Project Report

|  |  |
| --- | --- |
| Product Name | Applied Degree in Software Engineering |
| Qualification Name (NICF) | Advanced Certificate in Software Applications (Development and Deployment) |
| Product Name | Database Design and Implementation using MySQL |
| Module Name (NICF) | NICF Database Design and Implementation using MySQL |

|  |  |  |  |
| --- | --- | --- | --- |
| Student name | | Assessor name | |
| Tamim Hossain | | Arvinder Kaur | |
| Date issued | Completion date | | Submitted on |
| 03-08-2022 | 7-08-2022 | | 8-08-2022 |
|  | |  | |
| Project title | Design and implement 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: 06-08-2022 |

Table of Contents

[1. Project Background 4](#_Toc110913211)

[2. Project Objective 4](#_Toc110913212)

[2.1 Project Goal 5](#_Toc110913213)

[2.2 Scope of the Project 5](#_Toc110913214)

[2.3 Tools & Platform 6](#_Toc110913215)

[3. Database Requirements Specification 11](#_Toc110913216)

[3.1 System Requirements 11](#_Toc110913217)

[3.2 Hardware Requirements 11](#_Toc110913218)

[3.3 Software Requirements 12](#_Toc110913219)

[3.4 Database Requirements 12](#_Toc110913220)

[4. Database Design Document 13](#_Toc110913221)

[4.1 What is RDBMS? 13](#_Toc110913222)

[4.2 13](#_Toc110913223)

[4.3 RDBMS CONCEPTS 13](#_Toc110913224)

[4.1.1 What is Conceptual Design? 14](#_Toc110913225)

[4.1.2 What is Logical Design? 14](#_Toc110913226)

[4.1.3 What is Physical design? 14](#_Toc110913227)

[4.2 Entity and attributes for ABC Job Portal 15](#_Toc110913228)

[4.3 ERD Diagram for ABC Job Portal 17](#_Toc110913229)

[4.4 Conceptual Design for ABC Job Portal 18](#_Toc110913230)

[4.5 Logical Design for ABC Job Portal 19](#_Toc110913231)

[4.6 Normalization 19](#_Toc110913232)

[4.6.1 What is Normalization? 19](#_Toc110913233)

[4.6.2 What is 1NF? 20](#_Toc110913234)

[4.6.3 What is 2NF? 20](#_Toc110913235)

[4.6.4 What is 3NF? 20](#_Toc110913236)

[4.7 Normalization of profile\_tbl table 20](#_Toc110913237)

[5. Database Dictionary 25](#_Toc110913238)

[6. Database Design Implementation 32](#_Toc110913239)

[6.2 EERD 37](#_Toc110913240)

[7. Database Indexes & Backup 38](#_Toc110913241)

[7.1 Why create indexes? 38](#_Toc110913242)

[7.1.1 Indexes for username 39](#_Toc110913243)

[7.1.2 Indexes for Name 39](#_Toc110913244)

[7.1.3 Indexes for position on Job table 39](#_Toc110913245)

[7.1.4 Indexes for company\_name on company\_tbl 40](#_Toc110913246)

[7.2 Database Backup 40](#_Toc110913247)

[8. Community Portal Query & CSV Sample Data Import 43](#_Toc110913248)

[8.1 Steps to import CSV files 43](#_Toc110913249)

[8.2 5 useful queries to develop the application 45](#_Toc110913250)

[8.3 5 useful queries to meet the management requirements using joins 48](#_Toc110913251)

# Project Background

‘ABC Jobs Pte Ltd’ is an online based community portal where software programmer can find their preferred job and can post for job also. They can also communicate with each other by texting.  
In that case software programmer is a user and administrator will play the role as an admin on this community portal.  
The outcome of the project will be to store date of every user with their different, different name and id no and also could track every user data and storing their profile information, log-in information, Registration information also storing and tracking the conversation between user to user.  
  
The ‘ABC Jobs Pte Ltd’ presented themselves on this market to do some hand helping changes on the job finding and providing sector as well as better from the primitive way of finding jobs and also to make some communicational bonding between one software engineer to another software engineer and also ‘ABC Jobs Pte Ltd’ is determined to protect personal databases of a user so that an user never feel hesitate to provide their details on the community portal.  
  
Here is some key point to develop the project behind  
  
▸Providing the users to exclude every included steps to input data.  
▸Providing the users the safe inputting facility for their databases.  
▸Secure\_user\_data.  
▸Put back up of user data.

▸provide user different, different like as unique id and naming convention in case of access user data and also for maintain by admin.

▸Take data record of every steps user input data.

▸Serialize and designing the flow of user data.

▸Schedule the data backup system which will be allow to backup user input data automatically in a serialized and maintained manners.

# Project Objective

The pin point and main objective of the project is that the user could input their data frequently and easily. Also the data management system is also the objective of concern.  
Where all data of the user should be well organize and well managed, So that an user and their details can stay in solid track and proper way.  
  
In functional providence user can be done various task the way we designed is the solid implementation of DBMS(Data Base Management System)  
  
Also user should be able to store their data with free of cost where the all barriers of storage will bear by ‘ABC Jobs Pte Ltd’  
  
so the total inclination of the project objective is

We are giving the priority of the user database

▸most significant way  
▸Most safe way  
▸Free of cost storage.  
▸Easy management system.  
▸Easy way to access user own data.  
▸Responsive data backup system.

## 2.1 Project Goal

To design the conceptual and entity relationship diagram (ERD) using ERDPlus

To implement and develop the database using MySQL Workbench,Php My Admin and Command prompt also.

To run functional and non-functional testing

To create a backup database (using Task Scheduler) to run the batch file automatically

To create EERD (Enhanced entity-relationship diagrams) using MYSQL Workbench

To document the database using dbdesc.

## 2.2 Scope of the Project

The scope of this project is to design a Database which will be used for the storage of date for the user of ‘ABC Jobs Pte Ltd’.

▸Designing ERD (Entity Relationship Diagram) As per project requirement

▸Execute ERD Relational schema correctly.

▸Implementing the database as required and designed.

▸Convert database into EERD (Enhanced Entity Relationship Schema)

**Out of scope of the Project**

▸Giving the functionality system for driving users one table to another table

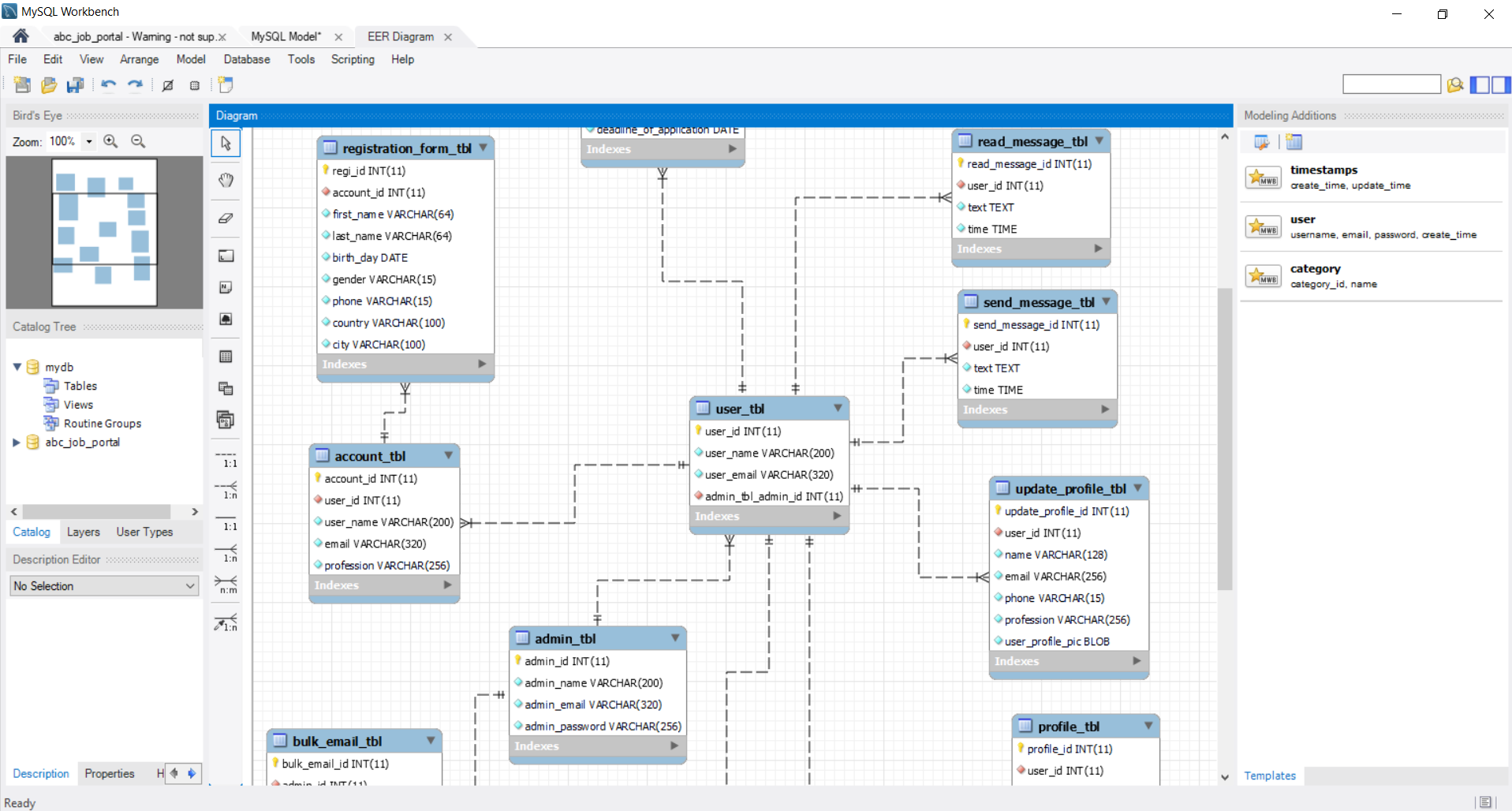
▸Connecting the database with community Portal.

▸Give the Access to the administrator to operate the data as per needed.

