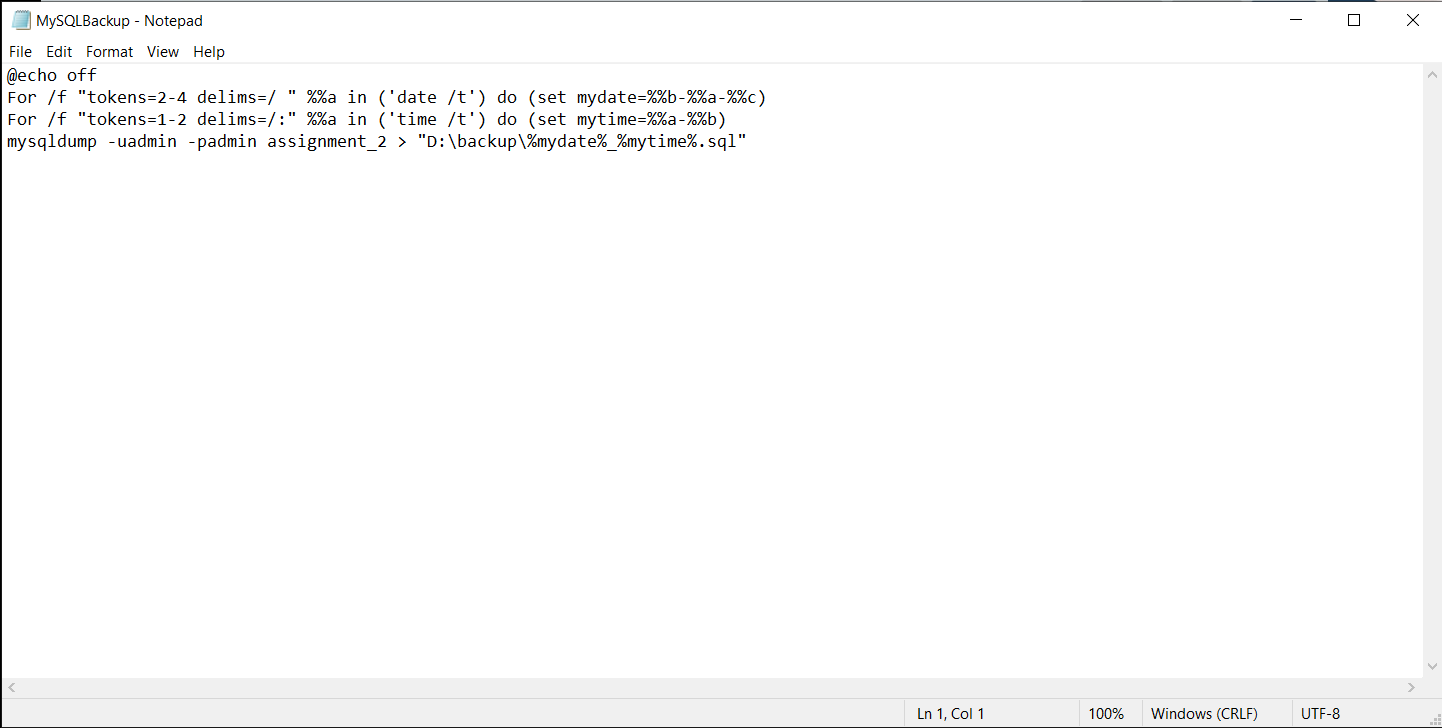
* MySQLWorkbench, to work on the table



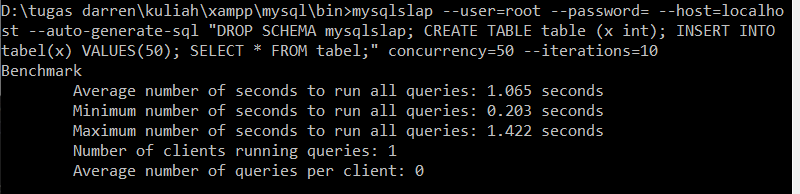
* ERD PLUS, to create logical and relational diagram



* Notepad for making batch file



* Command prompt for load testing



**Task 1**

**Task statement:**

**1. Create a database requirements specification document for the project scenario.**

**2. Include it as part of Project Report.**

1. Hardware requirement:

* Sufficient PC/Laptop

1. Software requirement:

* XAMPP (PHPMYADMIN, apache server)
* MySQLworkbench
* MySQL server
* Task scheduler
* Google chrome
* Command prompt
* Notepad

1. Database requirement

|  |  |  |
| --- | --- | --- |
| Entity | Description | Function |
| 1. Users | The user entity contains all the information about the user such as their name, email, etc. And they can update their profile | Registration page, Login page, Profile page, update profile |
| 1. Administrator | The administrator contains information for the admins such as their ID and password | Admin page |
| 1. Bulk Email | This is to send bulk emails to people | Admin page |
| 1. Jobs posted | This entity contains all the information about the jobs that are available for the users. | Job posting page |
| 1. Thread | This entity stores data about the thread that the user post | Thread posting page |
| 1. Messages | This entity stores data about messages that the users sent or received | Send message |

1. Entity relationship Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Entity one | Entity two | Relationship | Description |
| 1. | Users | Administrator | Many to many | Many users can be managed by many admin |
| 2. | Administrator | Bulk email | Many to many | Many admin can send many bulk emails |
| 3. | Users | Jobs posted | One to many | One users can apply to many jobs |
| 4. | Users | Messages | One to many | One users can send many messages |
| 5. | Users | thread | Many to many | Users can post and reply many threads |

**Task 2**

**Task statement:**

**1. Create a database design document.**

**2. Create a Conceptual Design for the proposed Database.**

**3. Create Logical Design for the proposed Database.**

**4. Create the ER Diagram for the project scenario.**

**5. Include the above as part of Project Presentation.**

**Solution:**

1. Database design document

RDBMS is a relational database management system for CRUD operations in database management. RDBMS usually consists of programs that helps the admin to create, read, update and delete in a database system.

1. Conceptual design

Conceptual design is the first step in creating a Database. The result of this phase is to create a design that is descriptive that explains the entity, its attributes, and the relationship that each entity has to each other

1. Logical design

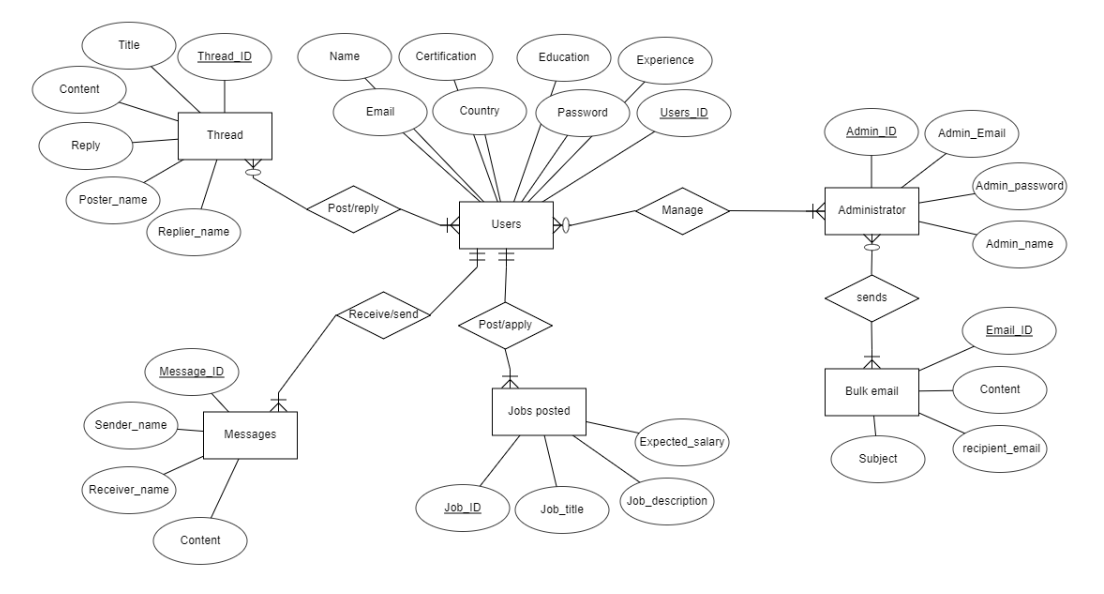
Logical design is a process of making an information model that uses the data model but does not necessarily be dependant on the database management system

