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**
Chathushi Jayarathna		MRS. Arvindar Kaur
Date issued	**Completion date**	**Submitted on**
9/07/2022	9/08/2022	8/8/2022

Project title	Design, Implement, Test & Document Community

Learner declaration
I certify that the work submitted for this assignment is my own and research sources are fully acknowledged. Student signature: Chathushi Date: 8/8/2022

Table of Contents

Table of Figures

1. Project Background

Project Definition

ABC Jobs Pte Ltd is an online-primarily based community portal in which software programmer can locate their preferred activity and may put up for activity also. They can create accounts, update profiles, view profiles, search for users, and communicate with others. The platform is primarily used for professional networking and career development, allowing job seekers to post their CVs and job to employers. Here, Portal users can search for other users based on various parameters such as Name, Company, City, Country, Job, Email & Username. After browsing, the user can view the user's public information. The portal allows users to log in, request forgotten passwords and update their profile information.

According to this project, databases should be created as background development. This will help the software developers to study the clients in a more convenient way and then update their reports. The ABC Jobs Pte Ltd was introduced in this market to support the changes in the job search and posting sector as well as to improve the primitive method of job search and also to provide a communication link between a software developer and other. The developers as well as ABC Jobs Pte Ltd are committed to protecting a user's personal databases, so a user never hesitates to provide their details on the community portal.

2. Project Objectives

2.1 Project Goals

- To create ERD (Entity Relationship Diagram) and Relational Schema using ERDPlus
- To create and implement databases using MYSQL Workbench
- 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

2.2 Project Scope

- Make the database include all the attributes, Relationships
- Making the ERD Relational Schema correctly
- Implementing the database
- Covert database into EERD

2.3 Project Out of Scope

- Connect the database with the community Portal
- Add administrator to the database

2.4 Tools & Platforms used

- **MYSQL Workbench 8.0.30** – To create Databases, EERD

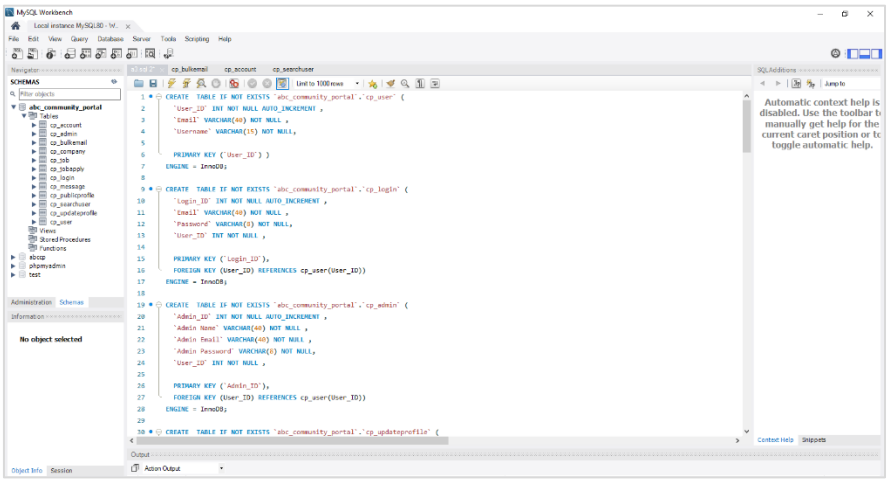


Figure 1:Screenshot of MySQL Workbench

- **XAMPP** – To connect with apache, MYSQL and connect MYSQL databases with PHP my admin

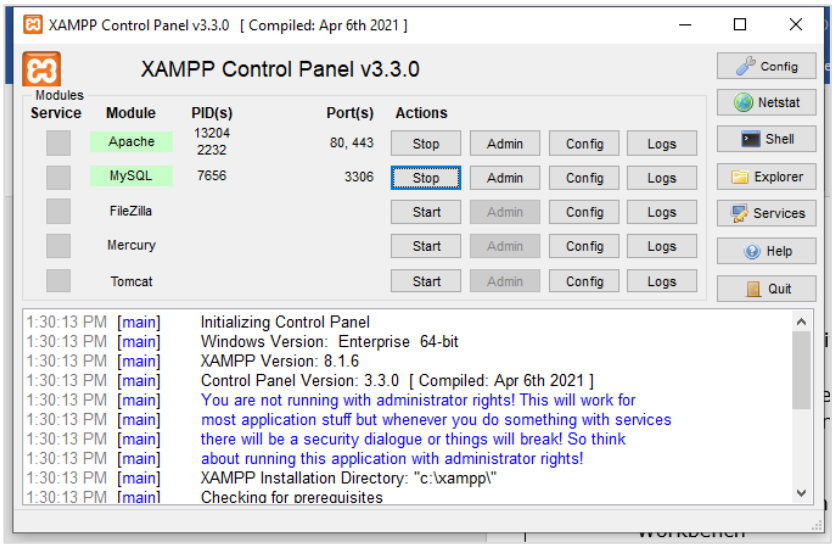


Figure 2:Screenshot of XAMPP

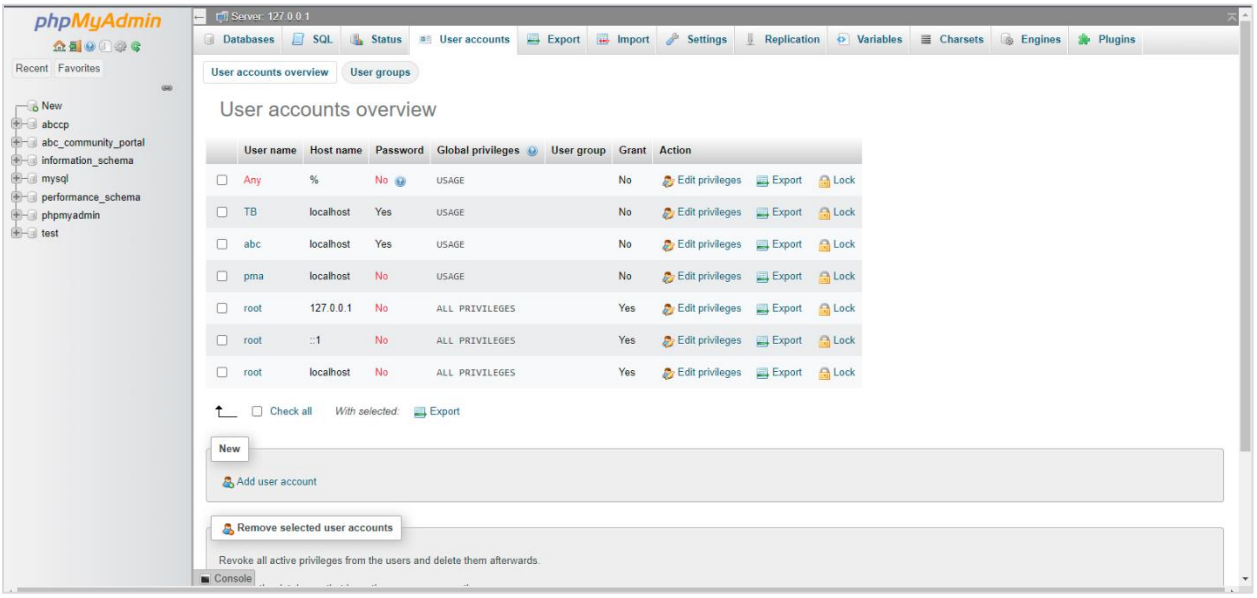


Figure 3:Screenshot of phpMyAdmin

- **ERDPlus** – To create ERD and Rational Schema

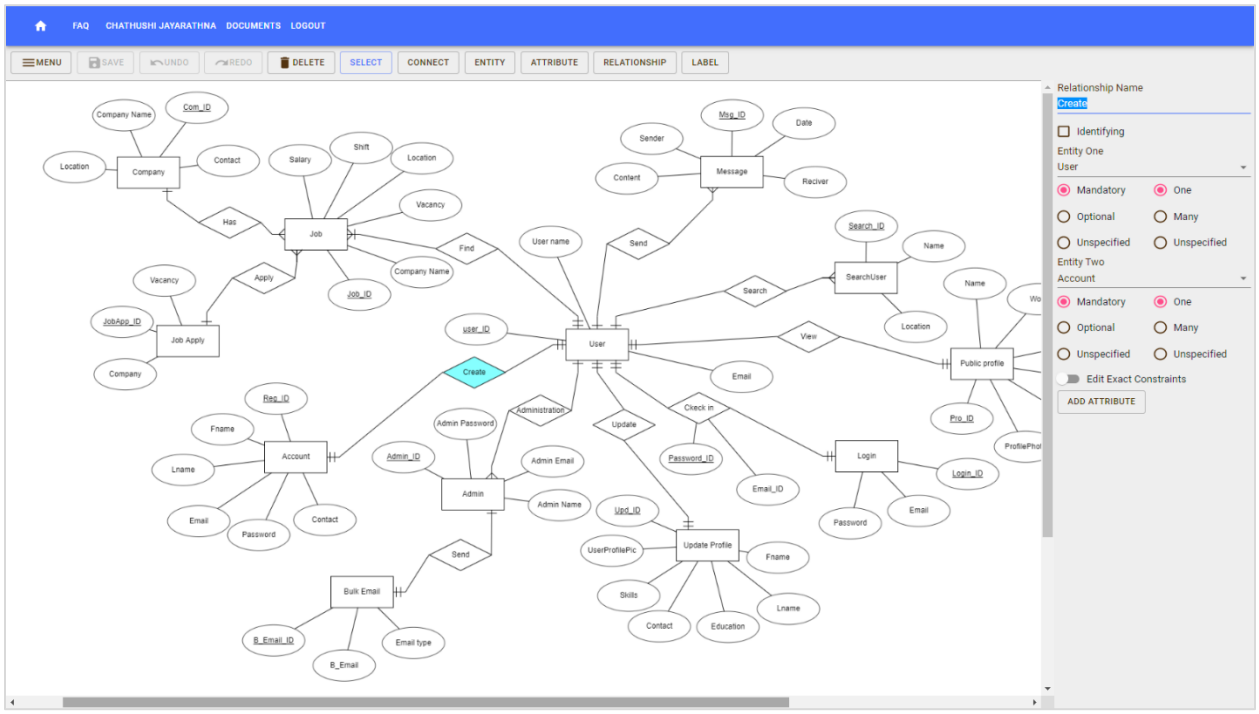


Figure 4:Screenshot of ERDPlus

- Microsoft Word – To create Documents (Ex :- Project Report)