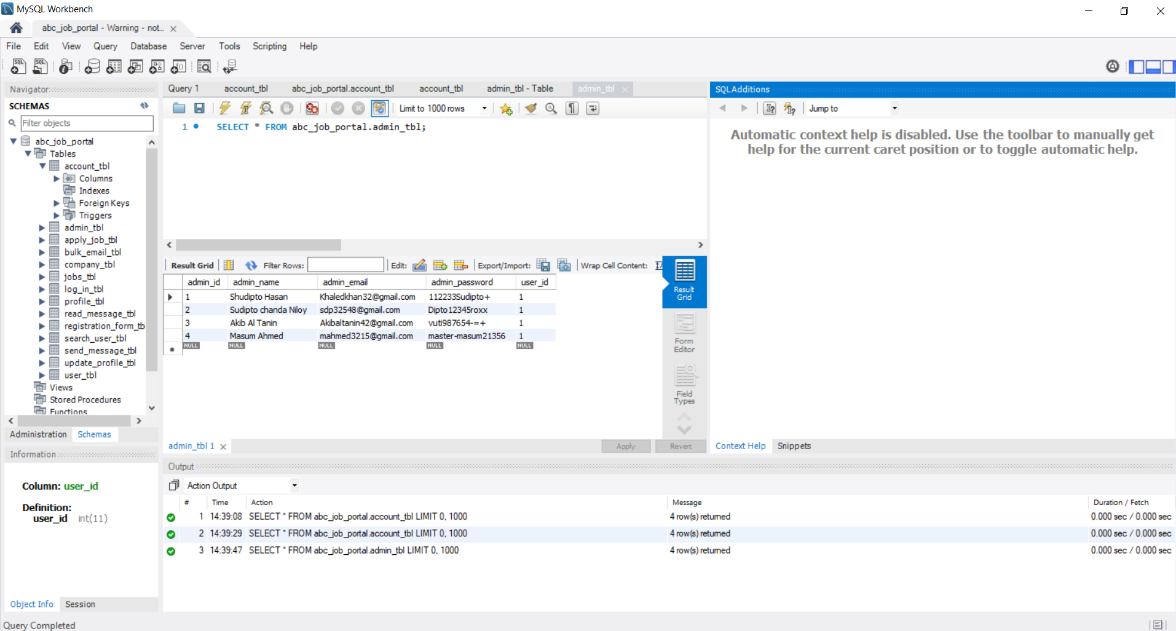
## 2.3 Tools & Platform

MySQL Workbench 8.0 – To create databases and EERD

Creating EERD



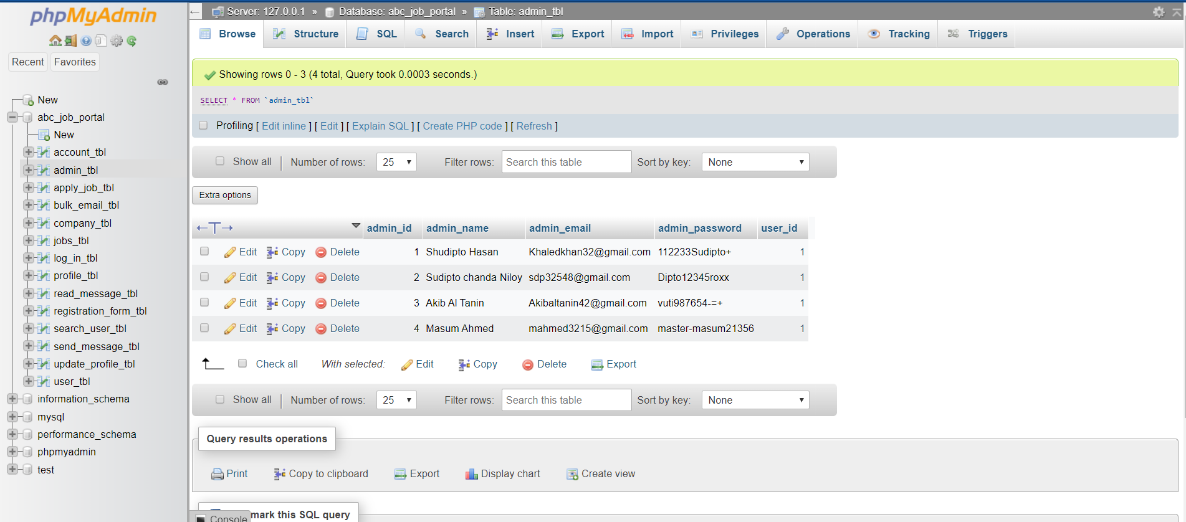
Creating Databases



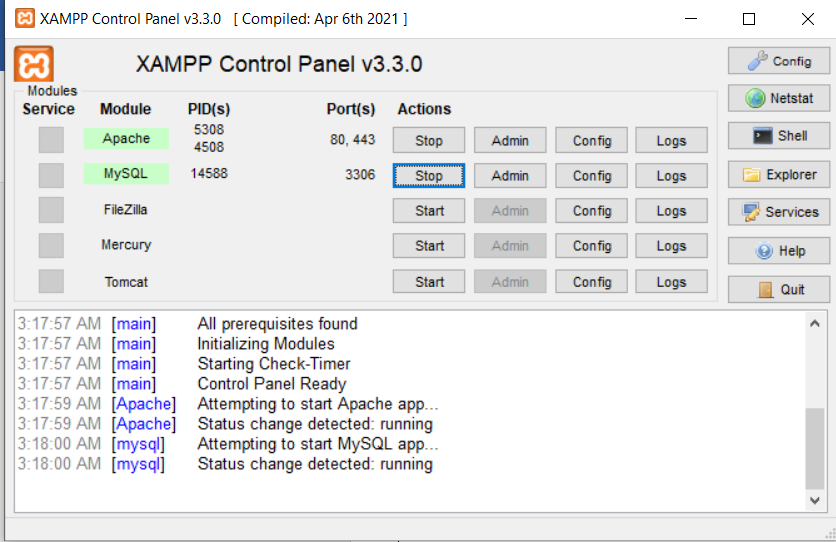
Apache 2.4 to With the use of local host.



phpMyAdmin- using PHPMyAdmin for creating databases.



XAMPP to get connected with Apache web server and My SQL.

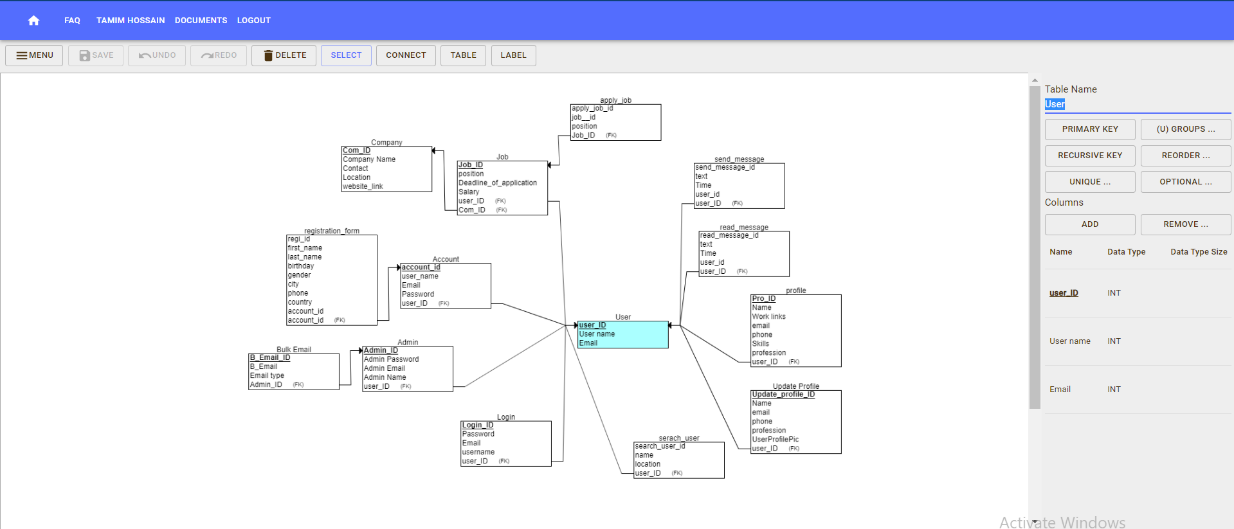


ERDPlus.com to design the ER Diagram and Relational Schema.

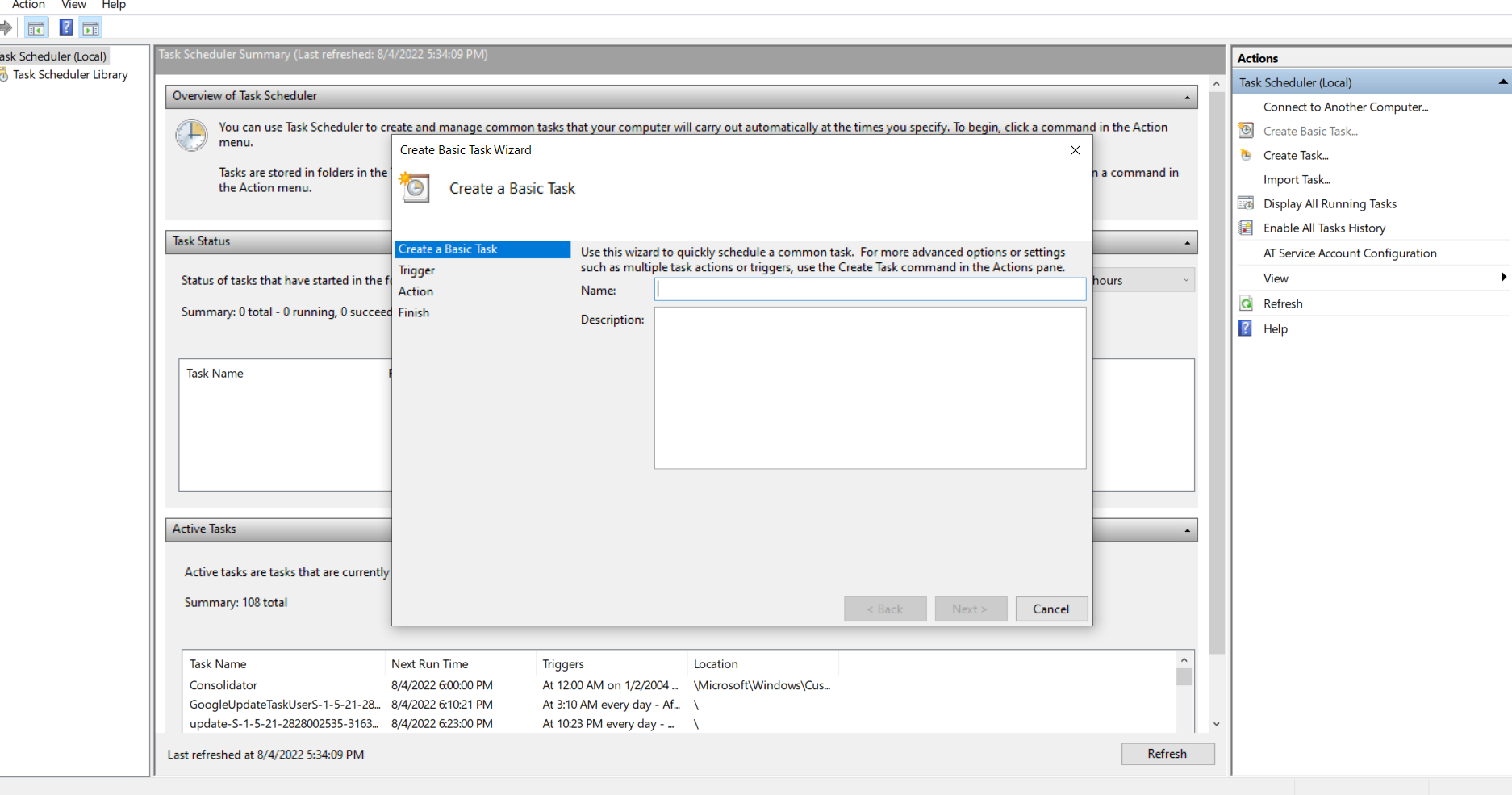
Creating ER Diagram



Creating Relational Schema

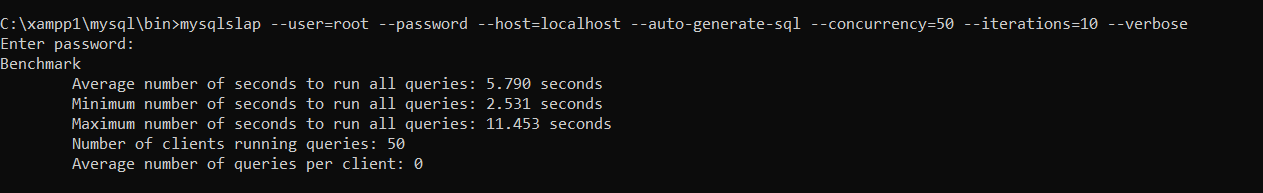


Windows Task Scheduler is used to run the batch script at a scheduled time for the backup database

