TASK MANAGEMENT SYSTEM

Database Management Systems

MSCS 542L

Team Name: The Unstoppables

Marist College.

School of Computer Science and Mathematics.

Submitted to:

Dr Reza Sadeghi

Fall 2022

TABLE OF CONTENTS

TABLE OF FIGURES

Desired Group:

Team Name: The Unstoppables

Team Members:

Yeshwanthy Puppala ....yeshwanthy.puppala1@marist.edu (Team Head)

Natraj Adepu.. natraj.adepu1@marist.edu (Team Member)

Chandra Shekar Reddy Ganna .... chandrashekarreddy.ganna1@marist.edu (Team Member)

Venkatesh Pendli ....venkateshwarlu.pendli1@marist.edu (Team Member)

Akhil Sai Baru ....akhilsai.baru1@marist.edu (Team Member)

Thirumala Rao ....Thirumalarao.yellisetti1@marist.edu (Team Member)

Pranay Reddy Kosireddy............................... Pranayreddy.kosireddy1@marist.edu (Team Member)

Description of Team Members:

Yeshwanthy Puppala:

My name is Yeshwanthy Puppala, I am from Hyderabad, India. I completed my Bachelors in the field of Computer Science and Engineering from Sreyas Institute of Engineering and Technology, Hyderabad in the year 2021. My interests are playing badminton, listening to music and podcasts. I worked as a Programmer Analyst Trainee for 9 months in Cognizant Company in the Automation domain. I would like to work with goal oriented, Optimistic people.

Natraj Adepu :

My name is Natraj Adepu i am from Hyderabad, India. i pursued my bachelors degree in the field of Computer Science & Engineering from lovely professional university which is in Punjab India , later I did masters of business administration from lovely professional university and got placed as a operations intern in start-up company which is in Delhi India but my interest is to work in an IT field so quit the job and started preparation to get job in MNC, I got job in cognizant as a programmer analyst. where is got exposed to git hub, Agile methodology and i got chance to learn advanced python. I want to work with dedicated, enthusiastic and fast learners.

Chandra Shekar Reddy Ganna:

My name is Chandra Shekar Reddy Ganna, I am from Hyderabad, India. I completed my B.Tech in Comp. science from Lovely Professional University, in 2020.my interests are listening to music and reading books. I have 2+ years of experience as a storage QA, I got a chance to work with multiple enterprise data storage technologies like RAID, SAN, DAS and I worked with different file systems like NTFS, NFS, CIFS and I worked with different Operating system like Windows, Linux, VMware ESXi. I tested Enterprise Storage System with different OS and file Systems I have experience working with Python. I would like to work with goal oriented, Optimistic people

Venkatesh Pendli:

My name is Venkateshwarlu pendli, I am from Hyderabad, India. I completed my under graduation in the field of mechanical engineering from SRM University Chennai in the year 2018. I have 3+ years of experience as a Quality analyst in Amazon development center, where I need to analyze the Kindle data which is received from content publishers by using both counter punch and kdp tools.I have some knowledge on java and c programming language. I like to work with people who always tries to learn new things and Optimistic People

Baru Akhil Sai:

My name is Akhil Sai Baru, I'm from Hyderabad, India. I completed my bachelors in the field of computer science from lovely professional university Jalandhar Punjab in the year 2020 my interests are playing chess, reading books and watching movies. I have one and half year experience as a cloud developer I used to work for an us based companies in different technologies. I would like to work with people who drive me forward and people who love to have fun.

Thirumala Rao:

My name is Thirumala Rao Yelisetti, I am from Hyderabad, India. I have completed my under graduation in the field of Electronics and Communication Engineering from Sreenidhi Institute of Science and Technology, Hyderabad in the year 2021.my interests are playing and watching cricket. I have 1 years of experience as a Project Engineer in Wipro. I have some knowledge on java and c programming language. My interests are watching and playing cricket, I like swimming. I like to work with people who always tries to learn new things and having good experience on various fields.

Pranay Reddy Kosireddy:

My name is Pranay Reddy Kosireddy, I am from Hyderabad, India. I completed my Bachelors in the field of Computer science from Osmania University, Hyderabad .my interests are listening to music and watching movies. I would like to work with focused, confident and creative mind people.

PROJECT OBJECTIVE:

Project Title: Task Management System

Summary: A Task Management System (TMS) displays a calendar for the desired week, month, or year. also, TMS organizes personal tasks of different users on a specific day. the users can see their individual calendar data & update them. TMS will store the data of different user types in distinct SQL tables. Task management system at minimum supports the following:

1. Admin user activities

a) Admin user can add and delete users.
b) Admin user can reset the password.
c) Admin user can add normal user to TMS by creating a new username and password. a normal user is not able to define or remove other users.
d) Admin user can remove users from TMS by removing their username, Password and Data.

2. User can be able to:

a) Add a task to TMS. the task contains: title, time, duration, and description. Remove a Task.
b) Edit a task's details.
c) Search will be TMS based on time, title, or duration and list the results on the screen.
d) For instance, it can list all scheduled works for one day.

3. TMS has user friendly GUI for users.

a) It shows a welcome page and provides a menu of all functions to the user in all pages.
b) It illustrates the reports in a tabular form. for instance, it displays a well-organized calendar of every month, or year.

REVIEW RELATED WORK:

There are many Task management Systems online. Wrike is one among them. Wrike provide features like Task Tracking, Task Removal, Task assigning but using Wrike we cannot update the task, Here the only way to update the task is to remove the task and then assign the new task. Wrike software is very complex to use, an ordinary person cannot use this software that easily and moreover Wrike software is a using an outdated way to track the task status. Wrike software is not using notification system to notify the person regarding their Task, this may result in missing the task deadline.[2]

Monday.com is a Task Management Software which is customizable. Monday.com can obtain calendars and other task data at a glance.Monday.com automatically updates timeline view when a task is assigned or updated or edited and, we can set priorities to tasks. But Monday.com is not providing a better notification system and the issue in Monday.com is Task Overlapping can occur. That means admin can assign a task to the user who is already working on some task at the same time.[3]

Time Tap is Task scheduling software. Time Tap software is providing a user-friendly interface. It can detect time zones automatically. It can integrate with Google Calendar. It has automated email system to remind about task related notifications. We can also assign a task to employee by checking if he is not allotted with any task. Task pending notification feature is not available in Time Tap software[3]

MERITS:

- Notification Functionality is used to notify about the task that is assigned to person.
- It helps to track all the assigned and updated tasks information instantly.
- It makes easier by displaying a calendar for the desired week, month, or year and organizes personal tasks of different users on a specific day.
- It ensures customers satisfaction.
- Task Management System provides the feature of task status review.
- Users can also Update the Task which is already Assigned.

GITHUB REPOSITORY ADDRESS:

https://github.com/yeshuuu/MSCS-542L_TASK-MANAGEMENT- SYSTEM_THE-UNSTOPPABLES#mscs-542l_task-management-system_the- Unstoppables

ENTITY RELATIONSHIP MODEL:

- 11 entities are chosen to establish a perfect relationship.
- These 11 entities include the task date, the task manager, the task category, the task status report, the admin, the clients, the task, the task employee, the task employee has a task, the task update, and the notifications.
- Relationship between task and manager is M:1, for clients and task M: N, for task and task category 1:1, employee and task relationship are N:1. task and task update the relationship is 1:1. These are all related to their fellow entities bringing about a perfect collaboration. There is a one-to-one, one-to-many, and many-to-many type of relationship.
- There are also a variety of attributes including names, specifications, data, passwords, user IDs, addresses, updated details, and many more.
- There is a one-to- many relationship that is experienced to the highest level among one-to-one and many-to-many relationships.

Entity	**Attributes**

admin	Aid(int)(PrimaryKey),Aname(varchar),
	Contact(varchar),Address(varchar)
	Username(varchar), Password(varchar)

Notification	Notification id(int)) (Primary key) T_id(int) (Foreign key), Notification name(varchar), Aid(int) (Foreign key), description(varchar)
task	T _id(int))(Primary key) Task name(varchar) Date id(int)(Foreignkey),Aid(int)(Foreignkey) status(varchar),Manager_id(int)(Foreignkey) Cid(int) (Foreign key)
Task employee	Eid(int))(Primarykey),Ename(varchar) email(varchar),username(varchar) password(varchar) contact(varchar)
Task category	Task_cat_id(int))(Primarykey),Category name(varchar), Aid(int) (Foreign key), Manager id(int) (Foreign key), T _ id((int) (Foreign key)
Task date	Date id(int)) (Primary key), T _id(int) (Foreign key), Cid(int) (Foreign key),Manager id(int) (Foreign key),Date(date),Eid(int)(Foreign key), Task cat id(int) (Foreign key)
Task manager	Manager_id(int))(Primarykey),Manager name(varchar),Specification(varchar), Experience(varchar), Date of joining(date), Date of joining(date),Username(varchar),

	Password(varchar)
Task status review	Task status id(int)) (Primary key), tid(int) (Foreign key), Task cat id(int) (Foreign key),cid(int) (Foreign key), Manager id(int) (Foreign key) , Task review(varchar), Date id(int) (Foreign key)
Task update	Task update id(int)) (Primary key), tid(int)(Foreign key), eid(int) (Foreign key), Update date(date), Update details(varchar)

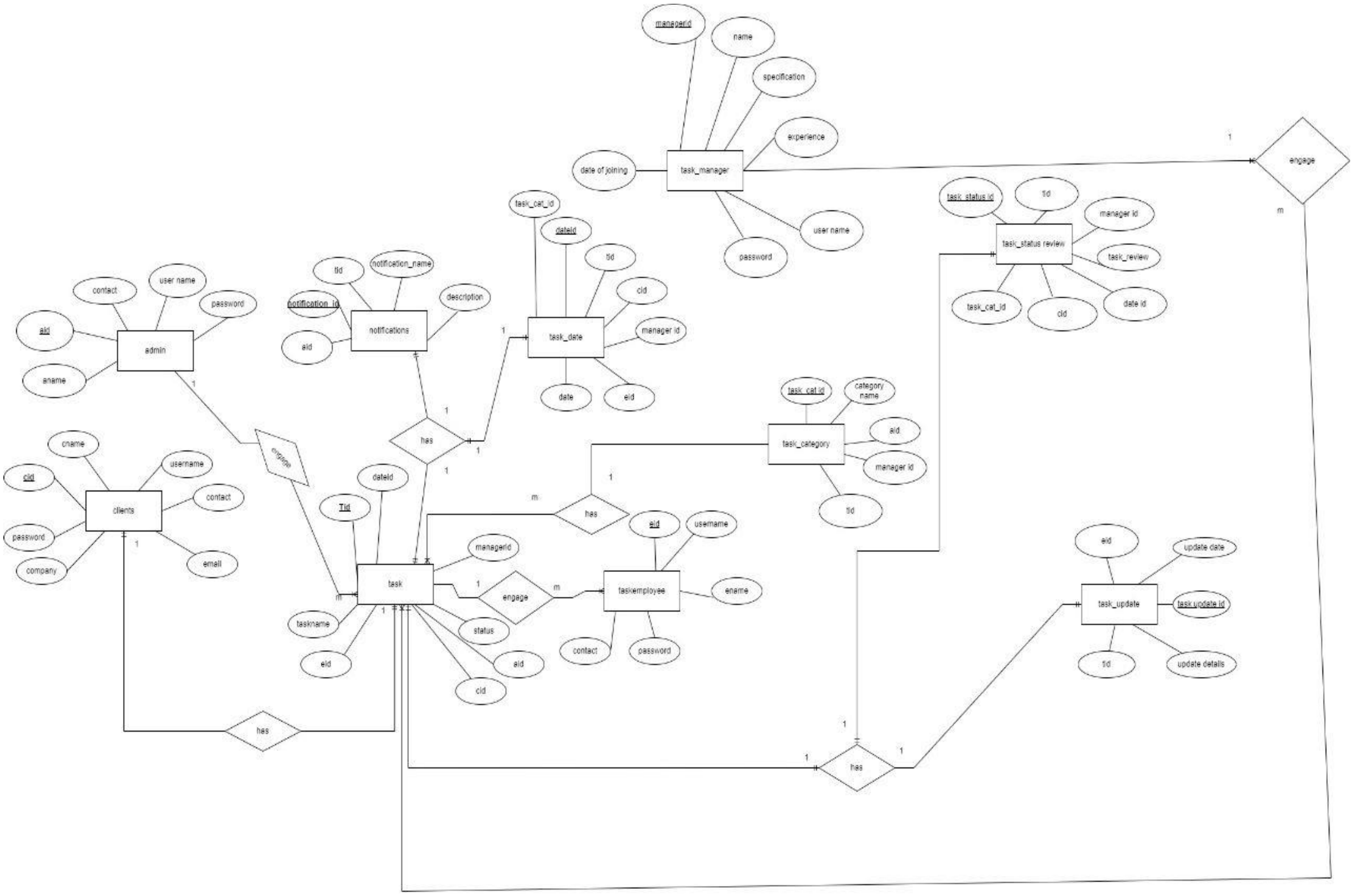
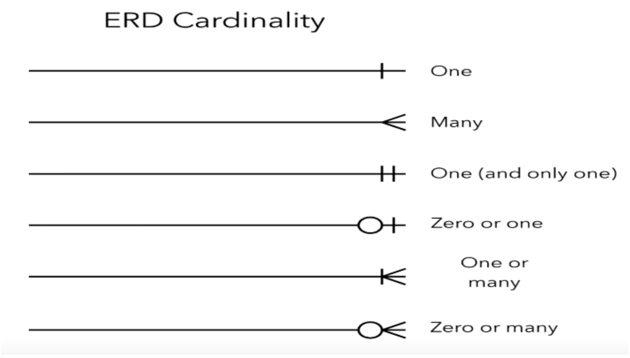


Figure 1: - Entity Relationship Model

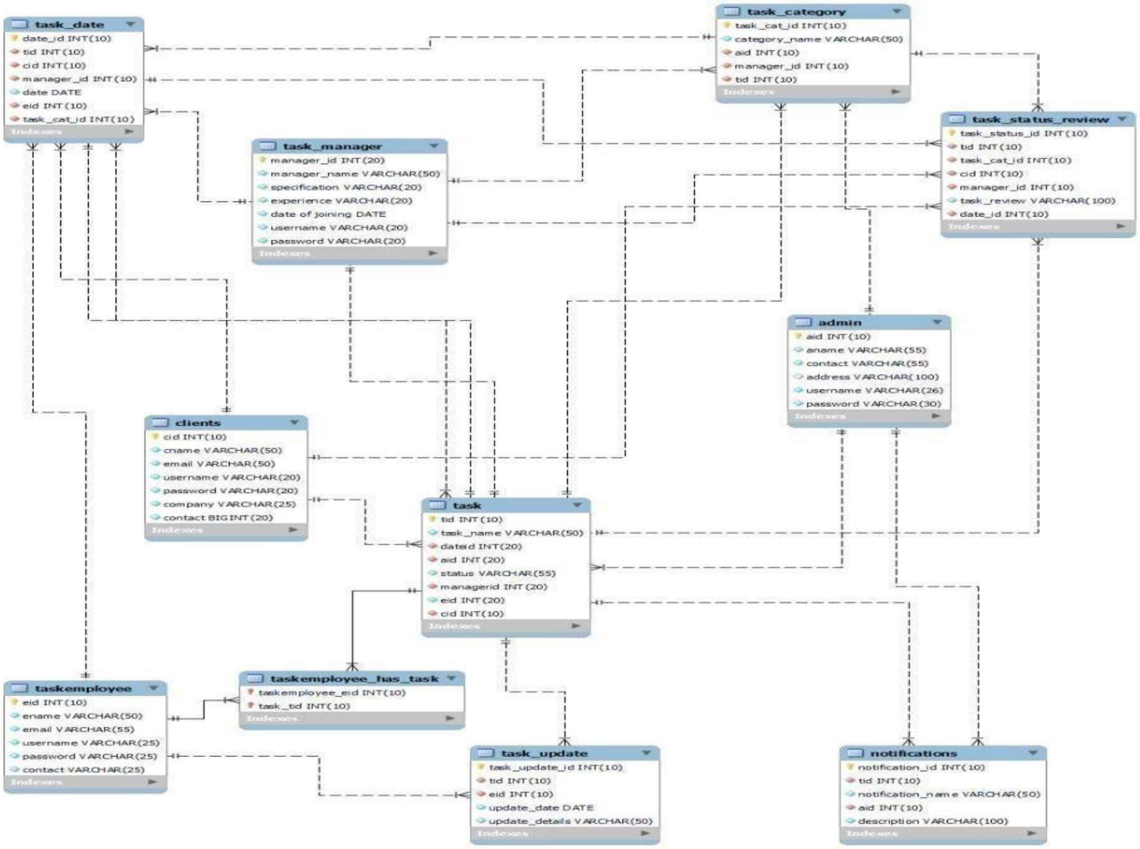
The Entity Relationship Diagram Cardinality for Task Management System is about the number of interactions between entities.