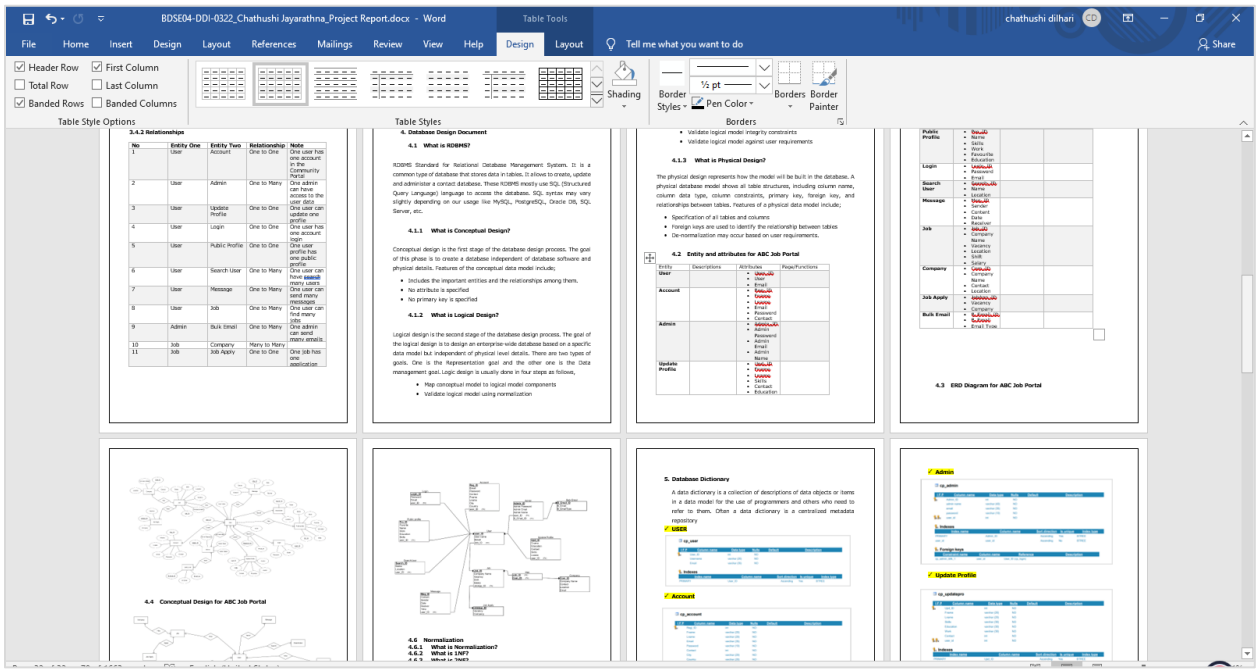


Figure 5:Screenshot of MS Word

- Microsoft Power Point – To create Project Presentation

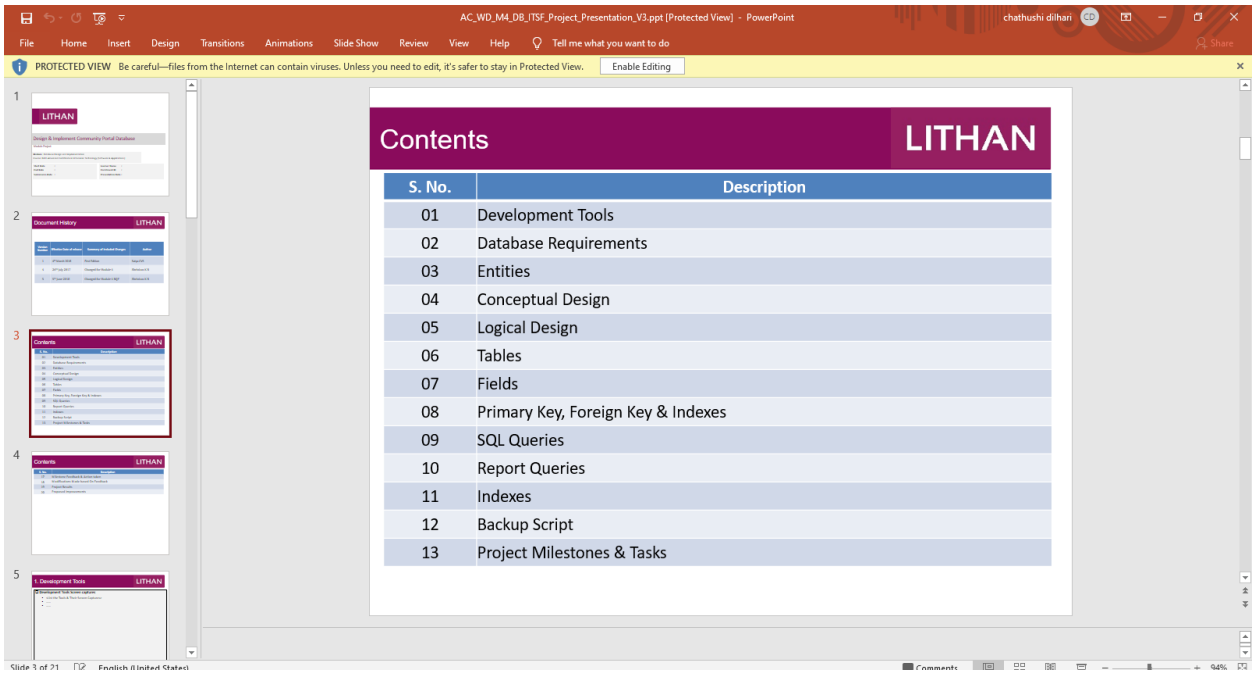


Figure 6:Screenshot of MS PowerPoint

- Command prompt

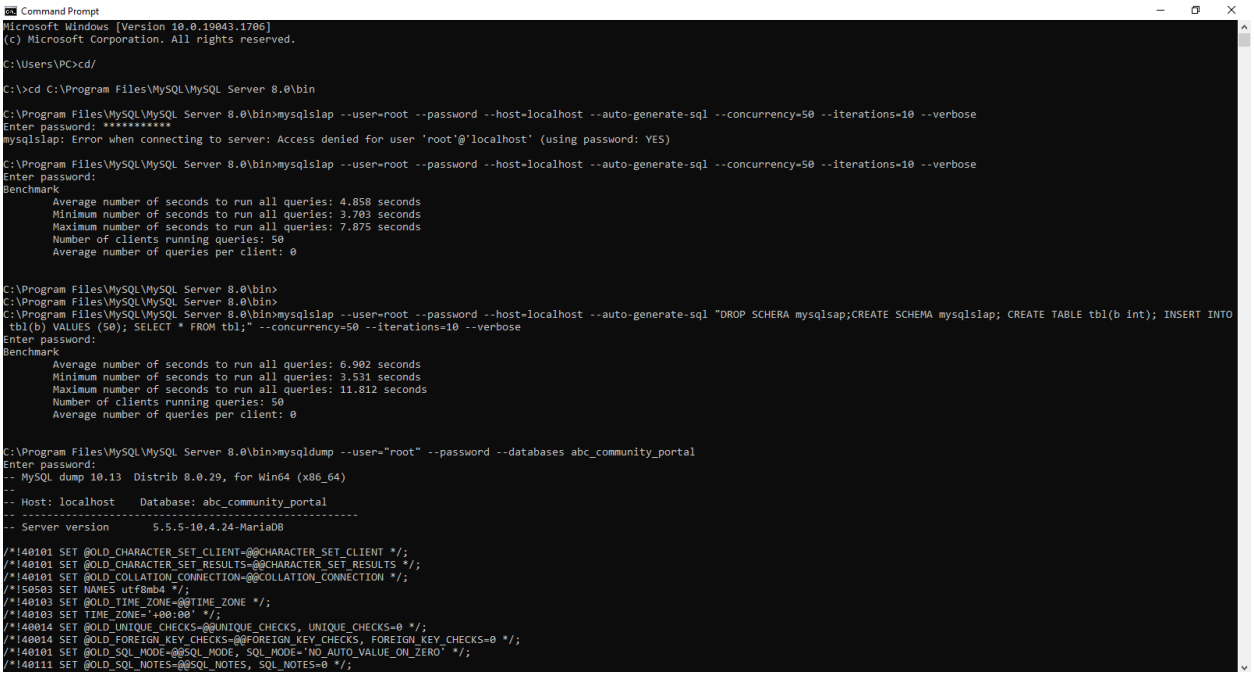


Figure 7:Screenshot of Command Prompt

- **Dbdesc –** To Create Database Documentation

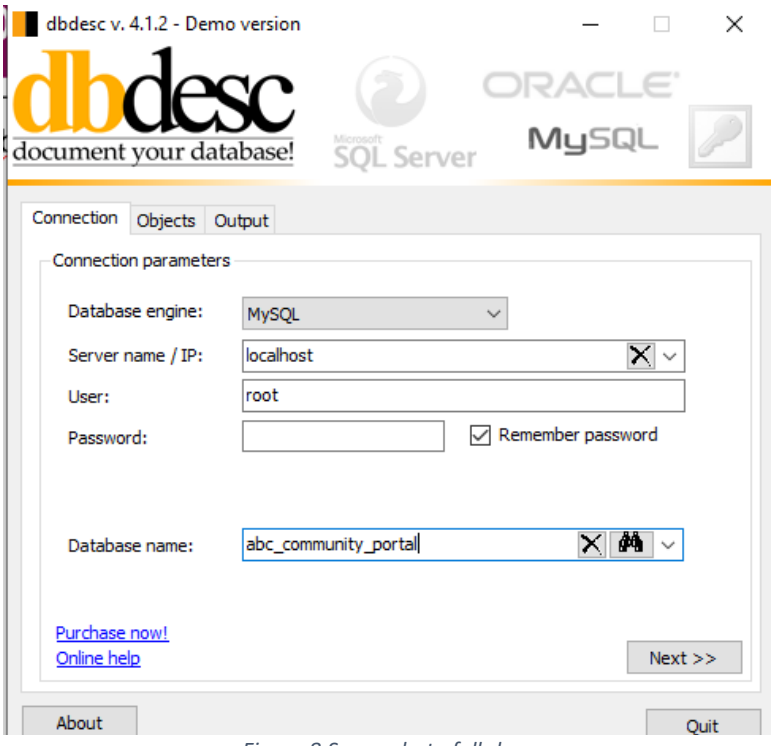


Figure 8:Screenshot of dbdesc

MS Excel – To create CSV Files

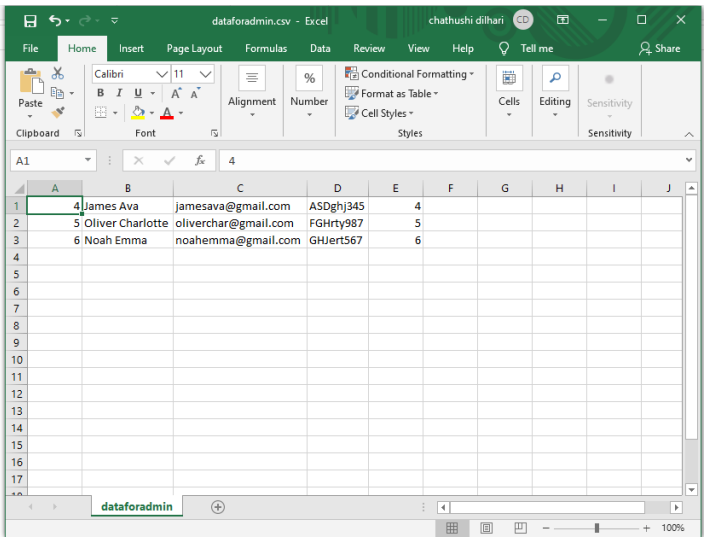


Figure 9:Screenshot of csv file

3. Database Specification Requirements

3.1 System Requirements

- Community Portal Home Page – Page with Login and Sign in button
- Registration page – Form to register to the community portal
- Registration thank you Page – Thank you page for registration
- Login Page – Login form to enter the community portal
- Forgot Password – user can request to reset their password
- Forgot password confirmation page – To confirm Email
- Public profile – Once users register to this community portal, they can see their profile
- Update Profile – Using this users can update their profile like profile picture, skills, education etc.
- Search user page – users can search other users using this page.
- MySQL Workbench 8.0 CE

3.2 Hardware Requirements

- Processer = i5
- RAM = 8GB
- Free Space on hard disk Minimum 4GB

3.3 Software Requirements

- MySQL Workbench 8.0.30
- MySQL Server 8.0.30
- dbdesc

3.4 Database Requirements

Process	Database Table
Registration	User, Account
Login	User, Login
Update Profile	User, Update Profile
Search & find	User, Search User
Send Messages	User, Message
Read Messages	User, Message
Post in Message Board	User, Post Message
Post Job Opportunities	User, Job
List Job Opportunities & Responses	User, Job, Apply Job
Administer user data	User, Profile, Admin
Send Bulk Email	Admin, Bulk Email

3.4.1 Relationships

No	Entity One	Entity Two	Relationship	Note
1	User	Account	One to One	One user has one account in the Community Portal
2	User	Admin	One to Many	One admin can have access to the user data
3	User	Update Profile	One to One	One user can update one profile
4	User	Login	One to One	One user has one account login
5	User	Public Profile	One to One	One user profile has one public profile
6	User	Search User	One to Many	One user can have search many users
7	User	Message	One to Many	One user can send many messages
8	User	Job	One to Many	One user can find many jobs
9	Admin	Bulk Email	One to Many	One admin can send many emails
10	Job	Company	Many to Many	Many jobs in many companies
11	Job	Job Apply	One to One	One job has one application

4. Database Design Document

4.1 What is RDBMS?

RDBMS Standard for Relational Database Management System. It is a common type of database that stores data in tables. It allows to create, update and administer a contact database. These RDBMS mostly use SQL (Structured Query Language) language to access the database. SQL syntax may vary slightly depending on our usage like MySQL, PostgreSQL, Oracle DB, SQL Server, etc.

4.1.1 What is Conceptual Design?

Conceptual design is the first stage of the database design process. The goal of this phase is to create a database independent of database software and physical details. Features of the conceptual data model include;

- Includes the important entities and the relationships among them.
- No attribute is specified
- No primary key is specified

4.1.2 What is Logical Design?

Logical design is the second stage of the database design process. The goal of the logical design is to design an enterprise-wide database based on a specific data model but independent of physical level details. There are two types of goals. One is the Representation goal and the other one is the Data management goal. Logic design is usually done in four steps as follows,

- Map conceptual model to logical model components
- Validate logical model using normalization

- Validate logical model integrity constraints
- Validate logical model against user requirements

4.1.3 What is Physical Design?

The physical design represents how the model will be built in the database. A physical database model shows all table structures, including column name, column data type, column constraints, primary key, foreign key, and relationships between tables. Features of a physical data model include;

- Specification of all tables and columns
- Foreign keys are used to identify the relationship between tables
- De-normalization may occur based on user requirements.

4.2 Entity and attributes for ABC Job Portal

Entity	Descriptions	Attributes	Page/Functions
User	A registered user is uniquely identified in the portal with a User ID	• User_ID • User • Email	• Registration • Login • Update Profile
Admin	Registered users can be an administrator to administer user data and send bulk email	• Admin_ID • Admin Password • Admin Email • Admin Name	• Administer user data
Update Profile	The users can edit their profile details using Public profile page.	• Upd_ID • Fname • Lname • Skills • Contact • Education	• Public Profile • Update Profile
Public Profile	1 user ID can have 1 profile that contains general information, contact	• Pro_ID • Name • Skills • Work • Favorite	• Public Profile • Update Profile

	information and is uniquely identified by the profile ID	• Education	
Login	All the users can log into the portal	• Login_ID • Password • Email	• Login • Forget password • Public Profile
Search user	Registered user can search other users, jobs	• Search_ID • Name • Location	• Search user
Message	Registered user can send and read messages to and from other users	• Msg_ID • Sender • Reciver • Date • Time	• Message • Read Message • Send Message
Job	Registered user can post job opportunities, apply or list job opportunities and responses	• Job_ID • Company Name • Vacancy • Location • Shift • Salary	• Job • Company • Job Apply
Company	Registered user can find companies	• Com_ID • Company Name • Contact • Location	• Job
Job Apply	Registered user can apply for job vacancies	• JobApp_ID • Vacancy • Company	• Job
Bulk Email	Bulk invite email can only be done by the administrator	• B_Email_ID • B_Email • Email Type	• Admin • Public Profile