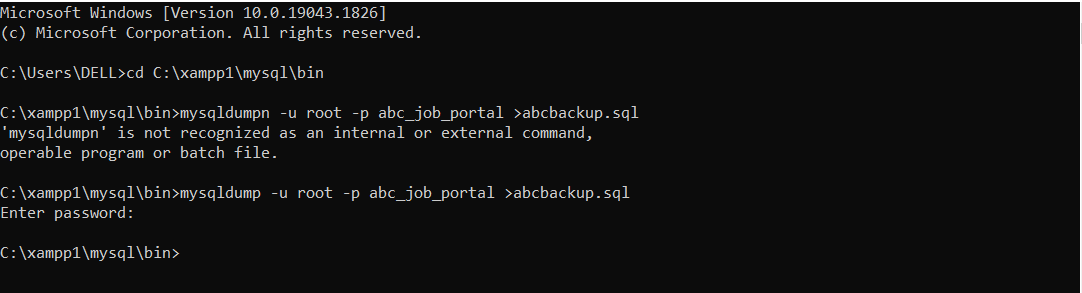


Command Prompt for create backup batch script, load test, and also for create users.

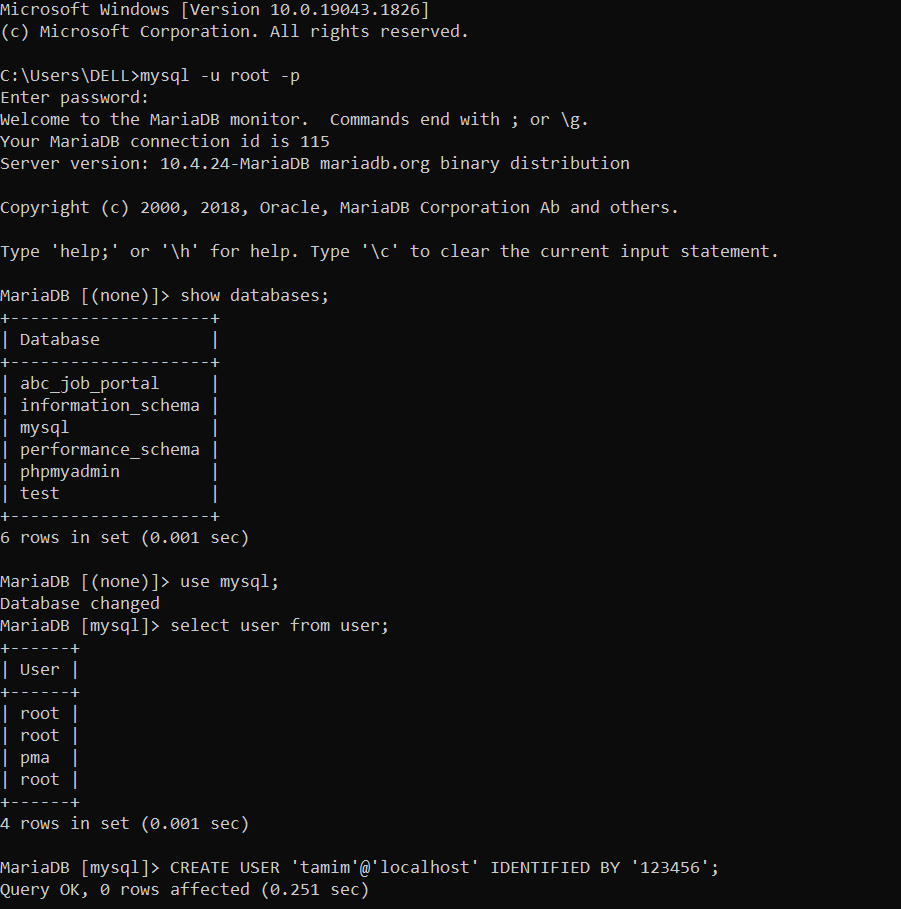
Applying Load Test



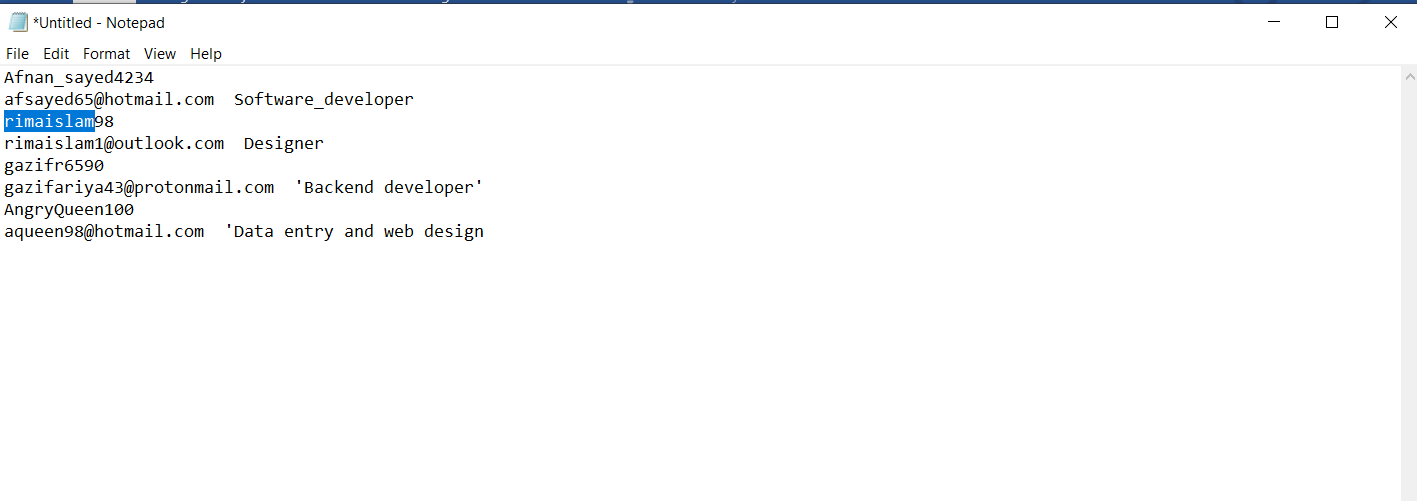
Creating Backup batch script



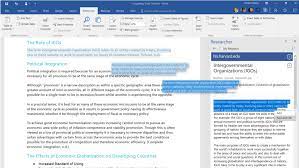
Creating Users.



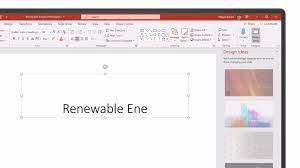
Notepad++ to note down the user information



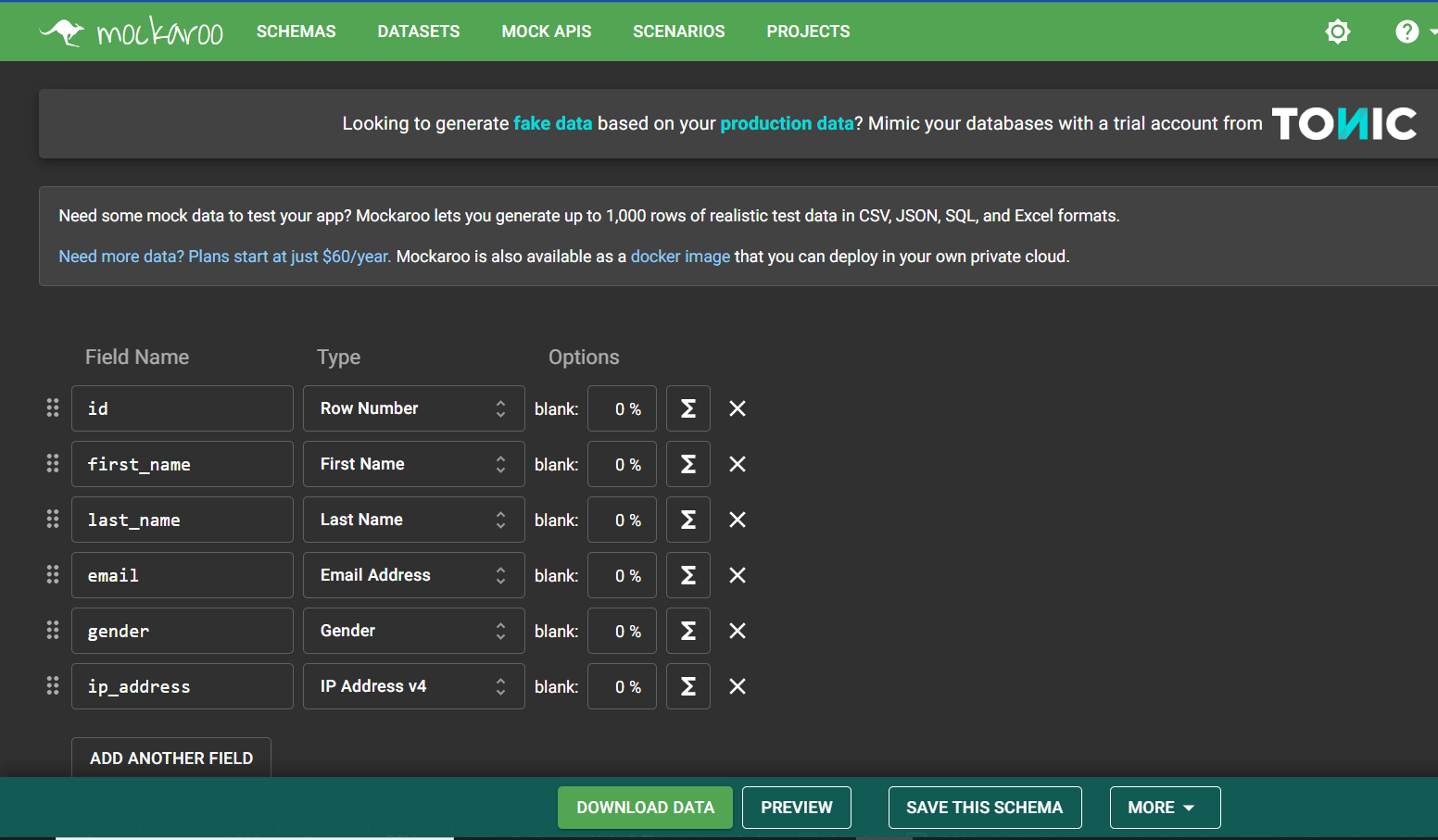
Microsoft word for Project Report



Microsoft PowerPoint for Project Presentation



**Mockaroo.com for generating random databases**



# **Database Requirements Specification**

Hence it is important to describe the requirement for completing and running the database