- One- to one: - One entity shares one event with another entity in a one-to-one (1:1) relationship.
- One-to-many: -In one-to-many (1: N) relationship, one entity only experiences an event once, but the other entity may experience the same event multiple times.
- Many-to-many: - When the same event or interaction occurs more than once between both entities, it is known as a many-to-many relationship.

ENHANCED ENTITY-RELATIONSHIP MODEL:

- We are treating all these relationships within this model. If suppose, the task date is the primary key, the other entities and attributes are linked to it. All the attributes also have a relationship that they share between each of them.
- For the admin table **aid** is primary key, for task manager **manager_id** is primary key. For task category **task_cat_id** is primary key. Aid and **manager_id** is foreign key. For task table **tid** is primary key and **cid, aid, manager id** is foreign key. For task status review **task status id** is primary key. **Date id, manager id, cid** is foreign key. Task employee table has **eid** is primary key. Clients table has **cid** primary key.
- This is not just about the entity base relationship, but it is also about the interconnectedness. The data type that comes along with these includes the basic context of the specification, manager id, task review, address, password, and many more.



- These include integers, Boolean values, and many more.

Figure2: - EER Model

DATABASE DEVELOPMENT

ADMIN TABLE: -

- The code created using DDL command with table name “ADMIN”.
- The admin table includes six attributes where aid is a primary key with the datatype of INT (10) and stated as NOT NULL so that this field cannot be null. The attributes aname, contact, username, password is declared using VARCHAR datatypes with NOT NULL and address is given as DEFAULT NULL.

```
CREATE TABLE IF NOT EXISTS `task_management`.`admin` (

  `aid` INT (10) NOT NULL,

  `aname` VARCHAR (55) NOT NULL,

  `contact` VARCHAR (55) NOT NULL,

  `address` VARCHAR (100) NULL DEFAULT NULL,

  `username` VARCHAR (26) NOT NULL,

  `password` VARCHAR (30) NOT NULL,

  PRIMARY KEY (`aid`));
```

CLIENTS TABLE: -

- The code created using DDL command with table name “ADMIN”.
- The admin table includes seven attributes where cid is a primary key with the datatype of INT and stated as NOT NULL so that this field cannot be null. The attributes cname, email, username, password, company is declared using VARCHAR datatypes with NOT NULL and contact with a datatype BIGINT with NOT NULL.

```
CREATE TABLE IF NOT EXISTS `task_management`.`clients` (
  `cid` INT (10) NOT NULL,
  `cname` VARCHAR (50) NOT NULL,
  `email` VARCHAR (50) NOT NULL,
  `username` VARCHAR (20) NOT NULL,
  `password` VARCHAR (20) NOT NULL,
  `company` VARCHAR (25) NOT NULL,
  `contact` BIGINT (20) NOT NULL,
  PRIMARY KEY (`cid`));
```

TASK_MANAGER TABLE: -

- The code created using DDL command with table name “task_manager”.
- The admin table includes seven attributes where manager_id with the datatype of INT and stated as NOT NULL so that this field cannot be null and also attributes manager_name, specification, experience, username, password is declared using VARCHAR datatype with NOT NULL. Date of joining datatype is DATE which indicates the date format.

```
CREATE TABLE IF NOT EXISTS `task_management`. ` task_manager` (

  `manager_id` INT (20) NOT NULL,

  `manager_name` VARCHAR (50) NOT NULL,

  `specification` VARCHAR (20) NOT NULL,

  `experience` VARCHAR (20) NOT NULL,

  `date of joining` DATE NOT NULL,

  `username` VARCHAR (20) NOT NULL,

  `password` VARCHAR (20) NOT NULL,

PRIMARY KEY (`manager_id`)) ;
```

TASKEMPLOYEE TABLE: -

- The code created using DDL command with table name “taskemployee”.
- The admin table includes six attributes where eid is a primary key with the datatype of INT and stated as NOT NULL so that this field cannot be null. The attributes ename, email, username, password, contact is declared using VARCHAR datatypes with NOT NULL.

```
CREATE TABLE IF NOT EXISTS `task_management`.`taskemployee` (

  `eid` INT(10) NOT NULL,

  `ename` VARCHAR(50) NOT NULL,

  `email` VARCHAR(55) NOT NULL,

  `username` VARCHAR(25) NOT NULL,
  `password` VARCHAR(25) NOT NULL,

  `contact` VARCHAR(25) NOT NULL,

  PRIMARY KEY (`eid`));
```

TASK_CATEGORY TABLE: -

- The code created using DDL command with table name “task_category”.
- The admin table includes five attributes where task_cat_id is a primary key with the datatype of INT and stated as NOT NULL so that this field cannot be null. The attributes, aid, manager_id, tid, is declared using INT and category_name as VARCHAR datatypes with NOT NULL. Manager_id is a foreign key which is reference to the table task_manager (`manager_id`), tid is a foreign key which is reference to the table task (`tid`) and aid is a foreign key which is reference to the table admin (`aid`) where NO ACTION are performed on it.

```
CREATE TABLE IF NOT EXISTS `task_management`.`task_category` (

  `task_cat_id` INT(10) NOT NULL,

  `category_name` VARCHAR(50) NOT NULL,

  `aid` INT(10) NOT NULL,

  `manager_id` INT(10) NOT NULL,

  `tid` INT(10) NOT NULL,

  PRIMARY KEY (`task_cat_id`),

  INDEX `task_fk1_idx` (`manager_id` ASC) VISIBLE,

  INDEX `task_fk2_idx` (`tid` ASC) VISIBLE,

  INDEX `task_fk3_idx` (`aid` ASC) VISIBLE,

  CONSTRAINT `task_fk1`

    FOREIGN KEY (`manager_id`)

    REFERENCES `task_management`. `task_manager` (`manager_id`)

    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `task_fk2`
    FOREIGN KEY (`tid`)
    REFERENCES `task_management`.`task` (`tid`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `task_fk3`
    FOREIGN KEY (`aid`)
    REFERENCES `task_management`.`admin` (`aid`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);
```

TASK_DATE TABLE: -

- The code created using DDL command with table name “task_date”.
- The admin table includes seven attributes where date_id is a primary key with the datatype of INT and stated as NOT NULL so that this field cannot be null. The attributes tid, cid, manager_id, eid is declared using INT and date using DATE datatypes with NOT NULL. Cid is a foreign key which is reference to the table clients(`cid), manager_id is a foreign key which is reference to the table task_manager (`manager_id`), .task_cat_id is a foreign key which is reference to the table task_category (`task_cat_id`) and tid is a foreign key which is reference to the table task (`tid`) where NO ACTION are performed on it.

```
CREATE TABLE IF NOT EXISTS `task_management`.`task_date` (
  `date_id` INT(10) NOT NULL,
  `tid` INT(10) NOT NULL,
  `cid` INT(10) NOT NULL,
  `manager_id` INT(10) NOT NULL,
  `date` DATE NOT NULL,
  `eid` INT(10) NOT NULL,

  `task_cat_id` INT(10) NOT NULL,

  PRIMARY KEY (`date_id`),

  INDEX `date_fk1_idx` (`cid` ASC) VISIBLE,

  INDEX `date_fk2_idx` (`manager_id` ASC) VISIBLE,

  INDEX `date_fk3_idx` (`eid` ASC) VISIBLE,

  INDEX `date_fk4_idx` (`task_cat_id` ASC) VISIBLE,

  INDEX `date_fk5_idx` (`tid` ASC) VISIBLE,

  CONSTRAINT `date_fk1`

    FOREIGN KEY (`cid`)

    REFERENCES `task_management`.`clients` (`cid`)

    ON DELETE NO ACTION

    ON UPDATE NO ACTION,

  CONSTRAINT `date_fk2`

    FOREIGN KEY (`manager_id`)

    REFERENCES `task_management`.`task_manager` (`manager_id`)

    ON DELETE NO ACTION

    ON UPDATE NO ACTION,

  CONSTRAINT `date_fk3`

    FOREIGN KEY (`eid`)

    REFERENCES `task_management`.`taskemployee` (`eid`)

    ON DELETE NO ACTION

    ON UPDATE NO ACTION,

  CONSTRAINT `date_fk4`

    FOREIGN KEY (`task_cat_id`)
    REFERENCES `task_management`.`task_category` (`task_cat_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `date_fk5`
    FOREIGN KEY (`tid`)
    REFERENCES `task_management`.`task` (`tid`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);
```

TASK TABLE: -

- The code created using DDL command with table name “task”.
- The admin table includes six attributes where tid is a primary key with the datatype of INT and stated as NOT NULL so that this field cannot be null. The attributes task_name, date_id, aid, status, manager_id, eid, cid is declared using INT and task_name, status is declared with using VARCHAR datatypes with NOT NULL. Aid is a foreign key which is reference to the table admin (`aid`), manager_id is a foreign key which is reference to the table task_manager (`manager_id`), cid is a foreign key which is reference to the table clients (`cid`) and Date_id is a foreign key which is reference to the table task_date (`date_id`) where NO ACTION is performed on it.

```
CREATE TABLE IF NOT EXISTS `task_management`.`task` (
 `tid` INT(10) NOT NULL,
 `task_name` VARCHAR(50) NOT NULL,
 `dateid` INT(20) NOT NULL,
 `aid` INT(20) NOT NULL,
 `status` VARCHAR(55) NOT NULL,
 `managerid` INT(20) NOT NULL,
 `eid` INT(20) NOT NULL,
 `cid` INT(10) NOT NULL,
 PRIMARY KEY (`tid`),
 INDEX `fk1_idx` (`aid` ASC) VISIBLE,
 INDEX `fk2_idx` (`managerid` ASC) VISIBLE,
 INDEX `fk4_idx` (`cid` ASC) VISIBLE,
 INDEX `fk5_idx` (`dateid` ASC) VISIBLE,
 CONSTRAINT `fk1`
   FOREIGN KEY (`aid`)
   REFERENCES `task_management`.`admin` (`aid`)
   ON DELETE NO ACTION
   ON UPDATE NO ACTION,
 CONSTRAINT `fk2`
   FOREIGN KEY (`managerid`)
   REFERENCES `task_management`.`task_manager` (`manager_id`)
   ON DELETE NO ACTION
   ON UPDATE NO ACTION,
 CONSTRAINT `fk4`
 FOREIGN KEY (`cid`)
   REFERENCES `task_management`.`clients` (`cid`)
   ON DELETE NO ACTION
   ON UPDATE NO ACTION,
 CONSTRAINT `fk5`
   FOREIGN KEY (`dateid`)
   REFERENCES `task_management`.`task_date` (`date_id`)
   ON DELETE NO ACTION
   ON UPDATE NO ACTION);
```

NOTIFICATION TABLE: -

- The code created using DDL command with table name “notifications”.
- The admin table includes five attributes where notification_id is a primary key with the datatype of INT and stated as NOT NULL so that this field cannot be null. The attributes tid, aid is declared using INT and notification_name, description are declared as VARCHAR datatypes with NOT NULL. Aid is a foreign key which is reference to the table admin (`aid`), tid is a foreign key which is reference to the table task (`tid`) where NO ACTION is performed on it.

```
CREATE TABLE IF NOT EXISTS `task_management`.`notifications` (
  `notification_id` INT(10) NOT NULL,
  `tid` INT(10) NOT NULL,
  `notification_name` VARCHAR(50) NOT NULL,
  `aid` INT(10) NOT NULL,
  `description` VARCHAR(100) NOT NULL,
  PRIMARY KEY (`notification_id`),
  INDEX `notification_fk1_idx` (`aid` ASC) VISIBLE,
  INDEX `notification_fk2_idx` (`tid` ASC) VISIBLE,
  CONSTRAINT `notification_fk1`
    FOREIGN KEY (`aid`)
    REFERENCES `task_management`.`admin` (`aid`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `notification_fk2`
    FOREIGN KEY (`tid`)
    REFERENCES `task_management`.`task` (`tid`)
    ON DELETE NO ACTION
  ON UPDATE NO ACTION);
```

