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 | |
| Agung Yuda Pratama | |  | |
| Date issued | Completion date | | Submitted on |
|  | 25th jan 2023 | |  |
|  | |  | |
| Project title | Design, Implement, Test & Document Community Portal Database. | | |

|  |
| --- |
| Learner declaration |
| I certify that the work submitted for this assignment is my own and research sources are fully acknowledged.  Student signature: Date: 20th jan 2023 |

TABLE OF CONTENTS

**Project Background**

**Project Purpose**

This Project is used for theSummative Assessment of student**s** in the Module ‘Database  
Design & Implementation of the ITSF Course ‘NICF-Advanced Certificate in Web  
Development”**.** This Project considers the skills required to design, develop & implement a MySQL  
Database.

**Existing system**

Making a developer community site is the final project for our UI Framework course. It indicates that we are currently prepared to create a methodical database system for the community portal.

**Need for database system**

The system developer will use the Relational Database Management System (RDBMS) to store and organize the data while creating databases. Therefore, in addition to creating the table, the developer organizes the data in the table. In the future, the system mightfeature a backup system for security.

**Project Objectives**

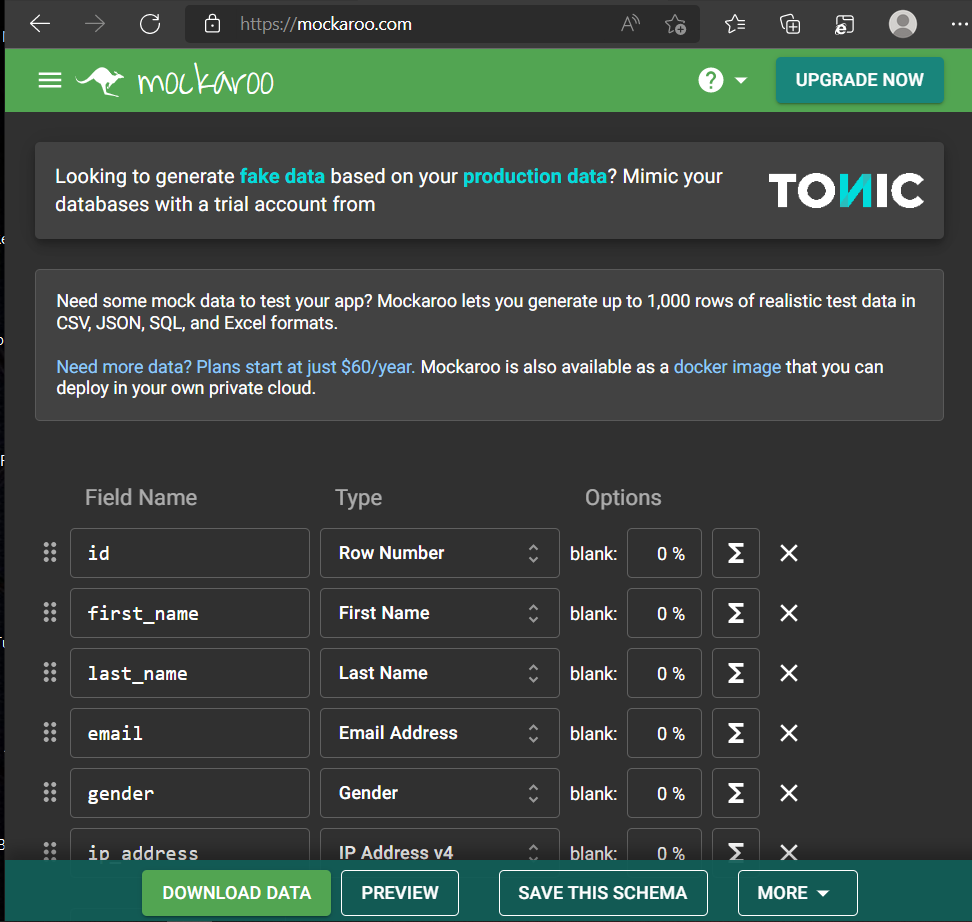
Project goal (2-3 lines)

We had to make a database management system for the community portal after the company had reached us.

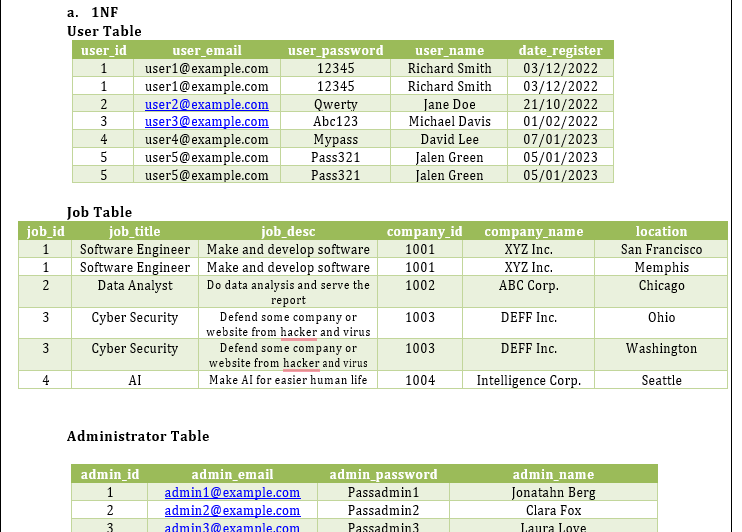
**Tools & platform used**:-

Mockaroo.com:- To generate sample data

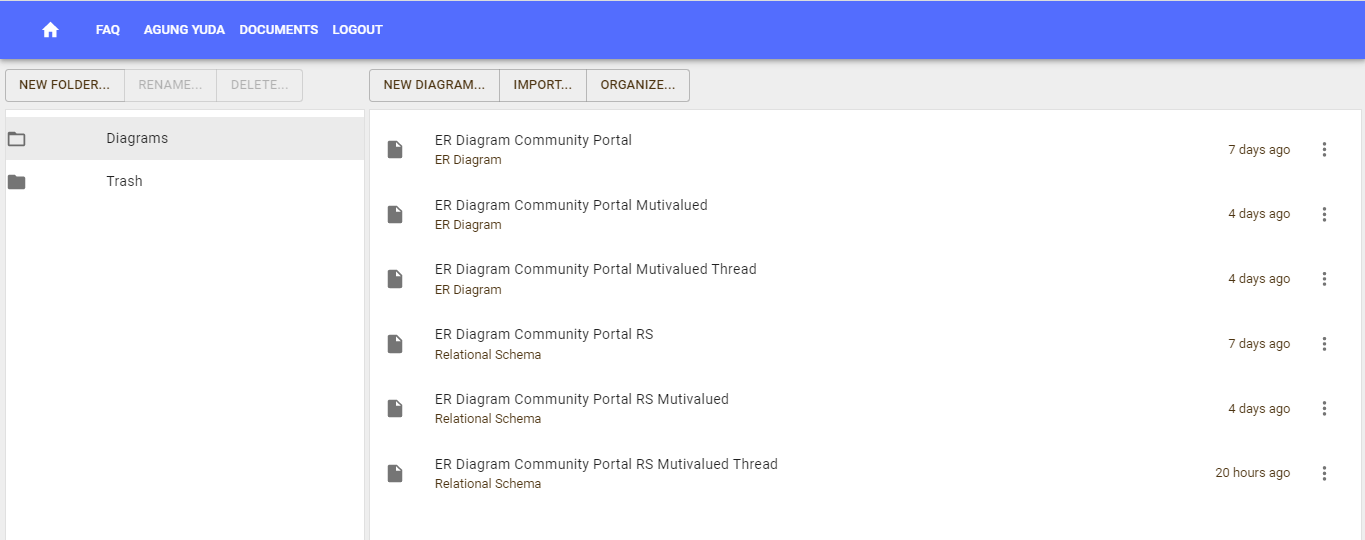


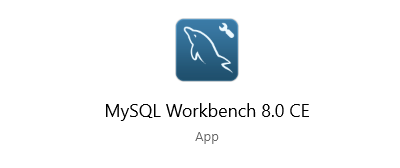


Word:- To prepare normalization records



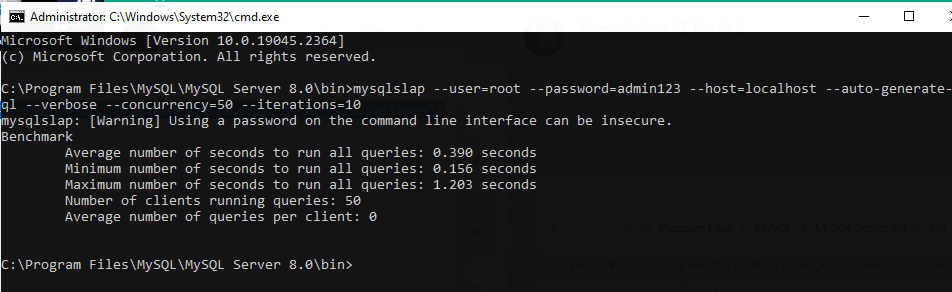
ERDPlus – for making ER Diagram and generating relational schema and sql query.