Here I have discussed the exact and presently used requirement for completing and running the Project

## 3.1 System Requirements

* User Login system
* User Registration System
* User Search System
* Message System
* Apply for job System

## 3.2 Hardware Requirements

* Processor = Intel COREi5 8th Generation
* RAM = 8GB
* Space on SSD/Hard Disk

## 3.3 Software Requirements

* PhpMyAdmin
* Xampp
* MY SQL Workbench 8.0

## 3.4 Database Requirements

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

|  |  |
| --- | --- |
| **Process** | **Database Table** |
| Primary Registration | User, Account, Registration |
| Final Registration | User, Account, Registration |
| Login | User, Login |
| Profile | User, profile |
| Update Profile | User, Update Profile |
| Search & find | User, Search User |
| Send Messages | User, Message |
| Read Messages | User, Message |
| Post Job Opportunities | User, Job |
| Find Job Opportunities | User, job |
| List Job Opportunities & Responses | User, Job, Apply Job |
| Administer user data | User, Profile, Admin |
| Send Bulk Email | Admin, Bulk Email |

Based on the requirement, the relationship information is:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Entity One** | **Entity Two** | **Relationship** | **Note** |
| 1 | User | Account | One to One | One user has one account in the Community Portal |
| 2 | Account | Registration Form | One to one | One Account has one registration form |
| 3 | User | Admin | One to Many | One admin can have access to the user data |
| 4 | User | Update Profile | One to One | One user can update one profile |
| 5 | User | Login | One to One | One user has one account login |
| 6 | User | Profile | One to One | One user has one profile |
| 7 | User | Search User | One to Many | One user can have search many users |
| 8 | User | Send message | One to Many | One user can send many messages |
| 9 | User | Read message | One to many | One user can read many messages |
| 10 | User | Job | One to Many | One user can find many jobs |
| 11 | Admin | Bulk Email | One to Many | One admin can send many emails |
| 12 | Job | Company | Many to Many | Many jobs has Many company |
| 13 | Job | Job Apply | One to One | One job has one application |

Task1

# 4. Database Design Document

* 1. What is RDBMS?

RDBMS stands for Relational Database Management System. It is an information management system that is oriented on a data model

\The software used to store, manage, query, and retrieve data stored in a relational database is called a relational database management system (RDBMS)

* 1. RDBMS CONCEPTS

Relation in the RDBMS is the deposit of tuples having similar characteristics. A relation in RDBMS means the set of the entities and data contained in them. The entities are different instances and their relation is organized in different rows and columns in the data table.  The related information is of identical domains and constraints. By deleting old data and by inserting new data, relations are altered in the database model.

### 4.1.1 What is Conceptual Design?

Conceptual design is an early phase of the design process, in which the broad outlines of function and form of something are articulated. It includes the design of interactions, experiences, processes, and strategies.

**Demonstration:**

Conceptual design is the first step of the multiphase process involved in creating a new product. Whether it’s a building, software application or gadget, it’s important to come up with a general concept before proceeding. The conceptual design phase is immediately followed by the schematic design phase. Conceptual design involves a team convincing the project owner that the idea is worth pursuing. Schematic design means ensuring the concept as sold is actually feasible.

That doesn’t mean that a team can’t determine feasibility before attempting to sell the concept. In fact, often design teams are working from an initial project brief, and the concept stage involves gathering information and researching the market. Many project plans now combine the conceptual and schematic design phases using the term “concept” to describe this stage of project development.

### 4.1.2 What is Logical Design?

A logical design is a conceptual, abstract design. You do not deal with the physical implementation details yet; you deal only with defining the types of information that you need.

The process of logical design involves arranging data into a series of logical relationships called entities and attributes. An entity represents a chunk of information. In relational databases, an entity often maps to a table. An attribute is a component of an entity and helps define the uniqueness of the entity. In relational databases, an attribute maps to a column.

### 4.1.3 What is Physical design?

Physical design is DBMS-specific whereas logical design by contrast is DBMS-independent. Logical design is concerned with the what; physical database design is concerned with the how. In short, physical design is a process of implementing a database on secondary storage with a specific DBMS.

**Definition**

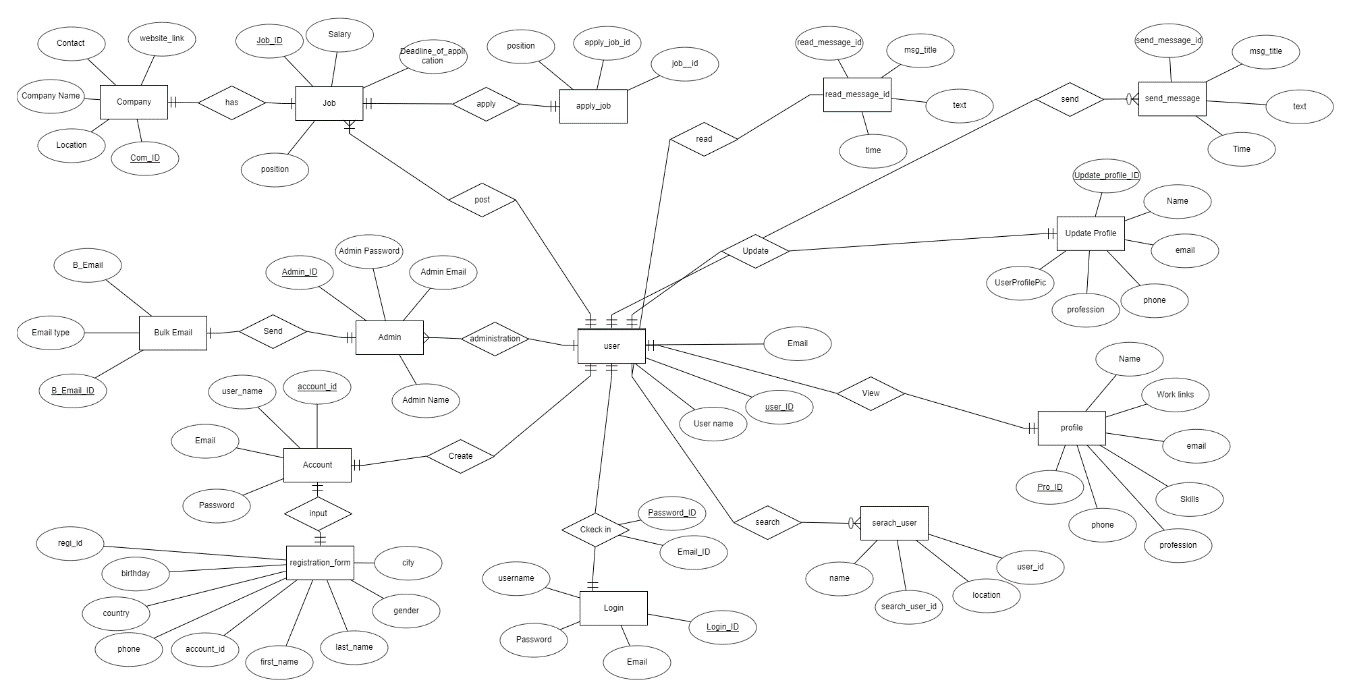
Physical database design represents the materialization of a database into an actual system. While logical design can be performed independently of the eventual database platform, many physical database attributes depend on the specifics and semantics of the target DBMS. Physical design is performed in two stages:

1. Conversion of the logical design into table definitions (often performed by an application developer): includes pre-deployment design, table definitions, normalization, primary and foreign key relationships, and basic indexing.
2. Post deployment physical database design (often performed by a database administrator): includes improving performance, reducing I/O, and streamlining administration tasks

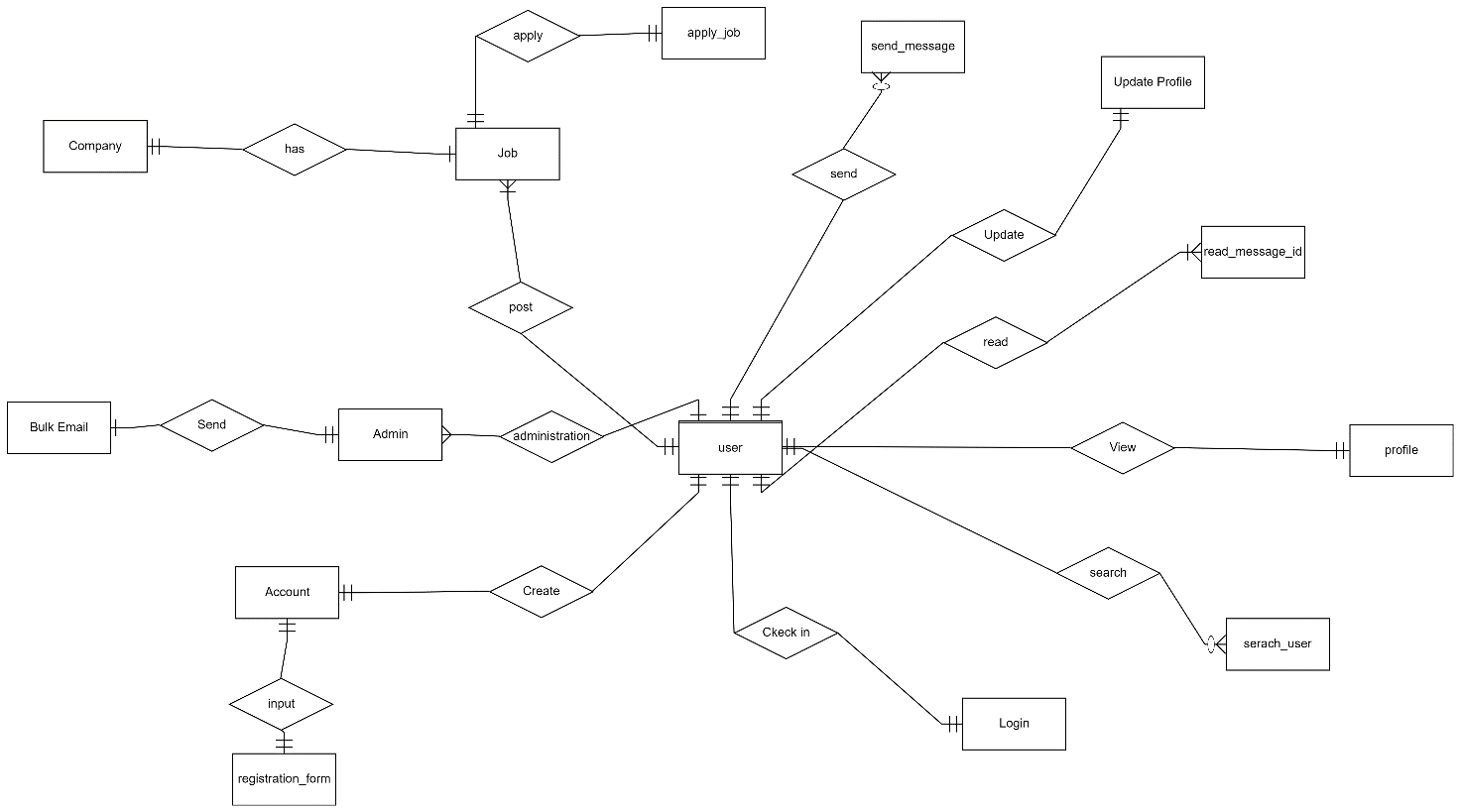
## 4.2 Entity and attributes for ABC Job Portal

