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 | |
| Wildan Luqmanul Hakim | |  | |
| Date issued | Completion date | | Submitted on |
| 2 August 2022 | 8 August 2022 | | 8 August 2022 |
|  | |  | |
| 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: 8 August 2022 |

Contents

[1. Project Background 2](#_Toc110699278)

[2. Project Objective 2](#_Toc110699279)

[2.1 Project Goal 2](#_Toc110699280)

[2.2 Scope of the Project 2](#_Toc110699281)

[2.3 Tools & Platform 2](#_Toc110699282)

[3. Database Requirements Specification 6](#_Toc110699283)

[3.1 System Requirements 6](#_Toc110699284)

[3.2 Hardware Requirements 6](#_Toc110699285)

[3.3 Software Requirements 6](#_Toc110699286)

[3.4 Database Requirements 7](#_Toc110699287)

[4. Database Design Document 8](#_Toc110699288)

[4.1 What is RDBMS? 8](#_Toc110699289)

[4.1.1 What is Conceptual Design? 8](#_Toc110699290)

[4.1.2 What is Logical Design? 8](#_Toc110699291)

[4.1.3 What is Physical design? 8](#_Toc110699292)

[4.2 Entity and attributes for ABC Job Portal 9](#_Toc110699293)

[4.3 ERD Diagram for ABC Job Portal 10](#_Toc110699294)

[4.4 Conceptual Design for ABC Job Portal 11](#_Toc110699295)

[4.5 Logical Design for ABC Job Portal 11](#_Toc110699296)

[4.6 Normalization 12](#_Toc110699297)

[4.6.1 What is Normalization? 12](#_Toc110699298)

[4.6.2 What is 1NF? 12](#_Toc110699299)

[4.6.3 What is 2NF? 12](#_Toc110699300)

[4.6.4 What is 3NF? 12](#_Toc110699301)

[4.7 Normalization of user table 12](#_Toc110699302)

[5. Database Dictionary 13](#_Toc110699303)

[6. Database Design Implementation 17](#_Toc110699304)

[6.1 MySQL Database Scripts & Screen Capture 17](#_Toc110699305)

[6Logical Design 23](#_Toc110699306)

[7. Database Indexes & Backup 23](#_Toc110699307)

[7.1 Why create indexes? 23](#_Toc110699308)

[7.1.1 Indexes for username 24](#_Toc110699309)

[7.1.2 Indexes for Name 24](#_Toc110699311)

[7.1.3 Indexes for Company 24](#_Toc110699312)

[7.2 Database Backup 24](#_Toc110699313)

[8. Community Portal Query & CSV Sample Data Import 27](#_Toc110699314)

[8.1 Steps to import CSV files 27](#_Toc110699315)

[8.2 5 useful queries to develop the application 30](#_Toc110699316)

[8.3 5 useful queries to meet the management requirements using joins 31](#_Toc110699317)

# Project Background

ABC Jobs is a website similar to LinkedIn.com, a Community Portal website where the website requires a database to store data from users, admins, companies and others.

Therefore, a database with the name abc\_jobs was created to store a lot of data from the ABC Jobs website, this database is stored and managed using MySQL. MySQL is a DBMS (Database Management System) using SQL (Structured Query Language) commands that are widely used today in making web-based applications. With MySQL we can manage databases using CLI or GUI, for CLI we use CMD and GUI using phpmyadmin

# Project Objective

## 2.1 Project Goal

* To design the conceptual and entity relationship diagram (ERD) using ERDPlus
* To implement and develop the database using MySQL Workbench

## 2.2 Scope of the Project

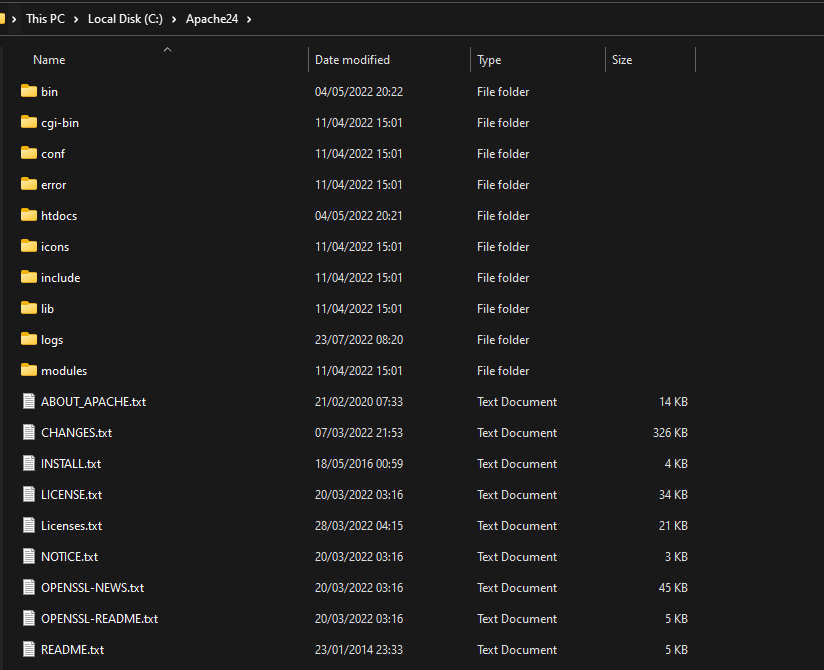
The scope of this project is to design a community portal similar to Linkedin.com. The website should allow users to register, login, change password, update profile, search and view other users’