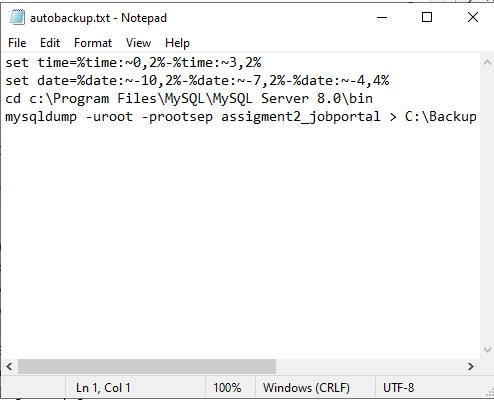
MySQL Workbench – for making database, EERD and phpMyAdmin – for making data dictionary



Command prompt – for load testing



Notepad – for making batch file



1. Task 1 Create a database requirements specification document for the project scenario, Include it as part of Project Report. (A1)

Task Statement

Database Requirements Specification Documentation

Solution

**Hardware requirements**

**-** Intel Core I3 Gen 8th

**-** Ram 8GB

**-** SSD 120GB

**Software requirements**

* MySQL Server (Database Server)
* MySQL Workbench (Database Client)
* XAMPP (Apache Server )
* phpMyAdmin (Data Dictionary)
* Web Browser (Google Chrome)
* Task Scheduler Windows
* Command Prompt
* Notepad

Database requirements (A1)

|  |  |  |
| --- | --- | --- |
| **Entity** | **Description** | **Page/Function** |
| **User** | It contains information about registering and login | Registration and Login Page |
| **Administrator** | Admin can administer users and send bulk email | Administrator Page |
| **Email** | To store the e-mail information of each user registered in the system. | Registration and Login Page |
| **Job** | It provides details about the user's job prospects. | Job Page |
| **Messages** | Users can send and receive messages | Messages Page |
| **Thread** | is a webpage or section of a website that displays a specific discussion or conversation, known as a "thread," between multiple users. | Thread Page |

Entities Relationship Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Entity one | Entity two | Relationship | Note/Description |
| 1 | User | Administrator | one to many | One Admin can administer many User |
| 2 | Admin | Bulk email | One to many | One admin can send many bulk emails |
| 3 | User | Thread | Many to many | One User can make and participate multiple Thread and One Thread can have multiple User |
| 4 | User | Messages | One to Many | One user can have multiple messages, but a One message can only have one user associated with it |
| 5 | User | Job | Many to Many | a one user can have multiple jobs and a one job can have multiple users associated with it |

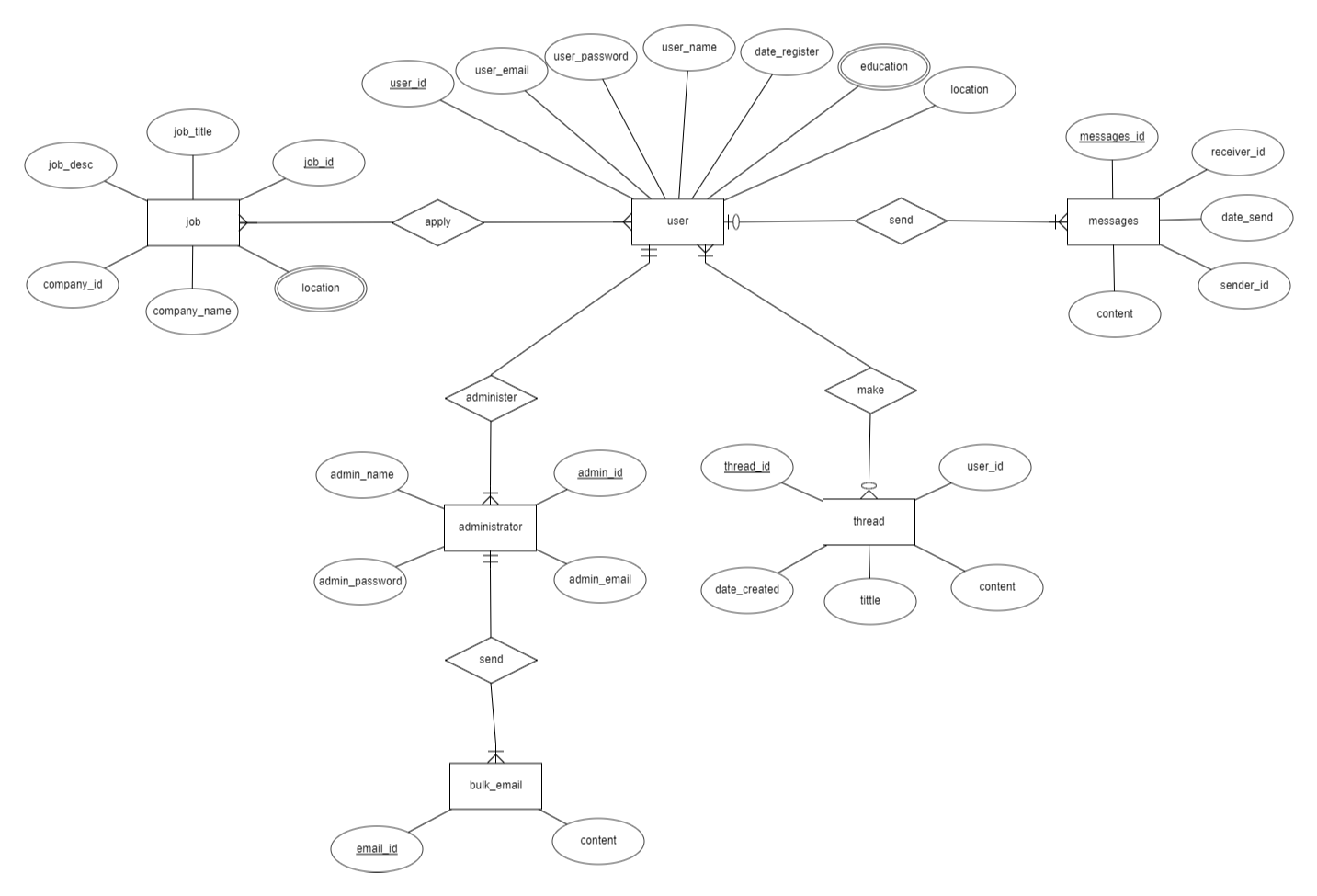
1. Task 2. Create a database design document.Create a Conceptual Design for the proposed Database. Create Logical Design for the proposed Database., Create the ER Diagram for the project scenario. Include the above as part of Project Presentation. (A2)

Task Statement

Solution

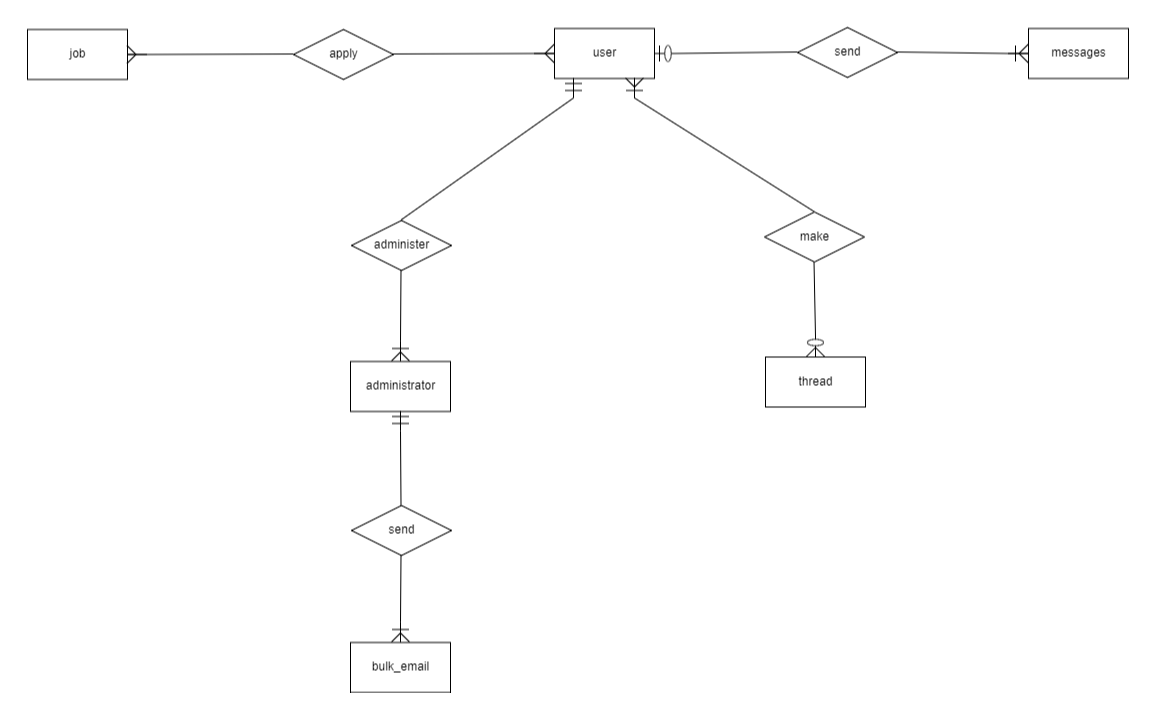
Explain what RDBMS is (explain in 1-2 lines)

Relational Database Management System (RDBMS) is a software package **that stores and manages** databases.It can Define,Construct & Manipulate the database. A DBMS is **a** software that allows **the** creation, definition and manipulation of **the** database, allowing users to **easily store, process and analyse data**.