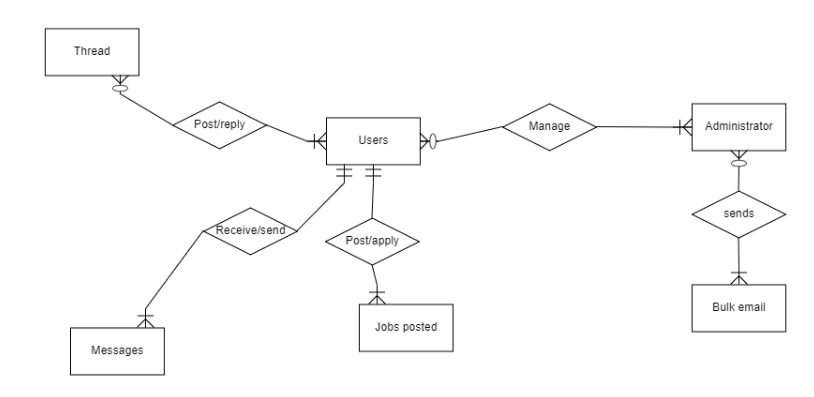
1. Physical design

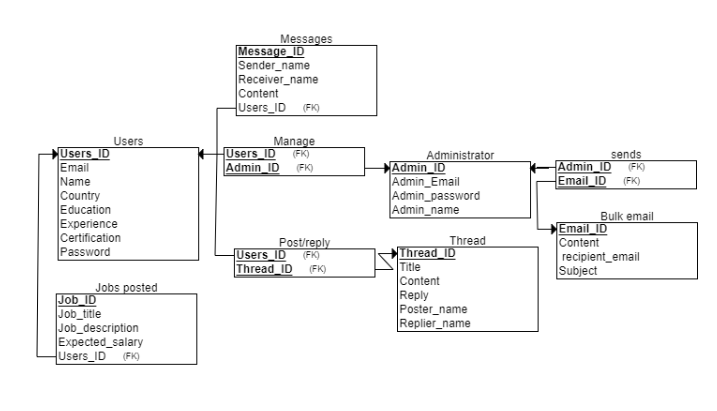
This is the last step in making a DBMS. On this phase the Database design is made. The data model is transformed into a structure of the DBMS.

Entities and attributes for ABC Portal

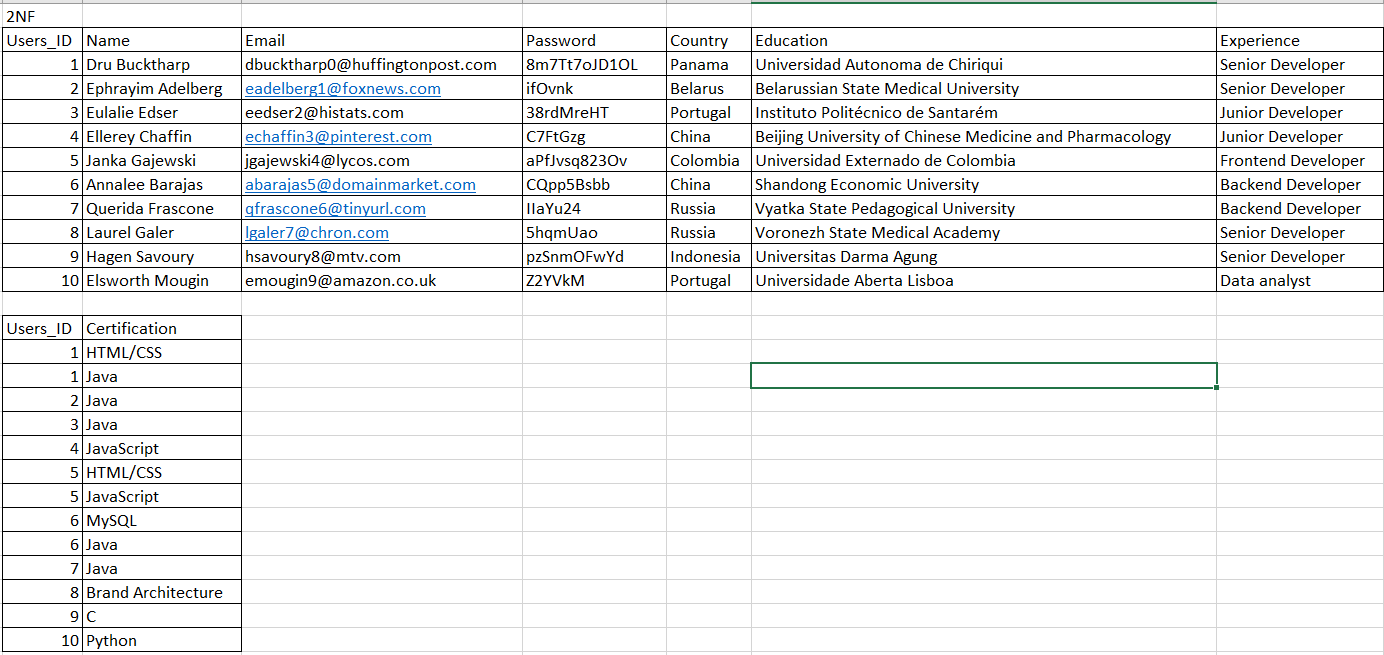
|  |  |  |
| --- | --- | --- |
| Entity | Description | Attributes |
| 1. Users | The user entity contains all the information about the user such as their name, email, etc. And they can update their profile | Users\_ID(PK)  Email  Name  Country Education Experience  Certification Password |
| 1. Administrator | The administrator contains information for the admins such as their ID and password | Admin\_ID(PK)  Admin\_Email Admin\_password Admin\_name |
| 1. Bulk Email | This is to send bulk emails to people | Email\_ID(PK)  Content  recipient\_email  Subject |
| 1. Jobs posted | This entity contains all the information about the jobs that are available for the users. | Job\_ID(PK)  Job\_title  Job\_description  Expected\_salary |
| 1. Thread | This entity stores data about the thread that the user post | Thread\_ID(PK)  Title  Content  Reply  Poster\_name  Replier\_name |
| 1. Messages | This entity stores data about messages that the users sent or received | Message\_ID(PK)  Sender\_name  Receiver\_name  Content |