4.3 ERD Diagram for ABC Job Portal

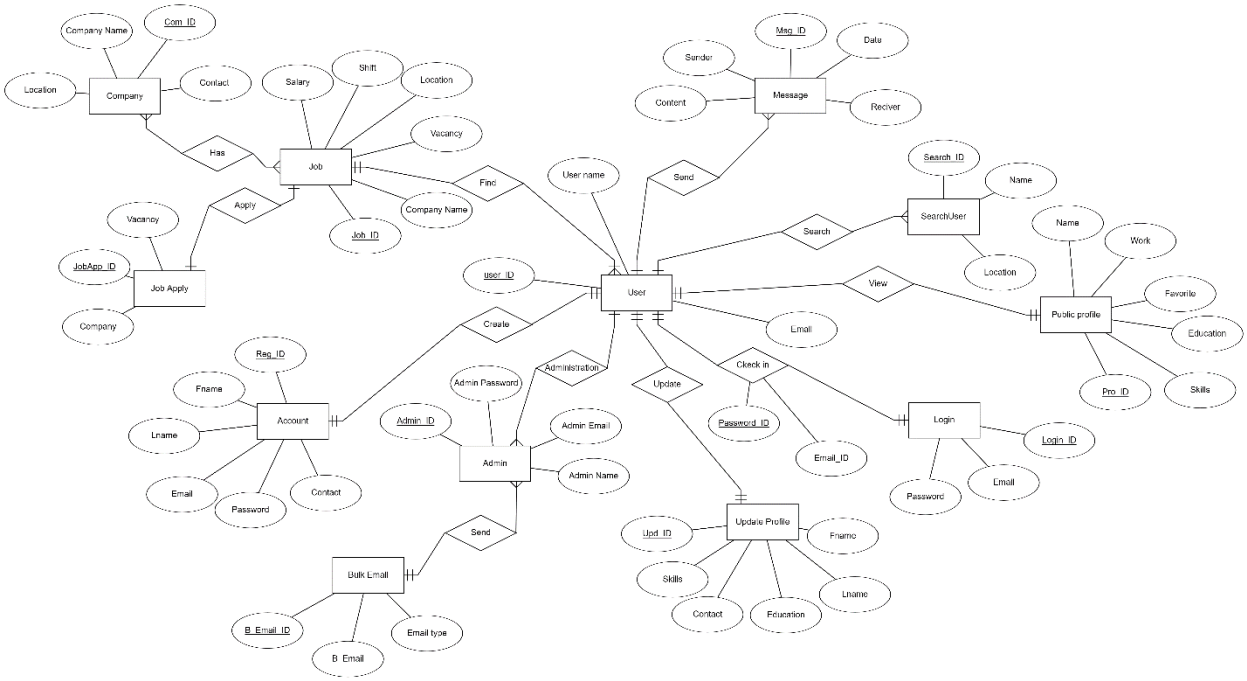


Figure 10:Screenshot of ERD

4.4 Conceptual Design for ABC Job Portal

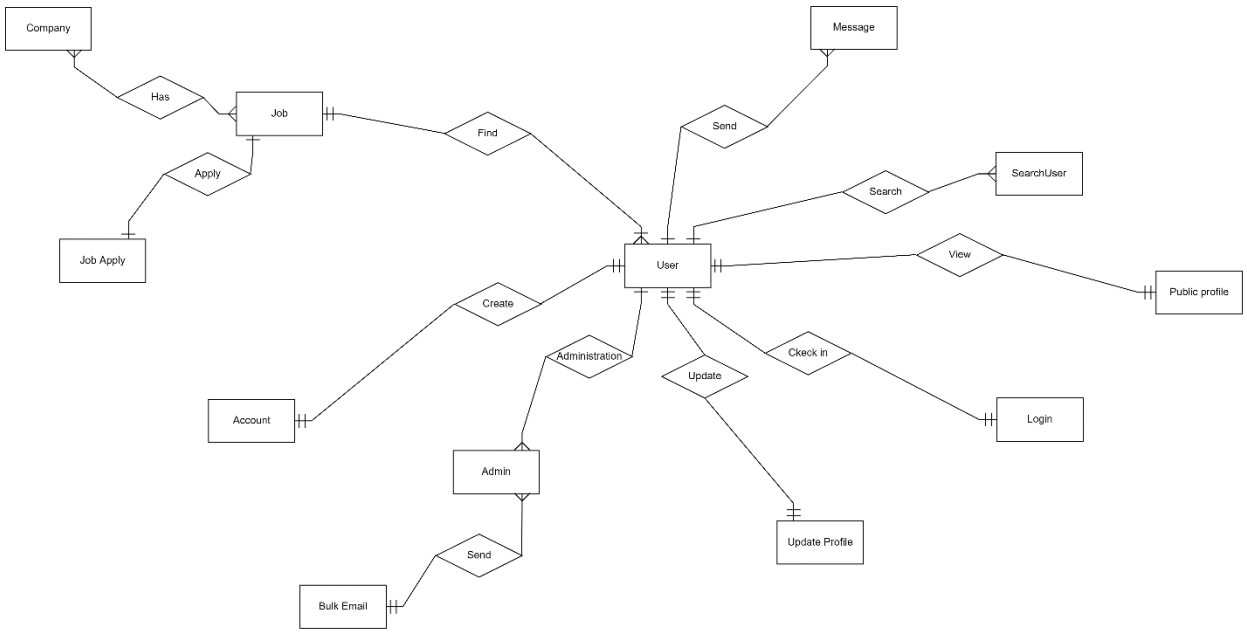


Figure 11:Screenshot of Conceptual Design

4.5 Logical Design for ABC Job Portal

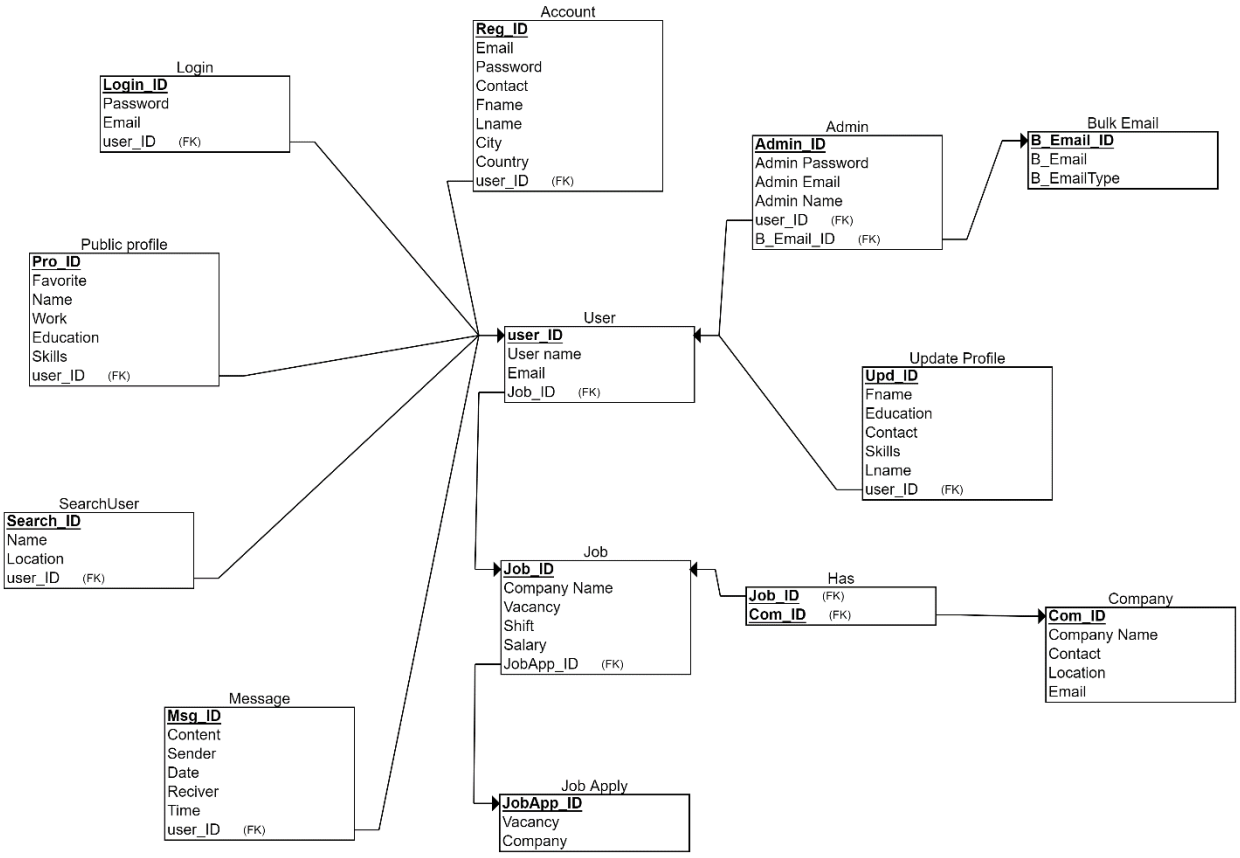


Figure 12:Screenshot of Logical Design

4.6 Normalization

4.6.1 What is Normalization?

Normalization is a database design that reduces facts redundancy and get rid of undesirable traits like Insertion, replace and Deletion Anomalies. Normalization policies divides larger table inti smaller tables and link them using relationships. The cause of Normalization in SQL is to do away with redundant (repetitive) records and ensure information is saved logically.

.

4.6.2 What is 1NF?

It's a level of normalization used in database management systems. A relation is said to be in 1NF if it consists of an atomic data type. In simpler words, a one-normal form states that a table's attributes can only contain a single value.

4.6.3 What is 2NF?

A relation is in second normal form in a DBMS (or 2NF) if it is in first normal form but has no non-primary attributes functionally dependent on the correct subset of candidate keys in the relation. A non-primary attribute of a relationship is an attribute that is not part of the candidate key of the relationship

4.6.4 What is 3NF?

Third Normal Form (3NF) is a database schema design methods for relational databases that uses normalization principles to reduce data duplication, avoid data anomalies, ensure referential integrity, and simplify data management

4.7 Normalization of Job table

✓ **UNF**

Vacancy_ID	Company	Vacancy Type	Location	Salary	Shift
1	Tec Sonic,IFS	Software Engineer	France	96000	Morning
2	Kingslake, George Bernard	Business Development Manager	Riyad	87000	Closing
3	IFS	Customer Care Executive	Venice	70000	Middle
4	George Bernard	Electrical Engineer	Remo	97000	Closing

Vacancy_ID	Company	Vacancy Type	Location	Salary	Shift
1	Tec Sonic	Software Engineer	France	96000	Morning
1	IFS	Software Engineer	France	96000	Morning
2	Kingslake	Business Development Manager	Riyad	87000	Closing
2	George Bernard	Business Development Manager	Riyad	87000	Closing
3	IFS	Customer Care Executive	Venice	70000	Middle
4	George Bernard	Electrical Engineer	Remo	97000	Closing

✓ **1NF**

Vacancy_ID	Com_ID	Vacancy Type	Location	Salary
1	1	Software Engineer	France	96000
1	2	Software Engineer	France	96000
2	3	Business Development Manager	Riyad	87000
2	4	Business Development Manager	Riyad	87000
3	5	Customer Care Executive	Venice	70000
4	6	Electrical Engineer	Remo	96000

Com_ID	Company
1	Tec Sonic
2	IFS
3	Kingslake
4	George Bernard

✓ **2NF**

Vacancy_ID	Com_ID	Shift
1	1	Morning
1	2	Morning
2	3	Closing
2	4	Closing
3	5	Middle
4	6	Closing

Com_ID	Company	Location	Salary ID
1	Tec Sonic	France	11
2	IFS	France	22
3	Kingslake	Riyad	33
4	George Bernard	Riyad	44
5	IFS	Venice	55
6	George Bernard	Remo	66

Vacancy id	Vacancy Type	Salary	Salary ID
1	Software Engineer	96000	1
1	Software Engineer	96000	2
2	Business Development Manager	87000	3
2	Business Development Manager	87000	4
3	Customer Care Executive	70000	5
4	Electrical Engineer	96000	6

✓ **3NF**

Com_ID	Company	Location
1	Tec Sonic	France
2	IFS	France
3	Kingslake	Riyad
4	George Bernard	Riyad
5	IFS	Venice
6	George Bernard	Remo

Vacancy id	Vacancy Type	Salary ID	Salary
1	Software Engineer	1	96000
1	Software Engineer	2	96000
2	Business Development Manager	3	87000
2	Business Development Manager	4	87000
3	Customer Care Executive	5	70000
4	Electrical Engineer	6	96000

Salary ID	Salary
1	96000
2	96000
3	87000
4	87000
5	70000
6	96000

4.8 Normalization of Account Table

✓ **UNF**

Regi_ID	Account	Name	Account_Type	Password
1	Facebook	Sara	Fake Accoount	789012
2	Instagram	Novel	Community Blog	345678
3	Skype	Ashley	Business Account, Personal Account	123457
4	Twitter	Leo	Spam Account	876543
5	Yahoo	Kevin	Professional Account	987654

Regi_ID	Account	Name	Account_Type	Password
1	Facebook	Sara	Fake Accoount	789012
2	Instagram	Novel	Community Blog	345678
3	Skype	Ashley	Business Account	123457
3	Skype	Ashley	Personal Account	123457
4	Twitter	Leo	Spam Account	876543
5	Yahoo	Kevin	Professional Account	987654

✓ **1NF**

Account_Type_ID	Account_Type
11	Fake Accoount
22	Community Blog
33	Business Account
33	Personal Account
44	Spam Account
55	Professional Account

Regi_ID	Account_Type_ID	Name	Password	Account
1	11	Sara	789012	Facebook
2	22	Novel	345678	Instagram
3	33	Ashley	123457	Skype
4	44	Leo	876543	Twitter
5	55	Kevin	987654	Yahoo

✓ **2NF**

Account_Type_ID	Account_Type
11	Fake Accoount
22	Community Blog
33	Business Account
33	Personal Account
44	Spam Account
55	Professional Account

Regi_ID	Account_Type_ID	Name	Password
1	11	Sara	789012
2	22	Novel	345678
3	33	Ashley	123457
4	44	Leo	876543
5	55	Kevin	987654

Regi_ID	Account
1	Facebook
2	Instagram
3	Skype
4	Twitter
5	Yahoo

✓ **3NF**

Account_Type_ID	Account_Type
11	Fake Accoount
22	Community Blog
33	Business Account
33	Personal Account
44	Spam Account
55	Professional Account

Regi_ID	Account_Type_ID	Name	Password
1	11	Sara	789012
2	22	Novel	345678
3	33	Ashley	123457
4	44	Leo	876543
5	55	Kevin	987654

Password_ID	Password
1111	789012
2222	345678
3333	123457
4444	876543
	987654

4.9 Normalization of Public Profile Table

✓ **UNF**

Pro_ID	UserName	Job_Name	Education	Company
1	Sara	Software Engineer	Stanford Univeristy	Kingslake
2	Leo	Technician	Harvard University	TecSonic, CEB,Telecome
3	Kevin	Business Development Manager	University of Cambridge	S&S Sons
4	Ashley	Electrical Engineer	UCL	CEB
5	Nethu	Customer Care Executive	Cornell	Telecome

Pro_ID	UserName	Job_Name	Education	Company
1	Sara	Software Engineer	Stanford Univeristy	Kingslake
2	Leo	Technician	Flora University	TecSonic
2	Leo	Technician	Harvard University	CEB
2	Leo	Technician	NSBM	Telecome
3	Kevin	Business Development Manager	University of Cambridge	S&S Sons

4	Ashley	Electrical Engineer	UCL	CEB
5	Nethu	Customer Care Executive	Cornell	Telecome

✓ **1NF**

Education_ID	Education
111	Stanford Univeristy
222	Flora University
222	Harvard University
222	NSBM
333	University of Cambridge
444	UCL
555	Cornell

Pro_ID	Education_ID	Username	Job_Name	Comapany
1	111	Sara	Software Engineer	Kingslake
2	222	Leo	Technician	TecSonic
3	333	Kevin	Business Development Manager	CEB
4	444	Ashley	Electrical Engineer	Telecome
5	555	Nethu	Customer Care Executive	S&S Sons

✓ **2NF**

Education_ID	Education
111	Stanford Univeristy
222	Harvard University
222	University of Cambridge
222	UCL
333	Cornell
444	UCL
555	Cornell

Pro_ID	Education_ID	Username
1	111	Sara
2	222	Leo
3	333	Kevin

4	444	Ashley
5	555	Nethu

Pro_ID	Job_Name	Comapany
1	Software Engineer	Kingslake
2	Technician	TecSonic
3	Business Development Manager	CEB
4	Electrical Engineer	Telecome
5	Customer Care Executive	S&S Sons

✓ 3NF

Education_ID	Education
111	Stanford Univeristy
222	Harvard University
222	University of Cambridge
222	UCL
333	Cornell
444	UCL
555	Cornell

Pro_ID	Education_ID	Username
1	111	Sara
2	222	Leo
3	333	Kevin
4	444	Ashley
5	555	Nethu

Pro_ID	Job_Name	Company_ID
1	Software Engineer	123
2	Technician	234
3	Business Development Manager	345
4	Electrical Engineer	456
5	Customer Care Executive	567