|  |  |  |  |
| --- | --- | --- | --- |
| **Entity** | **Description** | **Attributes** | **Page/Function** |
| user | Registered user is uniquely identified in the portal with a User ID | user\_id  user\_name  user\_email | Registration  Login Page  Forget Password |
| admin | Registered user can be an administrator to administer user data and send bulk email | admin\_id  user\_id  admin\_name  admin\_email  admin\_password | Administer user data |
| Log\_in | Registered user can log-in in the community portal with their registered email username and password | log\_in\_id  user\_id  user\_name  email  password | login page  Forget Password |
| read\_message | Registered user can read messages to other users | read\_message\_id  user\_id  msg\_title  text  time | Read Messages  Post in Message Board  List Message Board |
| send\_message | Registered user can send messages to other users | send\_message\_id  user\_id  msg\_title  text  time | Send Messages  Post in Message Board  List Message Board |
| profile | 1 User ID can have 1 profile with general information, contact information and uniquely identified by a Profile ID | profile\_id  user\_id  name  email  phone  skills  work\_links  peofession | Registration  Login  Update Profile |
| update\_profile | 1 user can update their profile from user profile page like as contact logs, name, job information and many more | Update\_profile\_id  user\_id  name  email  phone  profession  User\_profile\_pic | Update Profile |
| account | To be a registered user of the portal users need to create  account on this community portal | account\_id  user\_id  user\_name  email  password | registration page |
| registration\_form | One account has one registration form which will be include all the details of the user.. | regi\_id  account\_id  first\_name  last\_name  birth\_day  gender  phone  country  city | registration page  registration form page. |
| job | Registered user can post job opportunities, apply or list job opportunities and responses | job\_id  company\_id  user\_id  deadline\_of\_application  position  salary | Post Job Opportunities  View Job Opportunities |
| apply\_job | Registered user can apply or list job opportunities and responses | apply\_job\_id  job\_id  position | List Job Opportunities  List Job Responses  Apply |
| company | One posted job has one required company which will request user to apply | Company\_id  company\_name  contact  location  website\_link | post/view job  company |
| bulk\_email | Bulk invite email can only be done by the admin.  Admin will request user to register by bulk email | bulk\_email\_id  admin\_id  bulk\_email  bulk\_email\_type | send bulk email to the users. |
| search\_user | After registration and log in user can search for other user and view their profile information. | Search\_user\_id  user\_id  name  location | registration  log-in  user profile  search user. |

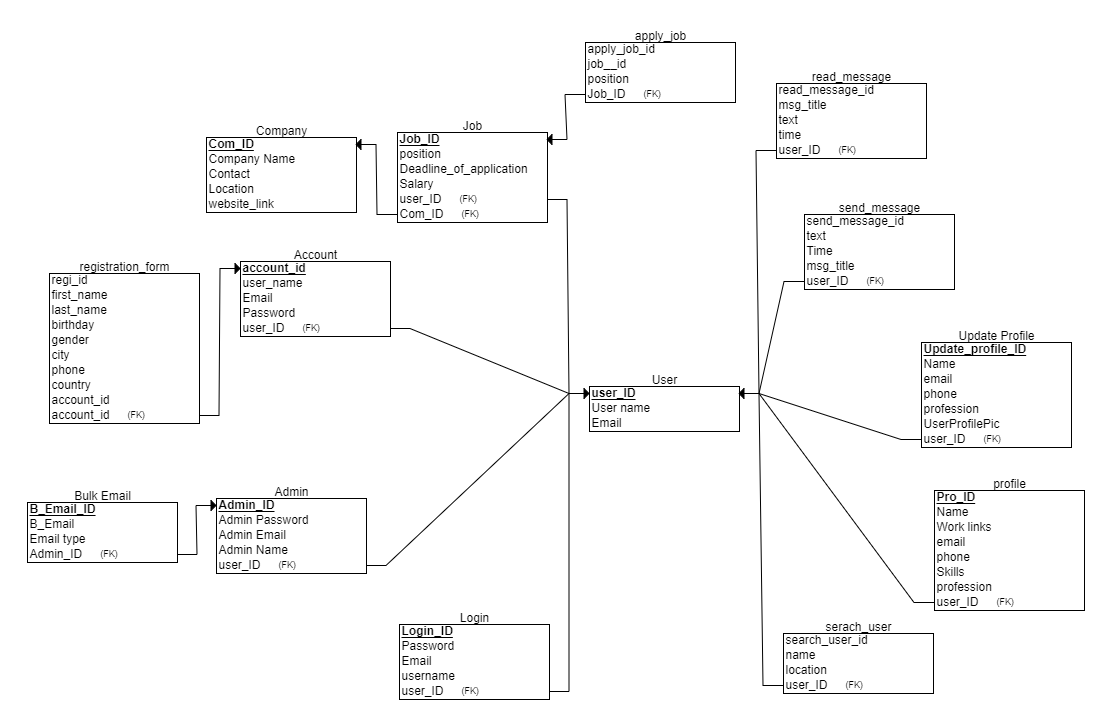
## 4.3 ERD Diagram for ABC Job Portal



## 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 the process of reorganizing data in a database so that it meets two basic requirements:

1. There is no redundancy of data, all data is stored in only one place.
2. Data dependencies are logical, all related data items are stored together.

Normalization is important for many reasons, but chiefly because it allows databases to take up as little disk space as possible, resulting in increased performance.

Normalization is also known as data normalization.

### 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.

If a relation contains a composite or multi-valued attribute, it violates the first normal form, or the relation is in first normal form if it does not contain any **composite** or **multi-valued attribute**. A relation is in first normal form if every attribute in that relation is singled valued attribute.

A table in 1 NF:

1. There are only Single Valued Attributes.
2. Attribute Domain does not change.
3. There is a unique name for every Attribute/Column.
4. The order in which data is stored does not matter.

### 4.6.3 What is 2NF?

Second Normal Form (2NF) is based on the concept of full functional dependency. Second Normal Form applies to relations with composite keys, that is, relations with a primary key composed of two or more attributes. A relation with a single-attribute primary key is automatically in at least 2NF. A relation that is not in 2NF may suffer from the update anomalies.

### 4.6.4 What is 3NF?

A given relation is said to be in its third normal form when it’s in 2NF but has no transitive partial dependency. Meaning, when no transitive dependency exists for the attributes that are non-prime, then the relation can be said to be in 3NF.

In simpler words,

In a relation that is in 1NF or 2NF, when none of the non-primary key attributes transitively depend on their primary keys, then we can say that the relation is in the third normal form of 3NF.

## 4.7 Normalization of profile\_tbl table

|  |
| --- |
| Profile table 1 NF    The table above is already in 1NF as each column has single, unique value and no repeated data. A primary key of Profile\_ID is also uniquely identified. |

|  |
| --- |
| Profile Table 2NF      The 1NF table is normalized to 2NF by splitting into two tables for profile and job with a primary key of job ID. skills, work\_links and profession are dependent upon primary key job\_id only. At this point, there are no partial dependencies. |

|  |
| --- |
| Profile Table 3NF        The 2NF table is normalized to 3NF by dividing the user user jobs table and creating a new table to store list of worklinks. The work code is the primary key in the work table and a foreign key to the primary key in user profile table. |

4.8 Normalization of job\_tbl Table

|  |
| --- |
| Job\_tbl Table 1NF    The table above is already in 1NF as each column has single, unique value and no repeated data. A primary key of Job\_ID is also uniquely identified. |

|  |
| --- |
| Job\_tbl Table 2NF    1NF Job table is already normalized to 2NF as there are no partial dependencies as all non-key attributes are dependent on the Job ID primary key and User ID foreign key. |

|  |
| --- |
| Job\_tbl Table 3NF    The job table is readily normalized to 3NF as it is already in 2NF with no partial and transitive dependencies. All non-primary key attributes position, salary, job salary, deadline of application are fully dependent on the primary key Job ID and foreign key User ID also company\_id |

4.9 Normalization of registration\_tbl Table

|  |
| --- |
| Registration\_tbl Table UNF    The above mentioned form is in Un normalize state as like first name last name are in different column |

|  |
| --- |
| Registration\_tbl Table 1 NF    The above 1NF table are transformed from un normalized form of registration form table Where first\_name and last\_name are combined as name and now the each and every column are unique and identified |

|  |
| --- |
| Registration\_tbl Table 2 NF    The 1NF table is normalized to 2NF by splitting into two tables as registration form table and user location table Where country and city are mentioned as location and both are identified by the unique primary key id called location id |

|  |
| --- |
| Registration\_tbl Table 3 NF    The 2NF table is normalized to 3NF by dividing driving the user location tables column called county to country code table now different kind of country will be identified by a unique key or primary key the primary key for country is country code . |