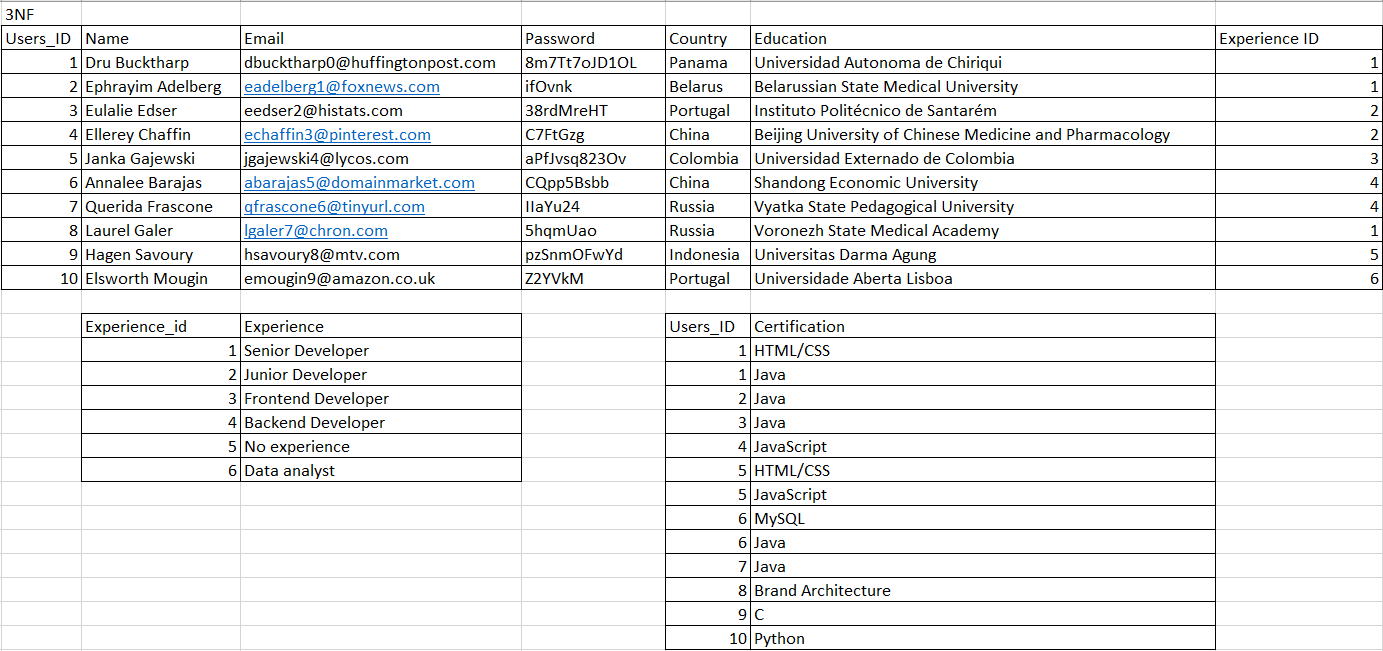
ER diagram

Conceptual design

Logical design

Include normalization as part of physical design



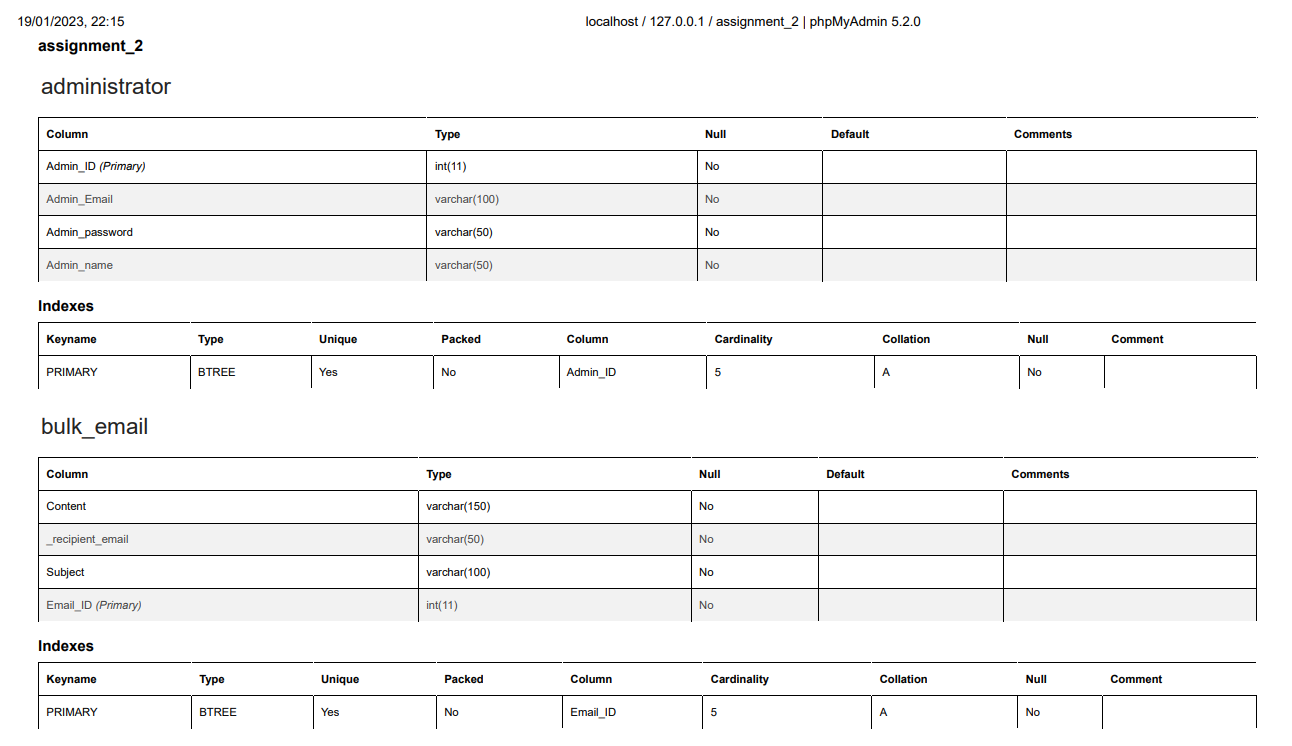
The table is in 3NF because it has reduced data redundancy for example the every table has a single value.