Company_ID	Comapany
123	Kingslake
234	TecSonic
345	CEB
456	Telecome
567	S&S Sons

5. Database Dictionary

A data dictionary is a collection of descriptions of data objects or items in a data model for the use of programmers and others who need to refer to them. Often a data dictionary is a centralized metadata repository

✓ USER

cp_user

I F P	Column name	Data type	Nulls	Default	Description
PK	User_ID	int	NO		
	Username	varchar (20)	NO		
	Email	varchar (35)	NO		

Indexes

Index name	Column name	Sort direction	Is unique	Index type
PRIMARY	User_ID	Ascending	Yes	BTREE

✓ Account

cp_account

I F P	Column name	Data type	Nulls	Default	Description
PK	Regi_ID	int	NO		
	Fname	varchar (20)	NO		
	Lname	varchar (20)	NO		
	Email	varchar (35)	NO		
	Password	varchar (10)	NO		
	Contact	int	NO		
	City	varchar (20)	NO		
	Country	varchar (20)	NO		
PK FK	user_id	int	NO		

Indexes

Index name	Column name	Sort direction	Is unique	Index type
PRIMARY	Regi_ID	Ascending	Yes	BTREE
user_id	user_id	Ascending	No	BTREE

Foreign keys

Constraint name	Column name	Reference	Description
cp_account_ibfk_1	user_id	User_ID (cp_login)	

✓ Admin

cp_admin

I F P	Column name	Data type	Nulls	Default	Description
I	Admin_ID	int	NO		
	admin name	varchar (40)	NO		
	email	varchar (35)	NO		
	password	varchar (10)	NO		
I F	user_id	int	NO		

Indexes

Index name	Column name	Sort direction	Is unique	Index type
PRIMARY	Admin_ID	Ascending	Yes	BTREE
user_id	user_id	Ascending	No	BTREE

Foreign keys

Constraint name	Column name	Reference	Description
cp_admin_ibfk_1	user_id	User_ID (cp_login)	

✓ Update Profile

cp_updatepro

I F P	Column name	Data type	Nulls	Default	Description
I	Upd_ID	int	NO		
	Fname	varchar (20)	NO		
	Lname	varchar (20)	NO		
	Skills	varchar (30)	NO		
	Education	varchar (30)	NO		
	Work	varchar (30)	NO		
	Contact	int	NO		
I F	user_id	int	NO		

Indexes

Index name	Column name	Sort direction	Is unique	Index type
PRIMARY	Upd_ID	Ascending	Yes	BTREE
user_id	user_id	Ascending	No	BTREE

Foreign keys

Constraint name	Column name	Reference	Description
cp_updatepro_ibfk_1	user_id	User_ID (cp_login)	

✓ Public Profile

cp_pro

I F P	Column name	Data type	Nulls	Default	Description
I	Pro_ID	int	NO		
	Name	varchar (35)	NO		
	Work	varchar (30)	NO		
	Favorite	varchar (30)	NO		
	Education	varchar (30)	NO		
	Skills	varchar (30)	NO		
I F	user_id	int	NO		

Indexes

Index name	Column name	Sort direction	Is unique	Index type
PRIMARY	Pro_ID	Ascending	Yes	BTREE
user_id	user_id	Ascending	No	BTREE

Foreign keys

Constraint name	Column name	Reference	Description
cp_pro_ibfk_1	user_id	User_ID (cp_login)	

✓ Login

cp_login

I F P	Column name	Data type	Nulls	Default	Description
I	Login_ID	int	NO		
	Email	varchar (40)	NO		
	Password	varchar (8)	NO		
I F	User_ID	int	NO		

Indexes

Index name	Column name	Sort direction	Is unique	Index type
PRIMARY	Login_ID	Ascending	Yes	BTREE
User_ID	User_ID	Ascending	No	BTREE

Foreign keys

Constraint name	Column name	Reference	Description
cp_login_ibfk_1	User_ID	User_ID (cp_login)	

✓ Search User

cp_searchuser

I F P	Column name	Data type	Nulls	Default	Description
I	search_id	int	NO		
	Name	varchar (35)	NO		
	Location	varchar (20)	NO		
I F	User_ID	int	NO		

Indexes

Index name	Column name	Sort direction	Is unique	Index type
PRIMARY	search_id	Ascending	Yes	BTREE
User_ID	User_ID	Ascending	No	BTREE

Foreign keys

Constraint name	Column name	Reference	Description
cp_searchuser_ibfk_1	User_ID	User_ID (cp_login)	

✓ Message

cp_message

I F P	Column name	Data type	Nulls	Default	Description
I	Msg_ID	int	NO		
	Content	varchar (60)	NO		
	Sender	varchar (20)	NO		
	Reciver	varchar (20)	NO		
	Date	date	NO		
	Time	int	NO		
I F	user_id	int	NO		

Indexes

Index name	Column name	Sort direction	Is unique	Index type
PRIMARY	Msg_ID	Ascending	Yes	BTREE
user_id	user_id	Ascending	No	BTREE

Foreign keys

Constraint name	Column name	Reference	Description
cp_message_ibfk_1	user_id	User_ID (cp_login)	

✓ Job

cp_job

I F P	Column name	Data type	Nulls	Default	Description
I	Vacancy_id	int	NO		
	Company Name	varchar (20)	NO		
	Vacancy	varchar (20)	NO		
	Salary	int	NO		
	Shift	varchar (20)	NO		
I F	user_id	int	NO		

Indexes

Index name	Column name	Sort direction	Is unique	Index type
PRIMARY	Vacancy_id	Ascending	Yes	BTREE
user_id	user_id	Ascending	No	BTREE

Foreign keys

Constraint name	Column name	Reference	Description
cp_job_ibfk_1	user_id	User_ID (cp_login)	

✓ Company

cp_company

I F P	Column name	Data type	Nulls	Default	Description
I	Com_ID	int	NO		
	Company Name	varchar (40)	NO		
	Email	varchar (35)	NO		
	Contact	int	NO		
	Location	varchar (20)	NO		
I F	Vacancy_id	int	NO		

Indexes

Index name	Column name	Sort direction	Is unique	Index type
PRIMARY	Com_ID	Ascending	Yes	BTREE
Vacancy_id	Vacancy_id	Ascending	No	BTREE

Foreign keys

Constraint name	Column name	Reference	Description
cp_company_ibfk_1	Vacancy_id	Vacancy_id (cp_company)	

✓ Job Apply

cp_jobapply

I F P	Column name	Data type	Nulls	Default	Description
I	JopApp_ID	int	NO		
	Vacancy	varchar (20)	NO		
	Company Name	varchar (20)	NO		
I F	Vacancy_id	int	NO		

Indexes

Index name	Column name	Sort direction	Is unique	Index type
PRIMARY	JopApp_ID	Ascending	Yes	BTREE
Vacancy_id	Vacancy_id	Ascending	No	BTREE

Foreign keys

Constraint name	Column name	Reference	Description
cp_jobapply_ibfk_1	Vacancy_id	Vacancy_id (cp_company)	

✓ Bulk Email

cp_bulkemail

I F P	Column name	Data type	Nulls	Default	Description
I	B_Email_id	int	NO		
	B_Email	varchar (35)	NO		
	B_EmailType	varchar (10)	NO		
I F	Admin_ID	int	NO		

Indexes

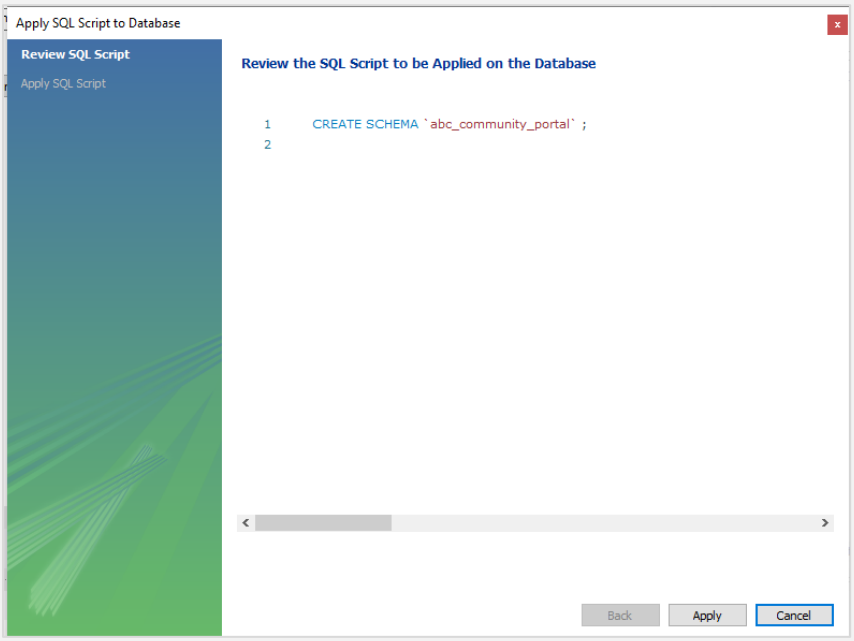
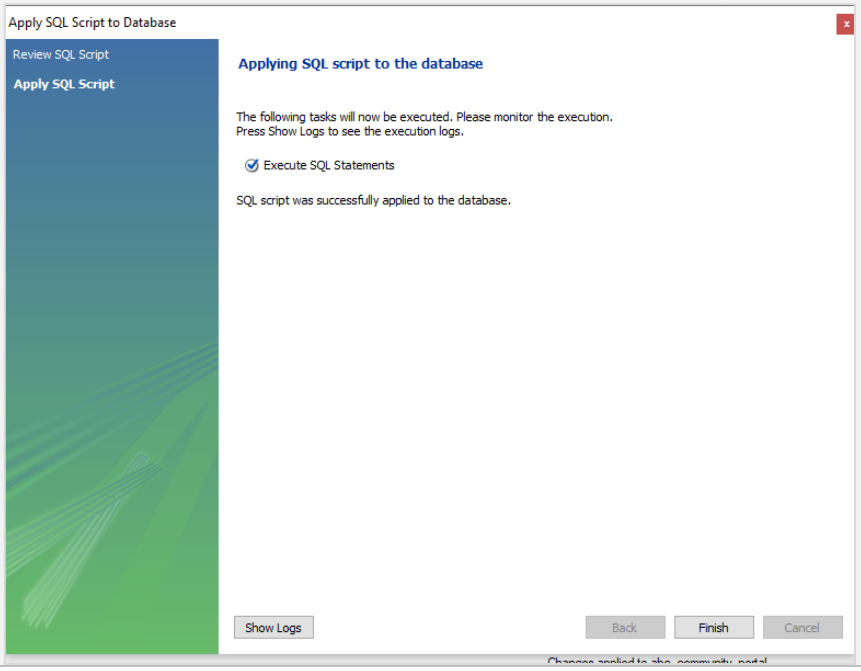
Index name	Column name	Sort direction	Is unique	Index type
PRIMARY	B_Email_id	Ascending	Yes	BTREE
Admin_ID	Admin_ID	Ascending	No	BTREE

Foreign keys

Constraint name	Column name	Reference	Description
cp_bulkemail_ibfk_1	Admin_ID	Admin_ID (cp_bulkemail)	

6. Database Design Implementation

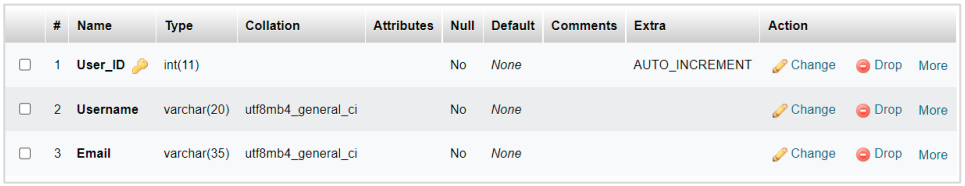
6.1 MySQL Database Scripts & Screen CAPTURE

Table / Database	Screen Capture
Community Portal Database	
	

User

```
CREATE  TABLE IF NOT EXISTS
`abc_community_portal`.`cp_user` (
  `User_ID` INT NOT NULL AUTO_INCREMENT ,
  `Username` varchar(20) NOT NULL,
  `Email` varchar(35) NOT NULL,

  PRIMARY KEY (`UserID`) )
ENGINE = InnoDB;
```



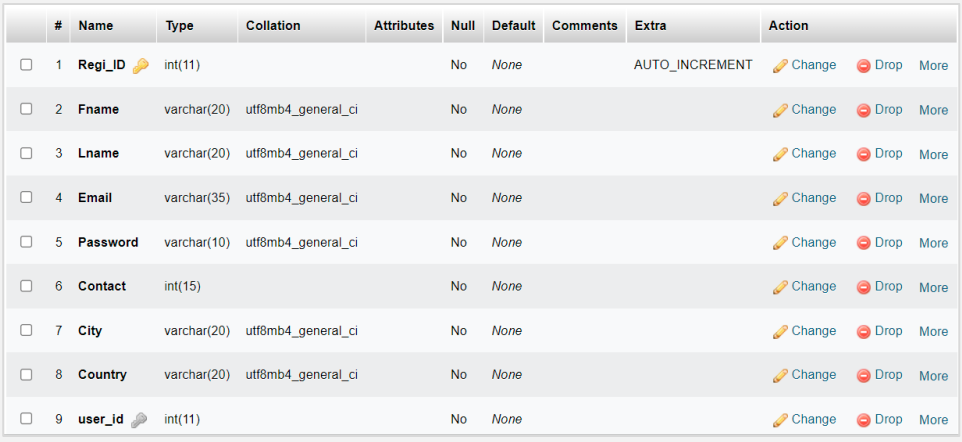
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	User_ID	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	Username	varchar(20)	utf8mb4_general_ci		No	None			Change Drop More
3	Email	varchar(35)	utf8mb4_general_ci		No	None			Change Drop More

Account

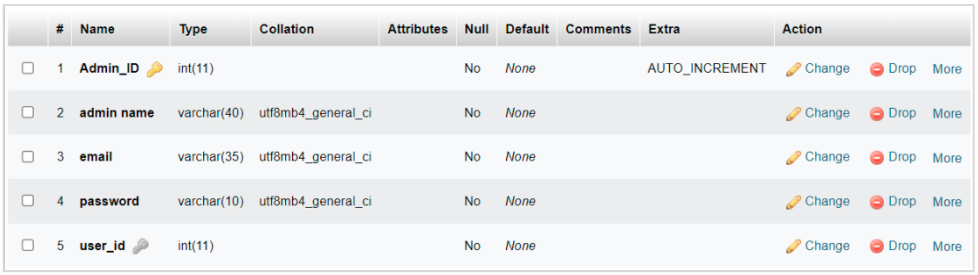
```
CREATE  TABLE IF NOT EXISTS
`abc_community_portal`.`cp_account` (
  `Regi_ID` INT NOT NULL AUTO_INCREMENT ,
  `Fname` varchar(20) NOT NULL,
  `Lname` varchar(20) NOT NULL,
  `Email` varchar(35) NOT NULL,
  `Password` varchar(10) NOT NULL,
  `Contact` int(15) NOT NULL,
  `City` varchar(20) NOT NULL,
  `Country` varchar(20) NOT NULL,
  `user_id` int(11) NOT NULL,

  PRIMARY KEY (`Regi_ID`) ,
  FOREIGN KEY (User_ID) REFERENCES cp_user(User_ID))

ENGINE = InnoDB;
```