## 2.3 Tools & Platform

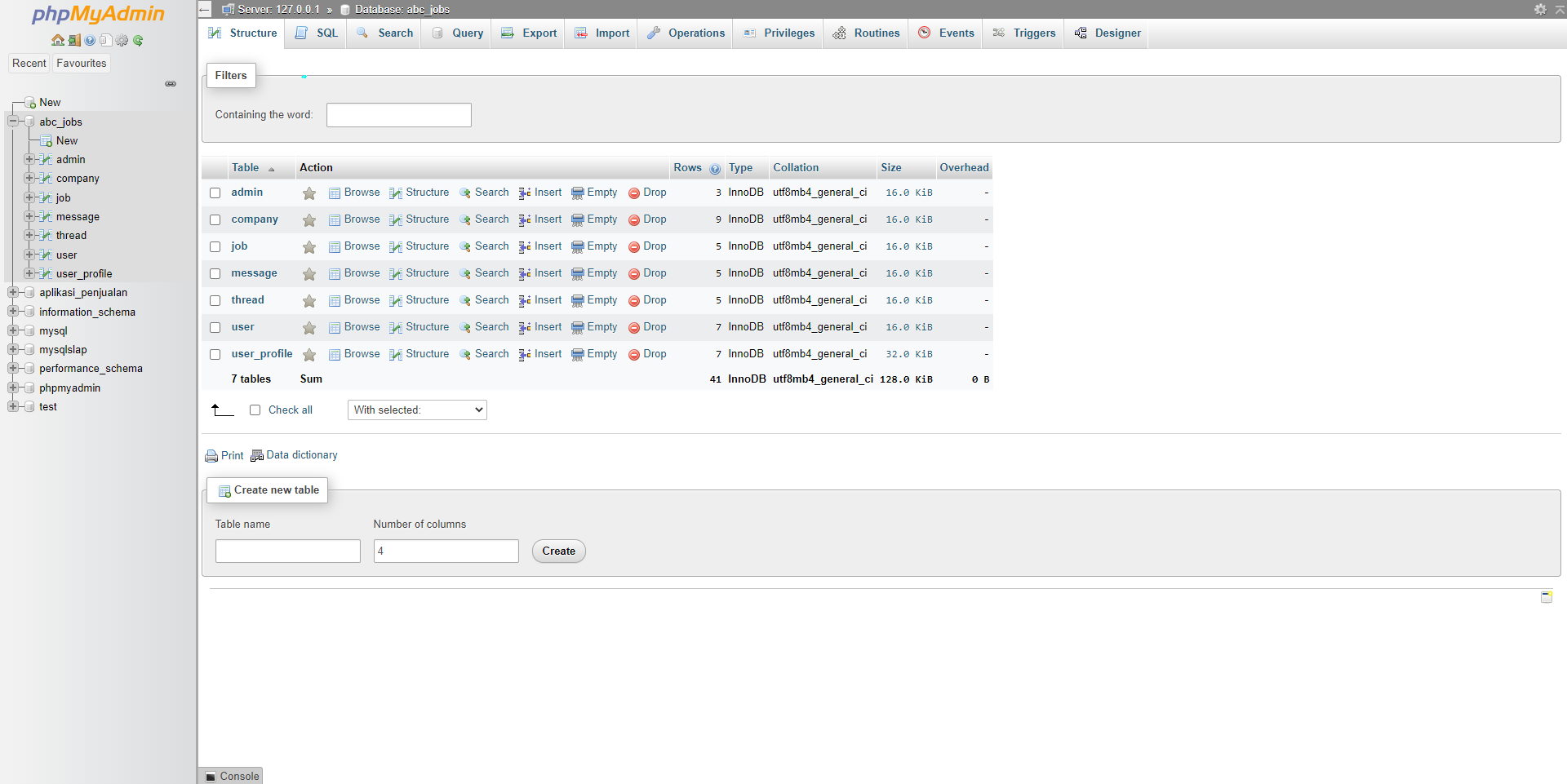
* MySQL Workbench 8.0



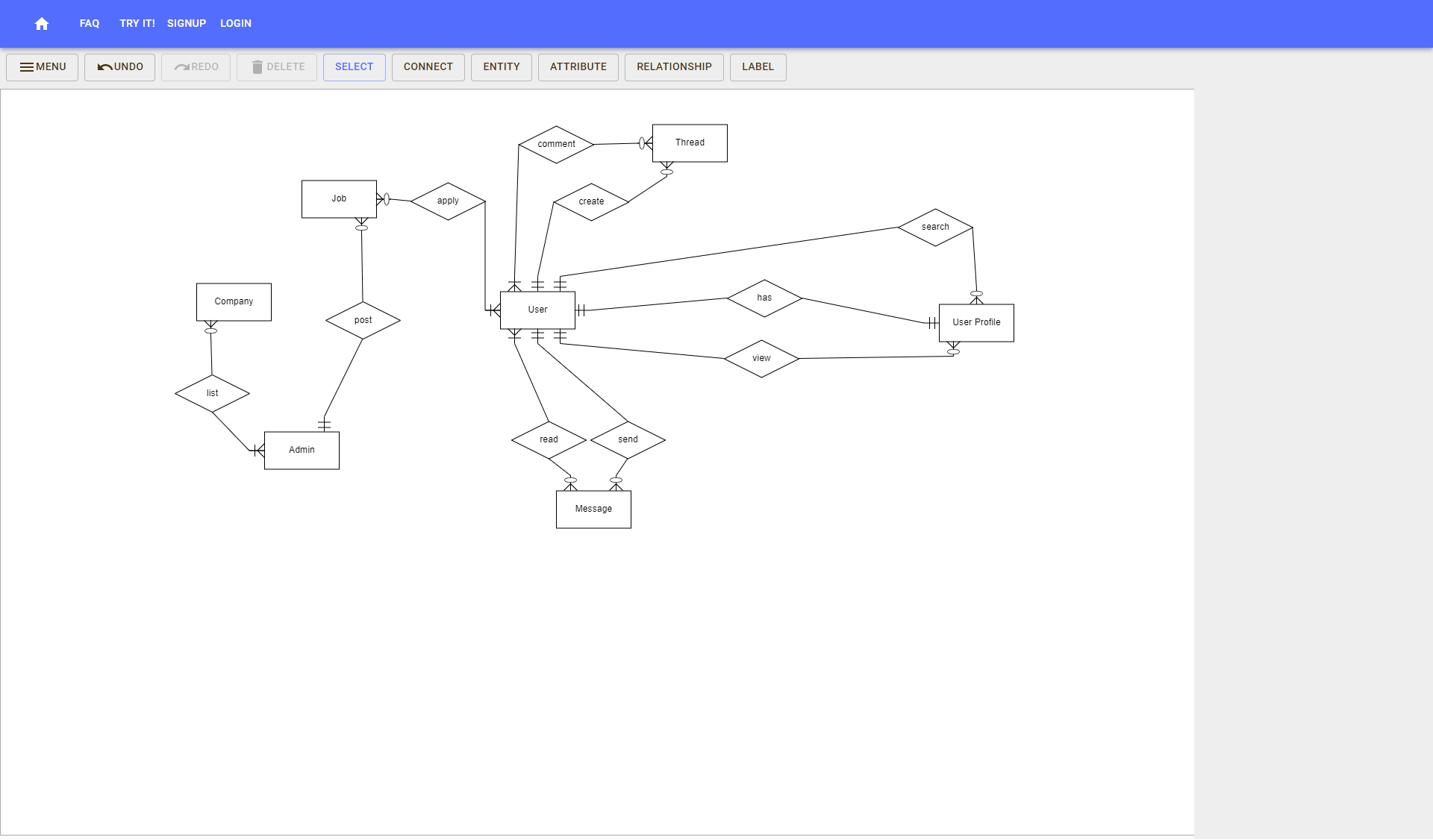
* Apache 2.4



* phpMyAdmin



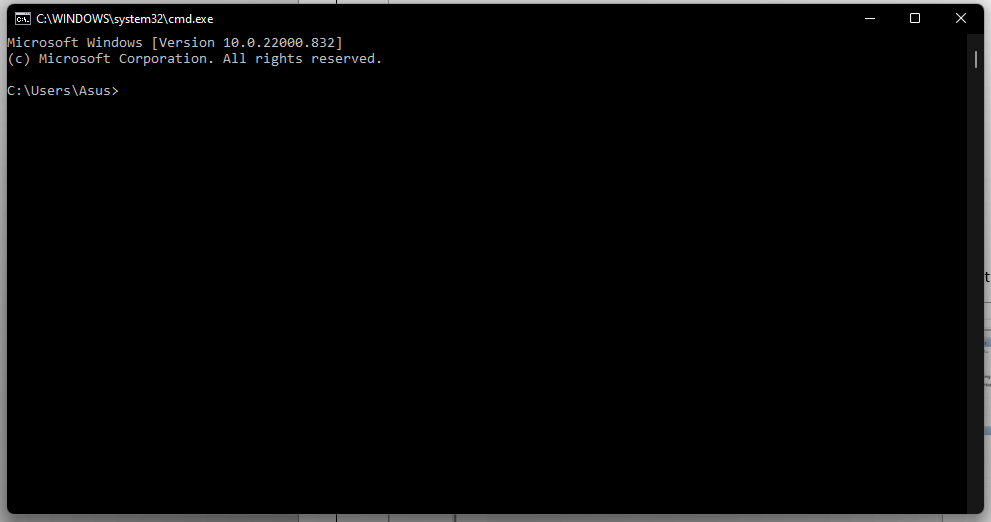
* ERDPlus.com



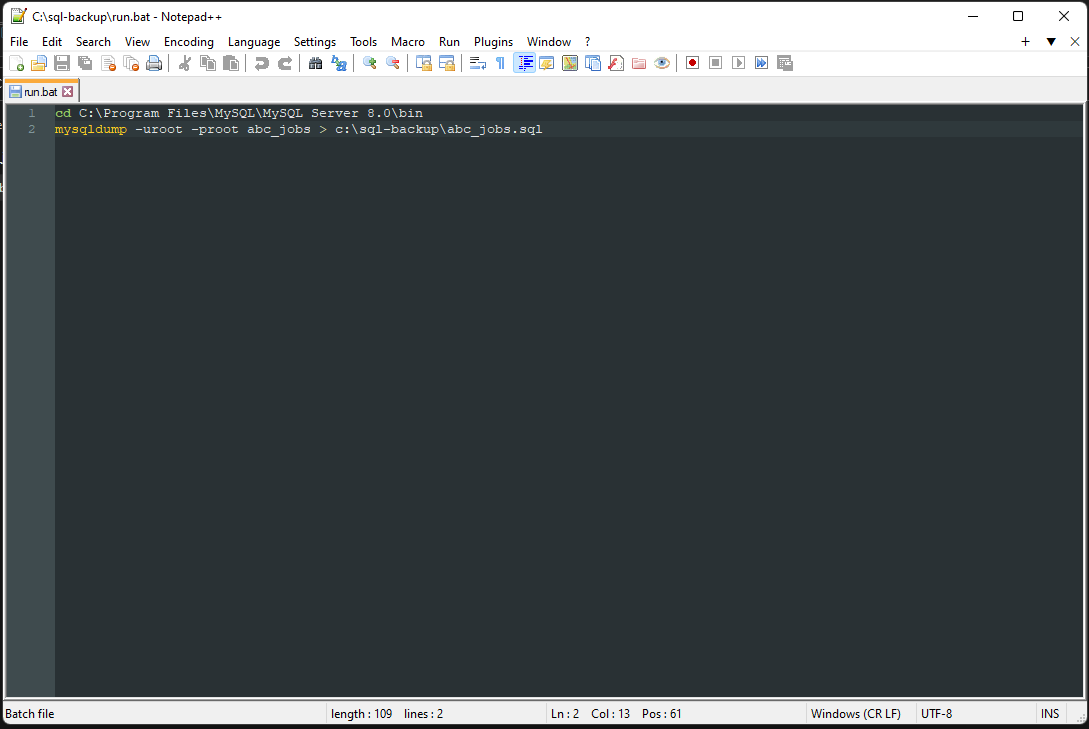
* Windows Task Scheduler



* Command Prompt



* Notepad++



# **Database Requirements Specification**

## 3.1 System Requirements

* Windows
* Linux
* Mac OS

## 3.2 Hardware Requirements

Processor : Intel Core or Xeon 3GHz (or Dual Core 2GHz) or equal AMD CPU.

Memory : 4 GB (6 GB recommended).

Graphic Accelerators : nVidia or ATI with support of OpenGL 1.5 or higher.

## 3.3 Software Requirements

* MySQL WorkBench
* XAMPP

## 3.4 Database Requirements

The following shows the database table required for each process in the community portal.

|  |  |
| --- | --- |
| **Process** | **Database Table** |
| Registration | user, user\_profile |
| Login | user |
| Forgot Password | user |
| Update Profile | user, user\_profile |
| Search & find | user, user\_profile |
| Send Messages | user, user\_profile, message |
| Read Messages | user, user\_profile, message |
| Post in Message Board | user, profile, message |
| List Message Board | user, profile, message |
| Create Threads | user, user\_profile, thread |
| Reply Threads | user, user\_profile, thread |
| Post Job Opportunities | user, user\_profile, job |
| List Job Opportunities & Responses | user, user\_profile, job |
| Administer user data | user, user\_profile, admin |

Based on the requirement, the relationship information is:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Entity One** | **Entity Two** | **Relationship** | **Note** |
| 1. | User | User\_Profile | One to one | One user has one profile page in the portal |
| 2. | User | Thread | One to zero or more (optional) | One user can create many or optional thread |
| 3. | User | Thread | One or more (optional) to zero or more (optional) | One user can comment to many or optional thread |
| 4. | Admin | Job | One to zero or more (optional) | One user can post many or optional job opportunities |
| 5. | User | Job | One or more (optional) to zero or more (optional) | One user or many can apply to many or optional job opportunities |
| 6. | User | Message | One to zero or more (optional) | One user can send many or optional message |
| 7. | User | Message | One or more (optional) to zero or more (optional) | One user or many can read many to optional messages |

Task1

# Database Design Document

## 4.1 What is RDBMS?

The software used to store, manage, query, and retrieve data stored in a relational database is called a relational database management system (RDBMS). The RDBMS provides an interface between users and applications and the database, as well as administrative functions for managing data storage, access, and performance.