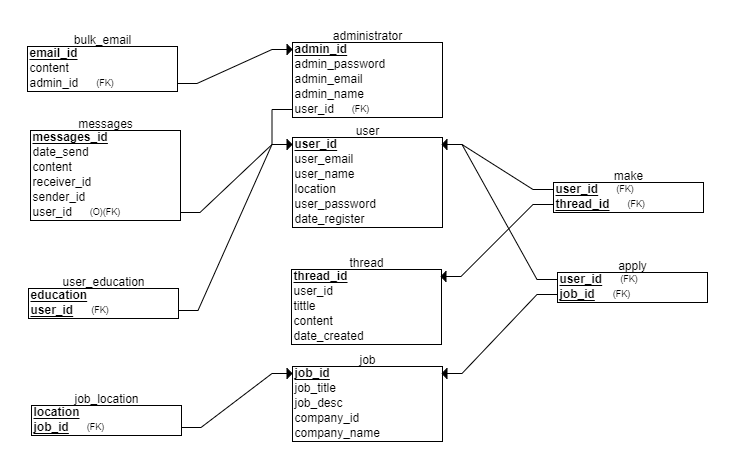
Conceptual design

The goal of **the** conceptual design phase is to create a conceptual model based on the criteria that **the developer has already made**. The result of conceptual design **conceptual** data model that **explains** entity data, **the** relationship between the entities and the attribute**.**



Logical design

Logical database design is the process of creating an information model based on a specific data model but not exactly depend on database management**.**



Physical design

Physical data refers to the idea of how data is organized and kept on storage devices, employing a number of tables to explain the data and their relationships.

1NF

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| user\_id | user\_email | user\_password | user\_name | date\_register |
| 1 | user1@example.com | 12345 | Richard Smith | 03/12/2022 |
| 2 | [user2@example.com](mailto:user2@example.com) | Qwerty | Jane Doe | 21/10/2022 |
| 3 | [user3@example.com](mailto:user3@example.com) | Abc123 | Michael Davis | 01/02/2022 |
| 4 | user4@example.com | Mypass | David Lee | 07/01/2023 |
| 5 | user5@example.com | Pass321 | Jalen Green | 05/01/2023 |

2NF

|  |  |  |  |
| --- | --- | --- | --- |
| user\_id | user\_email | user\_password | user\_name |
| 1 | user1@example.com | 12345 | Richard Smith |
| 2 | [user2@example.com](mailto:user2@example.com) | Qwerty | Jane Doe |
| 3 | [user3@example.com](mailto:user3@example.com) | Abc123 | Michael Davis |
| 4 | user4@example.com | Mypass | David Lee |
| 5 | user5@example.com | Pass321 | Jalen Green |

|  |  |  |
| --- | --- | --- |
| education | location | user\_id |
| Bachelor's | Illinois | 1 |
| Master’s | Illinois | 1 |
| Master’s | Los Angeles | 2 |
| Ph.D | New York | 3 |
| High School | Los Angeles | 4 |
| Bachelor’s | Illinois | 5 |
| Master’s | Illinois | 5 |

|  |  |
| --- | --- |
| user\_id | date\_register |
| 1 | 03/12/2022 |
| 2 | 21/10/2022 |
| 3 | 01/02/2022 |
| 4 | 07/01/2023 |
| 5 | 05/01/2023 |

3NF

|  |  |  |
| --- | --- | --- |
| user\_id | user\_email | user\_password |
| 1 | user1@example.com | 12345 |
| 2 | [user2@example.com](mailto:user2@example.com) | Qwerty |
| 3 | [user3@example.com](mailto:user3@example.com) | Abc123 |
| 4 | user4@example.com | Mypass |
| 5 | user5@example.com | Pass321 |

|  |  |
| --- | --- |
| user\_id | user\_name |
| 1 | Richard Smith |
| 2 | Jane Doe |
| 3 | Michael Davis |
| 4 | David Lee |
| 5 | Jalen Green |

|  |  |
| --- | --- |
| education | user\_id |
| Bachelor's | 1 |
| Master’s | 1 |
| Master’s | 2 |
| Ph.D | 3 |
| High School | 4 |
| Bachelor’s | 5 |
| Master’s | 5 |

|  |  |
| --- | --- |
| location | user\_id |
| Illinois | 1 |
| Illinois | 1 |
| Los Angeles | 2 |
| New York | 3 |
| Los Angeles | 4 |
| Illinois | 5 |
| Illinois | 5 |

|  |  |
| --- | --- |
| user\_id | date\_register |
| 1 | 03/12/2022 |
| 2 | 21/10/2022 |
| 3 | 01/02/2022 |
| 4 | 07/01/2023 |
| 5 | 05/01/2023 |