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	Regi_ID	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	Fname	varchar(20)	utf8mb4_general_ci		No	None			Change Drop More
3	Lname	varchar(20)	utf8mb4_general_ci		No	None			Change Drop More
4	Email	varchar(35)	utf8mb4_general_ci		No	None			Change Drop More
5	Password	varchar(10)	utf8mb4_general_ci		No	None			Change Drop More
6	Contact	int(15)			No	None			Change Drop More
7	City	varchar(20)	utf8mb4_general_ci		No	None			Change Drop More
8	Country	varchar(20)	utf8mb4_general_ci		No	None			Change Drop More
9	user_id	int(11)			No	None			Change Drop More

Admin	CREATE TABLE IF NOT EXISTS `abc_community_portal`.`cp_admin` (`admin_id` INT NOT NULL AUTO_INCREMENT , `admin name` varchar(40) NOT NULL, `email` varchar(35) NOT NULL, `password` varchar(10) NOT NULL, `user_id` int(11) NOT NULL, PRIMARY KEY (`admin_id`), FOREIGN KEY (User_ID) REFERENCES cp_user(User_ID)) ENGINE = InnoDB; 
Update Profile	CREATE TABLE IF NOT EXISTS `abc_community_portal`.`cp_updatepro` (`Upd_ID` INT NOT NULL AUTO_INCREMENT , `Fname` varchar(20) NOT NULL, `Lname` varchar(20) NOT NULL, `Skills` varchar(30) NOT NULL, `Education` varchar(30) NOT NULL, `Work` varchar(30) NOT NULL, `Contact` int(15) NOT NULL, `user_id` int(11) NOT NULL, PRIMARY KEY (`Upd_ID`) , FOREIGN KEY (User_ID) REFERENCES cp_user(User_ID)) ENGINE = InnoDB;

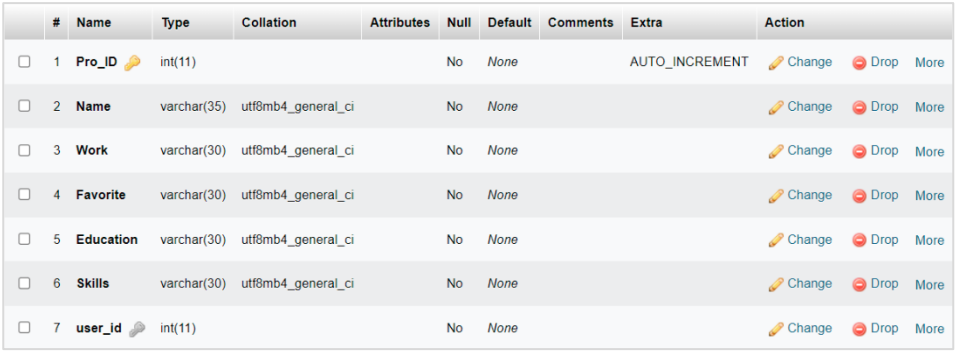
	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	Admin_ID	int(11)			No	None		AUTO_INCREMENT	Change Drop More
☐	2	admin name	varchar(40)	utf8mb4_general_ci		No	None			Change Drop More
☐	3	email	varchar(35)	utf8mb4_general_ci		No	None			Change Drop More
☐	4	password	varchar(10)	utf8mb4_general_ci		No	None			Change Drop More
☐	5	user_id	int(11)			No	None			Change Drop More

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	Upd_ID	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	Fname	varchar(20)	utf8mb4_general_ci		No	None			Change Drop More
3	Lname	varchar(20)	utf8mb4_general_ci		No	None			Change Drop More
4	Skills	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More
5	Education	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More
6	Work	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More
7	Contact	int(15)			No	None			Change Drop More
8	user_id	int(11)			No	None			Change Drop More

Public Profile

```
CREATE  TABLE IF NOT EXISTS
`abc_community_portal`.`cp_pro` (
  `Pro_ID` INT NOT NULL AUTO_INCREMENT ,
  `Name` varchar(35) NOT NULL,
  `Work` varchar(30) NOT NULL,
  `Favorite` varchar(30) NOT NULL,
  `Education` varchar(30) NOT NULL,
  `Skills` varchar(30) NOT NULL,
  `user_id` int(11) NOT NULL,

  PRIMARY KEY (`Pro_ID`) ,
  FOREIGN KEY (User_ID) REFERENCES cp_user(User_ID))
ENGINE = InnoDB;
```

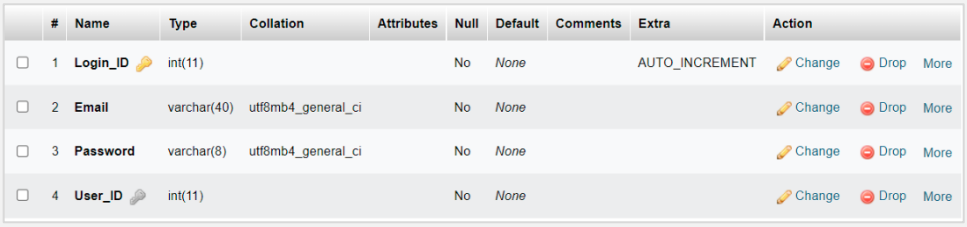
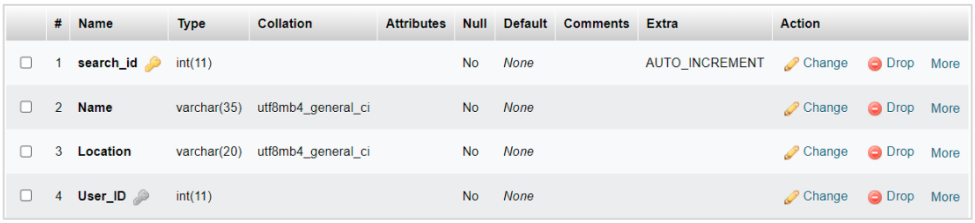


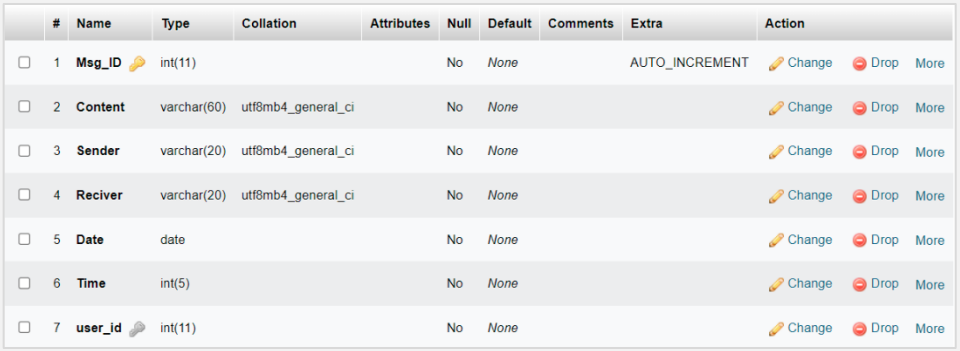
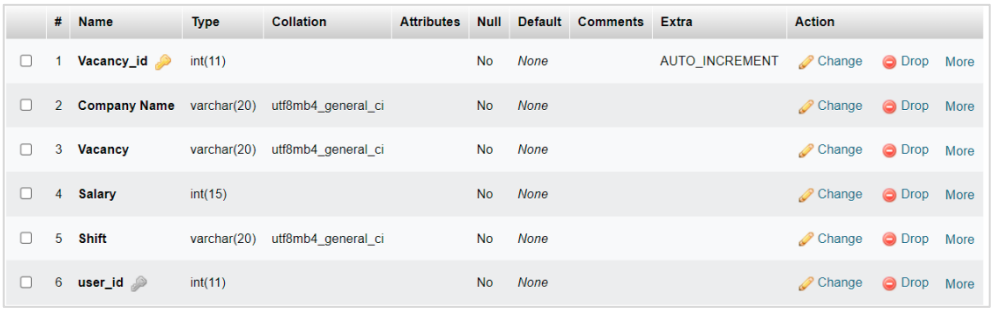
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	Pro_ID	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	Name	varchar(35)	utf8mb4_general_ci		No	None			Change Drop More
3	Work	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More
4	Favorite	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More
5	Education	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More
6	Skills	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More
7	user_id	int(11)			No	None			Change Drop More

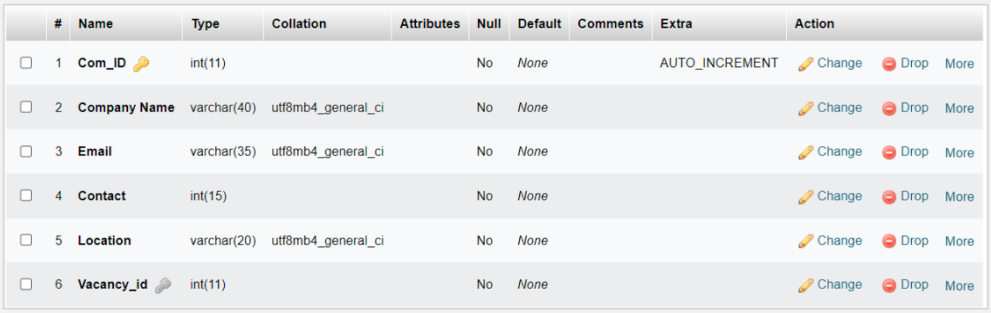
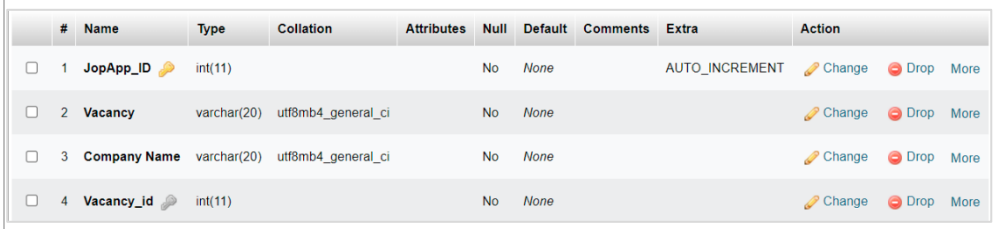
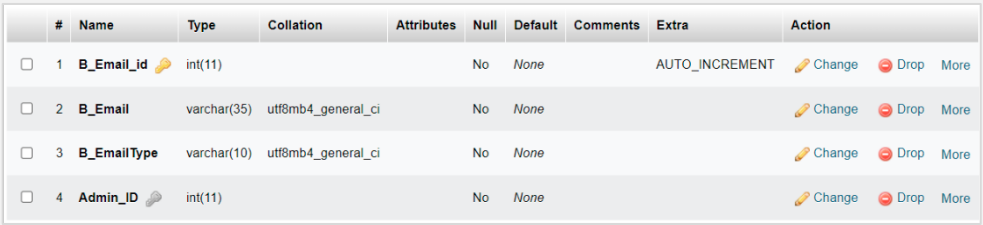
Login

```
CREATE  TABLE IF NOT EXISTS
`abc_community_portal`.`cp_login` (
  `Login_ID` INT NOT NULL AUTO_INCREMENT ,
  `Email` VARCHAR(40) NOT NULL ,
  `Password` VARCHAR(8) NOT NULL,
  `User_ID` INT NOT NULL ,

  PRIMARY KEY (`Login_ID`),
  FOREIGN KEY (User_ID) REFERENCES cp_user(User_ID))
```

	ENGINE = InnoDB;  | # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action | | 1 | Login_ID | int(11) | | | No | None | | AUTO_INCREMENT | Change Drop More | | 2 | Email | varchar(40) | utf8mb4_general_ci | | No | None | | | Change Drop More | | 3 | Password | varchar(8) | utf8mb4_general_ci | | No | None | | | Change Drop More | | 4 | User_ID | int(11) | | | No | None | | | Change Drop More |
Search User	CREATE TABLE IF NOT EXISTS `abc_community_portal`.`cp_searchuser` (`search_id` INT NOT NULL AUTO_INCREMENT , `Name` varchar(35) NOT NULL, `Location` varchar(20) NOT NULL, `User_ID` int(11) NOT NULL, PRIMARY KEY (`search_id`), FOREIGN KEY (user_id) REFERENCES cp_user(user_id)) ENGINE = InnoDB;  | # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action | | 1 | search_id | int(11) | | | No | None | | AUTO_INCREMENT | Change Drop More | | 2 | Name | varchar(35) | utf8mb4_general_ci | | No | None | | | Change Drop More | | 3 | Location | varchar(20) | utf8mb4_general_ci | | No | None | | | Change Drop More | | 4 | User_ID | int(11) | | | No | None | | | Change Drop More |
Message	CREATE TABLE IF NOT EXISTS `abc_community_portal`.`cp_message` (`Msg_ID` INT NOT NULL AUTO_INCREMENT , `Content` varchar(60) NOT NULL, `Sender` varchar(20) NOT NULL, `Reciver` varchar(20) NOT NULL, `Date` date NOT NULL, `Time` int(5) NOT NULL, `user_id` int(11) NOT NULL, PRIMARY KEY (`Msg_ID`), FOREIGN KEY (User_ID) REFERENCES cp_user(User_ID)) ENGINE = InnoDB;