### 4.1.1 What is Conceptual Design?

Conceptual design of the database includes entities and relationships without the attributes. Entities can be things, people or places and conceptually this design used the entity-relationship model.

### 4.1.2 What is Logical Design?

Logical design includes entities with attributes and relationships among them. At this stage, foreign key and primary key is identified and normalization is done.

### 4.1.3 What is Physical design?

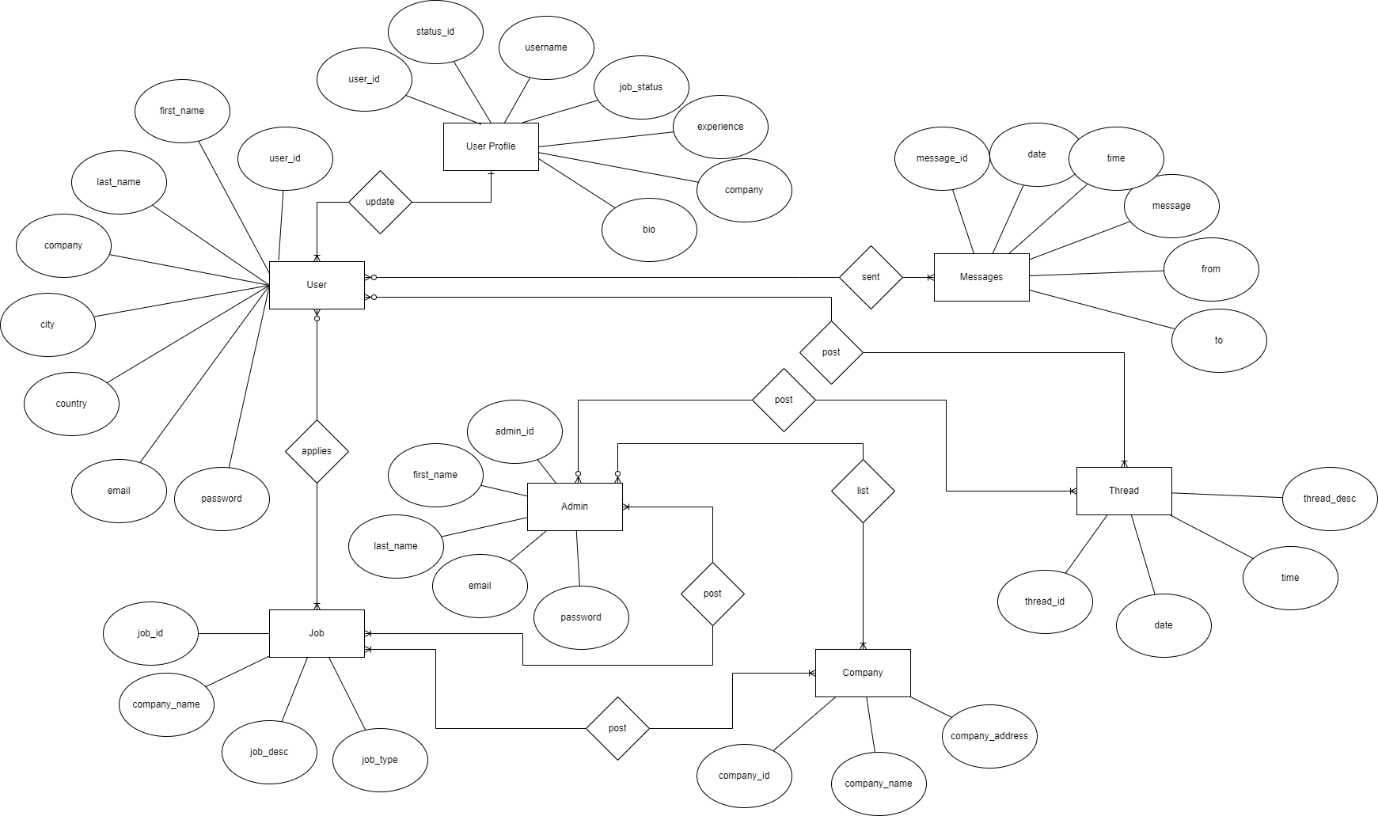
Physical design represents the actual database with table structures where the table name are the entities while the table column fields are the attributes. Each attributes have data types and constraints based on the business requirements.