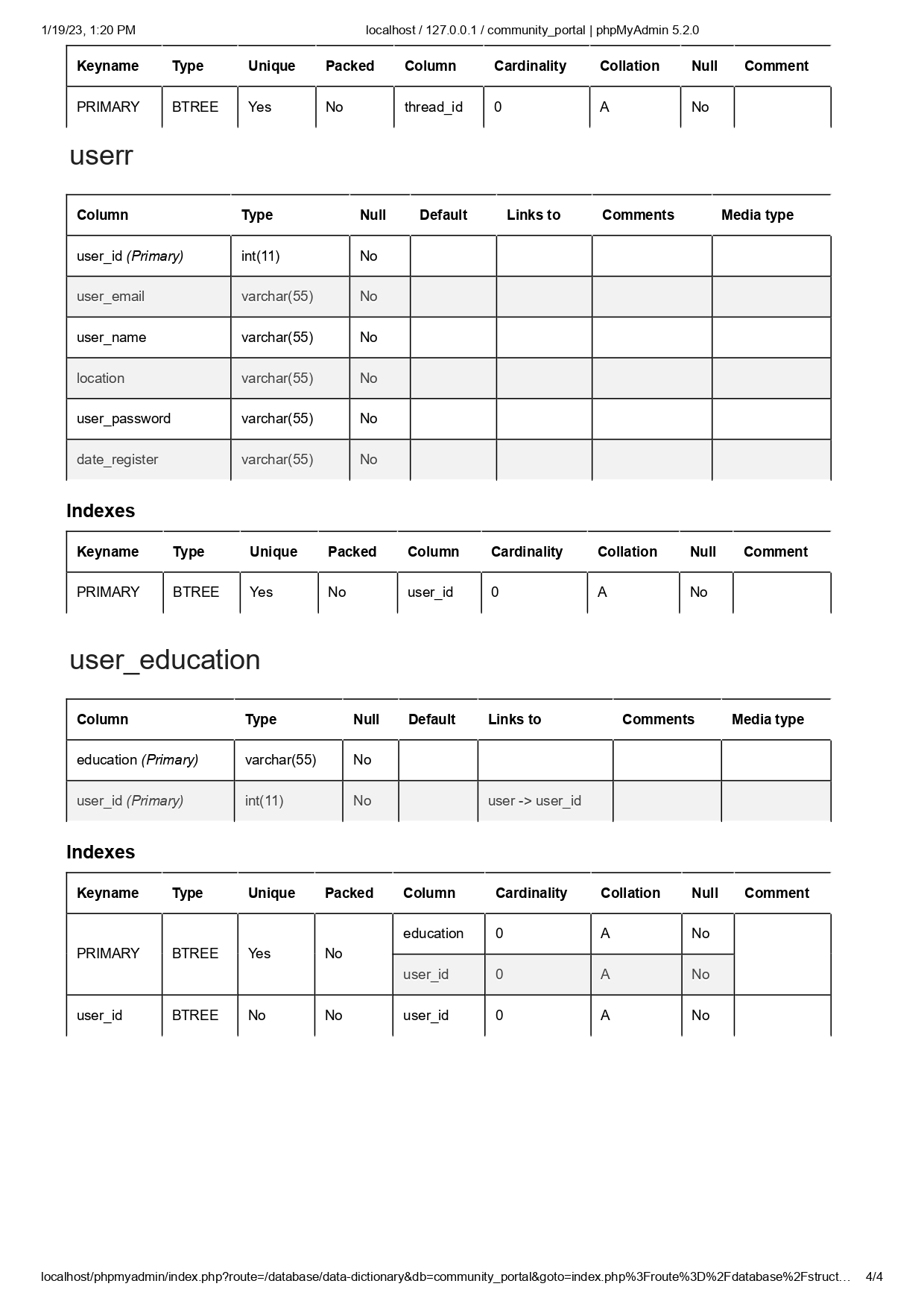
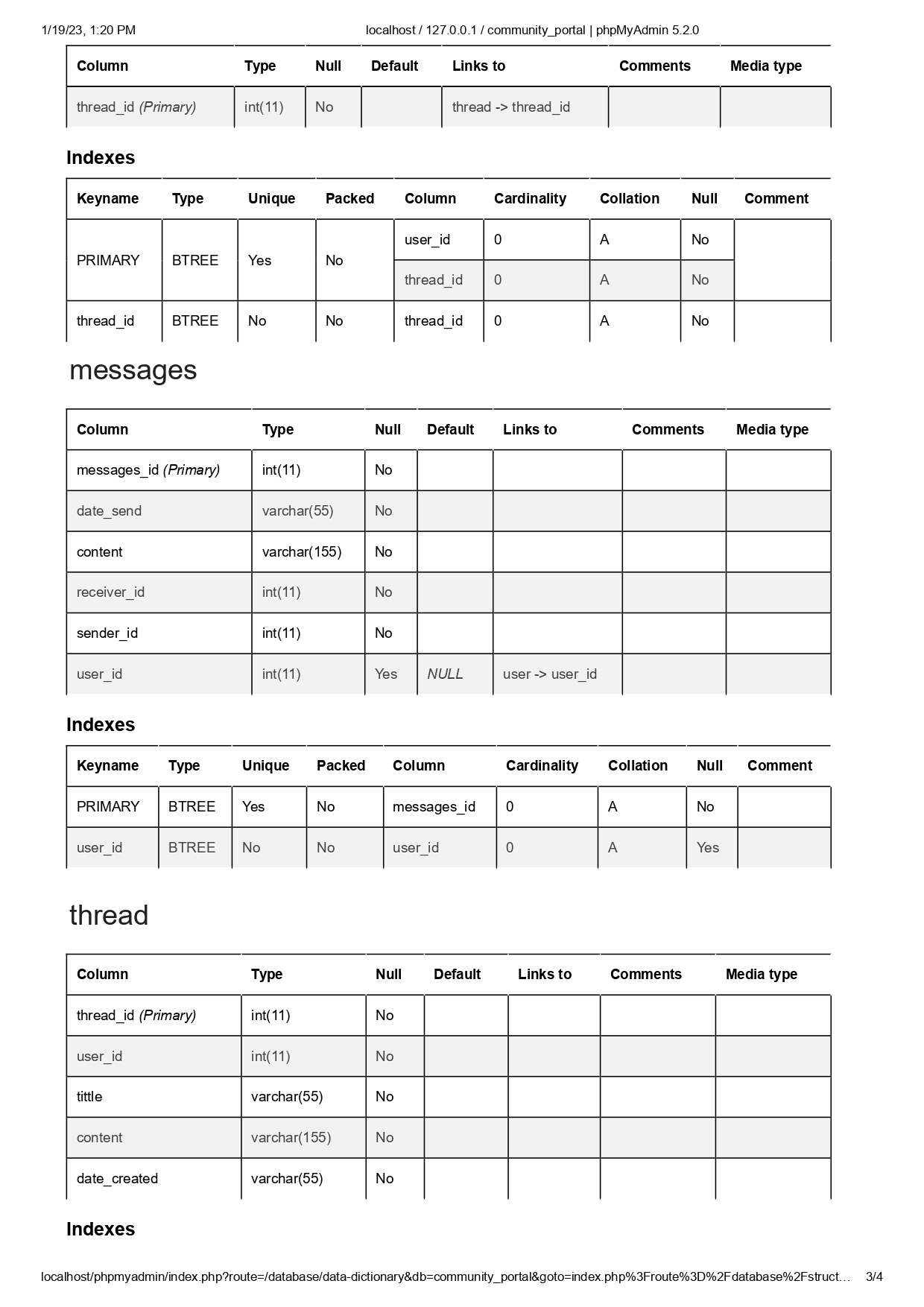
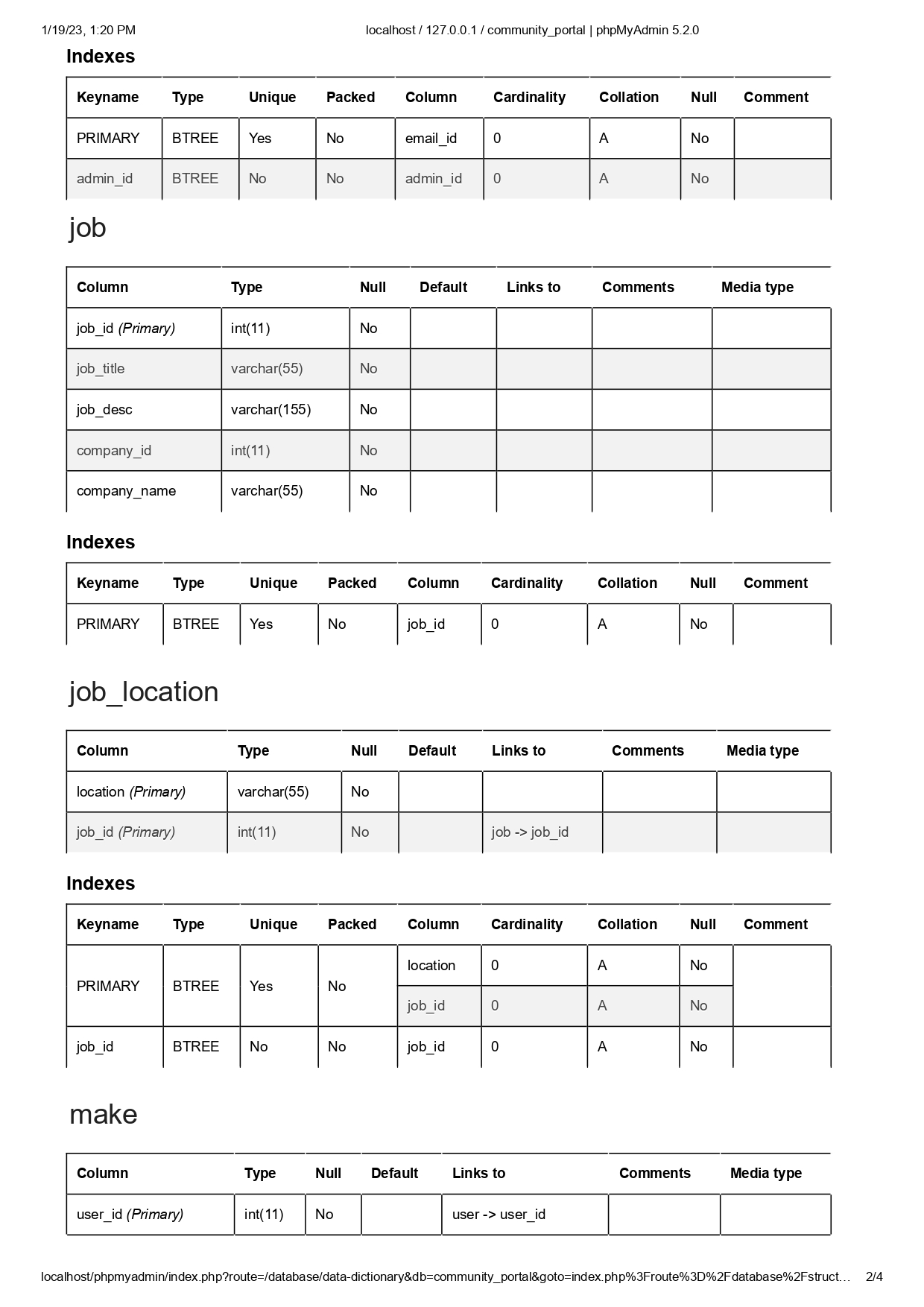
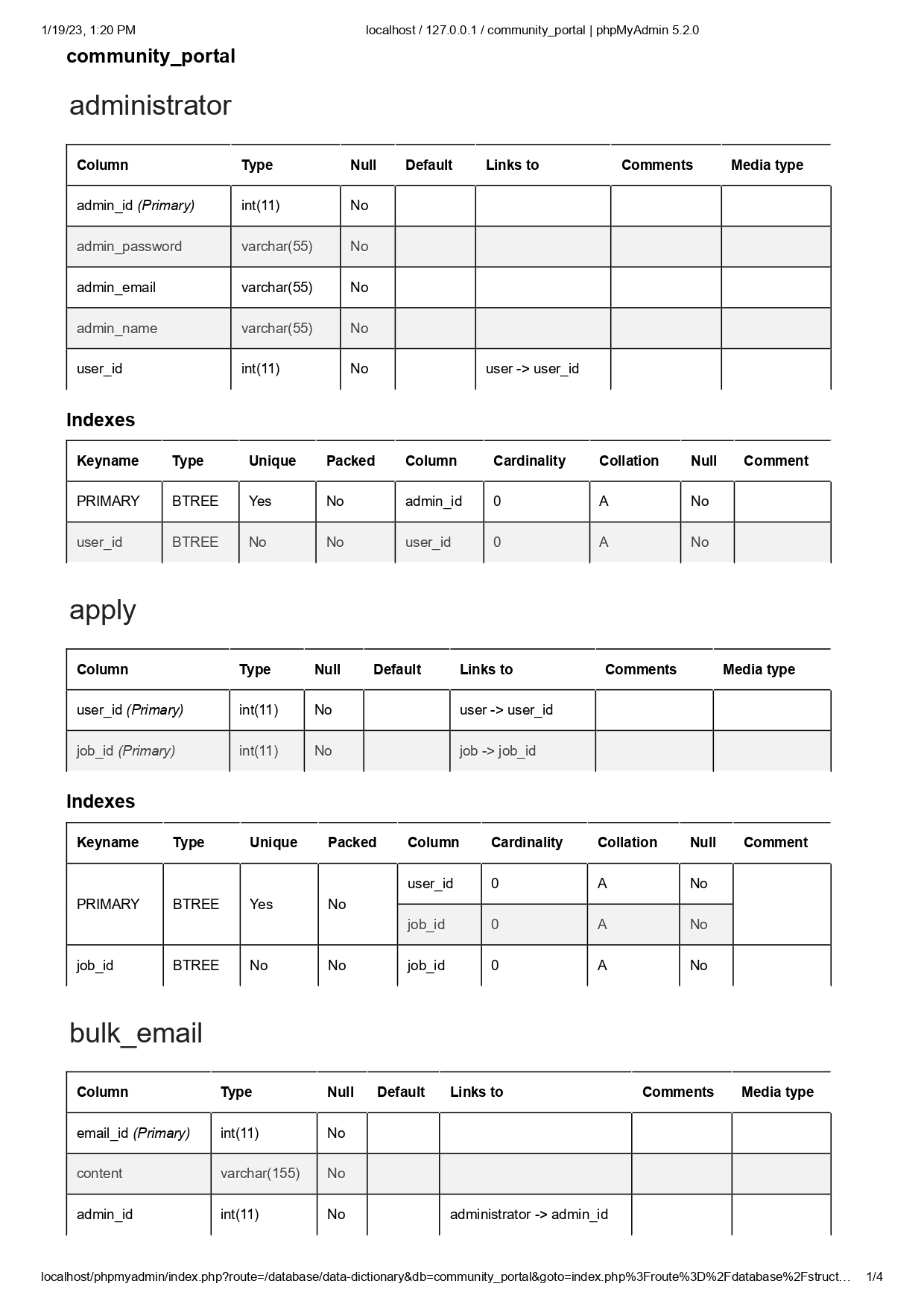
Entities and attributes for community Portal