TASK_STATUS _REVIEW TABLE: -

- The code created using DDL command with table name “task_status_review”.
- The admin table includes seven attributes where task_status_id is a primary key with the datatype of INT and stated as NOT NULL so that this field cannot be null. The attributes tid, task_cat_id, cid, manager_id, date_id is declared as INT and task_review declared using VARCHAR datatypes with NOT NULL. Cid is a foreign key which is reference to the table clients (`cid), Manager_id is a foreign key which is reference to the table task_manager (`manager_id`), date_id is a foreign key which is reference to the table task_date (`date_id`),Tid is a foreign key which is reference to the table task (`tid`) and Task_cat_id is a foreign key which is reference to the table task_category (`task_cat_id `) where NO ACTION is performed on it.

```
CREATE TABLE IF NOT EXISTS `task_management`.`task_status_review` (
 `task_status_id` INT(10) NOT NULL,
 `tid` INT(10) NOT NULL,
 `task_cat_id` INT(10) NOT NULL,
 `cid` INT(10) NOT NULL,
 `manager_id` INT(10) NOT NULL,
 `task_review` VARCHAR(100) NOT NULL,
 `date_id` INT(10) NOT NULL,
 PRIMARY KEY (`task_status_id`),
 INDEX `status_fk1_idx` (`cid` ASC) VISIBLE,
 INDEX `status_fk2_idx` (`manager_id` ASC) VISIBLE,
 INDEX `status_fk3_idx` (`date_id` ASC) VISIBLE,
 INDEX `status_fk4_idx` (`tid` ASC) VISIBLE,
 INDEX `status_fk5_idx` (`task_cat_id` ASC) VISIBLE,
 CONSTRAINT `status_fk1`
   FOREIGN KEY (`cid`)
```

```
  REFERENCES `task_management`.`clients` (`cid`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `status_fk2`
  FOREIGN KEY (`manager_id`)
  REFERENCES `task_management`.`task_manager` (`manager_id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `status_fk3`
  FOREIGN KEY (`date_id`)
  REFERENCES `task_management`.`task_date` (`date_id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `status_fk4`
  FOREIGN KEY (`tid`)
  REFERENCES `task_management`.`task` (`tid`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `status_fk5`
  FOREIGN KEY (`task_cat_id`)
  REFERENCES `task_management`.`task_category` (`task_cat_id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION);
```

TASK_UPDATE TABLE: -

- The code created using DDL command with table name “task_update”.
- The admin table includes six attributes where task_update_id is a primary key with the datatype of INT and stated as NOT NULL so that this field cannot be null. The attributes tid, eid are declared using INT, update_date is declared as DATE datatype and update_details declared as VARCHAR datatypes with NOT NULL. Eid is a foreign key which is reference to the table taskemployee (`eid `) and Tid is a foreign key which is reference to the table task (`tid `) where NO ACTION is performed on it.

```
CREATE TABLE IF NOT EXISTS `task_management`.`task_update` (

 `task_update_id` INT(10) NOT NULL,

 `tid` INT(10) NOT NULL,

 `eid` INT(10) NOT NULL,


 `update_date` DATE NOT NULL,

 `update_details` VARCHAR(50) NOT NULL,

 PRIMARY KEY (`task_update_id`),

 INDEX `update_fk1_idx` (`eid` ASC) VISIBLE,

 INDEX `update_fk2_idx` (`tid` ASC) VISIBLE,

 CONSTRAINT `update_fk1`

  FOREIGN KEY (`eid`)

  REFERENCES `task_management`.`taskemployee` (`eid`)

  ON DELETE NO ACTION

  ON UPDATE NO ACTION,

 CONSTRAINT `update_fk2`

  FOREIGN KEY (`tid`)

  REFERENCES `task_management`.`task` (`tid`)

  ON DELETE NO ACTION

  ON UPDATE NO ACTION);
```

TASKEMPLOYEE_HAS_TASK TABLE: -

- The code created using DDL command with table name “taskemployee_has_task”.
- The admin table includes two attributes where task_employee_eid and task_tid are primary keys with the datatype INT and stated as NOT NULL so that this field cannot be null. Taskemployee_eid is a foreign key which is reference to the table taskemployee (`taskemployee_eid `) and Tid is a foreign key which is reference to the table task (`tid `) where NO ACTION is performed on it.

```
CREATE TABLE IF NOT EXISTS `task_management`. `taskemployee_has_task` (
  `taskemployee_eid` INT (10) NOT NULL,
  `task_tid` INT (10) NOT NULL,
  PRIMARY KEY (`taskemployee_eid`, `task_tid`),
  INDEX `fk_taskemployee_has_task_task1_idx` (`task_tid` ASC) VISIBLE,
  INDEX `fk_taskemployee_has_task_taskemployee1_idx` (`taskemployee_eid` ASC) VISIBLE,
  CONSTRAINT `fk_taskemployee_has_task_taskemployee1`
    FOREIGN KEY (`taskemployee_eid`)
    REFERENCES `task_management`.`taskemployee` (`eid`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_taskemployee_has_task_task1`
    FOREIGN KEY (`task_tid`)
    REFERENCES `task_management`.`task` (`tid`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);
```

DATA INSERTION:

We use INSERT command which comes under Data Manipulation Language to insert one or more rows into a table by giving values.

INSERT:

INSERT INTO `clients`(`cid`, `cname`, `email`, `username`, `password`, `company`, `contact`) VALUES ('1','riju','riju@gmail.com','riju1','riju1','Qublix',673462726),('2','rahul','rahulu@gmail.com','rahul','rahul','Qublix',893583949),('3','pritidas','pdas@gmail.com','priti','priti','Accenture',887368393),('4','Sonam','sonam@gmail.com','sonam1','sonam1','Qublix',9845737832),('5','kasoul','kasoul@gmail.com','kasoul','kasoul1','Kanbilion',998735829),('6','Alex','alex@gmail.com','alex','alex','Qublix',893683683),('7','Anjali','anjali@gmail.com','anjali','anjali1','Qublix',8724638347),('8','Suhana','suhana@gmail.com','suhana1','suhana','Qublix',8765389330),('9','patidalgaikoyad','patidal@gmail.com','patidal1','patidal1','wipro',873936939),('10','Rex leo','rex@wipro.co.com','rex','leo','wipro',8937479379)

INSERT INTO `admin`(`aid`, `aname`, `contact`, `address`, `username`, `password`) VALUES ('1','admin','1234567890','kolkata','admin','admin'),('2','admin2','8663567123','Delhi','admin2','admin2'),('3','admin1','7612345670','Mumbai','admin1','admin1'),('4','admin3','9987123456','kolkata','admin3','admin3'),('5','akash','673658253','kolkata','akash','akash')

INSERT INTO `notifications`(`notification_id`, `tid`, `notification_name`, `aid`, `description`) VALUES ('1','1','a','1','java script project'),('2','2','b','2','. net module '),('3','3','c','3','java script project'),('4','4','d','4','Sap Bi'),('5','5','e','5','java full stack'),('6','6','f','6','Sap Abap'),('7','7','g','7','java script project'),('8','8','h','8','sap mm'),('9','9','i','9','java script project'),('10','10','j','10','java script project')

INSERT INTO `task`(`tid`, `task_name`, `dateid`, `aid`, `status`, `managerid`, `eid`, `cid`) VALUES ('1','client1','1','1','pending','1','2','2'),('2','client2','2','2','running','2','1','3'),('3','client3','3','3','pending','1','3','3'),('4','client4','4','4','submitted','4','4','4'),('5','client5','5','5','pending','5','5','5'),('6','client6','6','6','running','6','5','5'),('7','client7','7','7','pending','7','4','2'),('8','client8','8','8','pending','8','8','8'),('9','client 9','9','9','pending','9','3','2'),('10','client 10','10','10','submitted','10','5','2')

INSERT INTO `taskemployee`(`eid`, `ename`, `email`, `username`, `password`, `contact`) VALUES('1','rahul','rahul@gmail.com','rahul1','rahul',877267393),('2','ritul','ritul@gmail.com','ritul','ritul',9853783783),('3','sohan','sohan@gmail.com','sohan1','sohan',9853783273),('4','Ramesh','ramesh@gmail.com','ramesh','ramesh',9037937843),('5','Akansha','aku@gmail.com','aku1','aku1',9812345678),('6','Anmika','anamika@gmail.com','anamika','anamika',9812378933),('7','anjali','anjali@gmail.com','anu','anjali',8947940749)

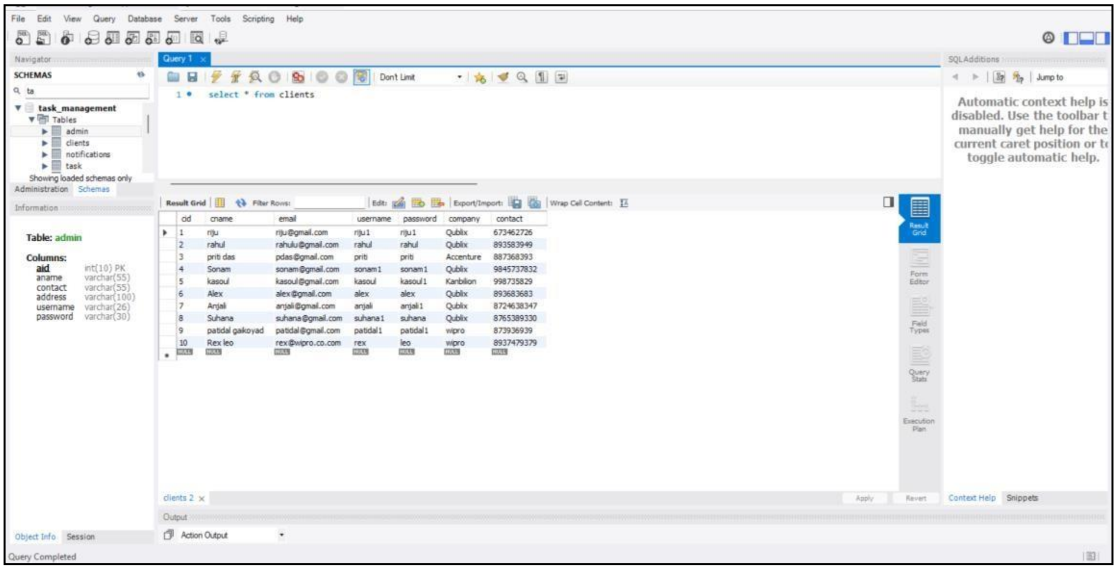
INSERT INTO `task_category`(`task_cat_id`,`category_name`, `aid`, `manager_id`, `tid`) VALUES('1','cat1','1','1','1'),('2','cat2','2','2','2'),('3','cat3','3','3','3'),('4','cat5','4','4','4'),('5','cat4','5','5',' 5'),('6','cat6','6','6','6'),('7','cat7','7','7','7'),('8','cat8','8','8','8'),('9','cat9','9','9','9'),('10','cat10','10','10','10')

INSERT INTO `task_date`(`date_id`, `tid`, `cid`, `manager_id`, `date`, `eid`, `task_cat_id`) VALUES ('1','1','1','1','1','1','1'),('2','2','3','4','7','2','6'),('3','3','3','4','3','3','3'),('4','4','4','4','4','4','4'),('5','5','5','5','5','5','5'),('6','6','6','6','6','6','6'),('7','7','7','7','7','7','7'),('8','8','8','8','8','8','8'),('9','9','9','9','9','9','9'),('10','10','10','10','1','8','8')

INSERT INTO `task_manager`(`manager_id`, `manager_name`, `specification`, `experience`, `date ofjoining`,`username`,`password`)VALUES('1','a','java','7','2010/10/12','man1','password'),('2','b','javascript','4','2010/11/12','admin1','password'),('3','c','php','3','2022/7/7','user2','password'),('4','d','java','5','2016/10/12','man5','password'),('5','e','sql','6','2012/10/12','man9','password'),('6','f','Sap','5','2015/10/12','man9','password');

SET FOREIGN_KEY_CHECKS=0;

Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (`task_management`.`task`, CONSTRAINT `fk1` FOREIGN KEY (`aid`) REFERENCES `admin` (`aid`))



```
select * from clients
```

cid	cname	email	username	password	company	contact
1	riju	riju@gmail.com	riju1	riju1	Qublix	673462726
2	rahul	rahulu@gmail.com	rahul	rahul	Qublix	893583949
3	priti das	pdas@gmail.com	priti	priti	Accenture	887368393
4	Sonam	sonam@gmail.com	sonam1	sonam1	Qublix	9845737832
5	kasoul	kasoul@gmail.com	kasoul	kasoul1	Kanbilion	998735829
6	Alex	alex@gmail.com	alex	alex	Qublix	893683683
7	Anjali	anjali@gmail.com	anjali	anjali1	Qublix	8724638347
8	Suhana	suhana@gmail.com	suhana1	suhana	Qublix	8765389330
9	patidal gaikoyad	patidal@gmail.com	patidal1	patidal1	wipro	873936939
10	Rex leo	rex@wipro.co.com	rex	leo	wipro	8937479379
NULL	NULL	NULL	NULL	NULL	NULL	NULL

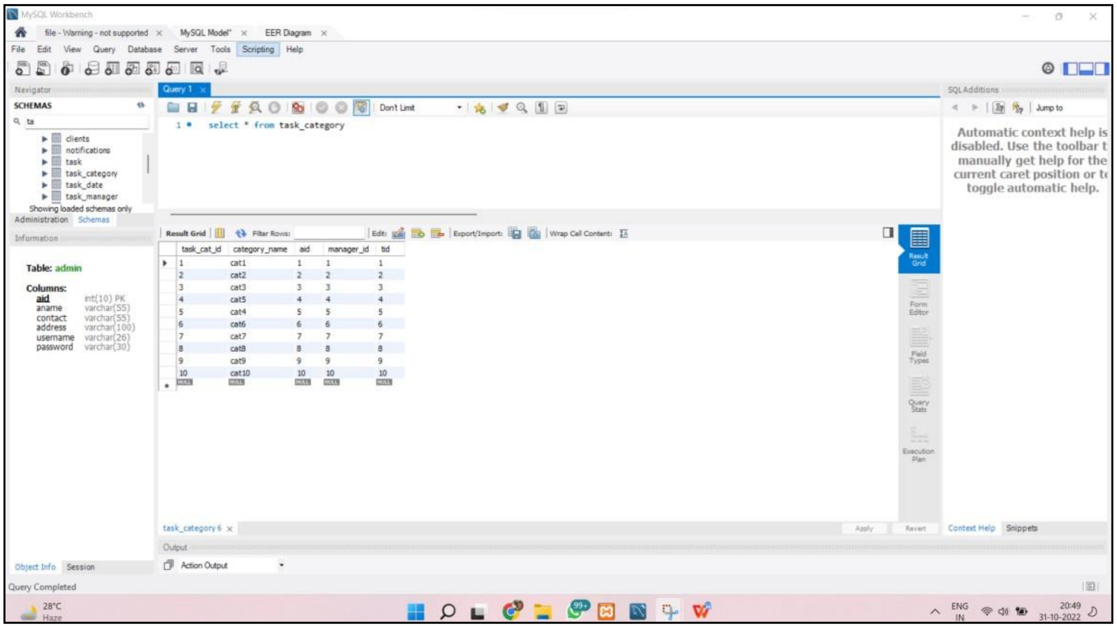
Table: admin

Columns:

aid	int(10) PK
aname	varchar(55)
contact	varchar(55)
address	varchar(100)
username	varchar(26)
password	varchar(30)

Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.