## 4.2 Entity and attributes for ABC Job Portal

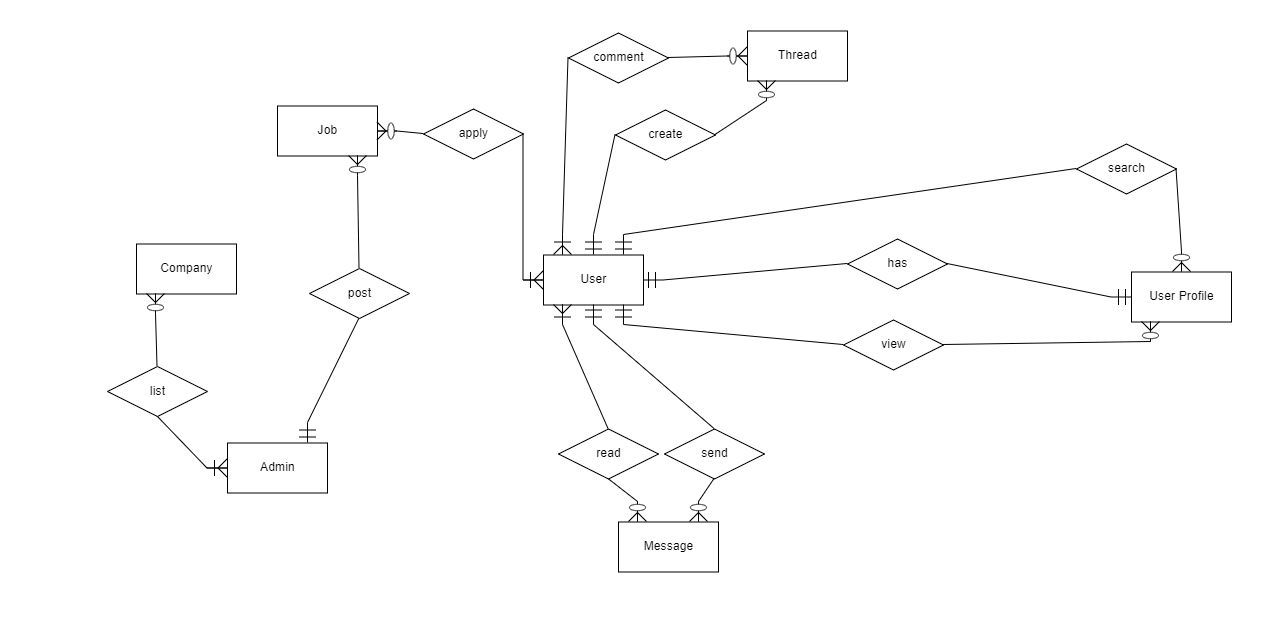
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Entities** | **Description** | **Attributes** | **Page/Function** |
| 1 | User | Registered user is uniquely identified in the portal with a User ID | * User\_id * First\_name * Last\_name * Company * City * Country * Email * Password | * Register Page * Login Page * Profile Page * Forget Password Page |
| 2 | User Profile | 1 User ID can have 1 profile with general information, contact information and uniquely identified by a Profile ID | * User\_id * Status\_id * Username * Experience * Company * bio | * Profile Page * Update Profile Page * Search User * View Users Public Profile * List Search Results |
| 3 | Admin | Admin can posts job and can list companies | * Admin\_id * First\_name * Last\_name * Email * Password | * Administater User Data |
| 4 | Company | Admin listed company in ABC Jobs | * Company\_id * Company\_name * Company\_address |  |
| 5 | Job | Registered user can post job opportunities, apply or list job opportunities and responses | * Job\_id * Company\_name * Job\_desc * Job\_type | * Post Job Opportunities * List Job Opportunities * List Job Responses |
| 6 | Thread | Registered user can create threads and comment to a thread | * Thread\_id * Date * Time * Thread\_desc * Comment | * Create Threads * Reply Threads * Read Threads |
| 7 | Message | Registered user can send and read messages to and from other users | * Message\_id * Date * Time * Message * To * From | * Send Messages * Read Messages * Post in Message Board * List Message Board |

Task2

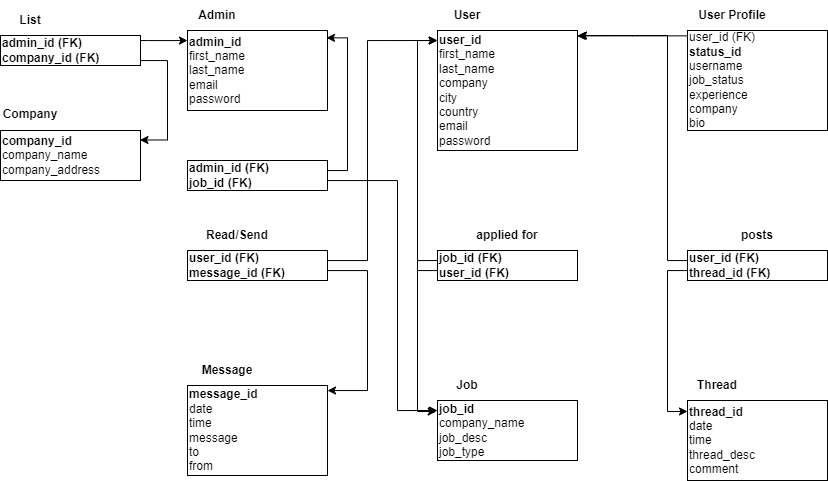
## 4.3 ERD Diagram for ABC Job Portal

****

## 4.4 Conceptual Design for ABC Job Portal



## 4.5 Logical Design for ABC Job Portal



## 4.6 Normalization

### 4.6.1 What is Normalization?

Normalization is a technique use to organize data by reducing repetitive data and eliminating irregularity in CRUD operation (Peterson, 2022). Data can be stored logically by conforming to a series of normal forms below which breaks up larger tables into smaller tables with a relationship-based links.

### 4.6.2 What is 1NF?

1NF or the first normal form does not have repeated or similar data with a primary key unique to the entity. Each table cell should be unique and only have a single value.

### 4.6.3 What is 2NF?

Table needs to be in 1NF and must have a relation of every non-primary key attribute to be fully dependent on the primary key. A table is in 2NF if it has no partial dependency and involve the removal of partial dependencies and placing them in a new relation (Upadhyay, 2019).

### 4.6.4 What is 3NF?

Table must be in 2NF and has no transitive dependencies. Transitive dependency is when non-primary key attribute is dependent on another non-primary key attribute. 3NF involve the removal of transitive dependencies and placing then in a new relation.