	
Job	CREATE TABLE IF NOT EXISTS `abc_community_portal`.`cp_Job` (`Vacancy_id` INT NOT NULL AUTO_INCREMENT , `Company Name` varchar(20) NOT NULL, `Vacancy` varchar(20) NOT NULL, `Salary` int(15) NOT NULL, `Shift` varchar(20) NOT NULL, `user_id` int(11) NOT NULL, PRIMARY KEY (`Vacancy_id`), FOREIGN KEY (user_id) REFERENCES cp_user(user_id)) ENGINE = InnoDB; 
Company	CREATE TABLE IF NOT EXISTS `abc_community_portal`.`cp_company` (`Com_ID` INT NOT NULL AUTO_INCREMENT , `Company Name` varchar(40) NOT NULL, `Email` varchar(35) NOT NULL, `Contact` int(15) NOT NULL, `Location` varchar(20) NOT NULL, `Vacancy_id` int(11) NOT NULL, PRIMARY KEY (`Com_ID`), FOREIGN KEY (Vacancy_id) REFERENCES cp_job(Vacancy_id)) ENGINE = InnoDB;

	 | # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1 | Com_ID | int(11) | | | No | None | | AUTO_INCREMENT | Change Drop More | | 2 | Company Name | varchar(40) | utf8mb4_general_ci | | No | None | | | Change Drop More | | 3 | Email | varchar(35) | utf8mb4_general_ci | | No | None | | | Change Drop More | | 4 | Contact | int(15) | | | No | None | | | Change Drop More | | 5 | Location | varchar(20) | utf8mb4_general_ci | | No | None | | | Change Drop More | | 6 | Vacancy_id | int(11) | | | No | None | | | Change Drop More |
Job Apply	CREATE TABLE IF NOT EXISTS `abc_community_portal`.`cp_jobapply` (`JopApp_ID` INT NOT NULL AUTO_INCREMENT , `Vacancy` varchar(20) NOT NULL, `Company Name` varchar(20) NOT NULL, `Vacancy_id` int(11) NOT NULL, PRIMARY KEY (`JopApp_ID`), FOREIGN KEY (Vacancy_id) REFERENCES cp_job(Vacancy_id)) ENGINE = InnoDB;  | # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1 | JopApp_ID | int(11) | | | No | None | | AUTO_INCREMENT | Change Drop More | | 2 | Vacancy | varchar(20) | utf8mb4_general_ci | | No | None | | | Change Drop More | | 3 | Company Name | varchar(20) | utf8mb4_general_ci | | No | None | | | Change Drop More | | 4 | Vacancy_id | int(11) | | | No | None | | | Change Drop More |
Bulk Email	CREATE TABLE IF NOT EXISTS `abc_community_portal`.`cp_bulkemail` (`B_Email_id` INT NOT NULL AUTO_INCREMENT , `B_Email` varchar(35) NOT NULL, `B_EmailType` varchar(10) NOT NULL, `Admin_ID` int(11) NOT NULL, PRIMARY KEY (`B_Email_id`), FOREIGN KEY (Admin_ID) REFERENCES cp_admin(Admin_ID)) ENGINE = InnoDB;  | # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1 | B_Email_id | int(11) | | | No | None | | AUTO_INCREMENT | Change Drop More | | 2 | B_Email | varchar(35) | utf8mb4_general_ci | | No | None | | | Change Drop More | | 3 | B_EmailType | varchar(10) | utf8mb4_general_ci | | No | None | | | Change Drop More | | 4 | Admin_ID | int(11) | | | No | None | | | Change Drop More |

6.2 EERD

EERD standard for Enhanced Entity-Relationship Diagrams. They are essential part of the modeling interface in MySQL Workbenches diagrams provide a visual representation of the relationships between tables in your model

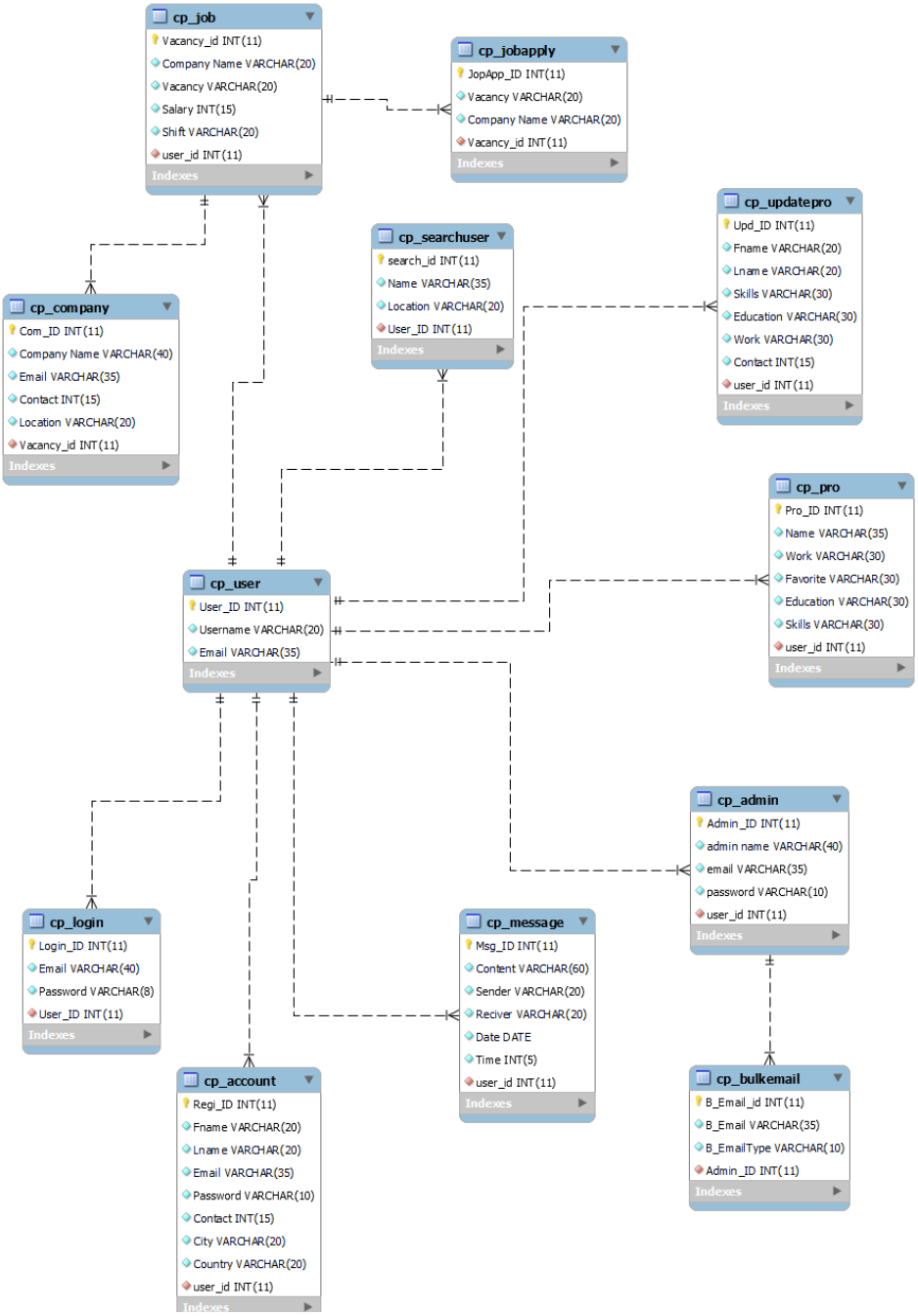


Figure 13:Screenshot of EERD

7. Database Indexes & Backups

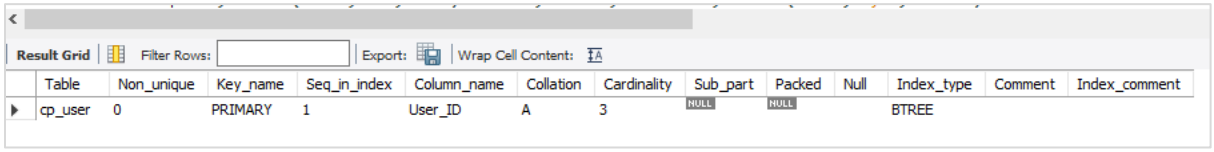
7.1 Why creates indexes?

Indexes are used to retrieve information from the database faster than in any other case. Indexes can't be visible by way of users, they may be truly used to speed up queries Indexes allow us to create looked after lists without developing all new sorted tables, which would soak up numerous storage area.

7.1.1 Indexes For users

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	**PRIMARY**	BTREE	Yes	No	User_ID	3	A	No	

```
use abc_community_portal;
show index from cp_user;
CREATE INDEX userx
ON users (Username,Email);
show index from cp_user;
```



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

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment
cp_user	0	PRIMARY	1	User_ID	A	3	NULL	NULL		BTREE		

7.1.2 Indexes for Accounts

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	**PRIMARY**	BTREE	Yes	No	Admin_ID	6	A	No	
Edit Rename Drop	**user_id**	BTREE	No	No	user_id	6	A	No	

```
use abc_community_portal;
show index from cp_account;
CREATE INDEX accountx
ON accounts (Fname,Email,Password);
show index from cp_account;
```

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

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment
cp_account	0	PRIMARY	1	Regi_id	A	3	NULL	NULL		BTREE		
cp_account	1	user_id	1	user_id	A	3	NULL	NULL		BTREE		

7.1.3 Indexes for Admins

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	**PRIMARY**	BTREE	Yes	No	Regi_ID	3	A	No	
Edit Rename Drop	**user_id**	BTREE	No	No	user_id	3	A	No	

```
use abc_community_portal;
show index from cp_admin;
CREATE INDEX adminx
ON admins (email,password);
show index from cp_admin;
```

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

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment
cp_admin	0	PRIMARY	1	admin_id	A	6	NULL	NULL		BTREE		
cp_admin	1	user_id	1	user_id	A	6	NULL	NULL		BTREE		

7.1.4 Indexes for Login

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	**PRIMARY**	BTREE	Yes	No	Login_ID	3	A	No	
Edit Rename Drop	**User_ID**	BTREE	No	No	User_ID	3	A	No	

```
use abc_community_portal;
show index from cp_login;
CREATE INDEX loginx
ON logins (Email,Password);
show index from cp_login;
```

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

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment
cp_login	0	PRIMARY	1	Login_id	A	3	NULL	NULL		BTREE		
cp_login	1	User_ID	1	User_ID	A	3	NULL	NULL		BTREE		

7.1.5 Indexes for Jobs

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	**PRIMARY**	BTREE	Yes	No	Vacancy_id	3	A	No	
Edit Rename Drop	**user_id**	BTREE	No	No	user_id	3	A	No	

```
use abc_community_portal;
show index from cp_job;
CREATE INDEX jobx
ON jobs (Salary,Shift);
show index from cp_job;
```

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

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment
cp_job	0	PRIMARY	1	Vacancy_id	A	3	NULL	NULL		BTREE		
cp_job	1	user_id	1	user_id	A	3	NULL	NULL		BTREE		

7.1.6 Indexes for Search Users

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	**PRIMARY**	BTREE	Yes	No	search_id	4	A	No	
Edit Rename Drop	**User_ID**	BTREE	No	No	User_ID	4	A	No	

```
use abc_community_portal;
show index from cp_searchuser;
CREATE INDEX searchx
ON searchs (Name,Location);
show index from cp_searchuser;
```

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

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment
cp_searchuser	0	PRIMARY	1	search_id	A	3	NULL	NULL		BTREE		
cp_searchuser	1	User_ID	1	User_ID	A	3	NULL	NULL		BTREE		

7.1.7 Indexes for Messages

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	**PRIMARY**	BTREE	Yes	No	Msg_ID	0	A	No	
Edit Rename Drop	**user_id**	BTREE	No	No	user_id	0	A	No	

```
use abc_community_portal;
show index from cp_message;
CREATE INDEX messagex
ON messages (Sender,Reciver);
show index from cp_message;
```

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

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment
cp_message	0	PRIMARY	1	Msg_ID	A	0	NULL	NULL		BTREE		
cp_message	1	user_id	1	user_id	A	0	NULL	NULL		BTREE		

7.1.8 Indexes for Companies

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	**PRIMARY**	BTREE	Yes	No	Com_ID	3	A	No	
Edit Rename Drop	**Vacancy_id**	BTREE	No	No	Vacancy_id	3	A	No	

```
use abc_community_portal;
show index from cp_company;
CREATE INDEX companyx
ON companys (Email,Contact,Location);
show index from cp_company;
```

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

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment
cp_company	0	PRIMARY	1	Com_id	A	3	NULL	NULL		BTREE		
cp_company	1	Vacancy_id	1	Vacancy_id	A	3	NULL	NULL		BTREE		

7.1.9 Indexes for Bulk Emails

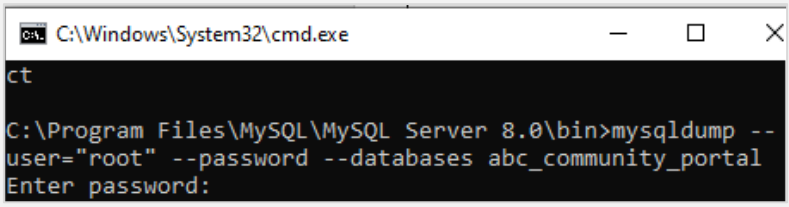
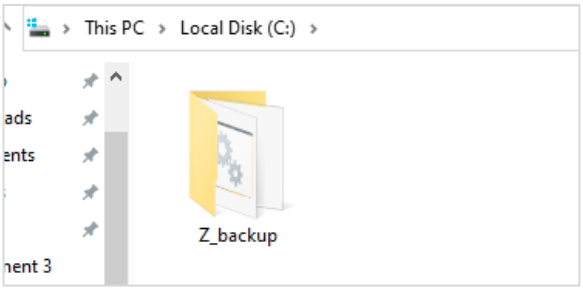
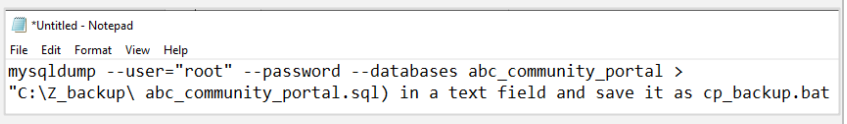
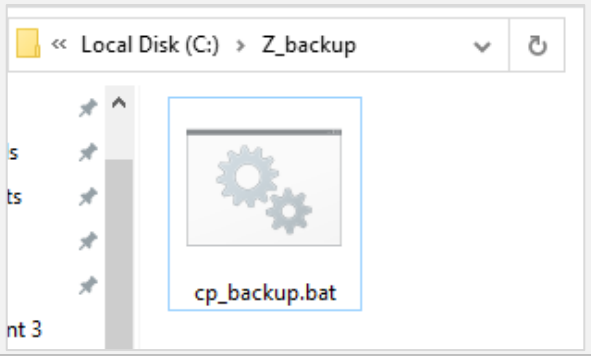
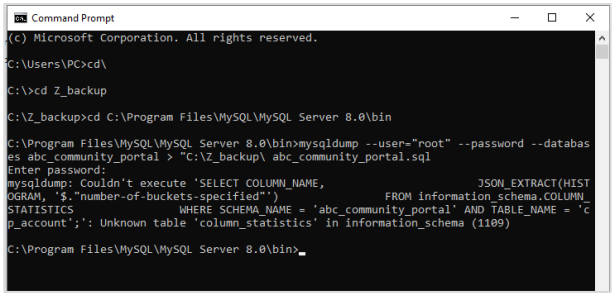
Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	**PRIMARY**	BTREE	Yes	No	B_Email_id	3	A	No	
Edit Rename Drop	**Admin_ID**	BTREE	No	No	Admin_ID	3	A	No	