Figure 3: Data inserted query



```
select * from task_category
```

Table: admin

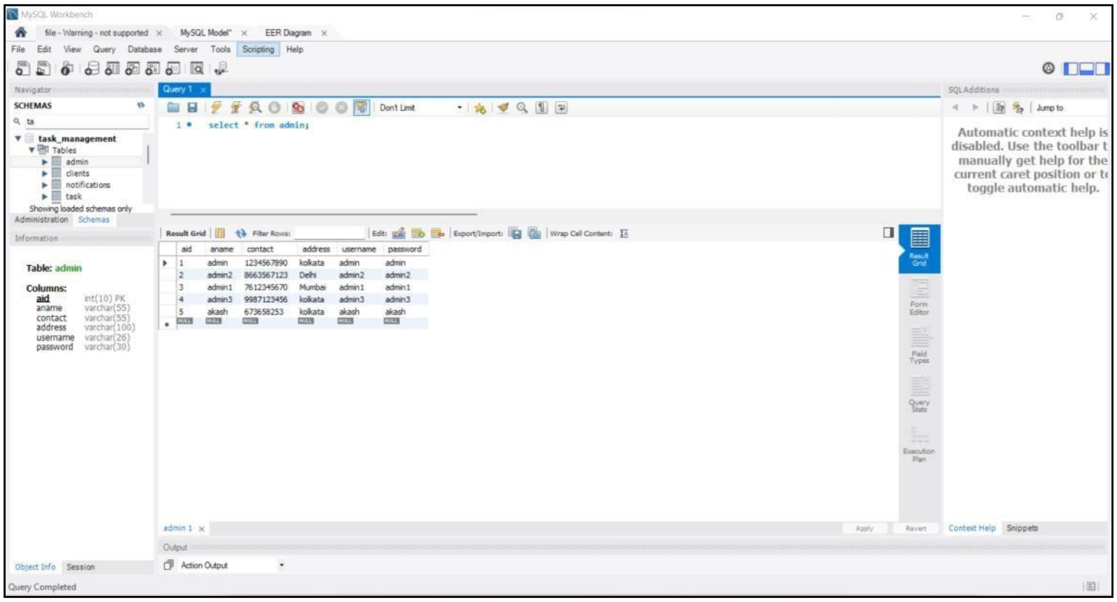
Columns:

aid	int(10) PK
aname	varchar(55)
contact	varchar(55)
address	varchar(100)
username	varchar(26)
password	varchar(30)

task_cat_id	category_name	aid	manager_id	tid
1	cat1	1	1	1
2	cat2	2	2	2
3	cat3	3	3	3
4	cat5	4	4	4
5	cat4	5	5	5
6	cat6	6	6	6
7	cat7	7	7	7
8	cat8	8	8	8
9	cat9	9	9	9
10	cat10	10	10	10
NULL	NULL	NULL	NULL	NULL

Query Completed

Figure 4: Select query for displaying Task _category table details



```
select * from admin;
```

Table: admin

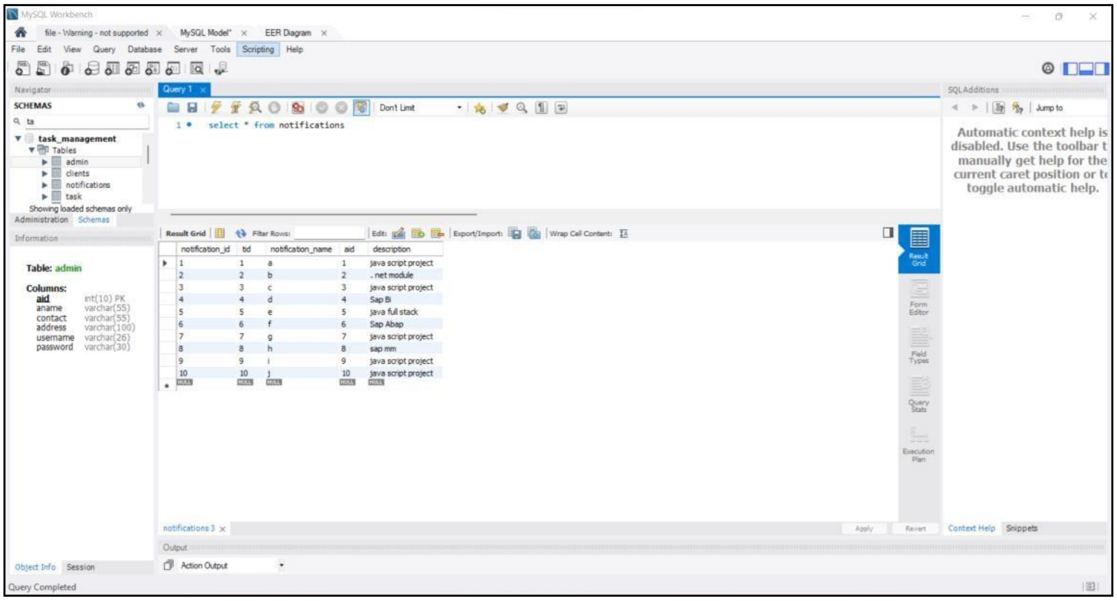
Columns:

aid	int(10) PK
aname	varchar(55)
contact	varchar(55)
address	varchar(100)
username	varchar(26)
password	varchar(30)

aid	aname	contact	address	username	password
1	admin	1234567890	kolkata	admin	admin
2	admin2	8663567123	Delhi	admin2	admin2
3	admin1	7612345670	Mumbai	admin1	admin1
4	admin3	9987123456	kolkata	admin3	admin3
5	akash	673658253	kolkata	akash	akash
NULL	NULL	NULL	NULL	NULL	NULL

Query Completed

Figure 5: Select query for displaying admin table details



```
select * from notifications
```

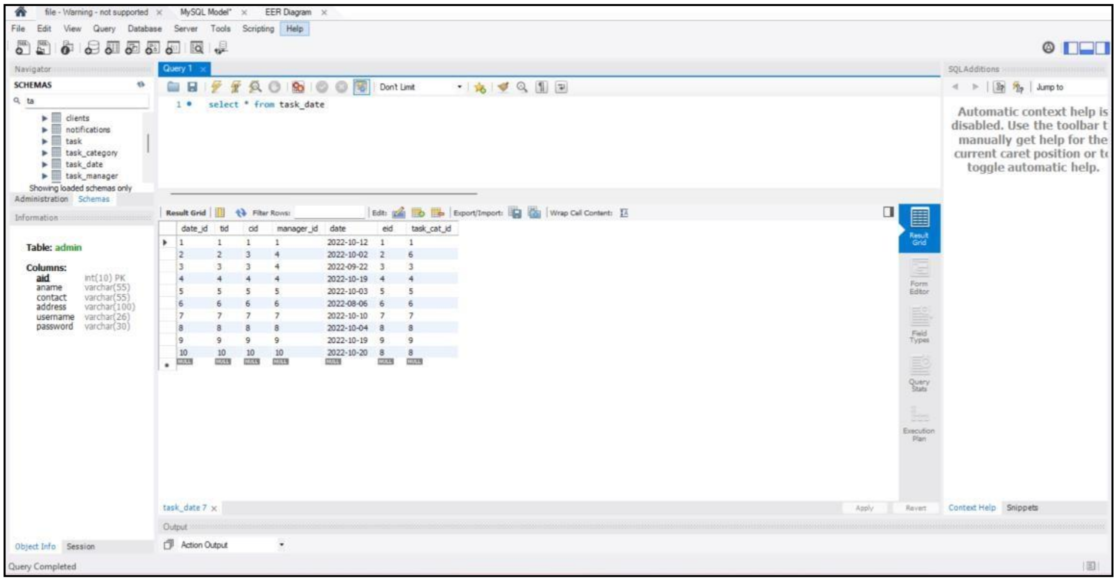
Table: admin

Columns:

aid	int(10) PK
aname	varchar(55)
contact	varchar(55)
address	varchar(100)
username	varchar(26)
password	varchar(30)

notification_id	tid	notification_name	aid	description
1	1	a	1	java script project
2	2	b	2	. net module
3	3	c	3	java script project
4	4	d	4	Sap Bi
5	5	e	5	java full stack
6	6	f	6	Sap Abap
7	7	g	7	java script project
8	8	h	8	sap mm
9	9	i	9	java script project
10	10	j	10	java script project
NULL	NULL	NULL	NULL	NULL

Figure 6: Select query for displaying notifications table details



```
select * from task_date
```

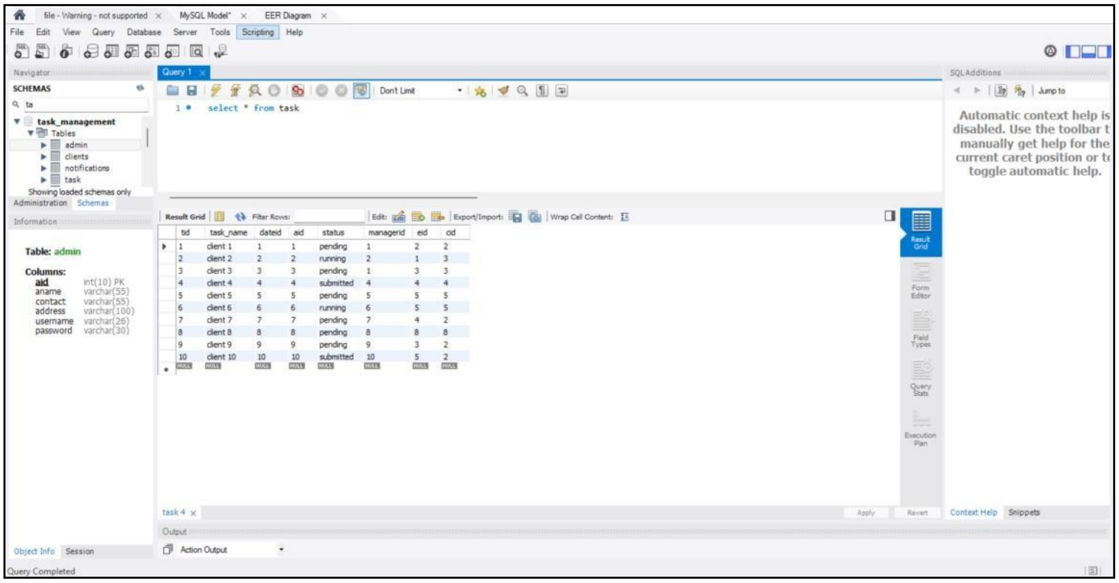
Table: admin

Columns:

aid	int(10) PK
aname	varchar(55)
contact	varchar(55)
address	varchar(100)
username	varchar(26)
password	varchar(30)

date_id	tid	cid	manager_id	date	eid	task_cat_id
1	1	1	1	2022-10-12	1	1
2	2	3	4	2022-10-02	2	6
3	3	3	4	2022-09-22	3	3
4	4	4	4	2022-10-19	4	4
5	5	5	5	2022-10-03	5	5
6	6	6	6	2022-08-06	6	6
7	7	7	7	2022-10-10	7	7
8	8	8	8	2022-10-04	8	8
9	9	9	9	2022-10-19	9	9
10	10	10	10	2022-10-20	8	8
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 7: Select query for displaying Task _date table details



```
select * from task
```

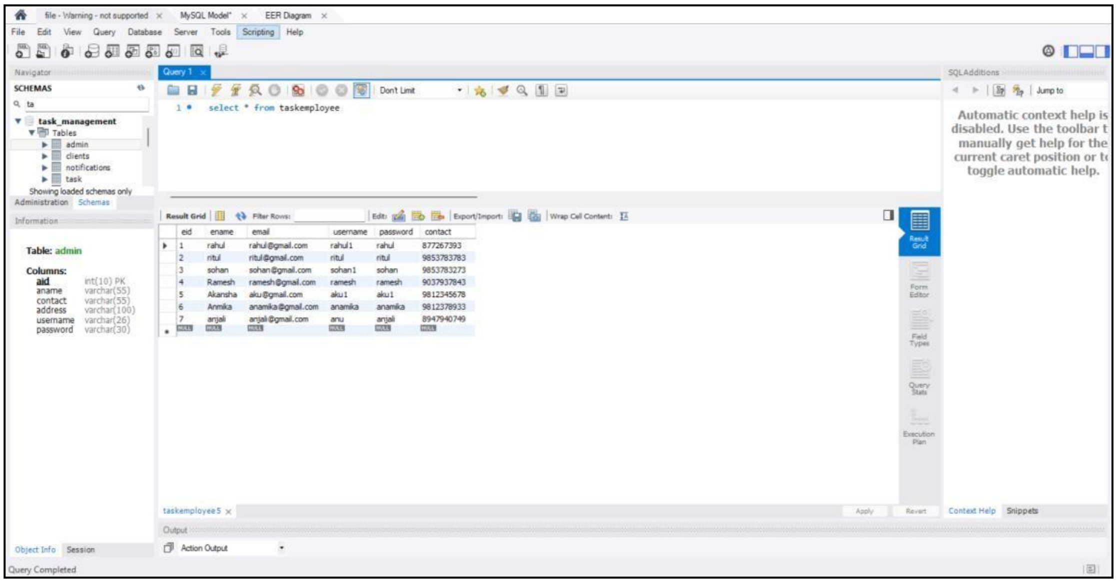
tid	task_name	dateid	aid	status	managerid	eid	cid
1	client 1	1	1	pending	1	2	2
2	client 2	2	2	running	2	1	3
3	client 3	3	3	pending	1	3	3
4	client 4	4	4	submitted	4	4	4
5	client 5	5	5	pending	5	5	5
6	client 6	6	6	running	6	5	5
7	client 7	7	7	pending	7	4	2
8	client 8	8	8	pending	8	8	8
9	client 9	9	9	pending	9	3	2
10	client 10	10	10	submitted	10	5	2
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Table: admin

Columns:

aid	int(10) PK
aname	varchar(55)
contact	varchar(55)
address	varchar(100)
username	varchar(26)
password	varchar(30)

Figure 8: Select query for displaying Task table details



```
select * from taskemployee
```