## 4.7 Normalization of user table

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table 1NF   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **user\_id** | **first\_name** | **last\_name** | **job\_id** | **company\_name** | **thread\_id** | **thread\_desc** | | user1 | Wildan | Luqmanul Hakim | job1 | Google | thread3 | I'm working on Google | | user1 | Wildan | Luqmanul Hakim | job2 | Amazon | thread1 | Very Happy | | user2 | Peter | Parker | job3 | Nvidia | thread6 | I like this Nvidia Company | | user2 | Peter | Parker | job3 | Nvidia | thread7 | The best experience | | user3 | Harry | Potter | job2 | Amazon | thread2 | It's time to coffe | | user3 | Harry | Potter | job3 | Nvidia | thread5 | I'm working on Amazon | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table 2NF  user job   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **user\_id** | **first\_name** | **last\_name** | **job\_id** | **company\_name** | | user1 | Wildan | Luqmanul Hakim | job1 | Google | | user1 | Wildan | Luqmanul Hakim | job2 | Amazon | | user2 | Peter | Parker | job3 | Nvidia | | user2 | Peter | Parker | job3 | Nvidia | | user3 | Harry | Potter | job2 | Amazon | | user3 | Harry | Potter | job3 | Nvidia |   user thread   |  |  | | --- | --- | | **thread\_id** | **thread\_desc** | | thread3 | I'm working on Google | | thread1 | Very Happy | | thread6 | I like this Nvidia Company | | thread7 | The best experience | | thread2 | It's time to coffe | | thread5 | I'm working on Amazon | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table 3NF  user job   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **user\_id** | **first\_name** | **last\_name** | **job\_id** | **company\_name** | | user1 | Wildan | Luqmanul Hakim | job1 | Google | | user1 | Wildan | Luqmanul Hakim | job2 | Amazon | | user2 | Peter | Parker | job3 | Nvidia | | user2 | Peter | Parker | job3 | Nvidia | | user3 | Harry | Potter | job2 | Amazon | | user3 | Harry | Potter | job3 | Nvidia |   user thread   |  |  | | --- | --- | | **thread\_id** | **thread\_desc** | | thread3 | I'm working on Google | | thread1 | Very Happy | | thread6 | I like this Nvidia Company | | thread7 | The best experience | | thread2 | It's time to coffe | | thread5 | I'm working on Amazon |   job   |  |  | | --- | --- | | **job\_id** | **company\_name** | | job1 | Google | | job2 | Amazon | | job3 | Nvidia | | job3 | Nvidia | | job2 | Amazon | | job3 | Nvidia | |

Task 3

# Database Dictionary

Database dictionary is a collection of table names, attributes, data types and information that describes each attribute as representation of the table names or entities.

**user**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Field Name (Column)** | **Data Type** | **Field Size** | **Key (Constraint)** | **References** | **Nullable** | **Description** | **Default Value** |
| user\_id | INT | 11 | Primary Key |  | No | Primary key for user table | Auto Increment |
| first\_name | VARCHAR | 48 | Not Null |  | No | User first name |  |
| last\_name | VARCHAR | 48 | Not Null |  | No | User last name |  |
| company | VARCHAR | 48 | Not Null |  | No | User company |  |
| city | VARCHAR | 48 | Not Null |  | No | User city |  |
| country | VARCHAR | 48 | Not Null |  | No | User country |  |
| email | VARCHAR | 64 | Not Null |  | No | User email for login |  |
| password | VARCHAR | 32 | Not Null |  | No | User password for login |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
| PRIMARY | BTREE | Yes | No | user\_id | 6 | A | No |  |

**user indexes**

**user\_profile**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Field Name (Column)** | **Data Type** | **Field Size** | **Key (Constraint)** | **References** | **Nullable** | **Description** | **Default Value** |
| user\_id | INT | 11 | Unique |  | No | Unique key for user\_profile table |  |
| status\_id | INT | 11 | Primary Key |  | No | Primary key for user\_profile table | Auto Increment |
| username | VARCHAR | 16 | Not Null |  | No | User username |  |
| job\_status | VARCHAR | 24 | Not Null |  | No | User job status |  |
| experience | VARCHAR | 24 | Not Null |  | No | User experience at job |  |
| company | VARCHAR | 48 | Not Null |  | No | User work in company |  |
| bio | VARCHAR | 500 | Not Null |  | No | User |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
| PRIMARY | BTREE | Yes | No | status\_id | 6 | A | No |  |
| user\_id | BTREE | Yes | No | user\_id | 6 | A | No |  |

**user\_profile indexes**

**admin**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Field Name (Column)** | **Data Type** | **Field Size** | **Key (Constraint)** | **References** | **Nullable** | **Description** | **Default Value** |
| admin\_id | INT | 11 | Primary Key |  | No | Primary key for admin table | Auto Increment |
| first\_name | VARCHAR | 48 | Not Null |  | No | Admin first name |  |
| last\_name | VARCHAR | 48 | Not Null |  | No | Admin last name |  |
| email | VARCHAR | 64 | Not Null |  | No | Admin email for login |  |
| password | VARCHAR | 32 | Not Null |  | No | Admin password for login |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
| PRIMARY | BTREE | Yes | No | admin\_id | 3 | A | No |  |

**admin indexes**

**company**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Field Name (Column)** | **Data Type** | **Field Size** | **Key (Constraint)** | **References** | **Nullable** | **Description** | **Default Value** |
| company\_id | INT | 11 | Primary Key |  | No | Primary key for company table | Auto Increment |
| company\_name | VARCHAR | 24 | Not Null |  | No | Company name |  |
| company\_address | VARCHAR | 255 | Not Null |  | No | Company address |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
| PRIMARY | BTREE | Yes | No | company\_id | 9 | A | No |  |

**company indexes**

**job**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Field Name (Column)** | **Data Type** | **Field Size** | **Key (Constraint)** | **References** | **Nullable** | **Description** | **Default Value** |
| job\_id | INT | 11 | Primary Key |  | No | Primary key for job table | Auto Increment |
| company\_name | VARCHAR | 24 | Not Null |  | No | Company name for job |  |
| job\_desc | VARCHAR | 500 | Not Null |  | No | Job description |  |
| job\_type | VARCHAR | 10 | Not Null |  | No | Type job |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
| PRIMARY | BTREE | Yes | No | job\_id | 5 | A | No |  |

**job indexes**

**thread**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Field Name (Column)** | **Data Type** | **Field Size** | **Key (Constraint)** | **References** | **Nullable** | **Description** | **Default Value** |
| thread\_id | INT | 11 | Primary Key |  | No | Primary key for thread table | Auto Increment |
| date | DATE |  | Not Null |  | No | Thread posted date |  |
| time | VARCHAR | 8 | Not Null |  | No | Thread posted time |  |
| thread\_desc | VARCHAR | 500 | Not Null |  | No | Thread description |  |
| comment | VARCHAR | 255 | Not Null |  | No | Thread comment |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
| PRIMARY | BTREE | Yes | No | thread\_id | 5 | A | No |  |

**thread indexes**

**message**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Field Name (Column)** | **Data Type** | **Field Size** | **Key (Constraint)** | **References** | **Nullable** | **Description** | **Default Value** |
| message\_id | INT | 11 | Primary Key |  | No | Primary key for message table | Auto Increment |
| date | DATE |  | Not Null |  | No | Message created date |  |
| time | VARCHAR | 8 | Not Null |  | No | Message created time |  |
| message | VARCHAR | 500 | Not Null |  | No | Message description |  |
| to | VARCHAR | 48 | Not Null |  | No | Message sender |  |
| from | VARCHAR | 48 | Not Null |  | No | Message receipent |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
| PRIMARY | BTREE | Yes | No | message\_id | 5 | A | No |  |