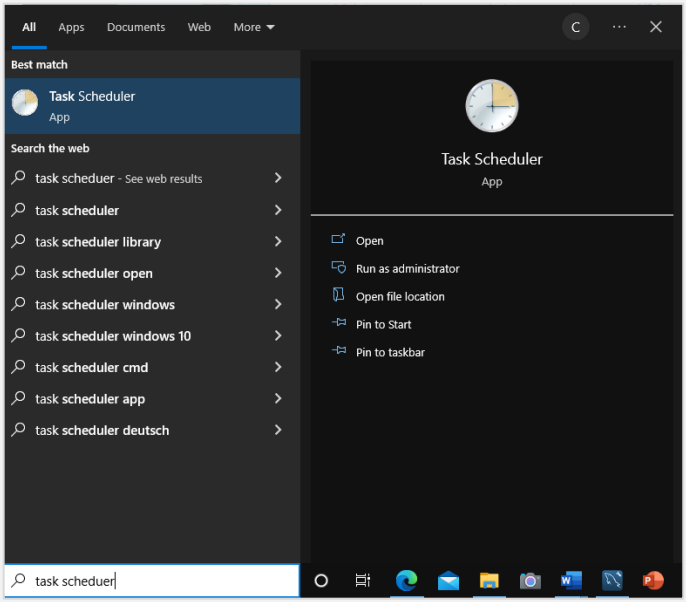
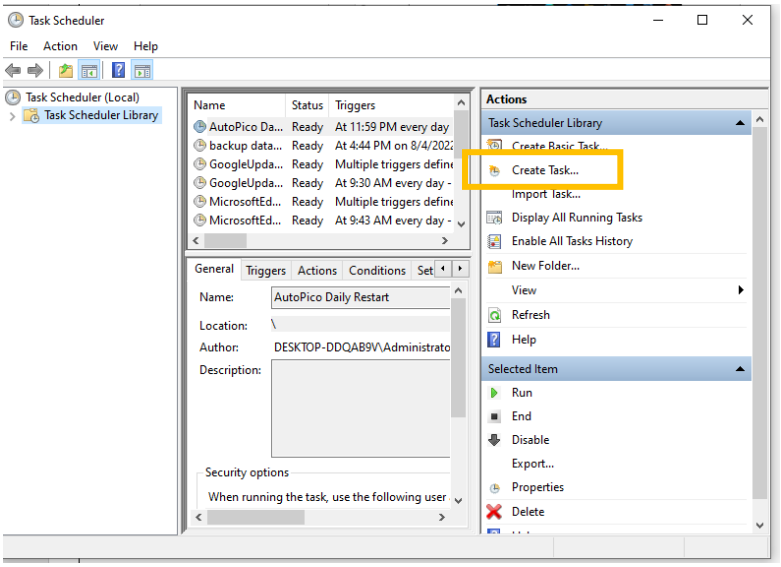
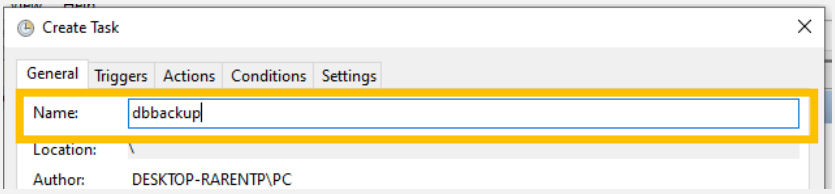
```
use abc_community_portal;
show index from cp_bulkemail;
CREATE INDEX bulkemailx
ON bulkemails (B_Email,B_EmailType);
show index from cp_bulkemail;
```

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

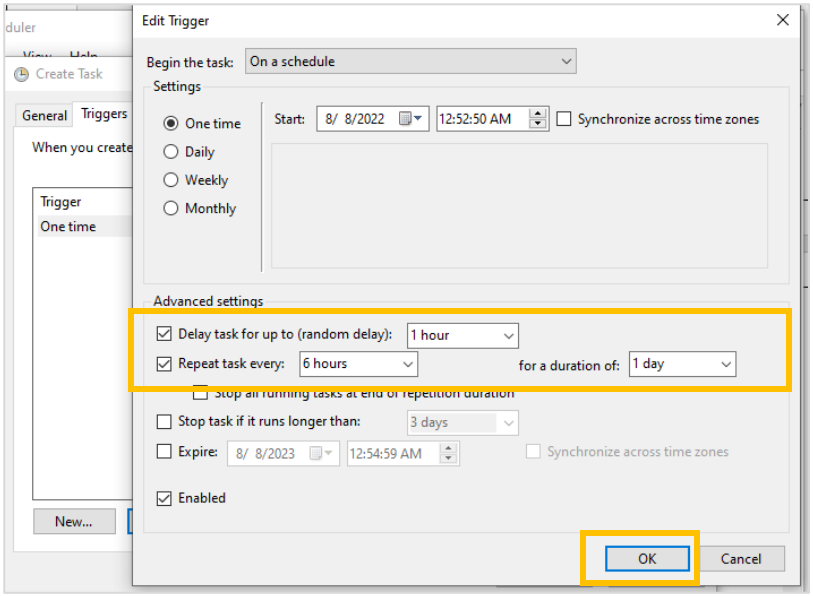
Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment
cp_bulkemail	0	PRIMARY	1	B_Email_id	A	3	NULL	NULL		BTREE		
cp_bulkemail	1	Admin_ID	1	Admin_ID	A	3	NULL	NULL		BTREE		

7.2 Database Backup

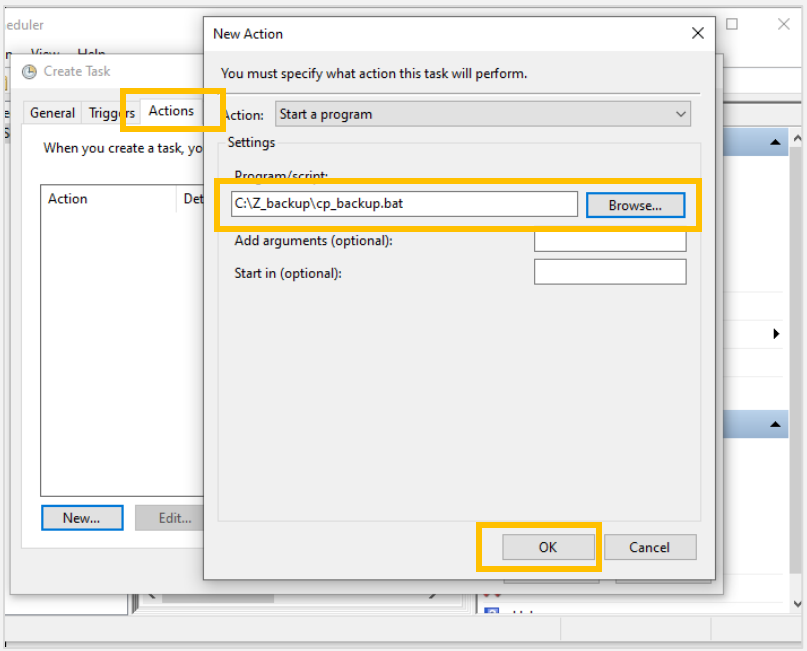
Steps	Screenshots
Step 1 - Choose your community portal database and test the command *(mysqldump --user="root" --password --databases abc_community_portal)*	
Step 2 - Create a folder in c drive name as Z_backup	
Step 3 - Create the script as executable file. Copy the command *(mysqldump --user="root" --password --databases abc_community_portal > "C:\Z_backup\ abc_community_portal.sql)* in a text field and save it as **cp_backup.bat**	 
Step 4 - Test the batch file	

Step 5 – Go to the Task Schedule	
Step 6 -Click on create task	
Step 7 -Enter the Take Name	

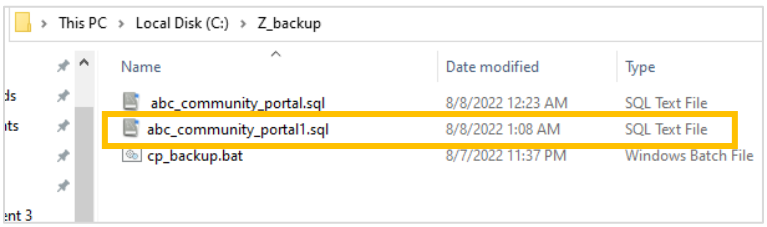
Step 8 - Create the trigger to run the task at the schedules time (every 6hours)



Step 9 - Set the action to be triggered. Browse and choose the batch file location

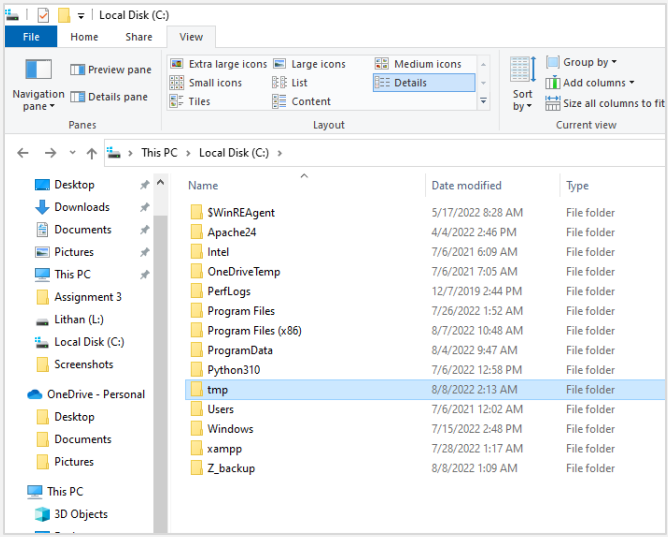
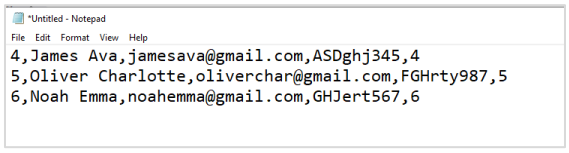
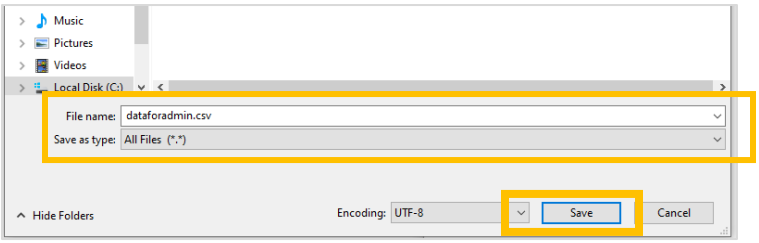
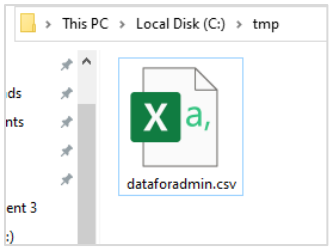