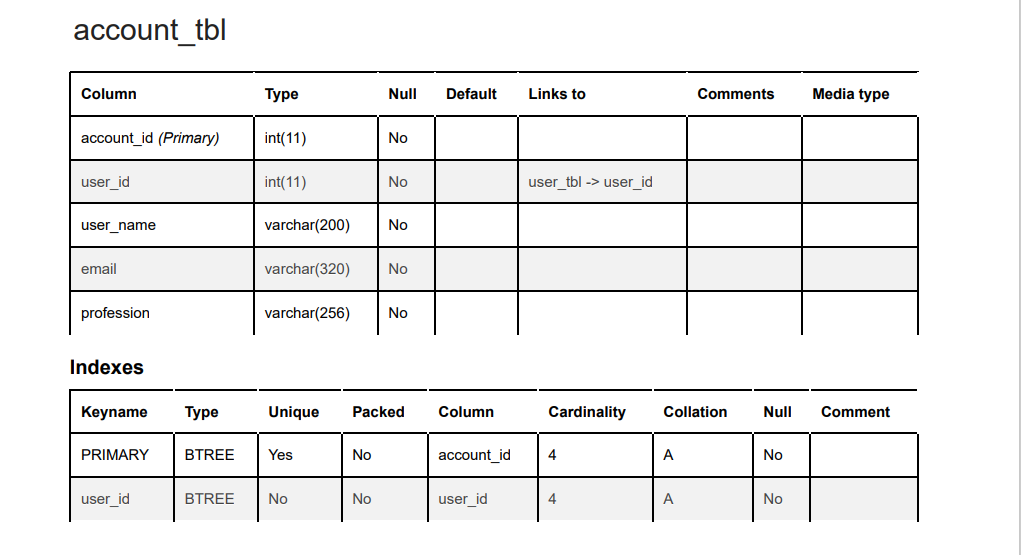
Task 3

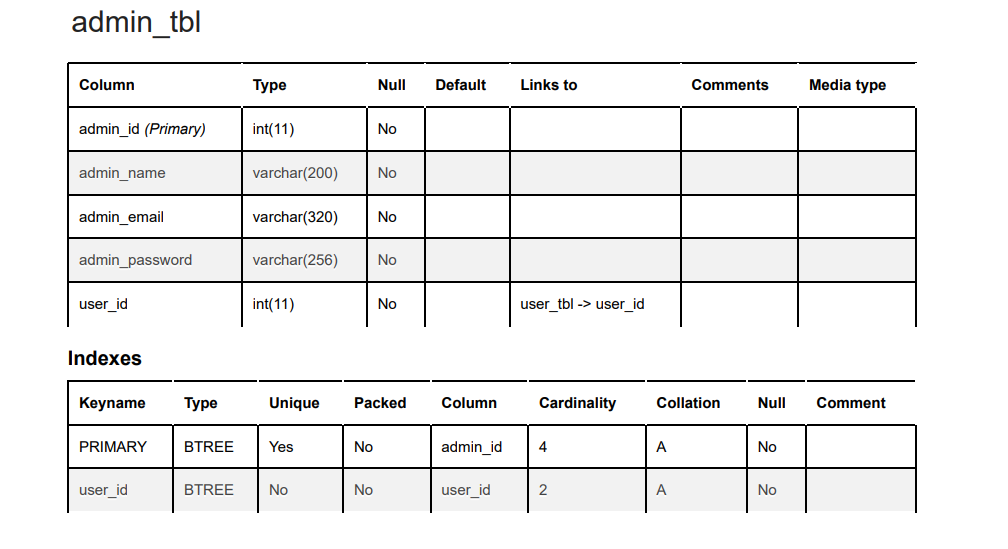
# Database Dictionary

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.

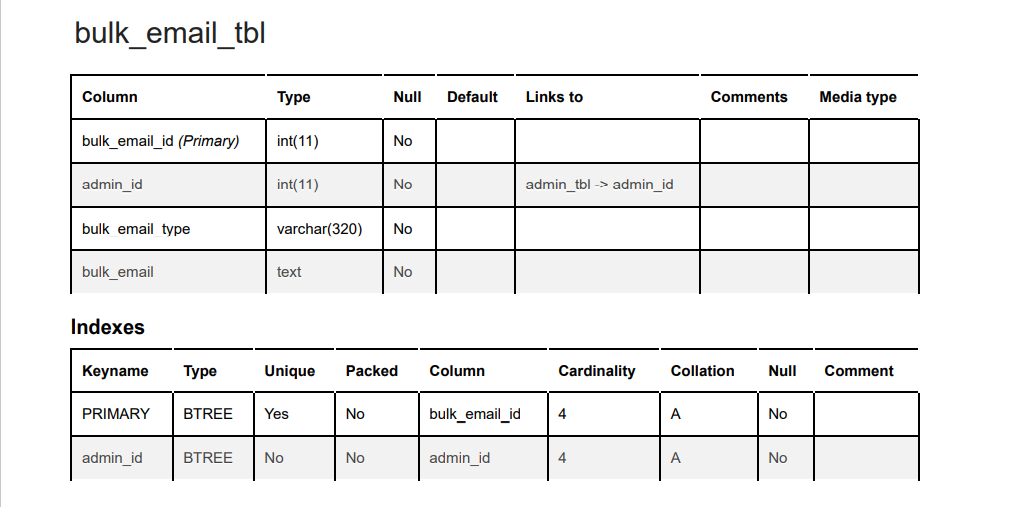
Here Is the Database Dictionary PDF mentioned.