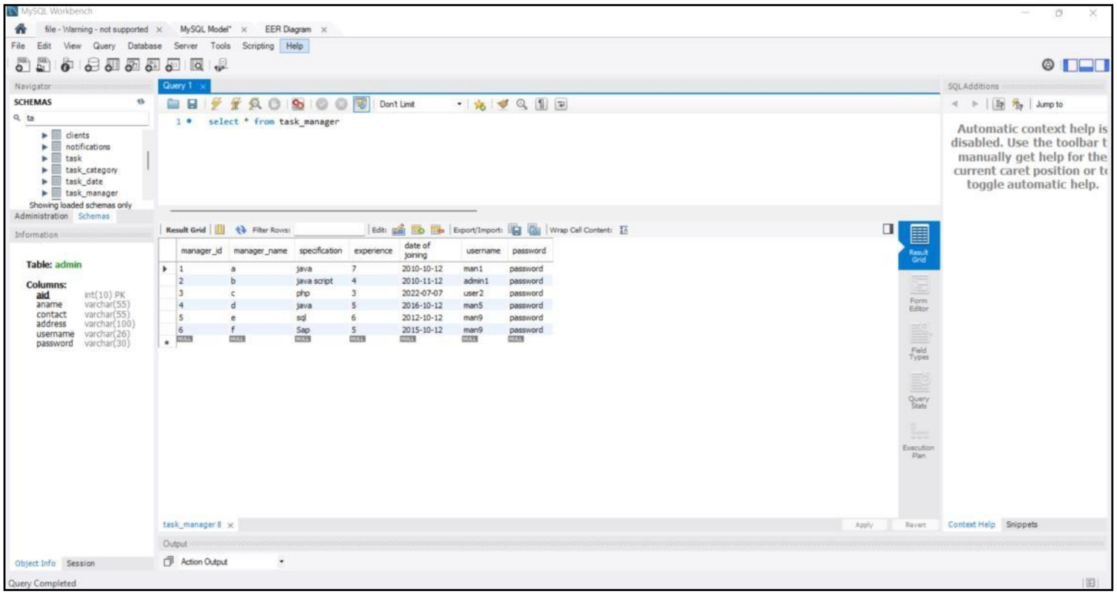
eid	ename	email	username	password	contact
1	rahul	rahul@gmail.com	rahul1	rahul	877267393
2	ritul	ritul@gmail.com	ritul	ritul	9853783783
3	sohan	sohan@gmail.com	sohan1	sohan	9853783273
4	Ramesh	ramesh@gmail.com	ramesh	ramesh	9037937843
5	Akansha	aku@gmail.com	aku1	aku1	9812345678
6	Anmika	anamika@gmail.com	anamika	anamika	9812378933
7	anjali	anjali@gmail.com	anu	anjali	8947940749
NULL	NULL	NULL	NULL	NULL	NULL

Table: admin

Columns:

aid	int(10) PK
aname	varchar(55)
contact	varchar(55)
address	varchar(100)
username	varchar(26)
password	varchar(30)

Figure 9: Select query for displaying Task _employee table details



```
select * from task_manager
```

manager_id	manager_name	specification	experience	date of joining	username	password
1	a	java	7	2010-10-12	man1	password
2	b	java script	4	2010-11-12	admin1	password
3	c	php	3	2022-07-07	user2	password
4	d	java	5	2016-10-12	man5	password
5	e	sql	6	2012-10-12	man9	password
6	f	Sap	5	2015-10-12	man9	password
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Table: admin

Columns:

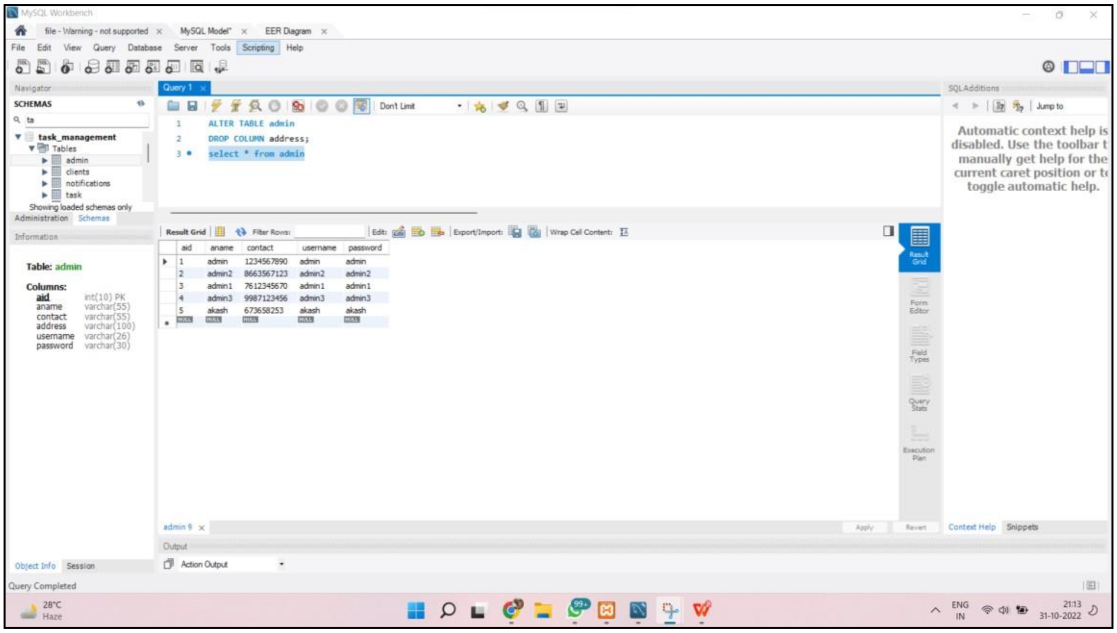
aid	int(10) PK
aname	varchar(55)
contact	varchar(55)
address	varchar(100)
username	varchar(26)
password	varchar(30)

Query Completed

Figure 10: Select query for displaying Task _manager table details

ALTER DATA:

We use ALTER command which comes under Data Definition Language to alter the database structure.



```
ALTER TABLE admin
DROP COLUMN address;
select * from admin
```

aid	aname	contact	username	password
1	admin	1234567890	admin	admin
2	admin2	8663567123	admin2	admin2
3	admin1	7612345670	admin1	admin1
4	admin3	9987123456	admin3	admin3
5	akash	673658253	akash	akash
NULL	NULL	NULL	NULL	NULL

Table: admin

Columns:

aid	int(10) PK
aname	varchar(55)
contact	varchar(55)
address	varchar(100)
username	varchar(26)
password	varchar(30)

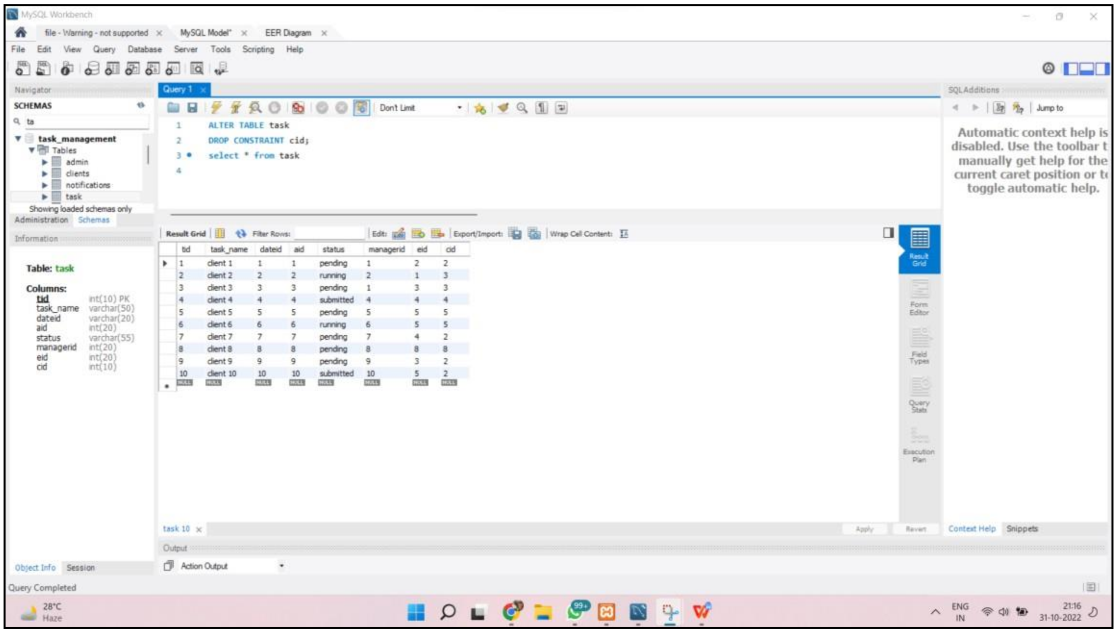
Query Completed

Figure 11: Alter query to drop the Column from Admin table

ALTER TABLE task

DROP CONSTRAINT cid;

select * from task



```
ALTER TABLE task
DROP CONSTRAINT cid;
select * from task
```

tid	task_name	dateid	aid	status	managerid	eid	cid
1	client 1	1	1	pending	1	2	2
2	client 2	2	2	running	2	1	3
3	client 3	3	3	pending	1	3	3
4	client 4	4	4	submitted	4	4	4
5	client 5	5	5	pending	5	5	5
6	client 6	6	6	running	6	5	5
7	client 7	7	7	pending	7	4	2
8	client 8	8	8	pending	8	8	8
9	client 9	9	9	pending	9	3	2
10	client 10	10	10	submitted	10	5	2
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Table: task

Columns:
tid int(10) PK
task_name varchar(50)
dateid varchar(20)
aid int(20)
status varchar(55)
managerid int(20)
eid int(20)
cid int(10)

Figure 12: Alter query to drop the Constraint from task table

UPDATE DATA:

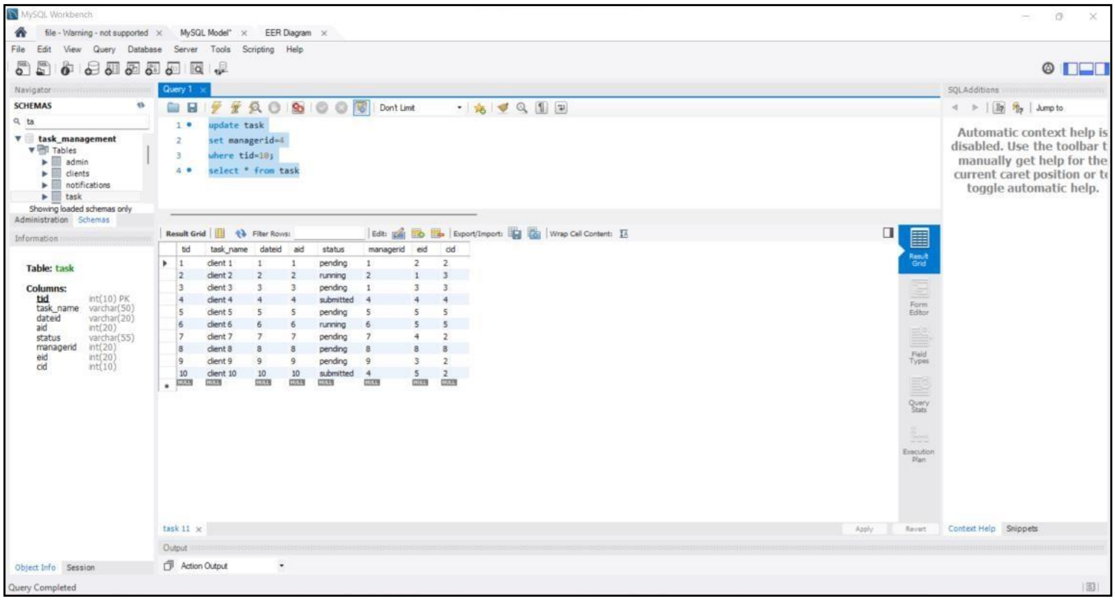
We use UPDATE command which comes under Data Manipulation Language to update the values in the database table.

update task

set managerid=4

where tid=10;

select * from task



```
update task
set managerid=4
where tid=10;
select * from task
```

tid	task_name	dateid	aid	status	managerid	eid	cid
1	client 1	1	1	pending	1	2	2
2	client 2	2	2	running	2	1	3
3	client 3	3	3	pending	1	3	3
4	client 4	4	4	submitted	4	4	4
5	client 5	5	5	pending	5	5	5
6	client 6	6	6	running	6	5	5
7	client 7	7	7	pending	7	4	2
8	client 8	8	8	pending	8	8	8
9	client 9	9	9	pending	9	3	2
10	client 10	10	10	submitted	4	5	2
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

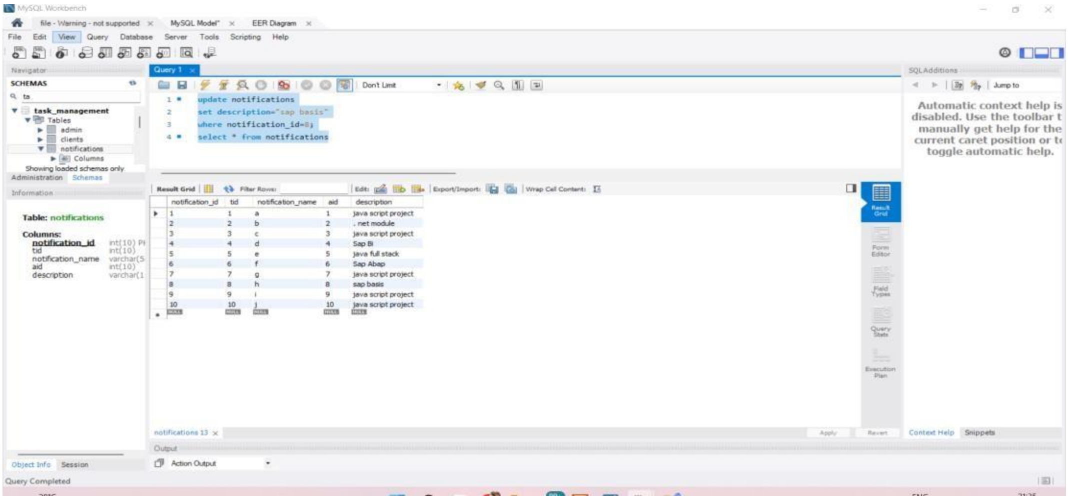
Figure 13: Update query using for manager _id on task table

update notifications

set description="sap basis"

where notification_id=8;

select * from notifications



```
update notifications
set description="sap basis"
where notification_id=8;
select * from notifications
```

notification_id	tid	notification_name	aid	description
1	1	a	1	java script project
2	2	b	2	. net module
3	3	c	3	java script project
4	4	d	4	Sap BI
5	5	e	5	java full stack
6	6	f	6	Sap Abap
7	7	g	7	java script project
8	8	h	8	sap basis
9	9	i	9	java script project
10	10	j	10	java script project
NULL	NULL	NULL	NULL	NULL