Step 10 - The backup file will appear automatically in the batch file directory

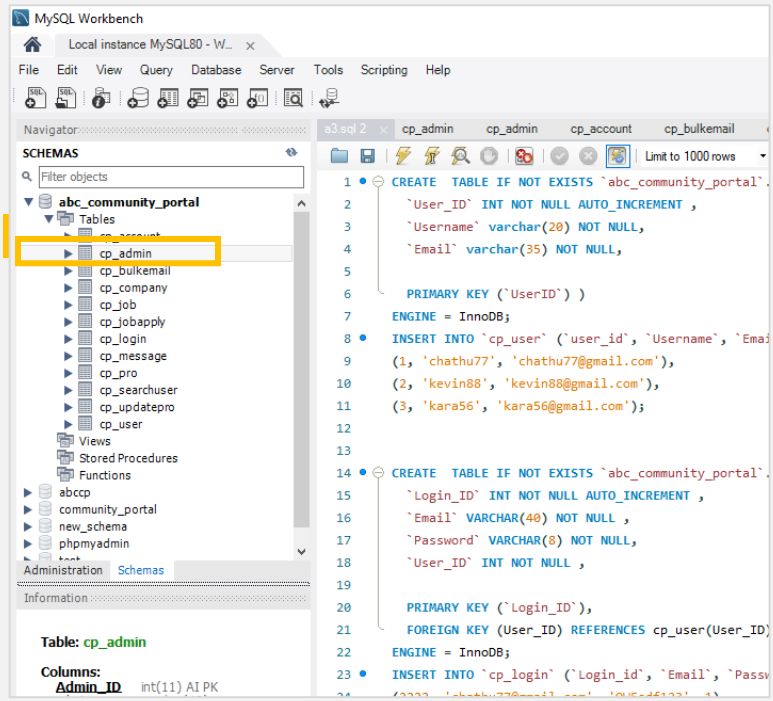


8. Community Portal Query & CSV Sample Data Import

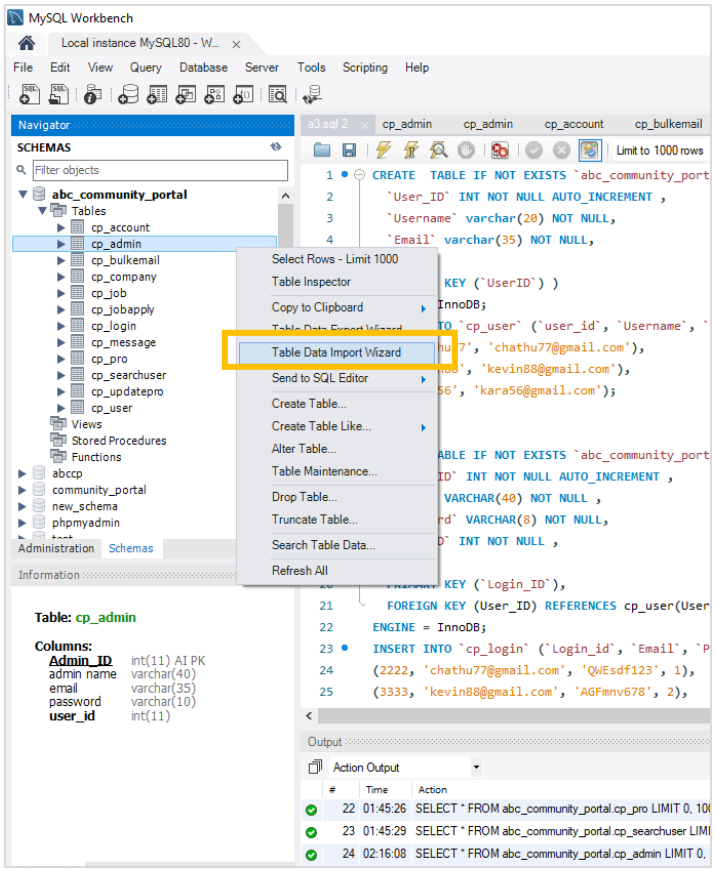
8.1 Steps to import CSV

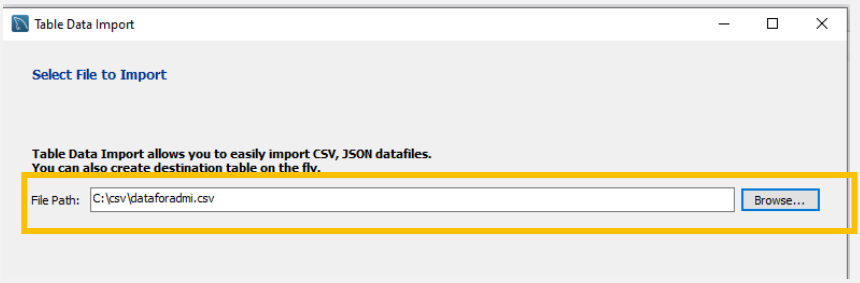
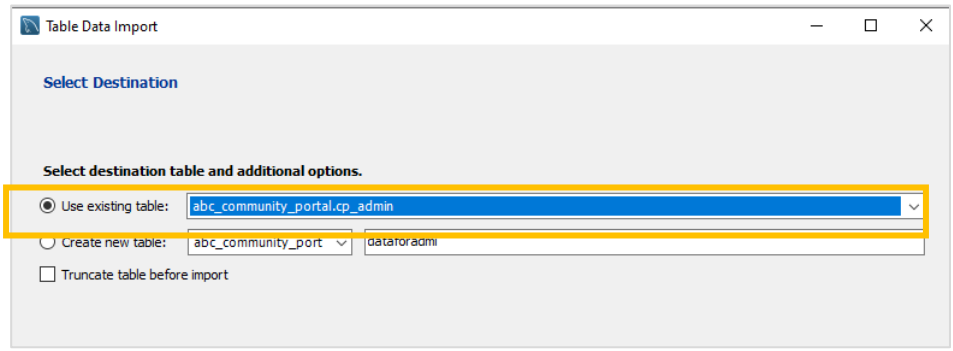
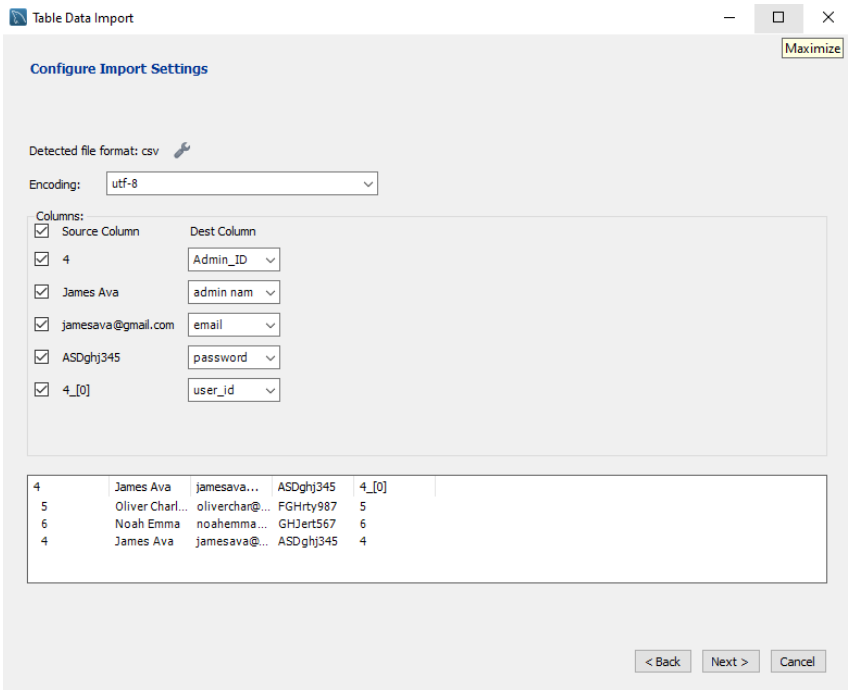
Steps	Screen Capture
Step 1 - Create tmp folder in C drive	
Step 2 - Add table details to the notepad and save it as a csv file in tmp folder	  

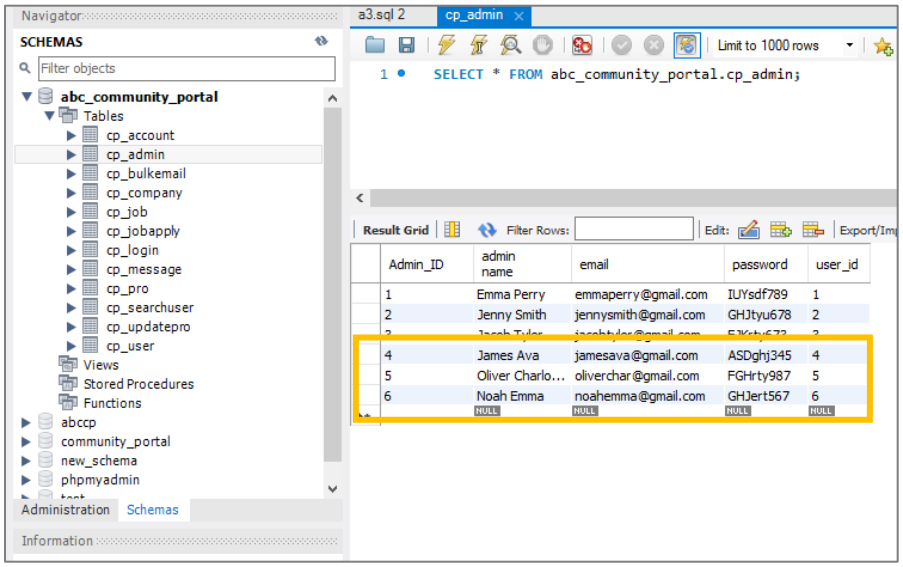
Step 3 - Go to the workbench and find the table that we need to add details



Step 4 - Right click on it (table) and click "Table data Import Wizard"



Step 5 - Browse the csv file that created	
Step 6 - Click Next and remained the default setting	 

Step 7 - The import is successful as the data was imported in the import results message	
Step 8 - Check the imported data using the SELECT query.	

8.2 5 Useful Queries to develop the application

Pages	Queries	Note	Evidence
Update Profile	INSERT INTO `cp_updatepro` (`Upd_id`, `Fname`, `Lname`, `Skills`, `Education`, `Work`, `Contact`, `user_id`) VALUES ('[value]', '[value]', '[value]', '[value]', '[value]', '[value]', '[value]', '[value]')	Storing user records for Skills, Education, Work, Contact to the database	Fig Q01
Job	INSERT INTO `cp_job` (`Vacancy_id`, `Company Name`, `Vacancy`, `Salary`, `Shift`, `user_id`) VALUES ('[value]', '[value]', '[value]', '[value]', '[value]', '[value]')	Storing job records for Company Name, Vacancy, Salary, Shift to the database	Fig Q02
Account (Registration)	INSERT INTO `cp_account` (`Regi_id`, `Fname`, `Lname`, `Email`, `Password`, `Contact`, `City`, `Country`, `user_id`) VALUES ('[value]', '[value]', '[value]', '[value]', '[value]', '[value]' , '[value]' , '[value]')	Storing user registration records for Email, Contact, Password, City, Country to the database	Fig Q03
Admin	INSERT INTO `cp_admin` (`admin_id`, `admin name`, `email`, `password`, `user_id`) VALUES ('[value]', '[value]', '[value]', '[value]', '[value]')	Storing admin details for Admin Name, Email and password to the database	Fig Q04
User	INSERT INTO `cp_user` (`user_id`, `Username`, `Email`) VALUES ('[value]', '[value]', '[value]')	Storing user record for username, Email to the database	Fig Q05

Fig Q01

```
INSERT INTO `cp_updatepro` (`Upd_id`, `Fname`, `Lname`, `Skills`, `Education`, `Work`, `Contact`, `user_id`) VALUES
(333, 'Chathu', 'Jayarathna', 'Flexible', 'Lithan Academy', 'AIA Finance', 742269978, 1),
(334, 'Kevin', 'Leo', 'Painting', 'Winston Hotel School', 'Thaj', 760528233, 2),
(335, 'Kara', 'Ashley', 'Goof Judgment', 'University of Stanford', 'S&S Holdings', 713800289, 3);
```

Upd_ID	Fname	Lname	Skills	Education	Work	Contact	user_id
333	Chathu	Jayarathna	Flexible	Lithan Academy	AIA Finance	742269978	1
334	Kevin	Leo	Painting	Winston Hotel School	Thaj	760528233	2
335	Kara	Ashley	Goof Judgment	University of Stanford	S&S Holdings	713800289	3

Fig Q02

```
INSERT INTO `cp_job` (`Vacancy_id`, `Company Name`, `Vacancy`, `Salary`, `Shift`, `user_id`) VALUES
(1, 'Virtusa', 'Software Engineer', 90000, 'Morning', 1),
(2, 'Telecome', 'Technician', 50000, 'Split', 2),
(3, 'Corella', 'Web Developer', 80000, 'Closing', 3);
```

Vacancy_id	Company Name	Vacancy	Salary	Shift	user_id
1	Virtusa	Software Engineer	90000	Morning	1
2	Telecome	Technician	50000	Split	2
3	Corella	Web Developer	80000	Closing	3

Fig Q03

```
INSERT INTO `cp_account` (`Regi_id`, `Fname`, `Lname`, `Email`, `Password`, `Contact`, `City`, `Country`, `user_id`) VALUES
(11, 'Chathu', 'Dil', 'chathu77@gmail.com', 'AGFmnv678', 742269978, 'Bandarawela', 'Sri Lanka', 1),
(22, 'Kevin', 'Leo', 'kevin88@gmail.com', 'QWEsdf123', 760528233, 'Tokyo', 'Japan', 2),
(33, 'Kara', 'Ashley', 'kara56@gmail.com', 'CVByui098', 713800289, 'Rome', 'Italy', 3);
```

Regi_id	Fname	Lname	Email	Password	Contact	City	Country	user_id
11	Chathu	Dil	chathu77@gmail.com	AGFmnv678	742269978	Bandarawela	Sri Lanka	1
22	Kevin	Leo	kevin88@gmail.com	QWEsdf123	760528233	Tokyo	Japan	2
33	Kara	Ashley	kara56@gmail.com	CVByui098	713800289	Rome	Italy	3

Fig Q04

```
INSERT INTO `cp_admin` (`admin_id`, `admin name`, `email`, `password`, `user_id`) VALUES
(1, 'Emma Perry', 'emmaperry@gmail.com', 'IUYsdf789', 1),
(2, 'Jenny Smith', 'jennysmith@gmail.com', 'GHJtyu678', 2),
(3, 'Jacob Tyler', 'jacobtyler@gmail.com', 'FJKrty673', 3);
```

admin_id	admin name	email	password	user_id
1	Emma Perry	emmaperry@gmail.com	IUYsdf789	1
2	Jenny Smith	jennysmith@gmail.com	GHJtyu678	2
3	Jacob Tyler	jacobtyler@gmail.com	FJKrty673	3

Fig Q05

```
INSERT INTO `cp_user` (`user_id`, `Username`, `Email`) VALUES
(1, 'chathu77', 'chathu77@gmail.com'),
(2, 'kevin88', 'kevin88@gmail.com'),
(3, 'kara56', 'kara56@gmail.com');
```

User_ID	Username	Email
1	chathu77	chathu77@gmail.com
2	kevin88	kevin88@gmail.com
3	kara56	kara56@gmail.com

8.3 Useful queries to meet the management requirements using joins

No	Table	Note	Query	Evidence
01	**Account**	All users' personal details, to fetch users, contact info information	SELECT regi_id, CONCAT(Fname, ' ',Lname) AS name, Email, Contact, CONCAT(City, ', ',Country) AS Location FROM cp_account ORDER BY name;	**T01**
02	**Update Profile + Public Profile**	All user's update profile details, to fetch users, Public Profile information	SELECT cp_updatepro.Upd_ID, CONCAT(cp_updatepro.FName, ' ' ,cp_updatepro.LName) AS FullName, cp_pro.Work, cp_pro.Favorite, cp_pro.Education FROM cp_updatepro JOIN cp_pro ON cp_updatepro.Upd_ID=cp_pro.Pro_ID ORDER BY FullName;ORDER BY name;	**T02**
03	**Company + Job**	All users update profile details, to fetch users, Public Profile information	SELECT cp_company.Com_id, CONCAT(cp_company.Email, ' ' ,cp_company.Contact) AS About, cp_job.Vacancy_id, cp_job.Salary, cp_job.Shift FROM cp_company JOIN cp_job ON cp_company.Com_id=cp_job.Vacancy_id ORDER BY About;	**T03**
04	**User + Job**	All users job details, to fetch users, Job information	SELECT cp_pro.Name, cp_job.Vacancy FROM cp_pro INNER JOIN cp_job ON cp_pro.user_id = cp_job.user_id;	**T04**

05	**Job + Search User**	All search users job details. To fetch search users, Job Information	SELECT cp_job.CompanyName, cp_searchuser.Name, cp_searchuser.Location FROM cp_job RIGHT JOIN cp_searchuser ON cp_job.User_ID = cp_searchuser.User_ID ORDER BY cp_job.CompanyName;	**T05**

T01

	regi_id	name ▲ 1	Email	Contact	Location
Edit Copy Delete	11	Chathu Dil	chathu77@gmail.com	742269978	Bandarawela, Sri Lanka
Edit Copy Delete	33	Kara Ashley	kara56@gmail.com	713800289	Rome, Italy
Edit Copy Delete	22	Kevin Leo	kevin88@gmail.com	760528233	Tokyo, Japan

T02

Upd_ID	FullName ▲ 1	Work	Favorite	Education
333	Chathu Jayarathna	AIA Finance	Music, Food	Lithan Academy
335	Kara Ashley	S&S Holdings	Reading	University of Stanford
334	Kevin Leo	Thaj	Travel	Winston Hotel School

T03

Com_id	About ▲ 1	Vacancy_id	Salary	Shift
3	corella@gmail.com 573800289	3	80000	Closing
2	telecom@gmail.com 570528233	2	50000	Split
1	virtusa@gmail.com 572269976	1	90000	Morning

T04

Name	Vacancy
Chathu Jayarathna	Software Engineer
Kevin Leo	Technician
Kara Ashley	Web Developer

T05

CompanyName ▲ 1	Name	Location
Corella	Joan Davies	Dubai
Telecome	Diana Alsop	Bangladesh
Virtusa	Blake Knox	France

9. References

- W3Schools Online Web Tutorials
- MySQL Tutorial (tutorialspoint.com)
- MySQL Tutorial for Beginners [Full Course] - YouTube