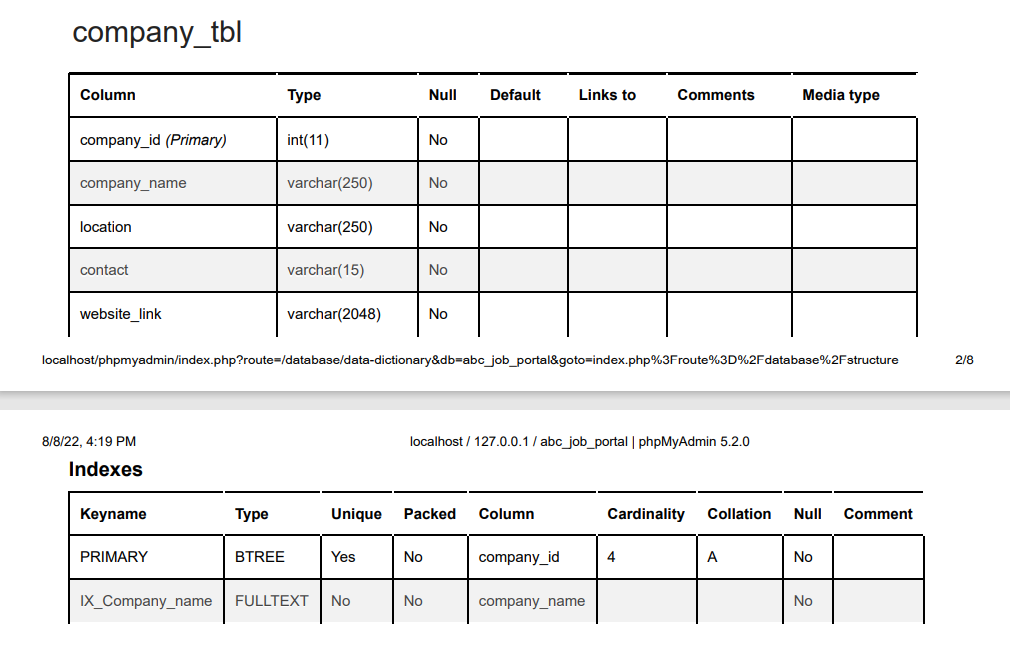




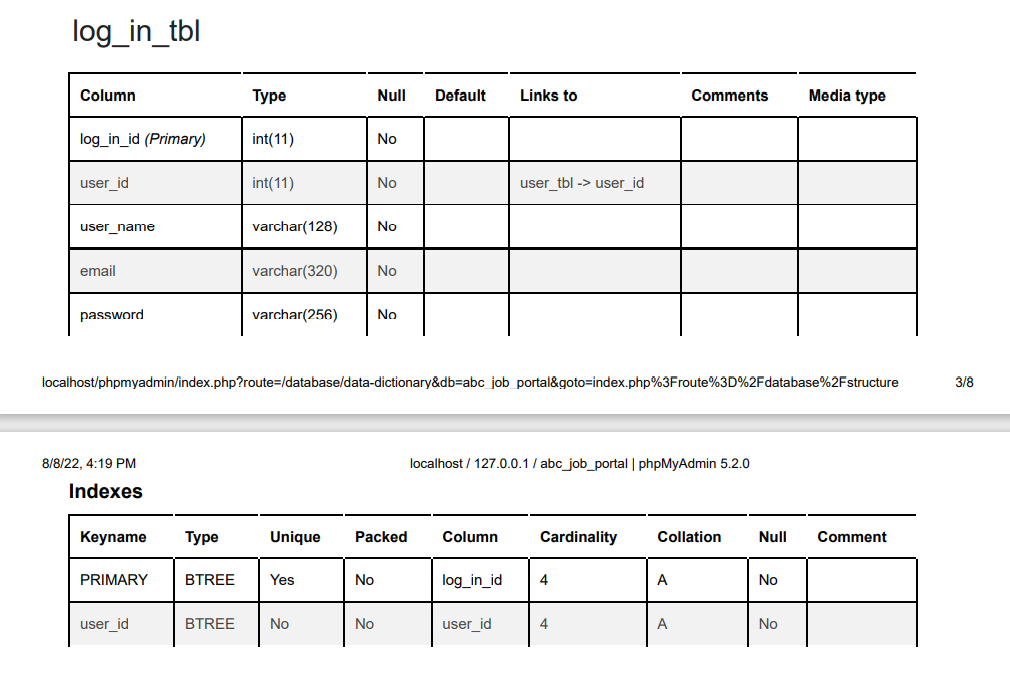


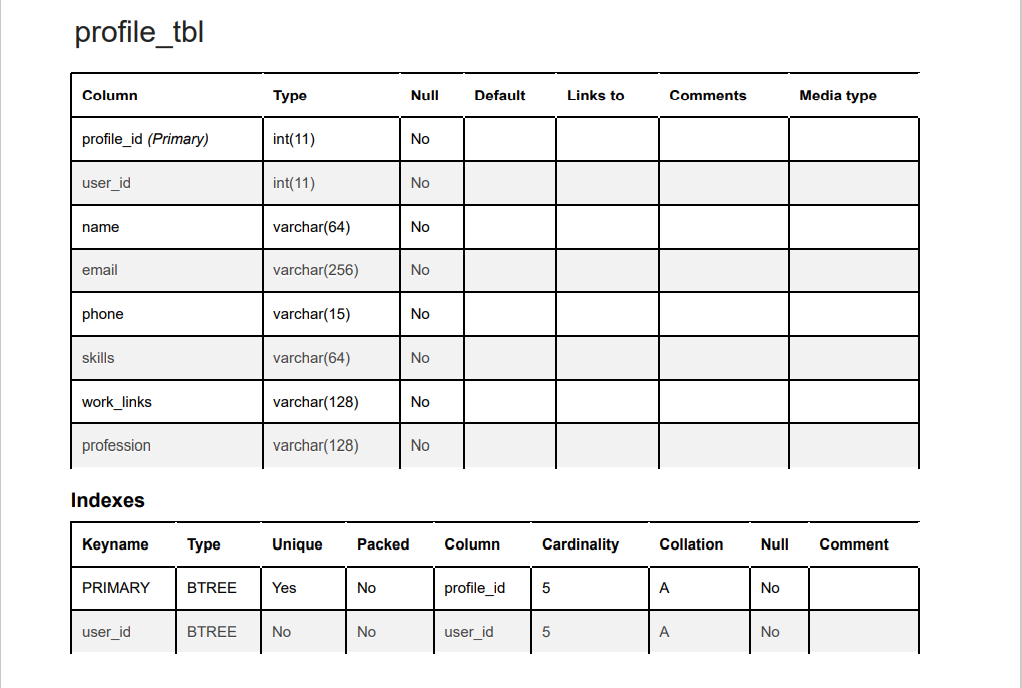


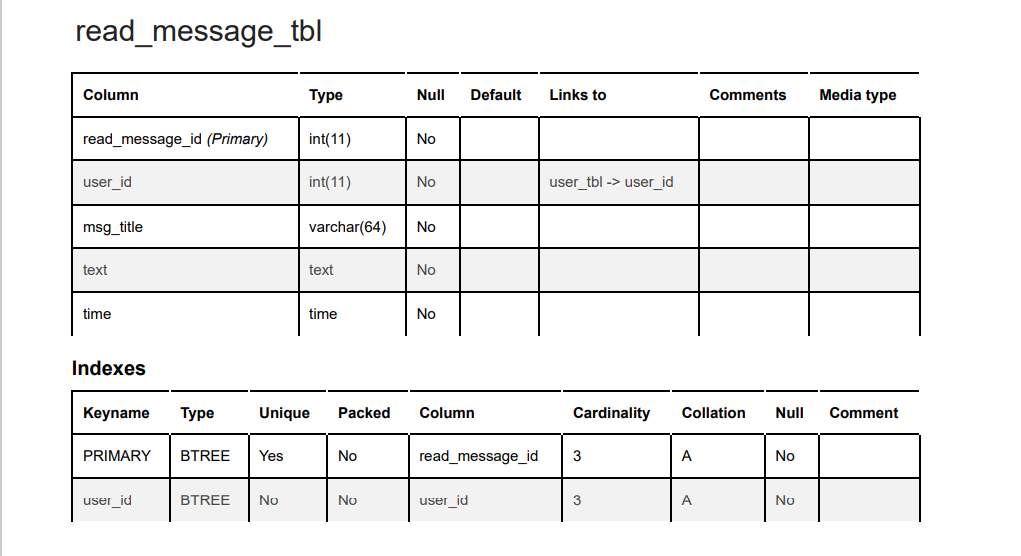




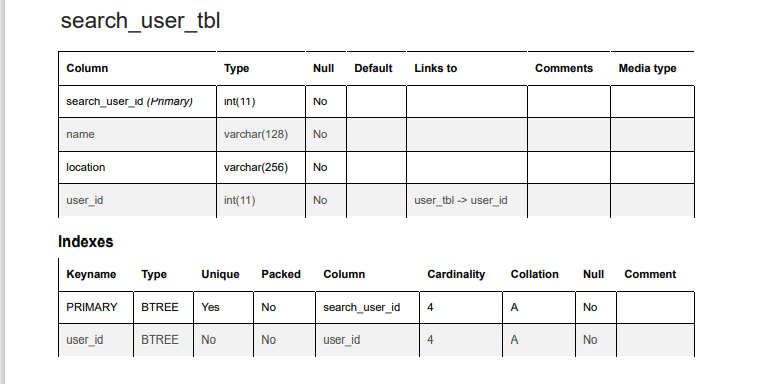




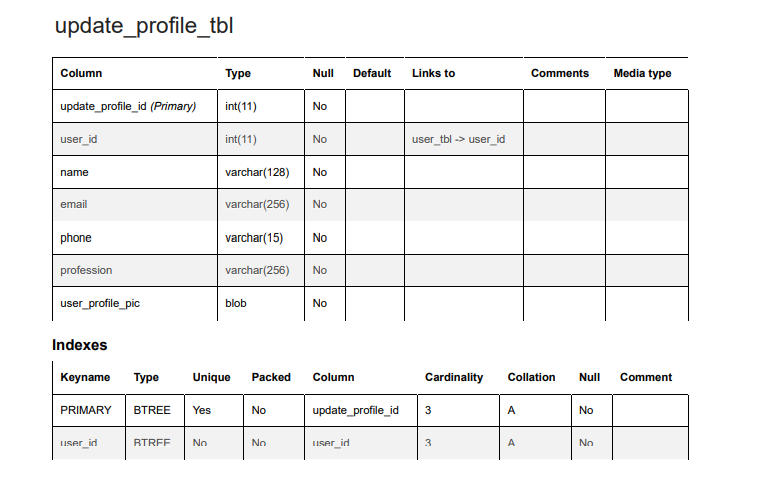


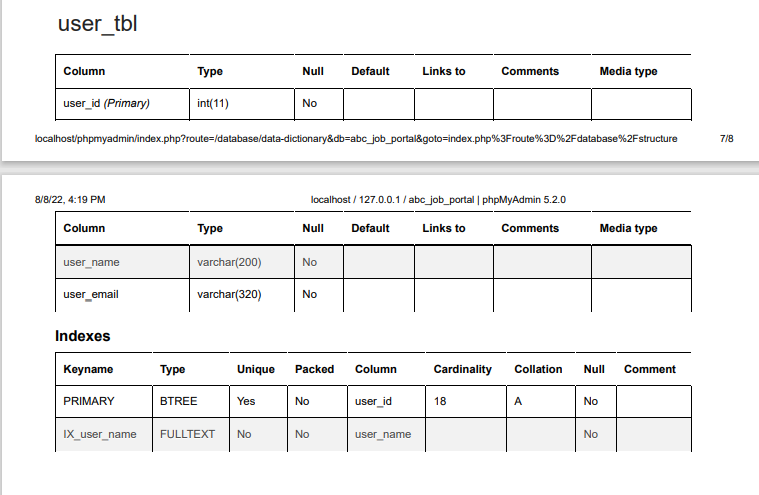












Task 4

# Database Design Implementation

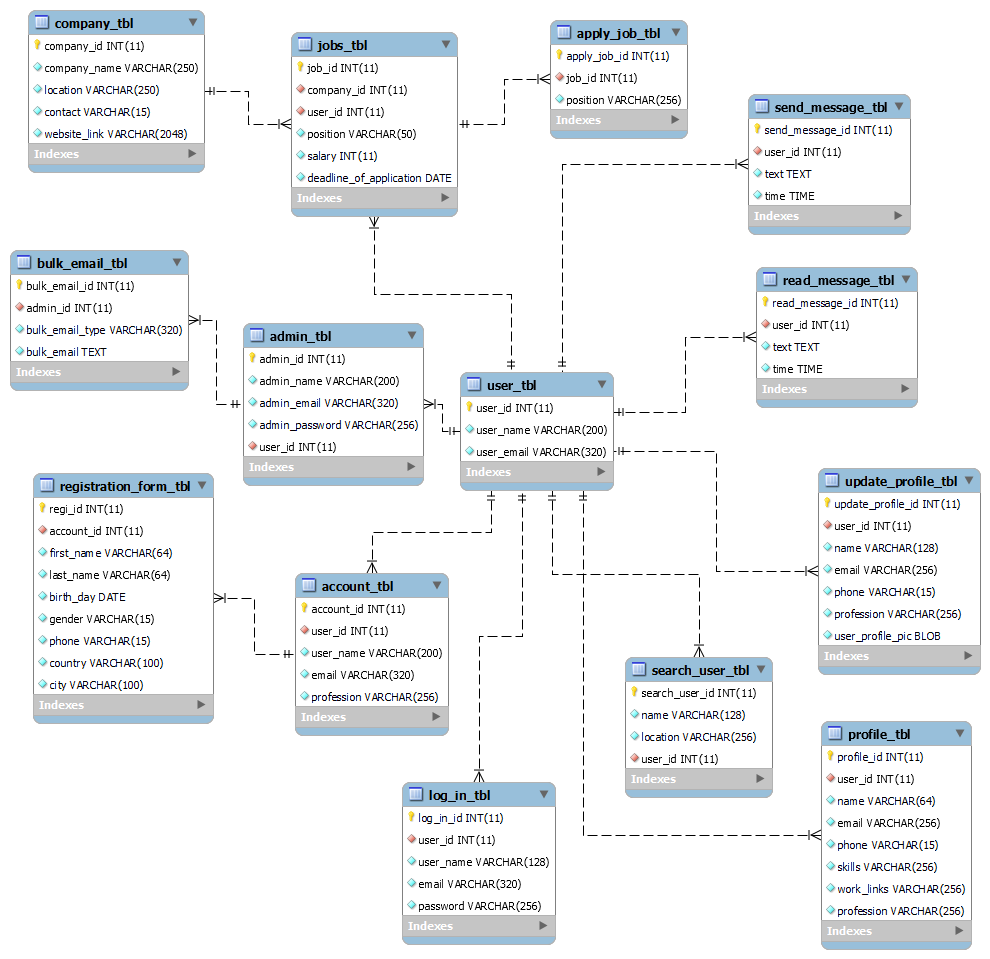
MySQL Database Scripts & Screen Capture

|  |  |
| --- | --- |
| **Table/ Database** | **Screen Capture** |
| abc\_job\_portal |  |
| account\_tbl |  |
| admin\_tbl |  |
| apply\_job\_tbl |  |
| Bulk\_email\_tbl |  |
| company\_tbl |  |
| Jobs\_tbl |  |
| Log\_in\_tbl |  |
| Profile\_tbl |  |
| read\_message\_tbl |  |
| registration\_form\_tbl |  |
| search\_user\_tbl |  |
| send\_message\_tbl |  |
| update\_profile\_tbl |  |
| User\_tbl |  |

## 6.2 EERD

The enhanced entity–relationship (EER) model (or extended entity–relationship model) in computer science is a high-level or conceptual data model incorporating extensions to the original entity–relationship (ER) model, used in the design of databases.

The EER model includes all of the concepts introduced by the ER model. Additionally it includes the concepts of a subclass and superclass (Is-a), along with the concepts of specialization and generalization. Furthermore, it introduces the concept of a union type or category, which is used to represent a collection of objects that is the union of objects of different entity types. EER model also includes EER diagrams that are conceptual models that accurately represent the requirements of complex databases.



Task5

1. Database Indexes & Backup

## 7.1 Why create indexes?

Indexes are special lookup tables that the database search engine can use to speed up data retrieval. Simply put, an index is a pointer to data in a table. An index in a database is very similar to an index in the back of a book.

For example, if you want to reference all pages in a book that discusses a certain topic, you first refer to the index, which lists all the topics alphabetically and are then referred to one or more specific page numbers.

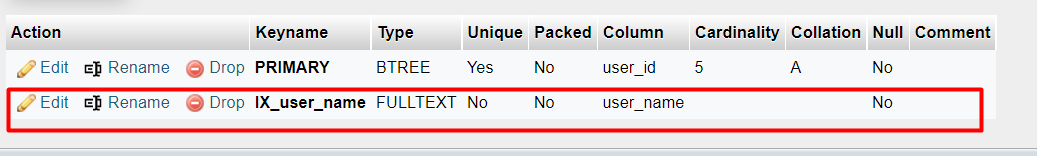
An index helps to speed up SELECT queries and WHERE clauses, but it slows down data input, with the UPDATE and the INSERT statements. Indexes can be created or dropped with no effect on the data.

Creating an index involves the CREATE INDEX statement, which allows you to name the index, to specify the table and which column or columns to index, and to indicate whether the index is in an ascending or descending order.

Indexes can also be unique, like the UNIQUE constraint, in that the index prevents duplicate entries in the column or combination of columns on which there is an index.