|  |  |  |
| --- | --- | --- |
| Entities | Description | Attributes |
| User | Information needed for registration or login | 1. User\_id (PK) 2. User\_email 3. User\_name 4. User\_password 5. Date\_register 6. Education 7. Location |
| Administrator | The administration site can  only access by the admin. | 1. Admin\_id 2. Admin\_email 3. Admin\_password 4. Admin\_name 5. User\_id (FK) |
| Bulk\_email | Admin sends bulk email to users, tells them about the new features, and updates them about information. | 1. Email\_id (PK) 2. Content 3. Admin\_id (FK) |
| Thread | is a webpage or section of a website that displays a specific discussion or conversation, known as a "thread," between multiple users. | 1. Thread\_id (PK) 2. User\_id 3. Title 4. Content 5. Date\_created |
| Messages | is a website section showing a list of user-to-user private communications. Instead of conversing in a public thread, this feature enables users to send and receive private messages from and to other users on the community site. | 1. Messages\_id (PK) 2. Date\_send 3. Content 4. Receiver\_id 5. Sender\_id 6. User\_id (FK) |
| Job | It provides details about the user's job prospects. | 1. Job\_id (PK) 2. Job\_title 3. Job\_desc 4. Company\_id 5. Company\_name 6. Location |

1. Task 3:- Create the Database dictionary with tables, fields & datatypes . Include the above as part of Project Report (A3)

Task Statement:

Create the database dictionary



Task 4:- Create a MySQL database & Implement the database design in that database. Implement Primary Key, Foreign Key & Constraints.Use SQL Scripts & Mysql or phpMyAdmin to create the database.Produce the Screen capture of created tables in phpMyAdmin or MySQL Command prompt. Include it as part of Project Presentation.

Implement the database in MySQL using SQL scripts

Solution

**Table for User**

CREATE TABLE user(

user\_id INT NOT NULL,

user\_email VARCHAR(55) NOT NULL,

user\_name VARCHAR(55) NOT NULL,

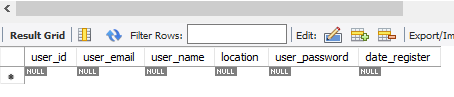
location VARCHAR(55) NOT NULL,

user\_password VARCHAR(55) NOT NULL,

date\_register VARCHAR(55) NOT NULL,

PRIMARY KEY (user\_id)

);



**Table for Administrator**

CREATE TABLE administrator(

admin\_id INT NOT NULL,

admin\_password VARCHAR(55) NOT NULL,

admin\_email VARCHAR(55) NOT NULL,

admin\_name VARCHAR(55) NOT NULL,

user\_id INT NOT NULL,

PRIMARY KEY (admin\_id),

FOREIGN KEY (user\_id) REFERENCES user(user\_id)

);



**Table for Bulk\_email**

CREATE TABLE bulk\_email(

email\_id INT NOT NULL,

content VARCHAR(155) NOT NULL,

admin\_id INT NOT NULL,

PRIMARY KEY (email\_id),

FOREIGN KEY (admin\_id) REFERENCES administrator(admin\_id)

);



**Table for Messages**

CREATE TABLE messages(

messages\_id INT NOT NULL,

date\_send VARCHAR(55) NOT NULL,

content VARCHAR(255) NOT NULL,

receiver\_id INT NOT NULL,

sender\_id INT NOT NULL,

user\_id INT,

PRIMARY KEY (messages\_id),

FOREIGN KEY (user\_id) REFERENCES user(user\_id)

);



**Table for Job**

CREATE TABLE job

(

job\_id INT NOT NULL,

job\_title VARCHAR(55) NOT NULL,

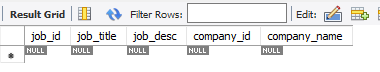
job\_desc VARCHAR(155) NOT NULL,

company\_id INT NOT NULL,

company\_name VARCHAR(55) NOT NULL,

PRIMARY KEY (job\_id)

);



**Table for Thread**

CREATE TABLE thread(

thread\_id INT NOT NULL,

user\_id INT NOT NULL,

tittle VARCHAR(55) NOT NULL,

content VARCHAR(155) NOT NULL,

date\_created VARCHAR(55) NOT NULL,

PRIMARY KEY (thread\_id)

);



**Table for Apply**

CREATE TABLE apply(

user\_id INT NOT NULL,

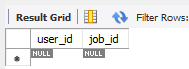
job\_id INT NOT NULL,

PRIMARY KEY (user\_id, job\_id),

FOREIGN KEY (user\_id) REFERENCES user(user\_id),

FOREIGN KEY (job\_id) REFERENCES job(job\_id)

);



**Table for Make**

CREATE TABLE make(

user\_id INT NOT NULL,

thread\_id INT NOT NULL,

PRIMARY KEY (user\_id, thread\_id),

FOREIGN KEY (user\_id) REFERENCES user(user\_id),

FOREIGN KEY (thread\_id) REFERENCES thread(thread\_id)

);



**Table for User\_education**

CREATE TABLE user\_education(

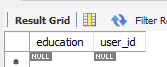
education VARCHAR(55) NOT NULL,

user\_id INT NOT NULL,

PRIMARY KEY (education, user\_id),

FOREIGN KEY (user\_id) REFERENCES user(user\_id)