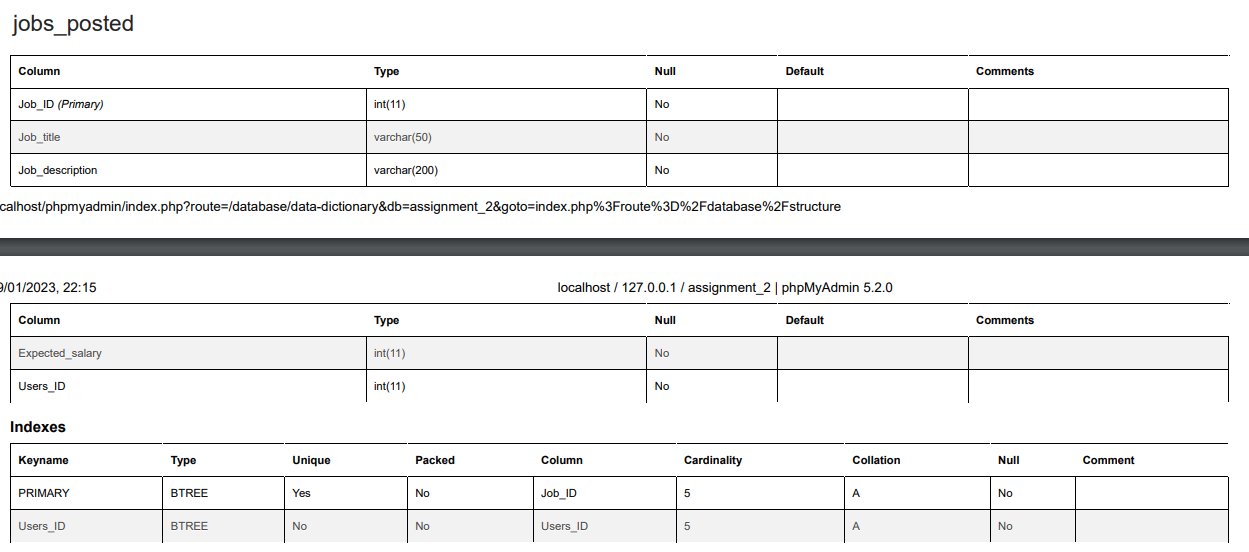
**Task 3**

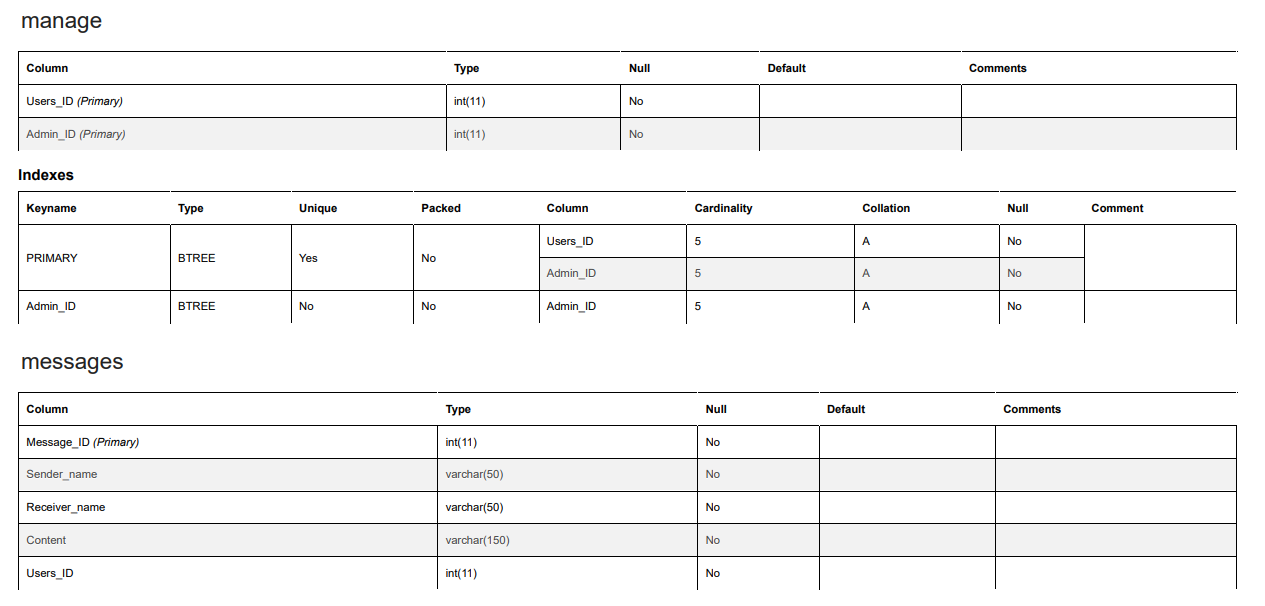
**Task statement:**

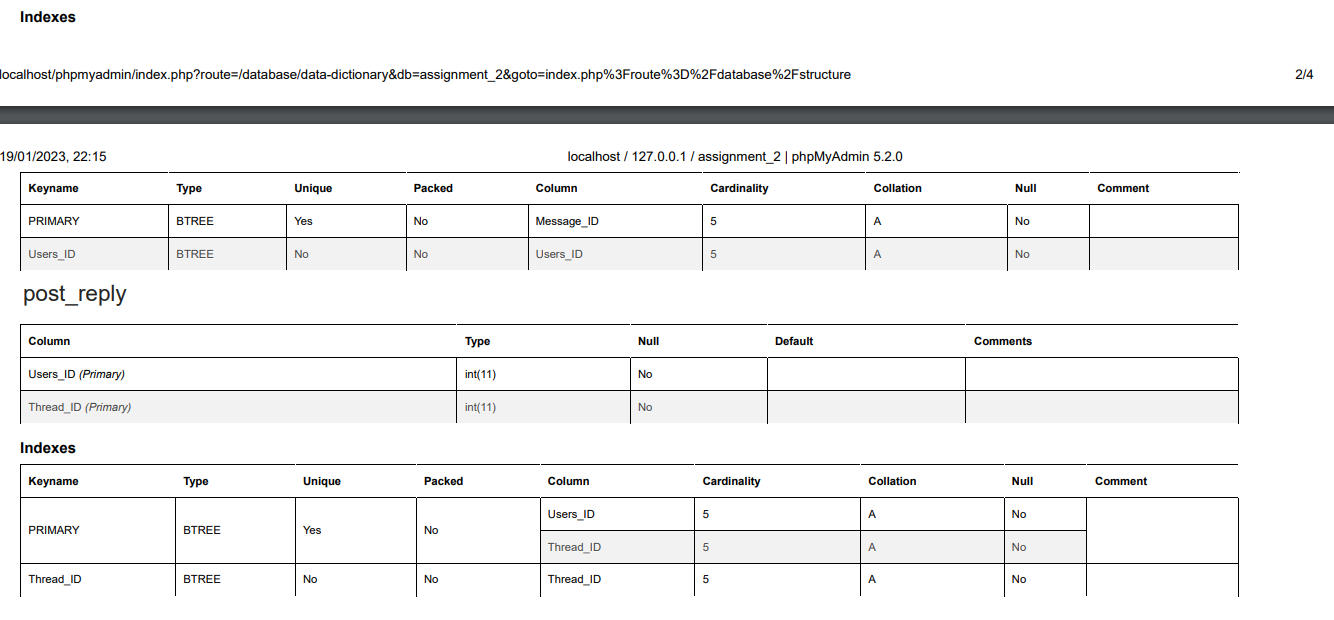
**1. Create the Database dictionary with tables, fields & datatypes**

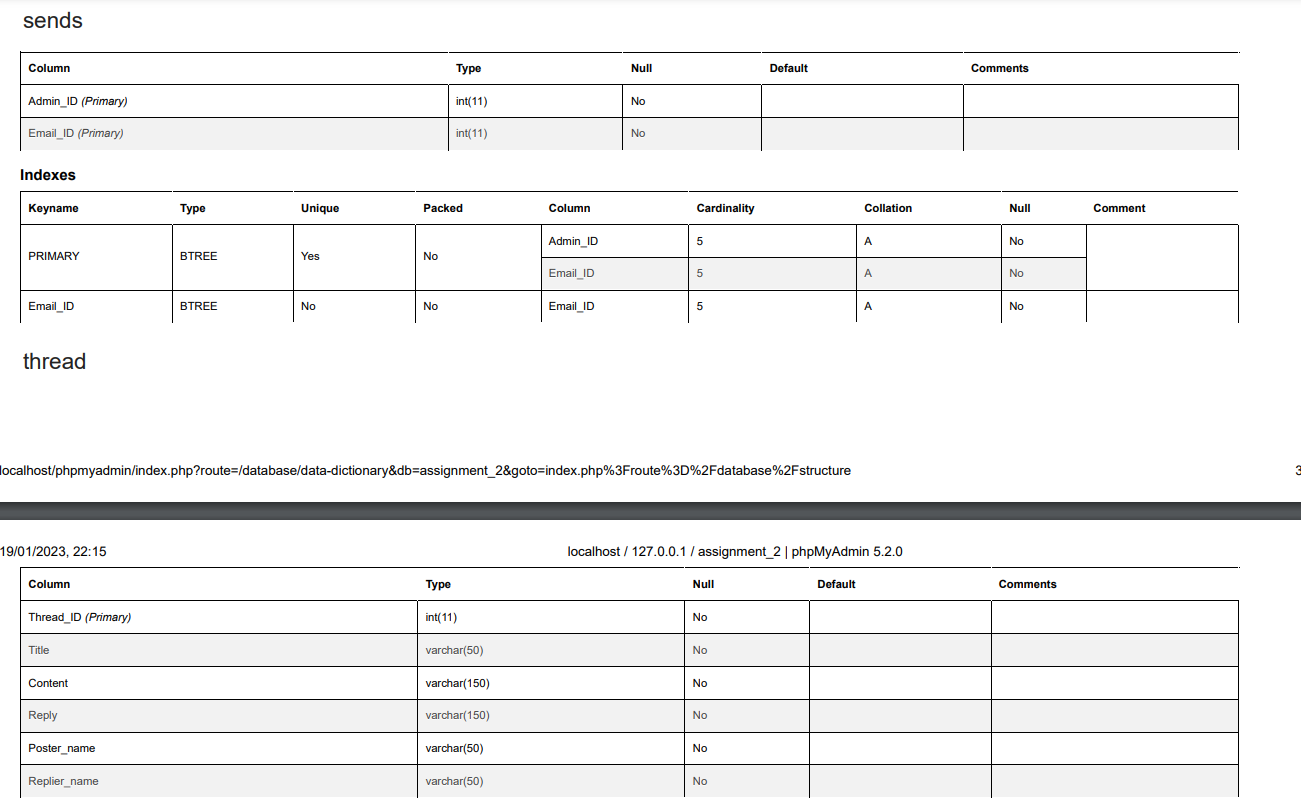
**2. Include the above as part of Project Report.**