**Message indexes**

Task 4

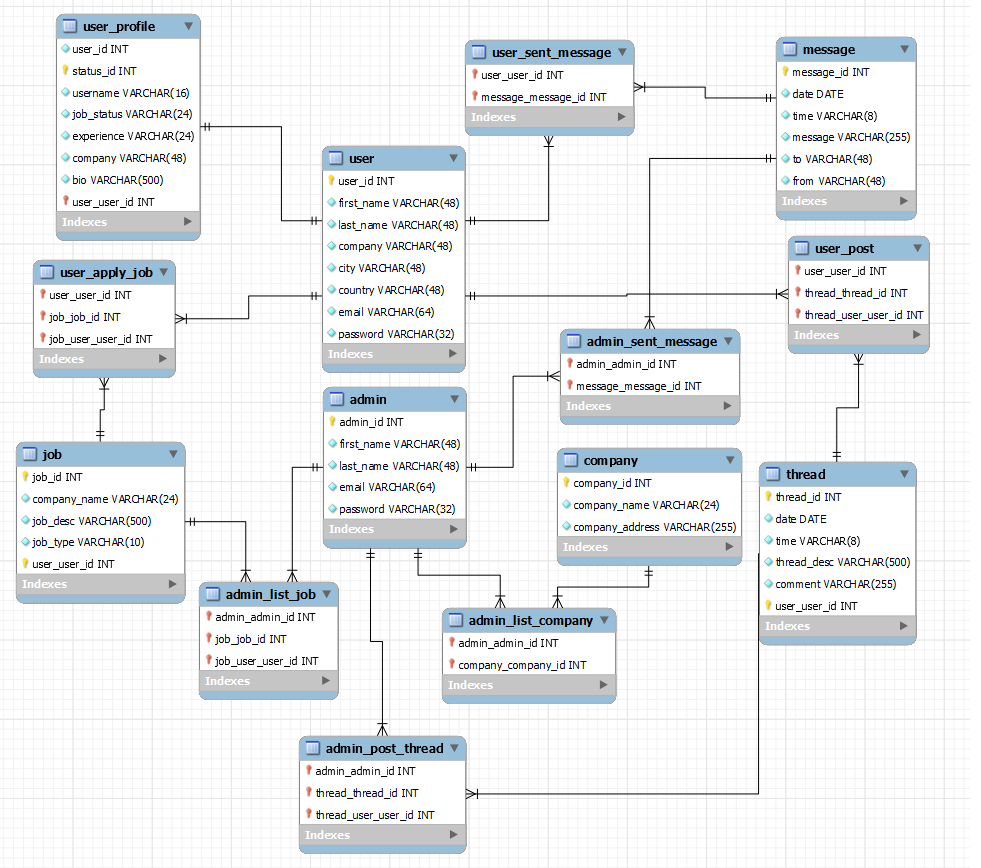
# Database Design Implementation

## 6.1 MySQL Database Scripts & Screen Capture

|  |  |
| --- | --- |
| Table/ Database | Screen Capture |
| abc database |  |
| user |  |
| user profile |  |
| admin |  |
| job |  |
| company |  |
| thread |  |
| message |  |

## 6Logical Design

EER



Task5

# Database Indexes & Backup

1.

## 7.1 Why create indexes?

Indexes is used to speed up the process of retrieving data from the database (SQL CREATE INDEX Statement, n.d.). Instead of scanning the whole table, indexes will select the exact row of the table.

Index is applied on information that is frequently search such as username, job title, email, name and more.

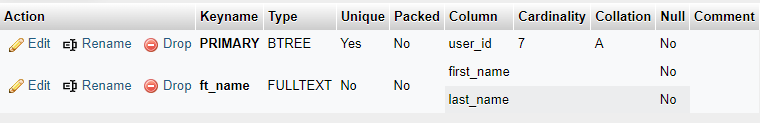
### 7.1.1 Indexes for username

CREATE FULLTEXT INDEX ft\_username ON user\_profile(username);

## 

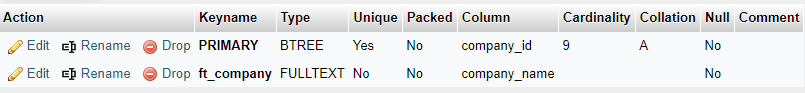
### 7.1.2 Indexes for Name

CREATE FULLTEXT INDEX ft\_name ON user(first\_Name, last\_Name);



### 7.1.3 Indexes for Company

CREATE FULLTEXT INDEX ft\_company ON company(company\_name);



## 7.2 Database Backup

|  |  |
| --- | --- |
| **Steps** | **Screenshot** |
| 1.Execute the mysqldump command in the command prompt |  |
| 2.Create a folder in c drive name as sql-backup and create batch file using notepad++, save as run.bat |  |
| 3. Run bat file with double click for make sure the program is work |  |
| 4.Automate the batch file using windows task scheduler. Click on Create Task |  |
| 5.Name the task as MySQLAutoBackuo6Hours |  |
| 6.In the Triggers section schedule the backup to run every 6 hours |  |
| 7.In Actions, specify the path of the batch file or run.bat file. Then, click OK. |  |
| 8.Active tasks will specify the next run time for the backup database task |  |
| 9. The backup file will appear automatically in the batch file directory |  |