);



**Table for Job\_location**

CREATE TABLE job\_location(

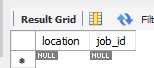
location VARCHAR(55) NOT NULL,

job\_id INT NOT NULL,

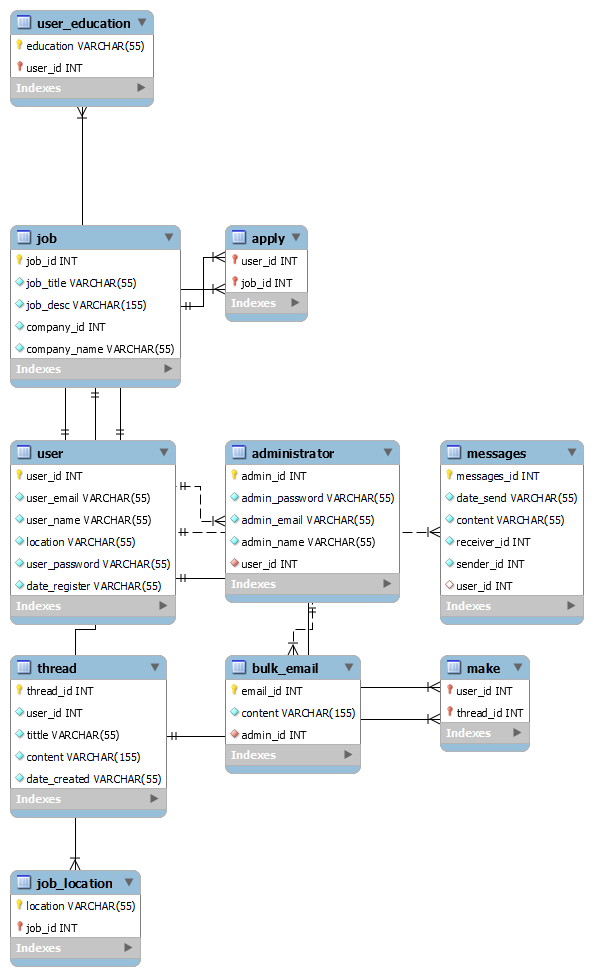
PRIMARY KEY (location, job\_id),

FOREIGN KEY (job\_id) REFERENCES job(job\_id)

);



EERD Diagram



Task 5:- Create indexes in table, provide script for creating the index(4 tables).Create a Backup script to back up the database every 6 hours. Include the mechanism and sources as part of Project Report

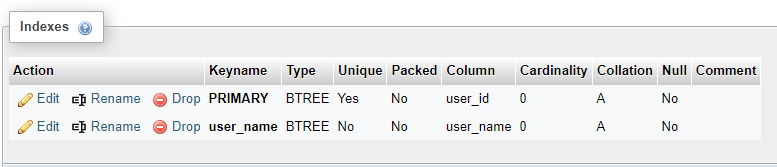
Task Statement:

Create indexes and backup script for daily backup

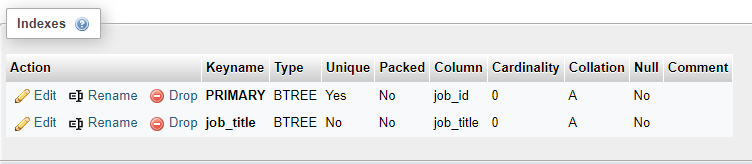
Solution

Part 1) Create Indexes for any 4 tables

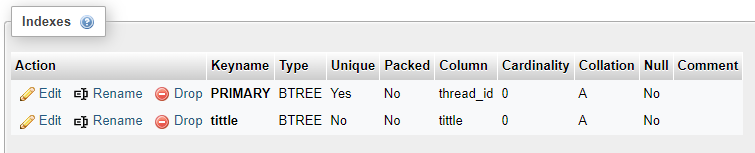
"ALTER TABLE `userr` ADD INDEX(`user\_name`);"



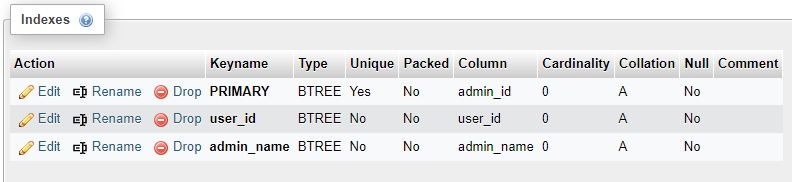
"ALTER TABLE `job` ADD INDEX(`job\_title`);"



"ALTER TABLE `thread` ADD INDEX(`tittle`);"



"ALTER TABLE `administrator` ADD INDEX(`admin\_name`);"



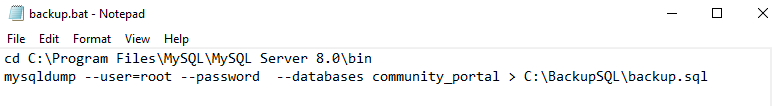
Part 2) Backup your database

Choose the community portal database.

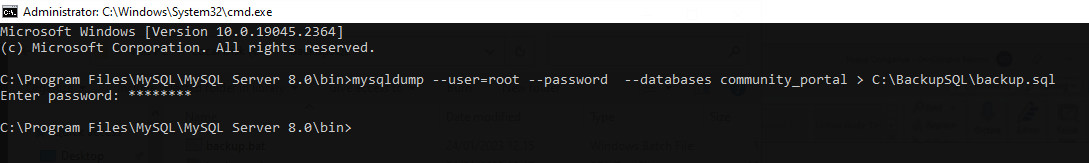


Create a new folder to store the backup data and than crate a script to backup data and save it as

.batch extensions in the folder.



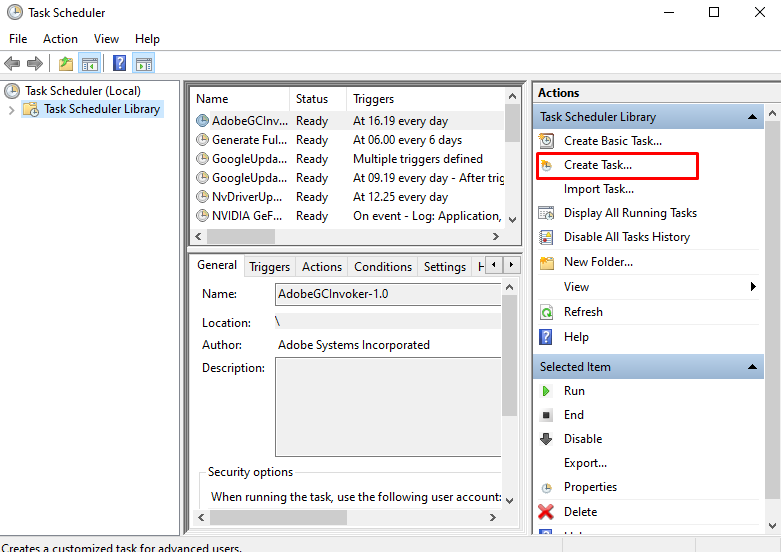
Then test the batch file



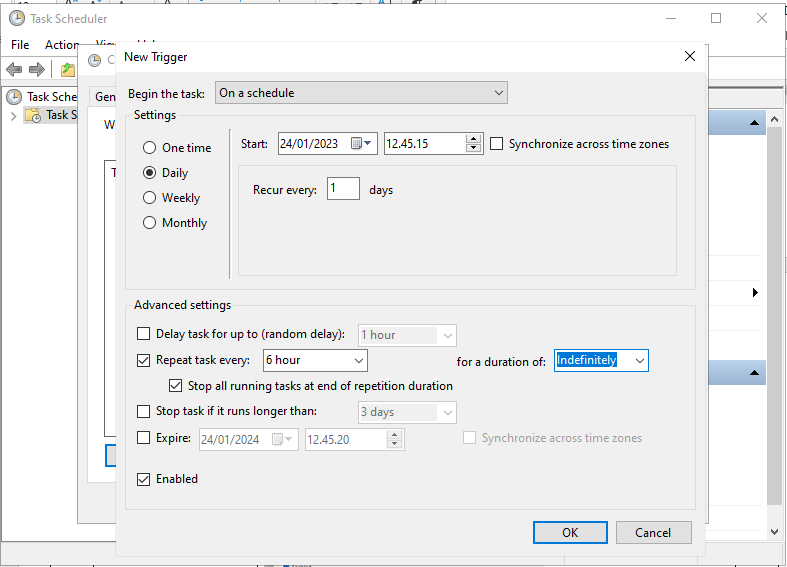
This is the output



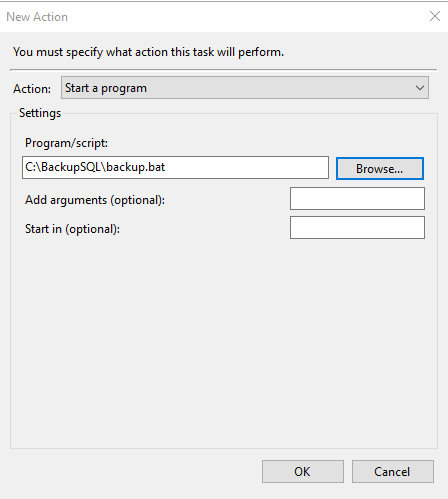
Now use windows task scheduler. Click on Create Task