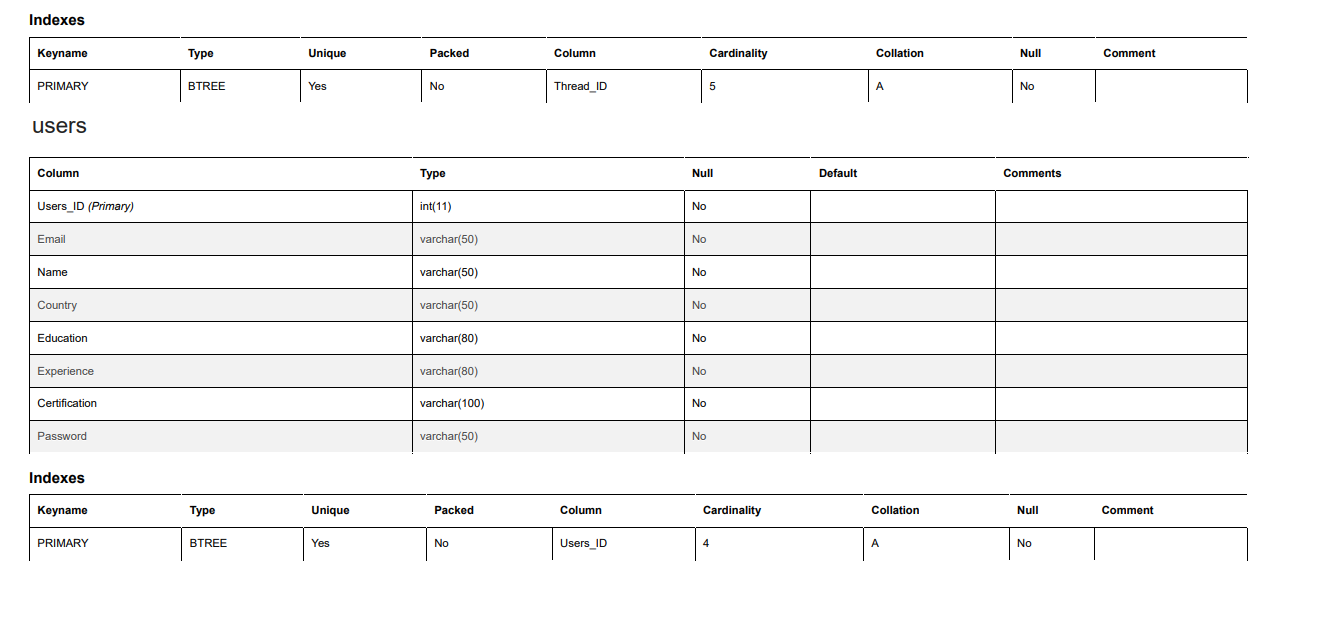












Database dictionary google drive links

<https://drive.google.com/file/d/1mitLs_koJx19AtlAOg7VJDkV8y142YZV/view?usp=sharing>

**Task 4**

**Task statement:**

**1. Create a MySQL database & Implement the database design in that database.**

**2. Implement Primary Key, Foreign Key & Constraints.**

**3. Use SQL Scripts & Mysql or phpMyAdmin to create the database.**

**4. Produce the Screen capture of created tables in phpMyAdmin or MySQL Command prompt.**

**5. Include it as part of Project Presentation.**

CREATE TABLE Users

(

Users\_ID INT NOT NULL,

Email VARCHAR(50) NOT NULL,

Name VARCHAR(50) NOT NULL,

Country VARCHAR(50) NOT NULL,

Education VARCHAR(80) NOT NULL,

Experience VARCHAR(80) NOT NULL,

Certification VARCHAR(100) NOT NULL,

Password VARCHAR(50) NOT NULL,

PRIMARY KEY (Users\_ID)

);

CREATE TABLE Administrator

(

Admin\_ID INT NOT NULL,

Admin\_Email VARCHAR(100) NOT NULL,

Admin\_password VARCHAR(50) NOT NULL,

Admin\_name VARCHAR(50) NOT NULL,

PRIMARY KEY (Admin\_ID)

);

CREATE TABLE Bulk\_email

(

Content VARCHAR(150) NOT NULL,

\_recipient\_email VARCHAR(50) NOT NULL,

Subject VARCHAR(100) NOT NULL,

Email\_ID INT NOT NULL,

PRIMARY KEY (Email\_ID)

);

CREATE TABLE Jobs\_posted

(

Job\_ID INT NOT NULL,

Job\_title VARCHAR(50) NOT NULL,

Job\_description VARCHAR(200) NOT NULL,

Expected\_salary INT NOT NULL,

Users\_ID INT NOT NULL,

PRIMARY KEY (Job\_ID),

FOREIGN KEY (Users\_ID) REFERENCES Users(Users\_ID)

);

CREATE TABLE Thread

(

Thread\_ID INT NOT NULL,

Title VARCHAR(50) NOT NULL,

Content VARCHAR(150) NOT NULL,

Reply VARCHAR(150) NOT NULL,

Poster\_name VARCHAR(50) NOT NULL,

Replier\_name VARCHAR(50) NOT NULL,

PRIMARY KEY (Thread\_ID)

);

CREATE TABLE Messages

(

Message\_ID INT NOT NULL,

Sender\_name VARCHAR(50) NOT NULL,

Receiver\_name VARCHAR(50) NOT NULL,

Content VARCHAR(150) NOT NULL,

Users\_ID INT NOT NULL,

PRIMARY KEY (Message\_ID),

FOREIGN KEY (Users\_ID) REFERENCES Users(Users\_ID)

);

CREATE TABLE Manage

(

Users\_ID INT NOT NULL,

Admin\_ID INT NOT NULL,

PRIMARY KEY (Users\_ID, Admin\_ID),

FOREIGN KEY (Users\_ID) REFERENCES Users(Users\_ID),

FOREIGN KEY (Admin\_ID) REFERENCES Administrator(Admin\_ID)

);

CREATE TABLE sends

(

Admin\_ID INT NOT NULL,

Email\_ID INT NOT NULL,

PRIMARY KEY (Admin\_ID, Email\_ID),

FOREIGN KEY (Admin\_ID) REFERENCES Administrator(Admin\_ID),

FOREIGN KEY (Email\_ID) REFERENCES Bulk\_email(Email\_ID)

);

CREATE TABLE Post\_reply

(

Users\_ID INT NOT NULL,

Thread\_ID INT NOT NULL,

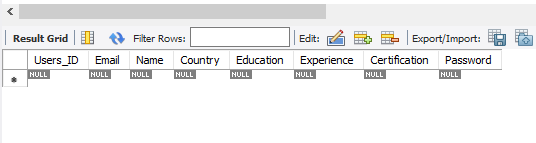
PRIMARY KEY (Users\_ID, Thread\_ID),

FOREIGN KEY (Users\_ID) REFERENCES Users(Users\_ID),

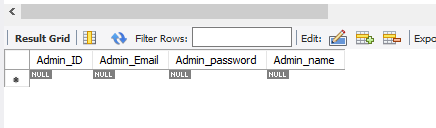
FOREIGN KEY (Thread\_ID) REFERENCES Thread(Thread\_ID)

);