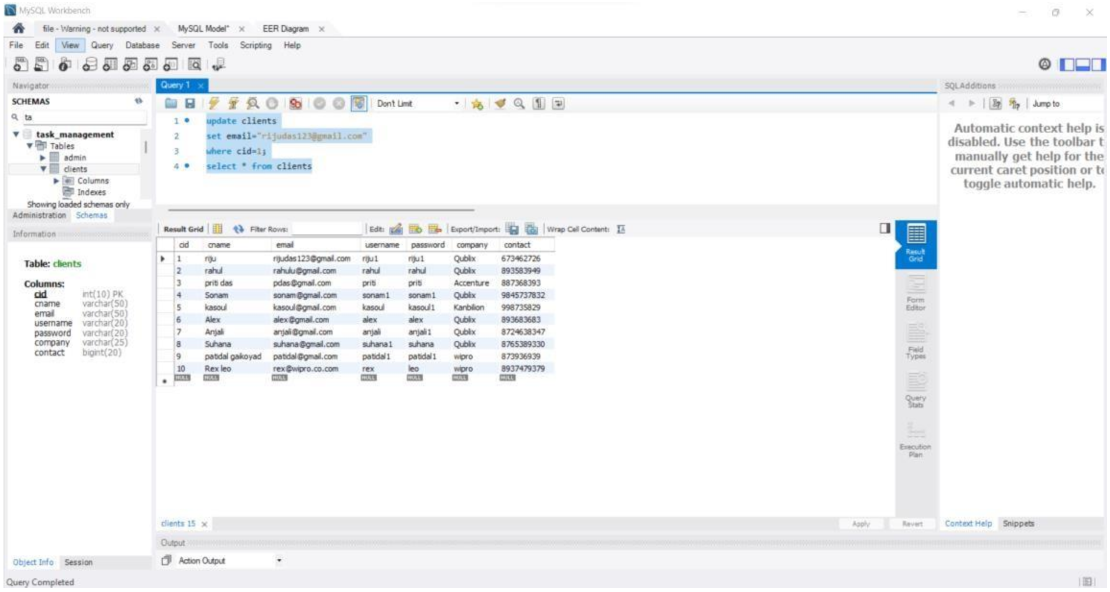
Figure 14: Update query using for description on notification table

update clients

set email="rijudas123@gmail.com"

where cid=1;

select * from clients



```
update clients
set email="rijudas123@gmail.com"
where cid=1;
select * from clients
```

Table: clients

Columns:

cid	int(10) PK
cname	varchar(50)
email	varchar(50)
username	varchar(20)
password	varchar(20)
company	varchar(25)
contact	bigint(20)

cid	cname	email	username	password	company	contact
1	riju	rijudas123@gmail.com	riju1	riju1	Qublix	673462726
2	rahul	rahulu@gmail.com	rahul	rahul	Qublix	893583949
3	priti das	pdas@gmail.com	priti	priti	Accenture	887368393
4	Sonam	sonam@gmail.com	sonam1	sonam1	Qublix	9845737832
5	kasoul	kasoul@gmail.com	kasoul	kasoul1	Kanbilion	998735829
6	Alex	alex@gmail.com	alex	alex	Qublix	893683683
7	Anjali	anjali@gmail.com	anjali	anjali1	Qublix	8724638347
8	Suhana	suhana@gmail.com	suhana1	suhana	Qublix	8765389330
9	patidal gaikoyad	patidal@gmail.com	patidal1	patidal1	wipro	873936939
10	Rex leo	rex@wipro.co.com	rex	leo	wipro	8937479379
NULL	NULL	NULL	NULL	NULL	NULL	NULL

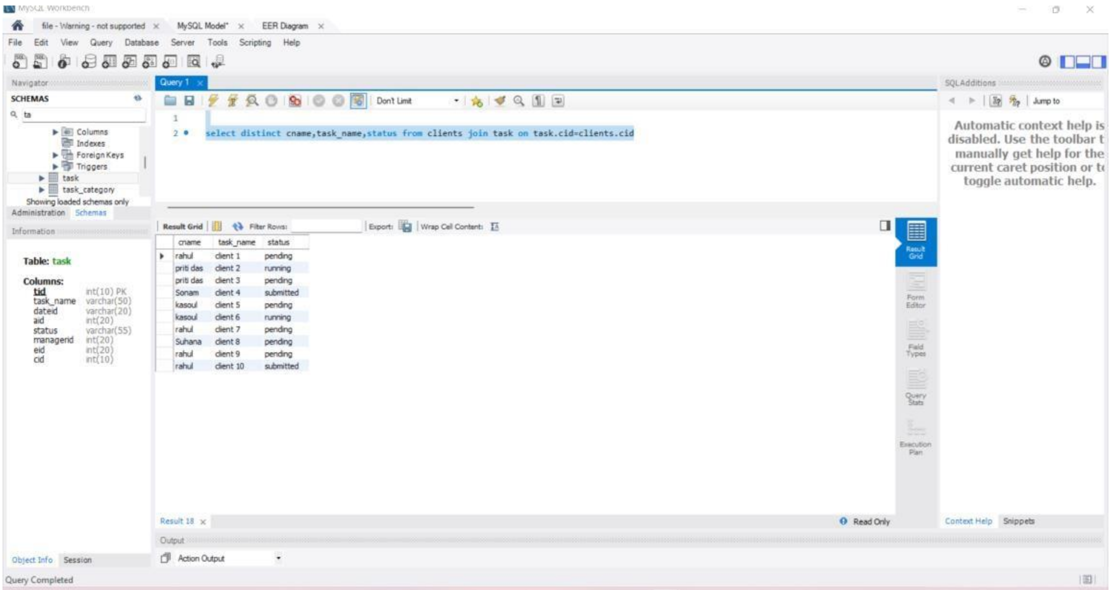
Query Completed

Figure 15: Update query using for emails on client table

OPTIMIZING DATABASE:

The optimizer is to merge, reorganize, and process the data in any order as SQL is a non-procedural language and also the process of selecting the most effective way to execute a SQL statement.

select distinct cname, task _name, status from clients join task on task.cid = clients.cid



```
select distinct cname,task_name,status from clients join task on task.cid=clients.cid
```

Table: task

Columns:

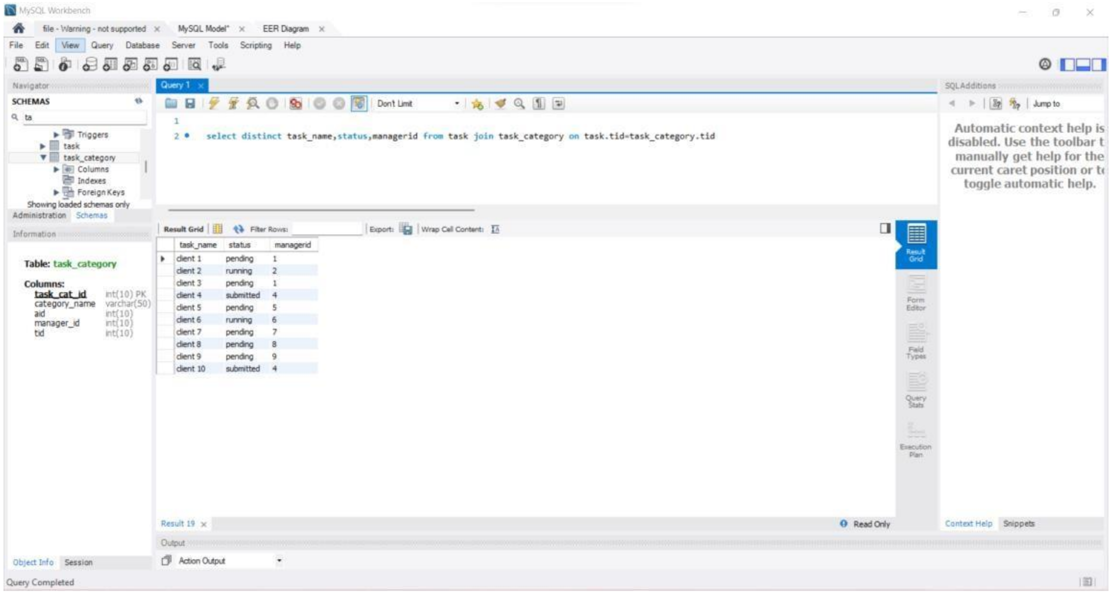
tid	int(10) PK
task_name	varchar(50)
dateid	varchar(20)
aid	int(20)
status	varchar(55)
managerid	int(20)
eid	int(20)
cid	int(10)

cname	task_name	status
rahul	client 1	pending
priti das	client 2	running
priti das	client 3	pending
Sonam	client 4	submitted
kasoul	client 5	pending
kasoul	client 6	running
rahul	client 7	pending
Suhana	client 8	pending
rahul	client 9	pending
rahul	client 10	submitted

Query Completed

Figure 16: Select query for Client table

select distinct task_name,status,managerid from task join task_category on task.tid=task_category.tid



```
select distinct task_name,status,managerid from task join task_category on task.tid=task_category.tid
```

Table: task_category

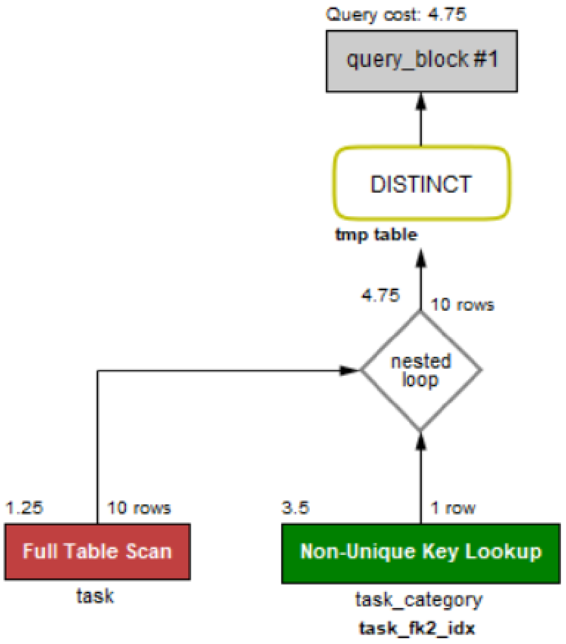
Columns:

task_cat_id	int(10) PK
category_name	varchar(50)
aid	int(10)
manager_id	int(10)
tid	int(10)

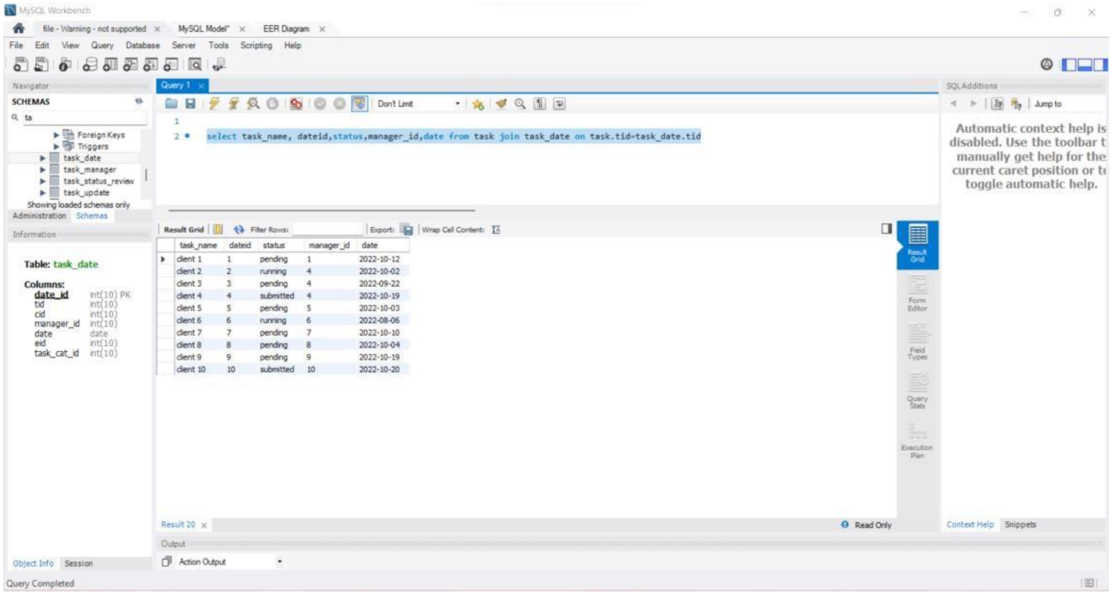
task_name	status	managerid
client 1	pending	1
client 2	running	2
client 3	pending	1
client 4	submitted	4
client 5	pending	5
client 6	running	6
client 7	pending	7
client 8	pending	8
client 9	pending	9
client 10	submitted	4

Query Completed

Figure 17: Join query for task and task _ category table



select task_name, dateid,status,manager_id,date from task join task_date on task.tid=task_date.tid



```
select task_name, dateid,status,manager_id,date from task join task_date on task.tid=task_date.tid
```

task_name	dateid	status	manager_id	date
client 1	1	pending	1	2022-10-12
client 2	2	running	4	2022-10-02
client 3	3	pending	4	2022-09-22
client 4	4	submitted	4	2022-10-19
client 5	5	pending	5	2022-10-03
client 6	6	running	6	2022-08-06
client 7	7	pending	7	2022-10-10
client 8	8	pending	8	2022-10-04
client 9	9	pending	9	2022-10-19
client 10	10	submitted	10	2022-10-20

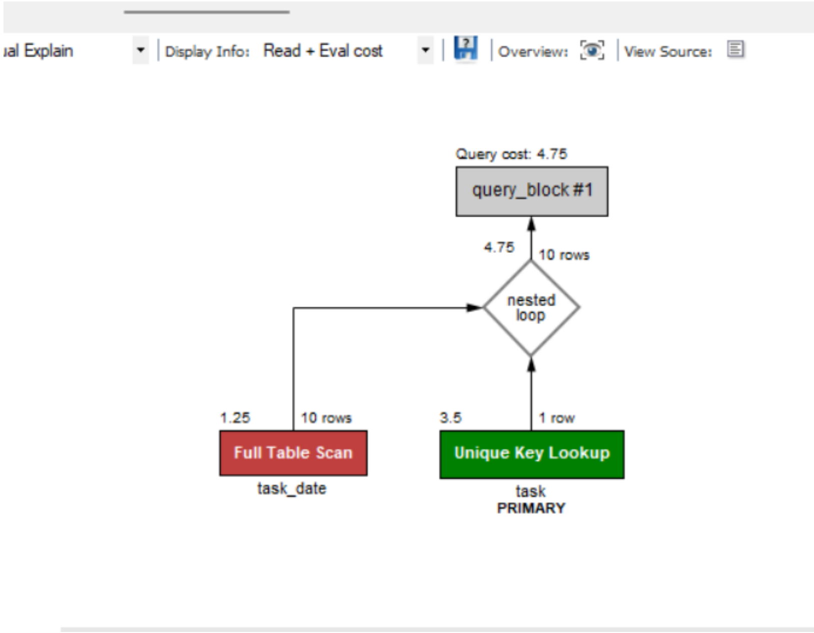
Table: task_date

Columns:

date_id	int(10) PK
tid	int(10)
cid	int(10)
manager_id	int(10)
date	date
eid	int(10)
task_cat_id	int(10)