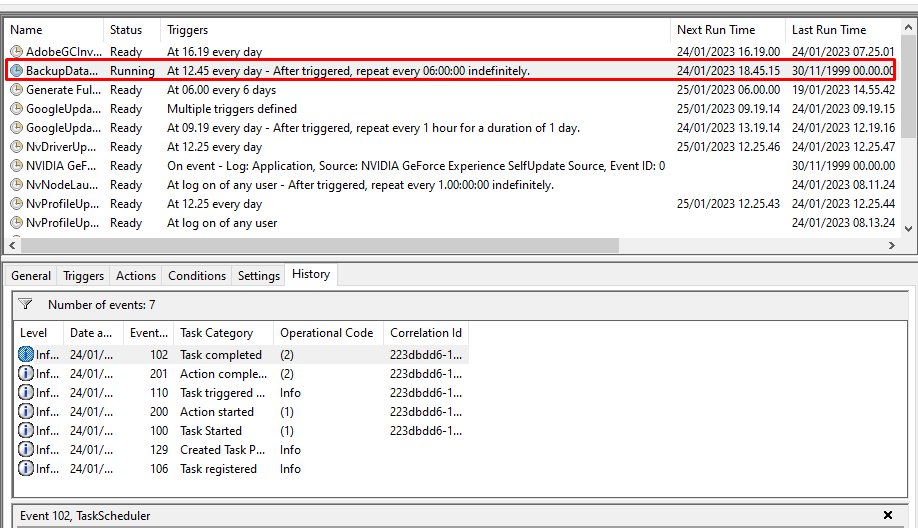
Give name BackupDatabaseEvery6Hours and move to trigger click new. Set task every 6 hours



In action specify the path of the batch file or run.bat file. Then, click OK.



Active tasks will specify the next run time for the backup database task



Task 6:- Create 8 SQL queries which will be used by the Community portal. 2. Import Sample data from CSV file in to the database. 3. Provide evidence of import as part of Project Presentation.

Task Statement:

Create query for the Community Portal and import sample data from CSV file

Solution

1. 8 queries

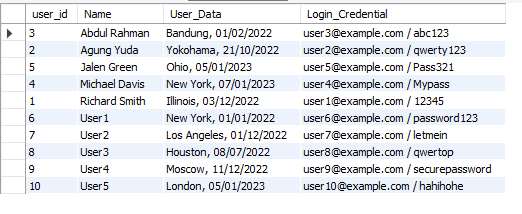
5 useful queries to develop the application

|  |  |  |  |
| --- | --- | --- | --- |
| Page | Tables | Function | Query |
| Registration Page | user | Create a new user | INSERT INTO `community\_portal`.`user` (`user\_id`, `user\_email`, `user\_name`, `location`, `user\_password`, `date\_register`) VALUES ('1', ‘some email’, 'Full name', 'location', 'password', 'date register'); |
| Login Page | user | Validate login Credential | SELECT email\_adress, password FROM user WHERE email = “some email” AND password = “something” |
| Edit Profile | user | Update Profile | UPDATE user SET user\_name = "someone", location = "something" WHERE user\_id = 1; |
| Message | Messages | Send a message between the user | INSERT INTO `community\_portal`.`messages` (`messages\_id`, `date\_send`, `content`, `receiver\_id`, `sender\_id`, `user\_id`) VALUES ('1', ’date send the message’, ‘something, '2', '1', '2'); |
| admin | bulk\_email | Send bulk email | INSERT INTO `community\_portal`.`bulk\_email` (`email\_id`, `content`, `admin\_id`) VALUES ('1',write content for bulk email', '5'); |

3 useful queries to meet the management requirements using joins

|  |  |  |  |
| --- | --- | --- | --- |
| No | Note | Query | Evidence |
| 1 | All users contact info.  To fetch users, contact info information | SELECT user\_id, user\_name AS Name, CONCAT(location,", ",date\_register) AS User\_Data, concat(user\_email, " / ", user\_password) AS Login\_Credential  FROM user  ORDER BY Name; | Report users contact log |
| 2 | All jobs description info, To fetch job type information | SELECT j.job\_id, concat(j.company\_id, " / ",j.company\_name, " / ", jl.location) AS Company, concat(j.job\_title," / ",j.job\_desc) AS Description  FROM job as j JOIN job\_location as jl ON j.job\_id = jl.job\_id  ORDER BY Job\_Desc; | Report jobs description log |
| 3 | All user that registered in 2022 | SELECT date\_register, concat(user\_id,' / ',user\_name) AS User\_data, location AS City  From user where date\_register between '01/01/2022' and '31/12/2022'  ORDER BY date\_register; | Report of User that registered in 2022 |