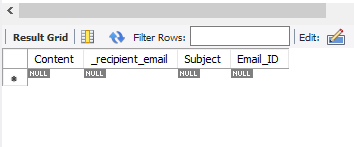
Users



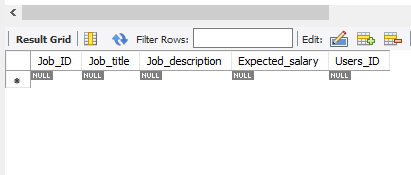
Administrator



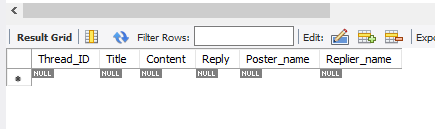
Bulk email



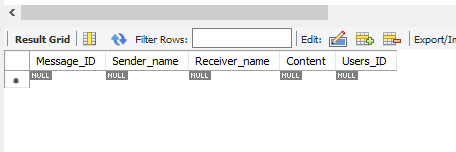
Job



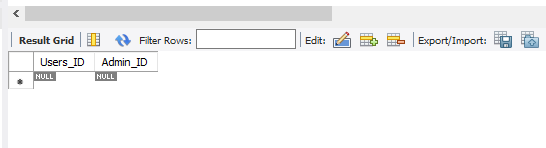
Thread



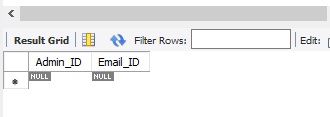
Messages



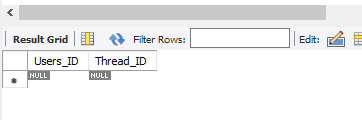
Manage



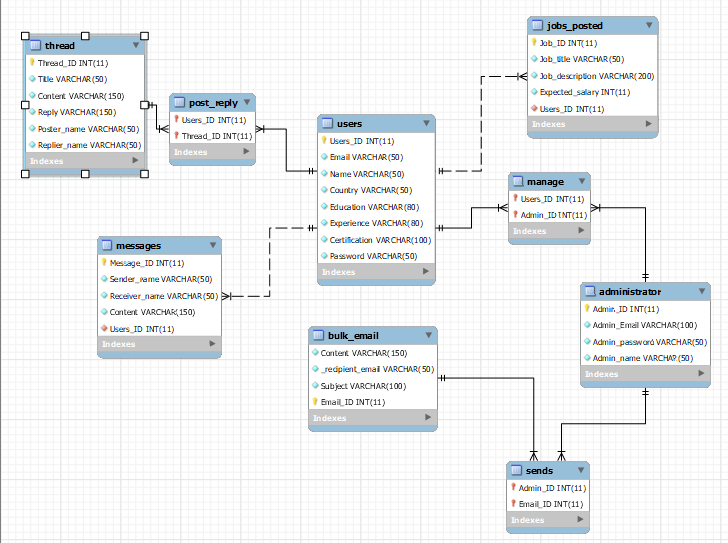
Sends



Post/reply



EER diagram



**Task 5**

**Task statement:**

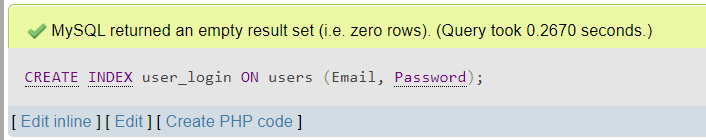
**1. Create indexes in table, provide script for creating the index.**

**2. Create a Backup script to back up the database every 6 hours.**

**3. Include the mechanism and sources as part of Project Report**

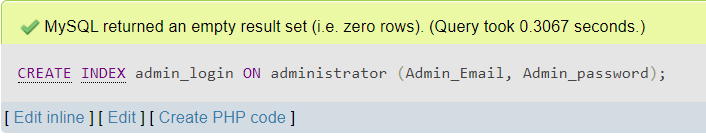
1. Index

CREATE INDEX user\_login ON users (Email, Password)



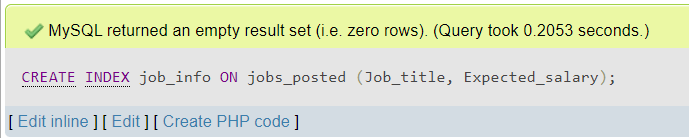


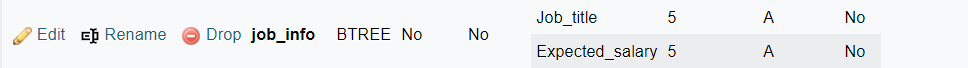
CREATE INDEX admin\_login ON administrator (Admin\_Email, Admin\_password);



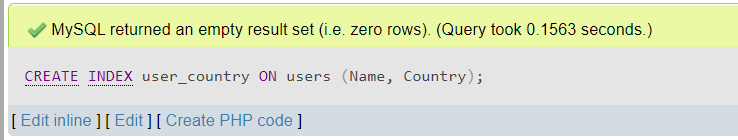


CREATE INDEX job\_info ON jobs\_posted (Job\_title, Expected\_salary)





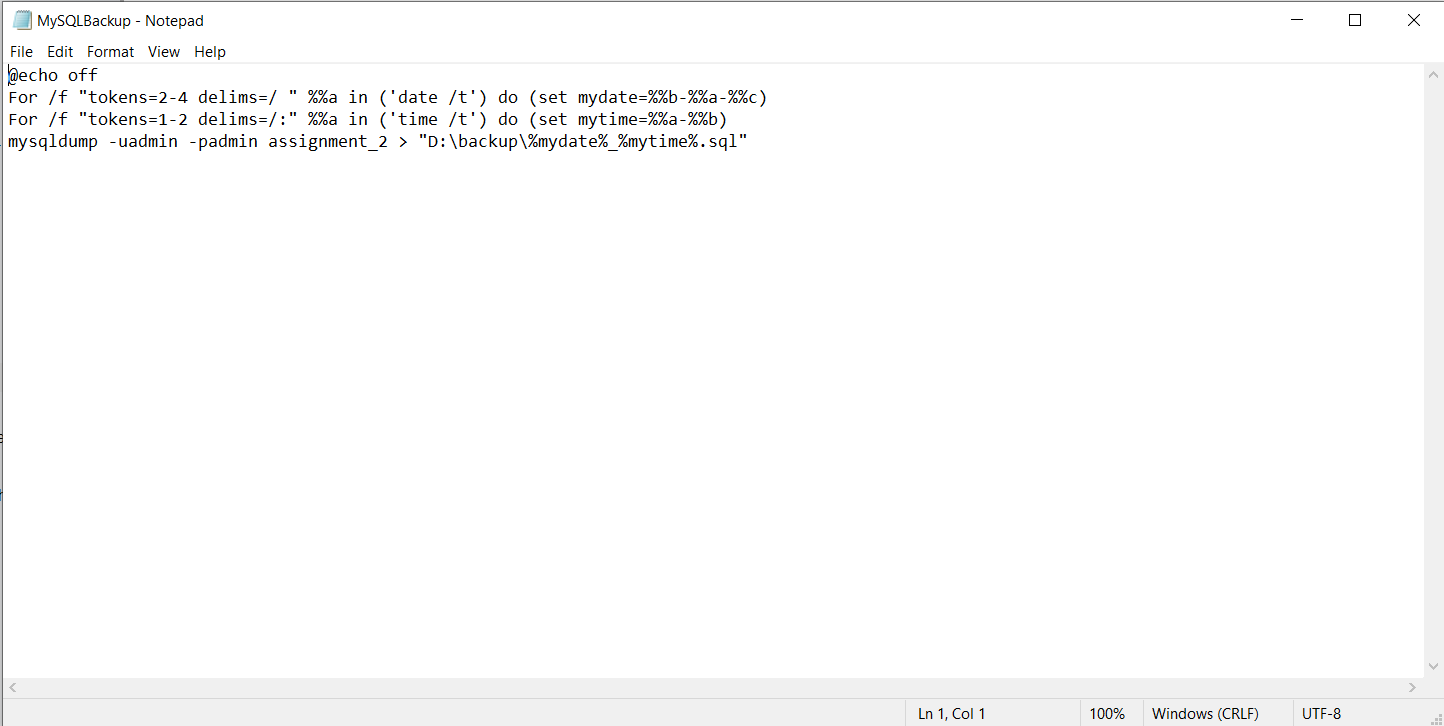
CREATE INDEX user\_country ON users (Name, Country)



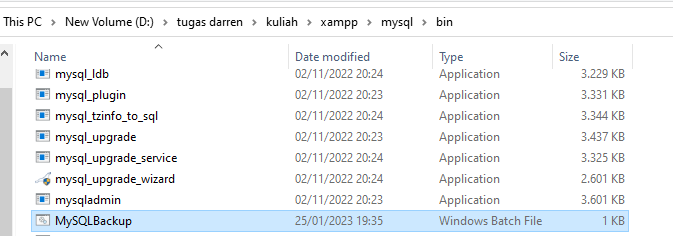


1. Backup script

* Make a script that backups the database in notepad



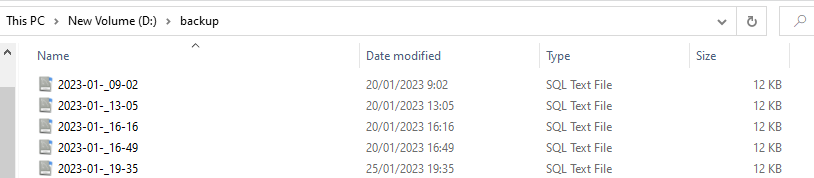
* Save it on your mysql bin path as bat file