Task6

# Community Portal Query & CSV Sample Data Import

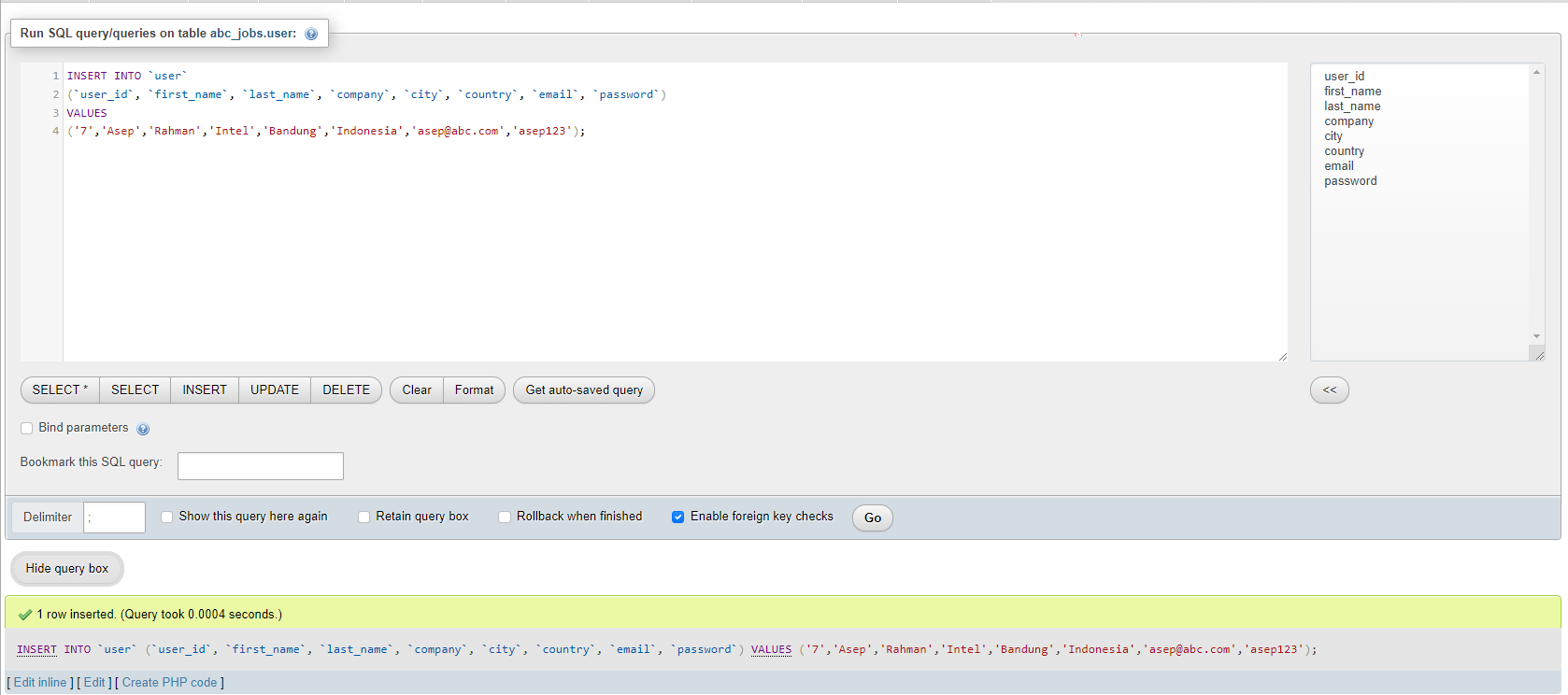
## 8.1 Steps to import CSV files

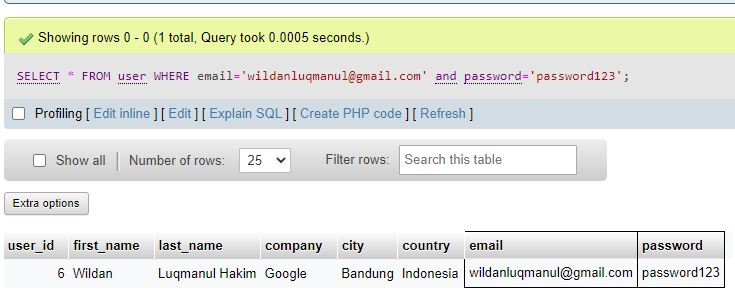
|  |  |
| --- | --- |
| **Steps** | **Screen Capture** |
| Generate csv file using mackaroo, and download it |  |
| Create folder mockaroo in C: and move the csv file into  c:\mackaroo and rename it to “admindata.csv” |  |
| open mysql workbench and open abc\_jobs schema, right click on admin table and select table data import wizard |  |
| Select the directory of the csv file in the mackaroo folder |  |
| Click Next and remained the default setting |  |
| The import is successful and check the imported data |  |
| Result  SELECT \* FROM abc\_jobs.admin; |  |

## 8.2 5 useful queries to develop the application

|  |  |  |  |
| --- | --- | --- | --- |
| Pages | Queries | Note | Evidence |
| Registration | INSERT INTO `user`(`user\_id`, `first\_name`, `last\_name`, `company`, `city`, `country`, `email`, `password`) VALUES ('[value-1]','[value-2]','[value-3]','[value-4]','[value-5]','[value-6]','[value-7]','[value-8]'); | Storing user register information for first name, last name, company, city, country email,  password to the database | T001 |
| Login | SELECT \* FROM user  WHERE email=’[value]’ and password=’[value]’; | Check email and password from user table, if user logging in | T002 |
| Update Profile | UPDATE `user` SET `user\_id`='[value-1]',`first\_name`='[value-2]',`last\_name`='[value-3]',`company`='[value-4]',`city`='[value-5]',`country`='[value-6]' WHERE 1; | update user information for first name, last name, company, city, country in the database | T003 |
| Create Thread | INSERT INTO `thread`(`thread\_id`, `date`, `time`, `thread\_desc`, `comment`) VALUES ('[value-1]','[value-2]','[value-3]','[value-4]','[value-5]'); | Storing user thread information for thread id, date, time, thread description, comment to the database | T004 |

**T001**

****

**T002**

**T003**

****

**T004**

****

## 8.3 5 useful queries to meet the management requirements using joins

|  |  |  |  |
| --- | --- | --- | --- |
| No | Note | Query | Evidence |
| 1 | All users contact info.  To fetch users, contact info information | SELECT user\_id, CONCAT(first\_name, " ", last\_name) AS Name, CONCAT(city,", ",country) AS Location, email AS Email  FROM user  ORDER BY Name; | Report users contact log |
| 2 | All jobs type info, To fetch job type information | SELECT job\_id, company\_name AS Company, job\_desc AS Description, job\_type AS Job\_Type  FROM job  ORDER BY Job\_Type; | Report jobs type log |
| 3 | All users thread.  To fetch users thread information | SELECT user.user\_id, CONCAT(user.first\_name, ' ' ,user.last\_name) AS Name, CONCAT(thread.date, ' ' ,thread.time) AS Date, thread.thread\_desc, thread.comment FROM user JOIN thread ON user.user\_id=thread.thread\_id ORDER BY Name; | Report users thread information |

**Report users contact log**



**Report job type log**



**Report users thread information**