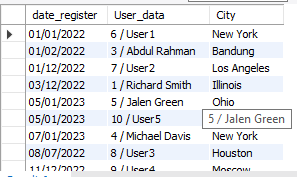
Report users contact log



Report jobs description log

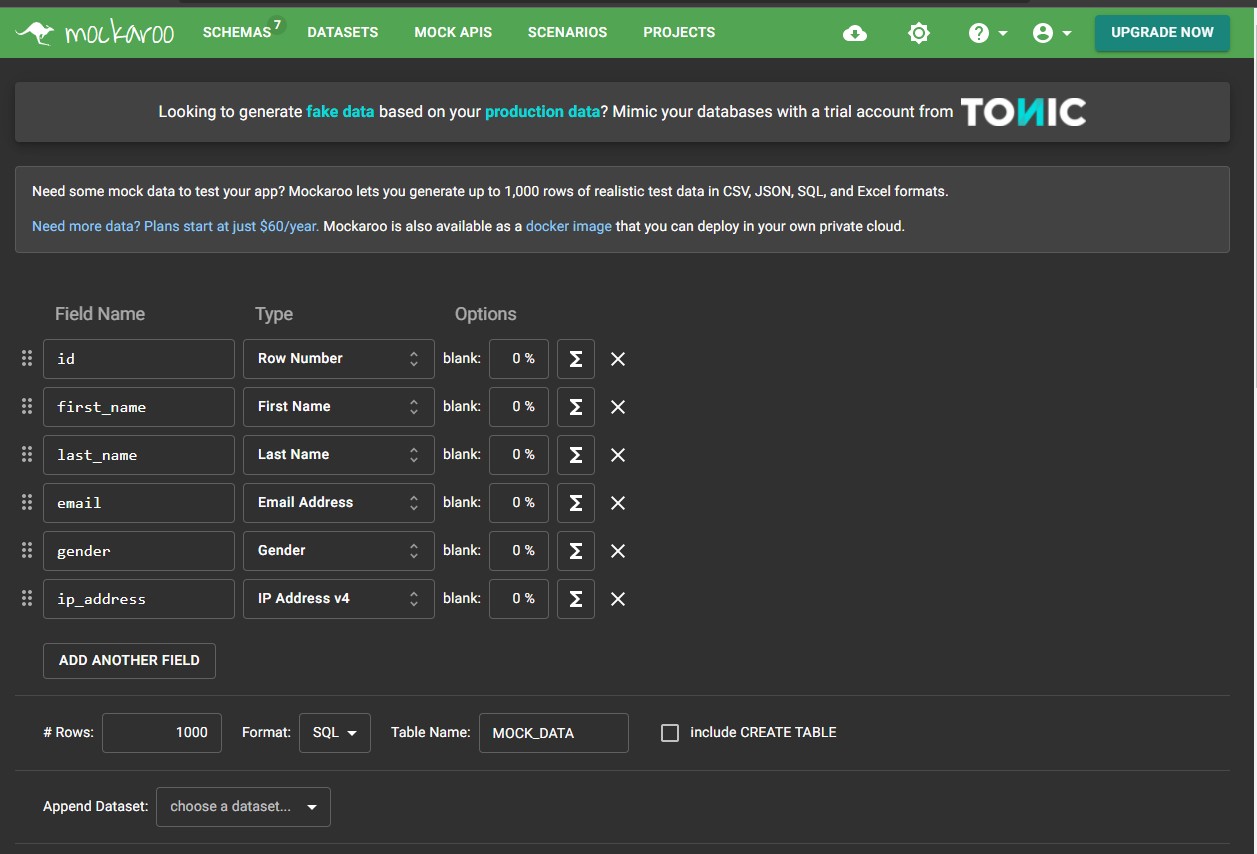


Report of User that registered in 2022

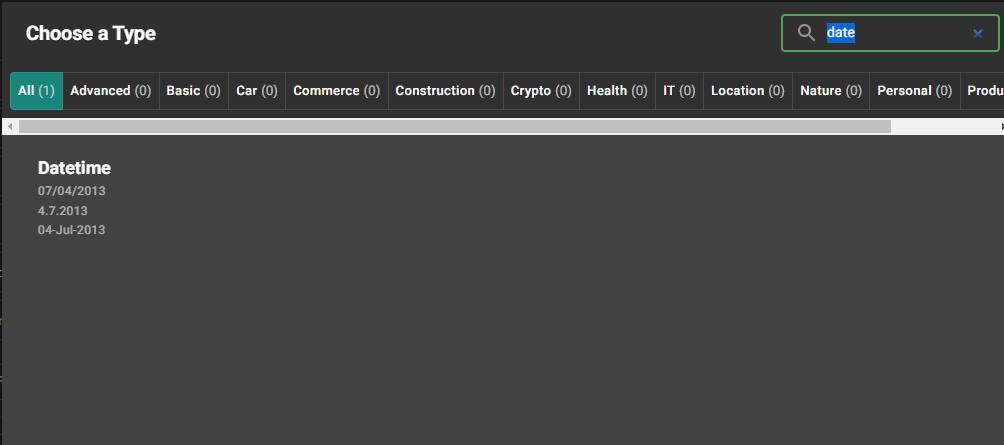


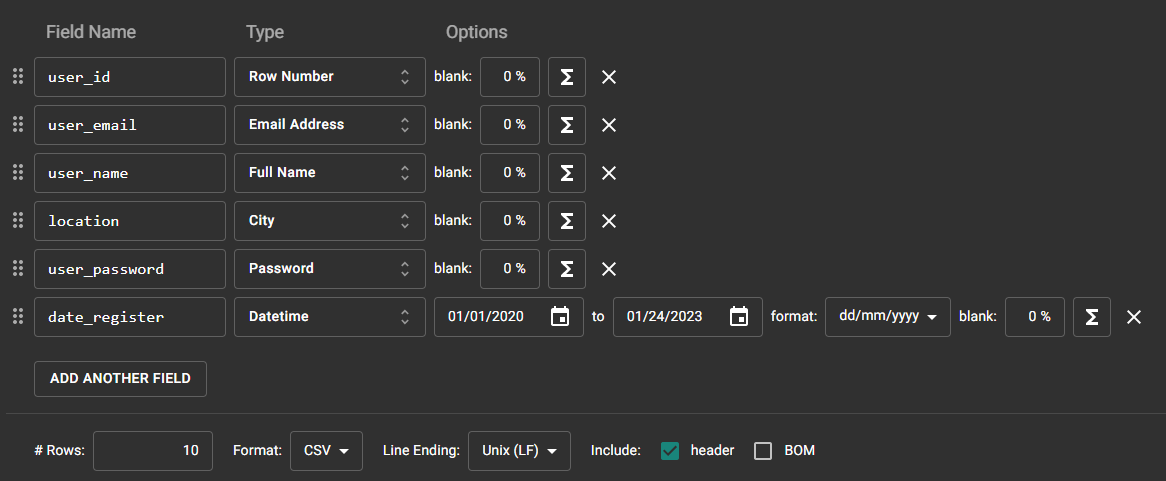
1. Steps to import CSV files

Open mockaroo.com

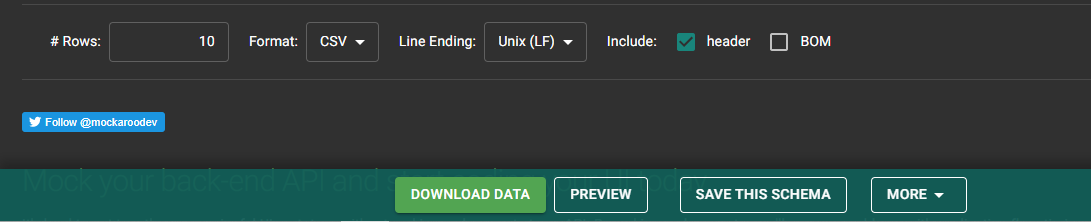
1. 

Edit the field as the requirements needed by click the X sign to delete the field, “Add another Filed” to add a new field. Change the name of the field, make it exactly the same just like your column table And choose the type of the field. When you click the “Choose Data Type” from the data type section at there can search and choose the data type that close enough to the sample data.

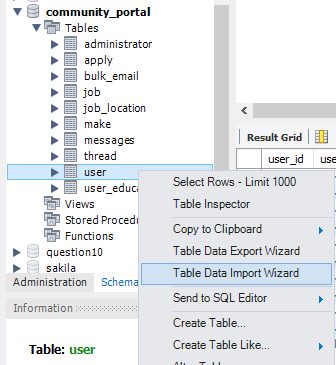


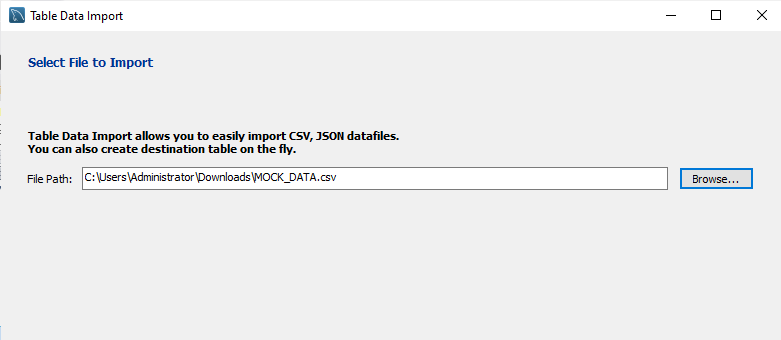


After that, set number of random sample data that you want to make and choose the format of the file to CSV and click donwload data

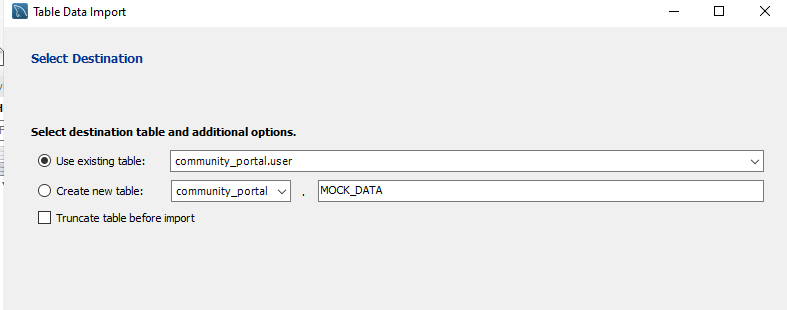


Open workbench and choose the database, and click in the table that you want to import

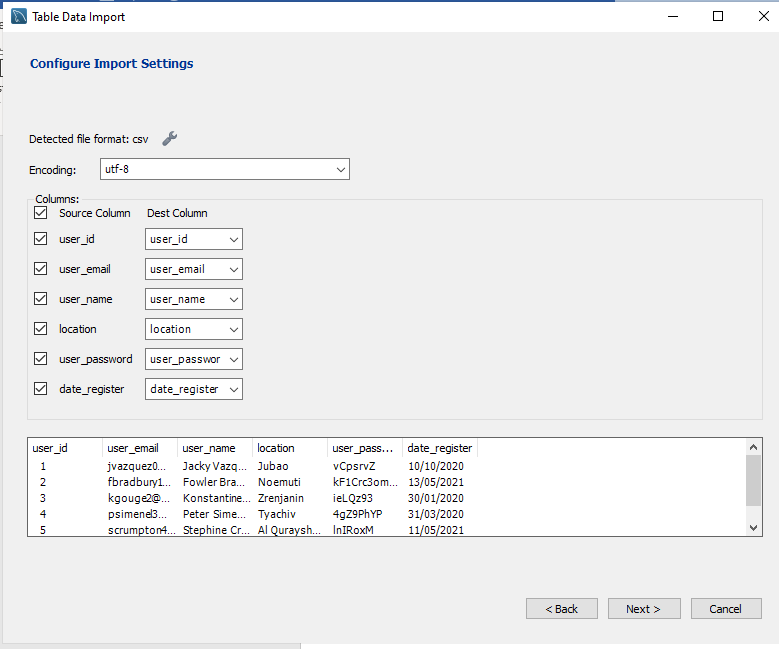




Click next



Click next again



And the data succesfully import