* Open task scheduler and make a task that runs the backup file every 6 hours



* Makes sure that your database is backed up in the designated place



**Task 6**

**Task statement:**

**1. Create 8 SQL queries which will be used by the Community portal.**

**2. Import Sample data from CSV file in to the database.**

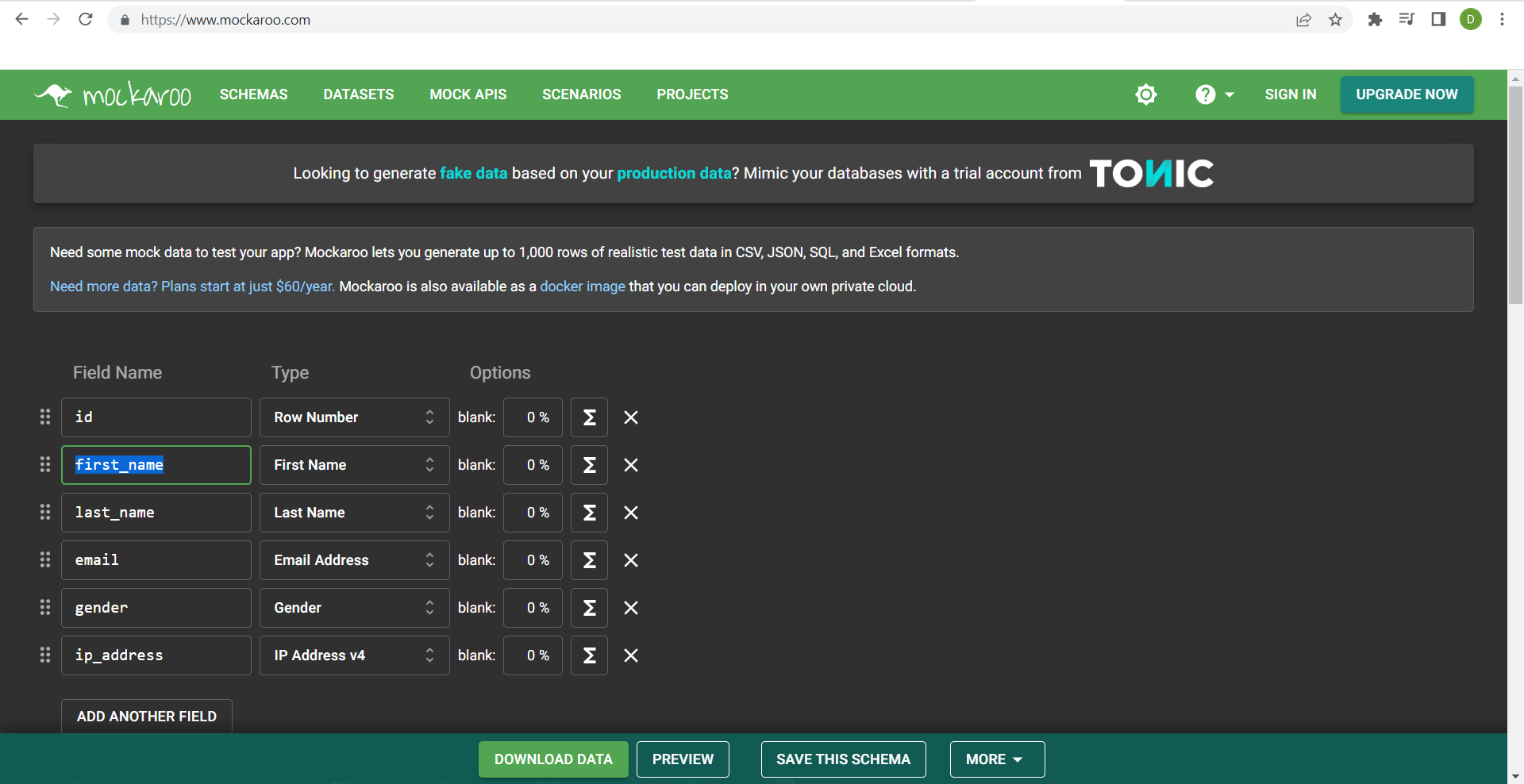
**3. Provide evidence of import as part of Project Presentation.**

|  |  |  |  |
| --- | --- | --- | --- |
| Page | Tables | Function | Query |
| Login | users | Validate Login | SELECT Email, Password FROM users WHERE Email = “something” AND Password = “something” |
| Registration | users | User Registration | INSERT INTO users (Name, Email, Password, …)VALUES (“name”,”email”,”password”,…) |
| Admin | administrator | Validate login as admin | SELECT Email, Password FROM administrator WHERE Email = “something” AND Password = “something” |
| Update profile | users | Update Profile | UPDATE users SET Name = “new name”, Experience = “new experience”, Certification = “new certification”, Education = “new education” WHERE (User\_id = “example: 1”); |
| Forget Password | users | Forget Password | UPDATE users SET Password = “new password” WHERE (Users\_id = “example: 1”); |
| Jobs | jobs\_posted | Post Job | INSERT INTO jobs\_posted (Job\_title, Job\_description, Expected\_Salary)VALUES(“The job title”, “the job description”, “the jobs expected salary”); |
| Profile | users | View Profile | SELECT Name, Country, Education, Experience, certification FROM users |
| Message | messages | Send Message | INSERT INTO messages (Content, Sender\_Name, Receiver\_Name)VALUES(“any message”, “Sender Name from users”, “Receiver name from users”); |
| Home page | thread | Read a Thread | SELECT \* FROM thread |
| Home page | users | Search User | SELECT \* FROM users WHERE Name Like ‘search name’; |
| Admin | bulk\_email | Send bulk email | INSERT INTO bulk\_email (\_recipient\_email, Content, Subject) VALUES(“The recipient email”, “the email content”, “the email subject”); |

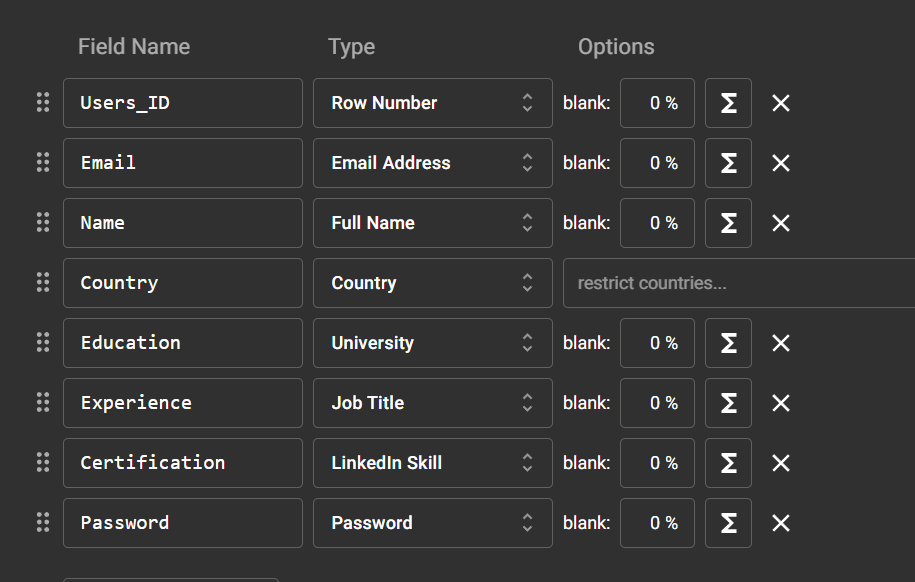
|  |  |  |  |
| --- | --- | --- | --- |
| No. | Note | Query | Evidence |
| 1 | View users login information | SELECT CONCAT(Users\_ID, " " , Email, " ", Password)AS "Login Info" FROM users ORDER BY Users\_ID ASC; |  |
| 2 | View Job and its expected salary | SELECT CONCAT(Job\_title, " ", Expected\_salary)AS "Job info" FROM jobs\_posted ORDER BY Job\_ID ASC; |  |
| 3 | User qualification | SELECT CONCAT(Name, " ", Education, " ", Certification, " ", Experience)AS "Qualification" FROM users ORDER BY Users\_ID ASC; |  |
| 4 | View admins login information | SELECT CONCAT(Admin\_ID, " " , Admin\_Email, " ", Admin\_password)AS "Login Info" FROM administrator ORDER BY Admin\_ID ASC; |  |

Import sample data from mockaroo:

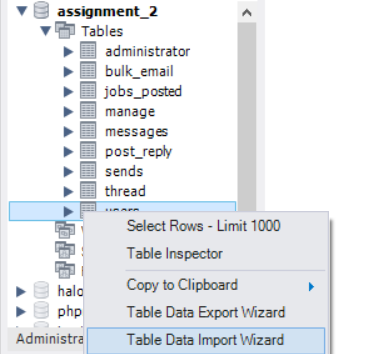
* Open mockaroo



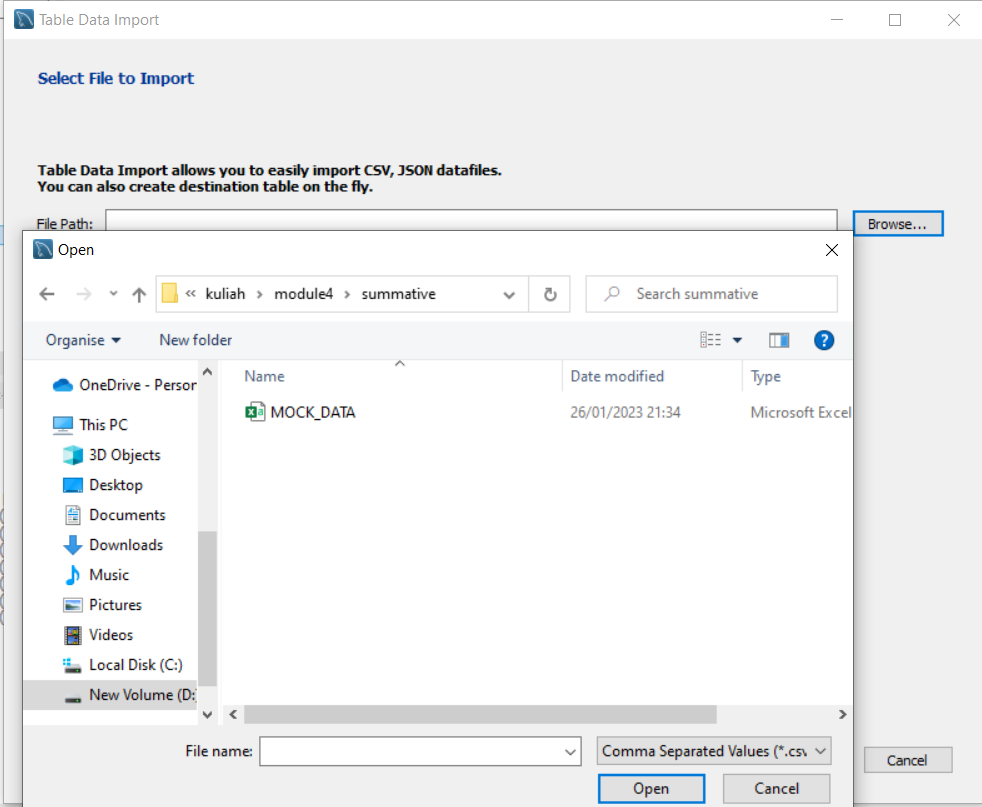
* Put the data the same as your table and download how many data you want (up to 1000)



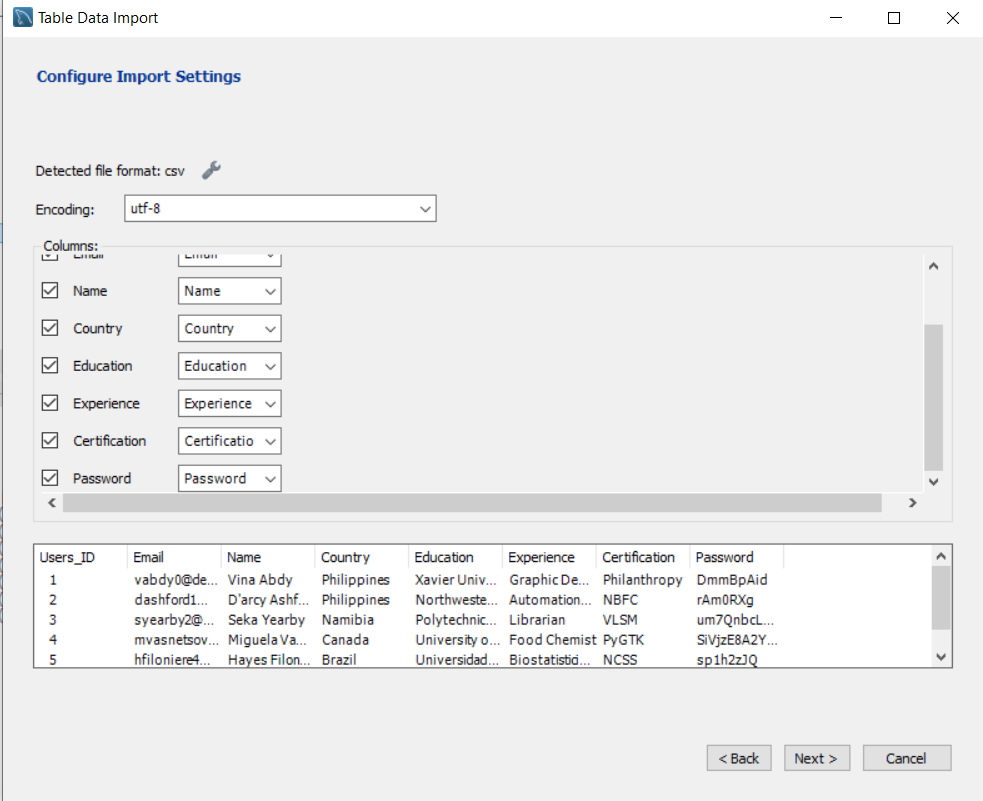
* Open mysql workbench click right on the table and click on import wizard



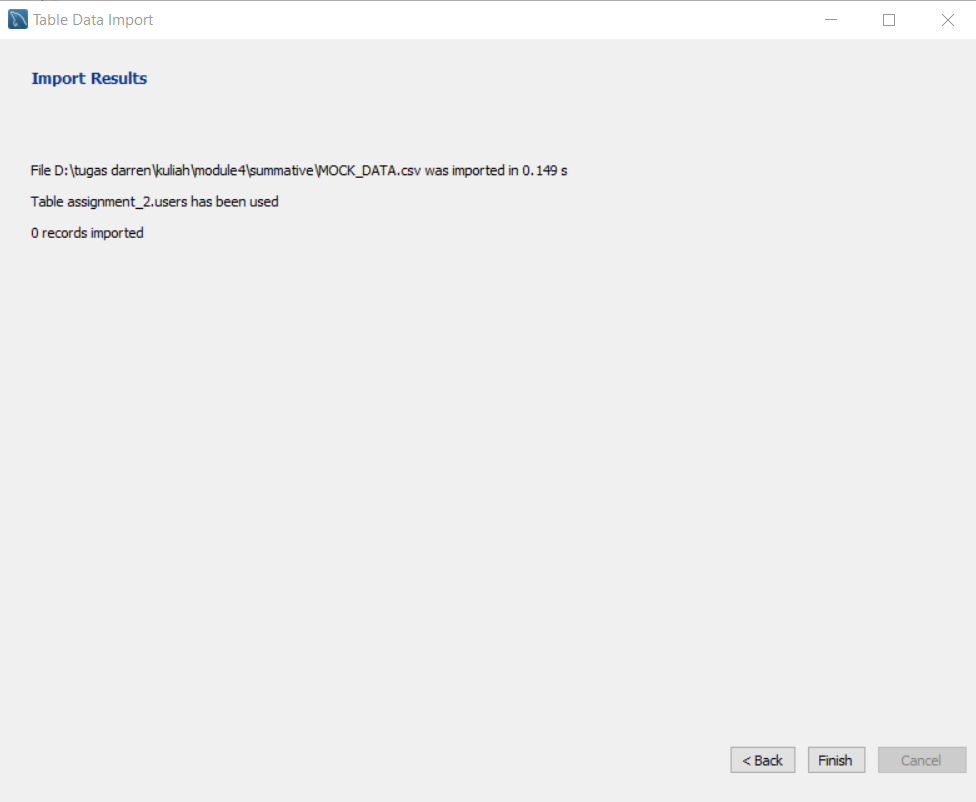
* Select the data you have downloaded



* Click on next



* The data will be imported if there isn’t any data yet in your table



* Since I have already have a mock data I put it in a different table