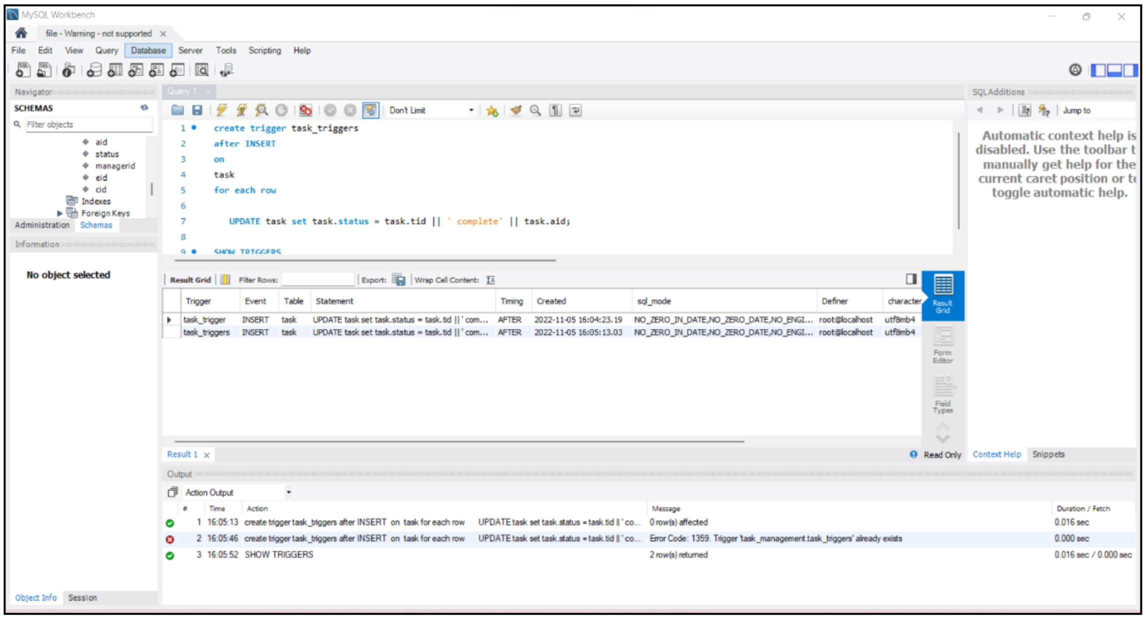
Query Completed

Figure 18: Join query for task and task _ date table



Triggers

```
create trigger task_triggers

after INSERT

on

task

for each row

  UPDATE task set task.status = task.tid || ' complete' || task.aid;

SHOW TRIGGERS
```



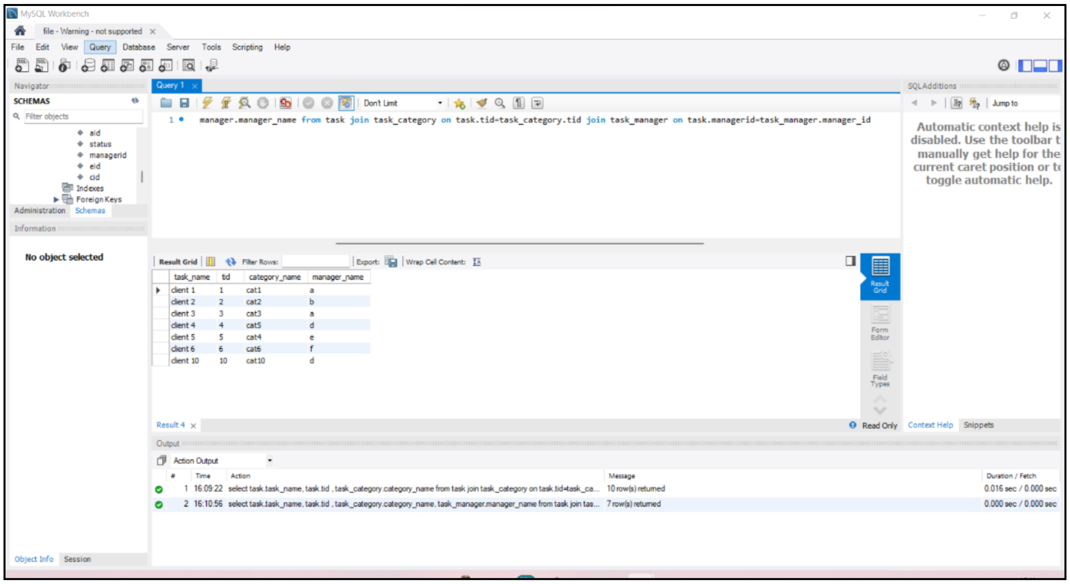
```
create trigger task_triggers
after INSERT
on
task
for each row

   UPDATE task set task.status = task.tid || ' complete' || task.aid;

SHOW TRIGGERS
```

	Trigger	Event	Table	Statement	Timing	Created	sql_mode	Definer	character
▶	task_trigger	INSERT	task	UPDATE task set task.status = task.tid || ' com...	AFTER	2022-11-05 16:04:23.19	NO_ZERO_IN_DATE,NO_ZERO_DATE,NO_ENGI...	root@localhost	utf8mb4
	task_triggers	INSERT	task	UPDATE task set task.status = task.tid || ' com...	AFTER	2022-11-05 16:05:13.03	NO_ZERO_IN_DATE,NO_ZERO_DATE,NO_ENGI...	root@localhost	utf8mb4

	#	Time	Action		Message	Duration / Fetch
✓	1	16:05:13	create trigger task_triggers after INSERT on task for each row	UPDATE task set task.status = task.tid || ' co...	0 row(s) affected	0.016 sec
☒	2	16:05:46	create trigger task_triggers after INSERT on task for each row	UPDATE task set task.status = task.tid || ' co...	Error Code: 1359. Trigger 'task_management.task_triggers' already exists	0.000 sec
✓	3	16:05:52	SHOW TRIGGERS		2 row(s) returned	0.016 sec / 0.000 sec

Figure.19 Trigger Query

select task.task_name, task.tid , task_category.category_name, task_manager.manager_name from task join task_category on task.tid=task_category.tid join task_manager on task.managerid=task_manager.manager_id



```
1 ●  manager.manager_name from task join task_category on task.tid=task_category.tid join task_manager on task.managerid=task_manager.manager_id
```

task_name	tid	category_name	manager_name
dient 1	1	cat1	a
dient 2	2	cat2	b
dient 3	3	cat3	a
dient 4	4	cat5	d
dient 5	5	cat4	e
dient 6	6	cat6	f
dient 10	10	cat10	d

```
#  Time      Action                                                                                              Message             Duration / Fetch
1  16:09:22  select task.task_name, task.tid , task_category.category_name from task join task_category on task.tid=task_ca...  10 row(s) returned  0.016 sec / 0.000 sec
2  16:10:56  select task.task_name, task.tid , task_category.category_name, task_manager.manager_name from task join tas...   7 row(s) returned   0.000 sec / 0.000 sec
```

Figure 20. Join Query

B.

Optimizing queries may be a component of database performance tuning since inefficient queries are frequently the cause of performance issues.

Indexes are another crucial component of a database, along with queries. Your data will remain structured thanks to indexing, which also makes it simpler to find.

The efficiency of data retrieval is increased by proper indexing, which may boost database speed while also saving users and the systems time and resources.

One of the best strategies that can use to improve database speed is to defragment the information.

The database will unavoidably get splintered as a result of the frequent writing and deleting of data, which might make it more difficult or slower to retrieve data or execute a query plan.

Graphical user experience design :

This GUI has login option, there the user has to put user name and password, if the user is new user then he has to register first then login to the system. For connection part we have to use the php my admin database and vs code for IDE. After create the connection with the database the user can view the homepage. After login to the system he can see action items, there he can update name, password etc.

This is the flow chart of the process

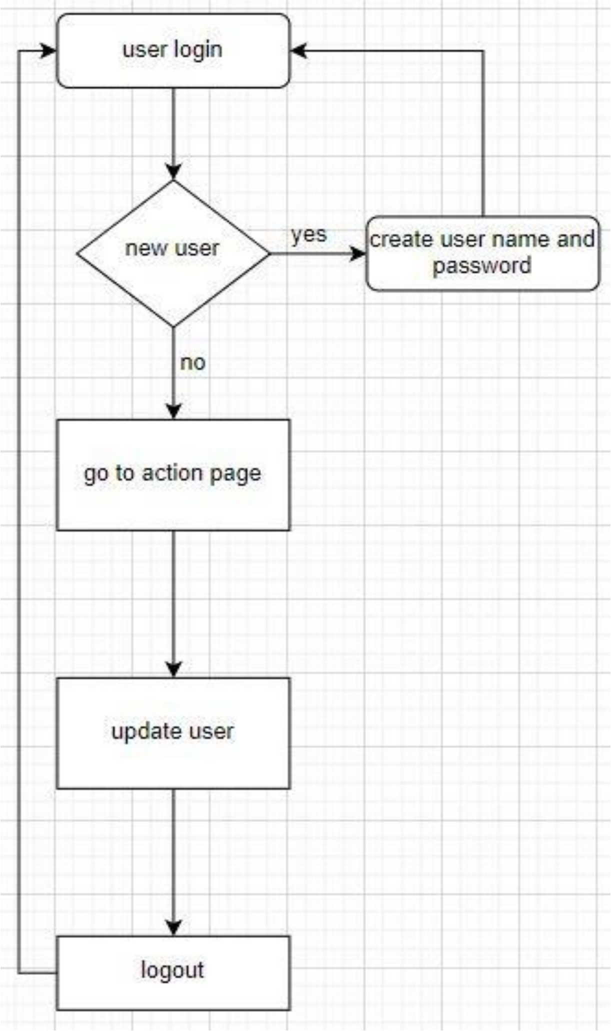


Fig .Flow chart

Graphical user interface design:

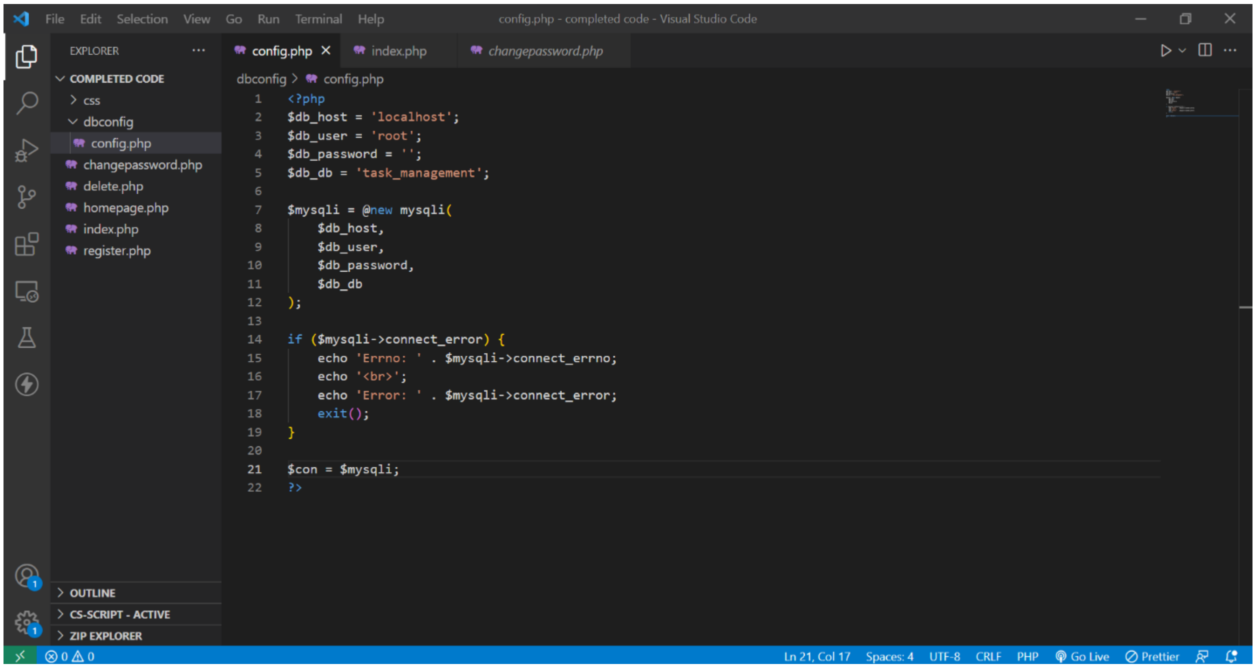
```
 <?php

$db_host = 'localhost';

$db_user = 'root';

$db_password = 'root';

$db_db = 'task_management';

//create connection 1

/* $con = mysqli_connect($db_host,$db_user,$db_password,$db_db);

   if (mysqli_connect_error()) {

    echo 'Error!';

    exit();

    }

    echo 'Connection done';  */


/* $mysqli = @new mysqli(

    $db_host,

    $db_user,

    $db_password,

    $db_db

); */


// connection create type 2
```

```
try{
    $con = new PDO("mysql:host=$db_host;dbname=$db_db",$db_user,$db_password );
    $con->setAttribute(PDO::ATTR_ERRMODE,PDO::ERRMODE_EXCEPTION);
     echo "Connection done";
}
catch(PDOException $e){
    echo "Error".$e->getMessage();
}
?>


<?php
    require”login.php”;
?>
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <title>Login</title>
        <link type="text/css" rel="stylesheet" href="css/style.css">
        <script src="https://kit.fontawesome.com/b26b33266f.js" crossorigin="anonymous"></script>
    </head>
    <body>
        <?php
    $username = $password = "";
    if(isset($_POST["login"])){
        $username = $_POST["username"];
        $password = $_POST["password"];

        $query = "SELECT * from admin WHERE username = '$username' AND password = '$password' ";
        $query_run = mysqli_query($con, $query);

        if(mysqli_num_rows($query_run) > 0){
            header("location:homepage.php?user=$username");
        }else{
            echo "<script> alert('Username or Password is incorrect.')</script>";
        }
    }
?>.

<div>
    <h1>
        LOGIN
    </h1>
    <h2>
        <form method = "post" action = "<?php echo htmlspecialchars($_SERVER['PHP_SELF']);?>" class = "loginbox">
```