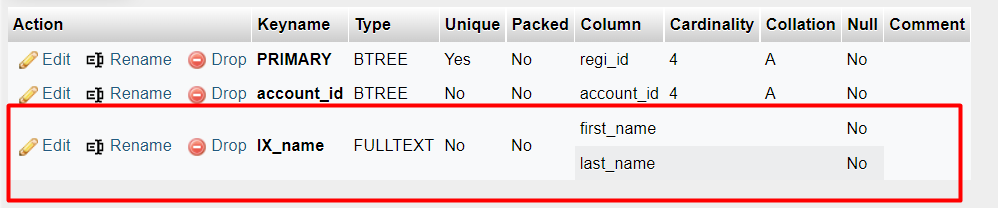
### 7.1.1 Indexes for username

CREATE FULLTEXT INDEX IX\_user\_name ON user\_tbl(user\_name);



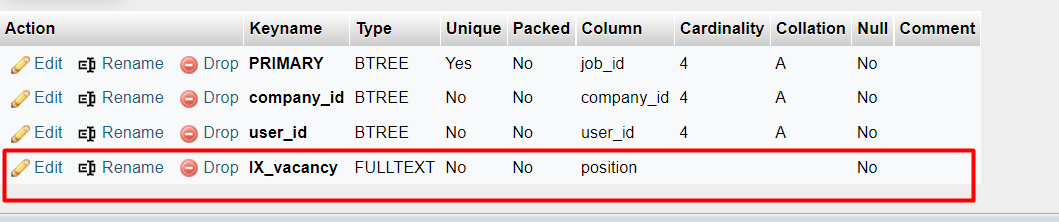
### 7.1.2 Indexes for Name

CREATE FULLTEXT INDEX IX\_name ON registration\_form\_tbl(first\_name, last\_name);



### 7.1.3 Indexes for position on Job table

CREATE FULLTEXT INDEX IX\_vacancy ON job\_tbl(position);



### 7.1.4 Indexes for company\_name on company\_tbl

CREATE FULLTEXT INDEX IX\_company\_name ON company\_tbl(position);



## 7.2 Database Backup

|  |  |
| --- | --- |
| **steps** | **screenshot** |
| 1. Locate My SQL Bin folder First |  |
| 2.Create a Batch script backup file on the bin folder using the following command |  |
| 3.Automate the batch file using window Task scheduler |  |
| 4.Name the task as backup database |  |
| 5.In the Triggers section schedule the backup to run every 6 hours |  |
| 6.In Actions, specify the path of the batch file or abcbackup.sql file. Then, click OK. |  |
| 7.Active tasks will specify the next run time for the backup database task |  |

Task6

# Community Portal Query & CSV Sample Data Import

## 8.1 Steps to import CSV files

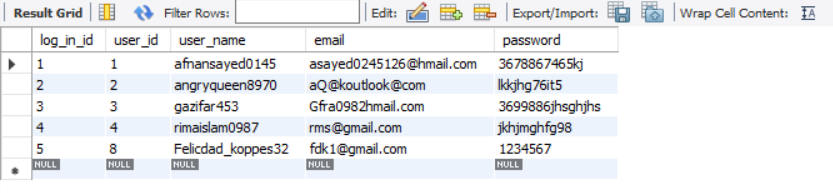
|  |  |
| --- | --- |
| 1.Generate or create a csv file in mockaroo for user\_id, username and email. Once downloaded mockaroo will display the data in excel. |  |
| Save the excel file as Comma delimited |  |
| Now go to the php my Admin and go the user tableand and press import Then u will see Choose file then choose your preffered CSV file |  |
| Set the importing file as CSV |  |
| Files successfully  uploaded |  |
| Click on Browse to see the list of the table… |  |

## 8.2 5 useful queries to develop the application

|  |  |  |  |
| --- | --- | --- | --- |
| **Pages** | **Queries** | **Note** | **Evidence** |
| Log-in | INSERT INTO  `log\_in\_tbl`  (`log\_in\_id`, `user\_id`, `user\_name`, `email`, `password`)  VALUES  (‘[value]’, ‘[value]’, ‘[value]’, ‘[value]’, ‘[value]’,) | Storing users  Log-in log-in details  user name, Email,  Password,  to the database | Fig Q01 |
| Account | INSERT INTO `account\_tbl` (`account\_id`, `user\_id`, `user\_name`, `email`, `password`) VALUES  (‘[value]’, ‘[value]’, ‘[value]’, ‘[value]’, ‘[value]’,) | Storing users Registration details  user name, Email,  Password,  to the database | Fig Q02 |
| Profile | INSERT INTO `profile\_tbl` (`profile\_id`, `user\_id`, `name`, `email`, `phone`, `skills`, `work\_links`, `profession`) VALUES  (‘[value]’, ‘[value]’, ‘[value]’, ‘[value]’, ‘[value]’, ‘[value]’, ‘[value]’, ‘[value]’,) | Storing users profile details  name,  Email,  Phone,  Skills,  Work links,  profession  to the database | Fig Q03 |
| User | INSERT INTO `user\_tbl` (`user\_id`, `user\_name`, `user\_email`) VALUES  (‘[value]’, ‘[value]’, ‘[value]’,) | Storing  users details  user\_name,  user\_mail,  to the database | Fig Q04 |
| Admin | INSERT INTO `admin\_tbl` (`admin\_id`, `admin\_name`, `admin\_email`, `admin\_password`, `user\_id`) VALUES (NULL, 'khan\_jamal', 'khanjamal@gmail.com', '021154785', '4'); | Storing admin details Admin Name, Email  and password to the database | Fig Q05 |

**Fig Q01**





**Fig Q02**



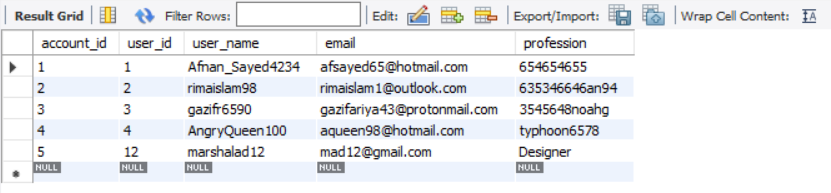


Fig Q03



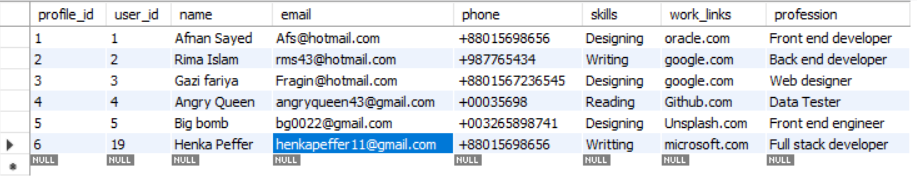


Fig Q04

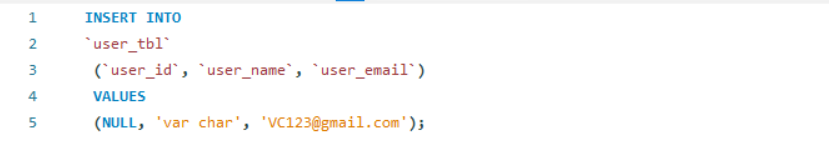
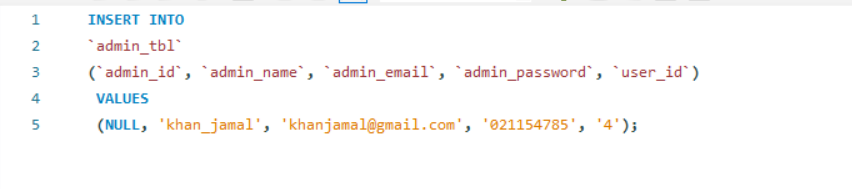
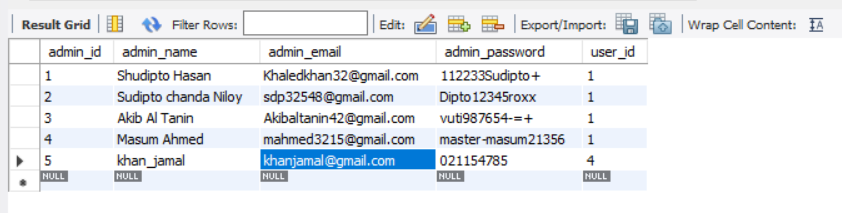




Fig Q05





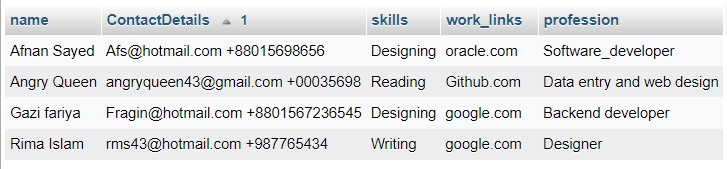
## 8.3 5 useful queries to meet the management requirements using joins

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Note** | **Query** | **Evidence** |
| 1 | All users personal details info.  To fetch users, registration information | SELECT  regi\_id,  CONCAT(first\_name, ' ',last\_name) AS name,  birth\_day, gender,phone, CONCAT(city, ', ',country)  AS location  FROM  registration\_form\_tbl ORDER BY name; | Report users contact log |
| 2 | All users profile and update profile  information to fetch user work information | SELECT  profile\_tbl.name,  CONCAT(profile\_tbl.email, ' ' ,profile\_tbl.phone)  AS ContactDetails,  profile\_tbl.skills, profile\_tbl.work\_links, update\_profile\_tbl.profession  FROM  profile\_tbl JOIN update\_profile\_tbl  ON  profile\_tbl.profile\_id=update\_profile\_tbl.update\_profile\_id ORDER BY ContactDetails; | Report user contact and work information |
| 3 | Company and job information to fetch total job details | SELECT  jobs\_tbl.job\_id, jobs\_tbl.user\_id, jobs\_tbl.position, jobs\_tbl.salary, jobs\_tbl.  deadline\_of\_application,company\_tbl.company\_name  FROM  jobs\_tbl  JOIN  company\_tbl ON jobs\_tbl.job\_id = company\_tbl.  company\_id  ORDER BY jobs\_tbl.deadline\_of\_application; | Report job details |
| 4 | Send message and read message information for fetching a total message information of a user. | SELECT  read\_message\_tbl.read\_message\_id, send\_message\_tbl.send\_message\_id,  read\_message\_tbl.user\_id,read\_message\_tbl.  text,read\_message\_tbl.time  FROM  read\_message\_tbl  JOIN  send\_message\_tbl ON read\_message\_tbl.read\_message\_id = send\_message\_tbl.send\_message\_id  ORDER BY read\_message\_tbl.read\_message\_id; | Report message information |
| 5 | Admin and bulk email information for fetching Bulk email details | SELECT  admin\_tbl.admin\_name, admin\_tbl.admin\_email, bulk\_email\_tbl.bulk\_email\_type,bulk\_email\_tbl.bulk\_email  FROM  admin\_tbl  JOIN  bulk\_email\_tbl ON admin\_tbl.admin\_id = bulk\_email\_tbl.bulk\_email\_id  ORDER BY admin\_tbl.admin\_id; | Report bulk email  sender details |

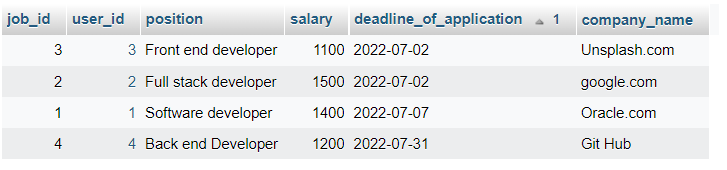
**Report users contact log**



**Report user contact and work information**



**Report job details**



**Report Message Information**



**Report Bulk email sender details**