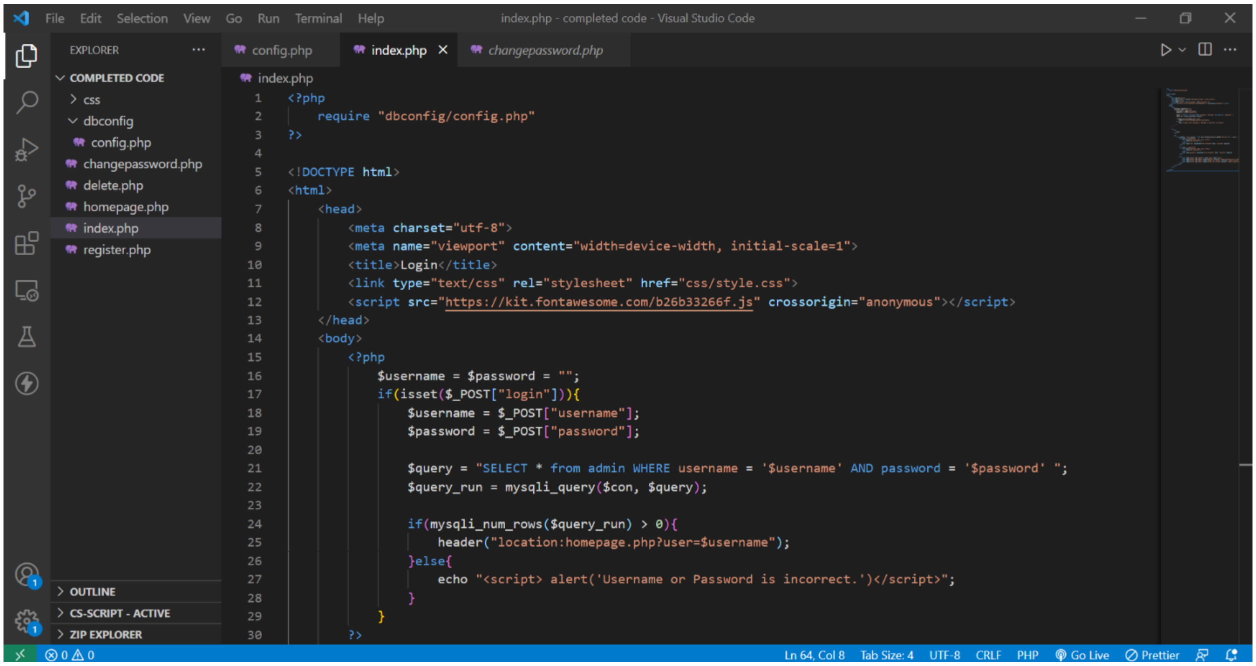
```
                <div class = "loginfield">
                    <span style="font-size: 20px; color: white;">
                        <i class="fas fa-user"></i>
                    </span>
                    <input type="text" placeholder="Your Username" name = "username" required>
                </div>

                <div class = "loginfield">
                    <span style="font-size: 20px; color: white;">
                        <i class="fas fa-lock"></i>
                    </span>
                    <input type="password" placeholder="Your Password" name = "password" required>
                </div>
                <div>
                    <input type="submit" id="loginbtn" value="Login" name="login">
                    <input type="button" id="regbtn" value="Register" onclick= "window.location.href = 'register.php'">
                    <input type="button" id="chgbtn" value="Change Your Password" onclick= "window.location.href = 'changepassword.php'">
                    <input type="button" id="delbtn" value="Delete Your Account" onclick= "window.location.href = 'delete.php'">
                </div>
            </form>
        </h2>
```

```
    </div>
  </body>
</html>
```

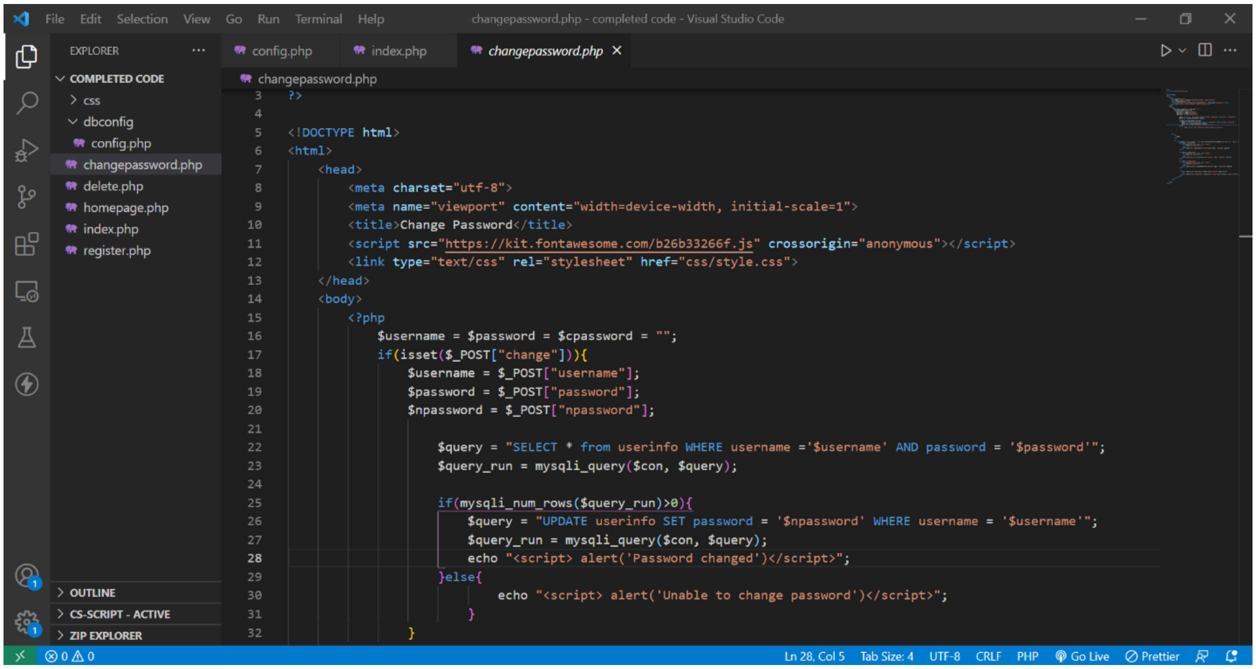
- The main menu provides the list of all available options to the end user. These options include insertion, deletion, modification, searching, and printing all the available items in the database. If the user has the role of admin, he/she should also have access to user administration as well.
- The user administration page should provide the capability of adding, removing, and updating users' information: username, password, role/privilege.
- The regular users should only be allowed to change their passwords
- This page should get the username and password values from the end user and check with the database. If the user was a legitimate one, the program should allow the user to proceed with the other pages. Otherwise, it should show a warning of incorrect username or password.[4]



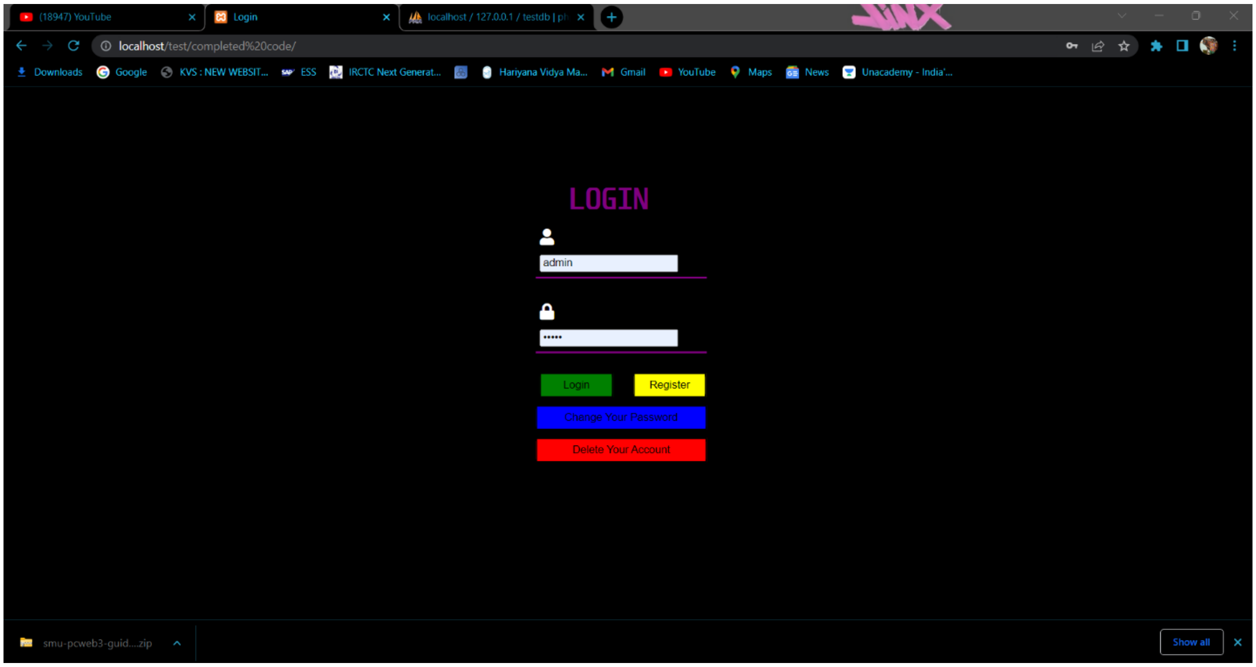
```php
<?php
$db_host = 'localhost';
$db_user = 'root';
$db_password = '';
$db_db = 'task_management';

$mysqli = @new mysqli(
    $db_host,
    $db_user,
    $db_password,
    $db_db
);

if ($mysqli->connect_error) {
    echo 'Errno: ' . $mysqli->connect_errno;
    echo '<br>';
    echo 'Error: ' . $mysqli->connect_error;
    exit();
}

$con = $mysqli;
?>
```



```
<?php
    require "dbconfig/config.php"
?>

<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <title>Login</title>
        <link type="text/css" rel="stylesheet" href="css/style.css">
        <script src="https://kit.fontawesome.com/b26b33266f.js" crossorigin="anonymous"></script>
    </head>
    <body>
        <?php
            $username = $password = "";
            if(isset($_POST["login"])){
                $username = $_POST["username"];
                $password = $_POST["password"];

                $query = "SELECT * from admin WHERE username = '$username' AND password = '$password' ";
                $query_run = mysqli_query($con, $query);

                if(mysqli_num_rows($query_run) > 0){
                    header("location:homepage.php?user=$username");
                }else{
                    echo "<script> alert('Username or Password is incorrect.')</script>";
                }
            }
        ?>
```

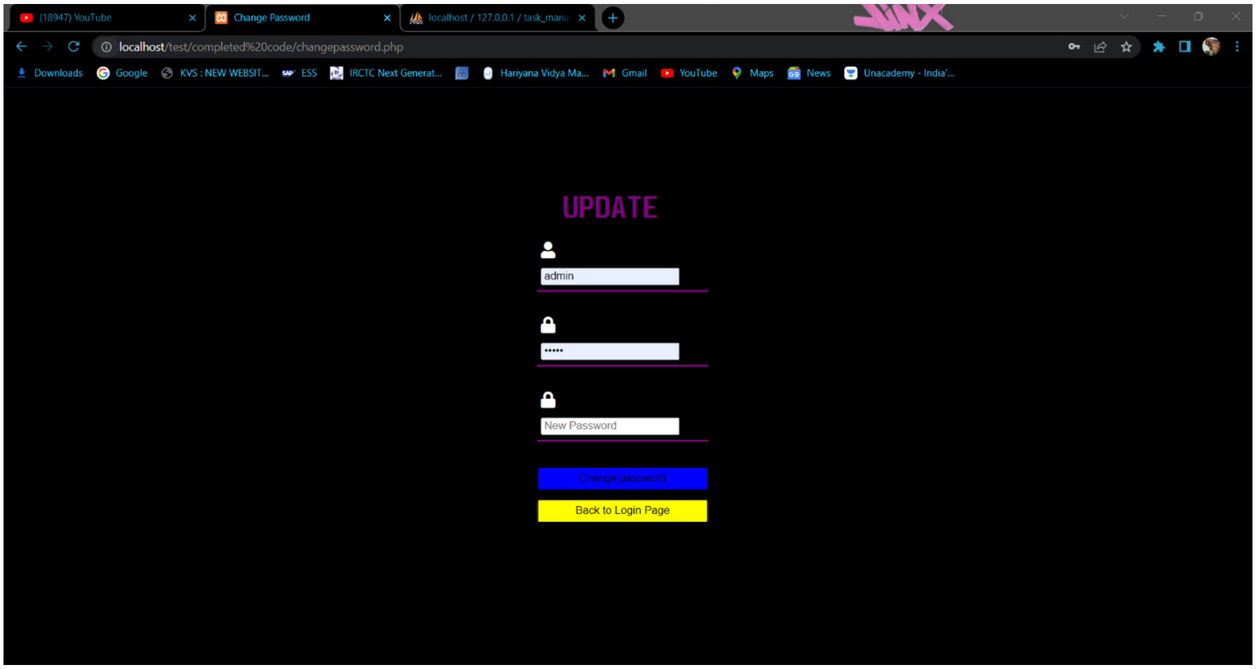


```
?>

<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <title>Change Password</title>
        <script src="https://kit.fontawesome.com/b26b33266f.js" crossorigin="anonymous"></script>
        <link type="text/css" rel="stylesheet" href="css/style.css">
    </head>
    <body>
        <?php
            $username = $password = $cpassword = "";
            if(isset($_POST["change"])){
                $username = $_POST["username"];
                $password = $_POST["password"];
                $npassword = $_POST["npassword"];

                    $query = "SELECT * from userinfo WHERE username ='$username' AND password = '$password'";
                    $query_run = mysqli_query($con, $query);

                    if(mysqli_num_rows($query_run)>0){
                        $query = "UPDATE userinfo SET password = '$npassword' WHERE username = '$username'";
                        $query_run = mysqli_query($con, $query);
                        echo "<script> alert('Password changed')</script>";
                    }else{
                            echo "<script> alert('Unable to change password')</script>";
                        }
                }
```


(18947) YouTube
Login
localhost / 127.0.0.1 / testdb | ph
localhost/test/completed%20code/
Downloads
Google
KVS : NEW WEBSIT...
ESS
IRCTC Next Generat...
Hariyana Vidya Ma...
Gmail
YouTube
Maps
News
Unacademy - India'...
LOGIN
Login
Register
Change Your Password
Delete Your Account
smu-pcweb3-guid....zip
Show all

admin

•••••


(18947) YouTube
Change Password
localhost / 127.0.0.1 / task_mana
localhost/test/completed%20code/changepassword.php
Downloads
Google
KVS : NEW WEBSIT...
ESS
IRCTC Next Generat...
Hariyana Vidya Ma...
Gmail
YouTube
Maps
News
Unacademy - India'...
UPDATE
New Password
Change password
Back to Login Page

admin

•••••

References:

- [1] Cheng, D. R., & South, M. (2020). Electronic task management system: a pediatric institution's experience. *Applied Clinical Informatics*, *11*(05), 839- 845.
- Mikelsone, E., Spilbergs, A., Segers, J. P., Volkova, T., & Liela, E. (2022). Better Ideation Task

  Results in Web-Based Idea Management Systems. *Businesses*, *2*(2), 129-140.
- [4]https://www.javatpoint.com/php-mysql-login-system
- [2] https://www.consumervoice.org/wrike-review
- [3] https://project-management.com/calendar-software/