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

Project Report of TASK MANAGEMENT SYSTEM

Team Name:

The Unstoppables

Team Members:

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

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

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

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

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

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

Pranay Reddy Kosireddy and 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

TABLE OF CONTENTS

TABLE OF FIGURES

OBJECTIVE:

- The main work of the task management system is to record various details it comes along with the contacts.
- It is used by an organization or sometimes even an individual to efficiently manage and complete different projects and also relate to it with respect to the prioritization of the task.
- It is not just about the project management, but it has several other applications especially when it comes to an organization as it gets complete control of the human resources, the infrastructures, the software, and different applications.[1]

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]

MERITS:

- Allows you to set goals, keep track of deliverables, and manage deadlines.
- Easy access to the software for everyone to check task updates.
- Seamless collaborations are driven together for better communication of the management system with a visual-based to heightened productivity.
- It helps to track all the assigned and updated tasks information instantly.
- Moreover, the merit of this project is that 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.
- Notification Functionality is used to notify about the task that is assigned to person.

- It ensures customers satisfaction.
- It gives effective functionality for remote working.
- It is user friendly and east to use software.
- 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

DATA DESCRIPTION:

This sql database has ten tables like admins, task update review, task category, task manager, task status, clients, task, task employee, task update, notification.

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.

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 is N:1. task and task update the relationship is 1:1.

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.
- 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.

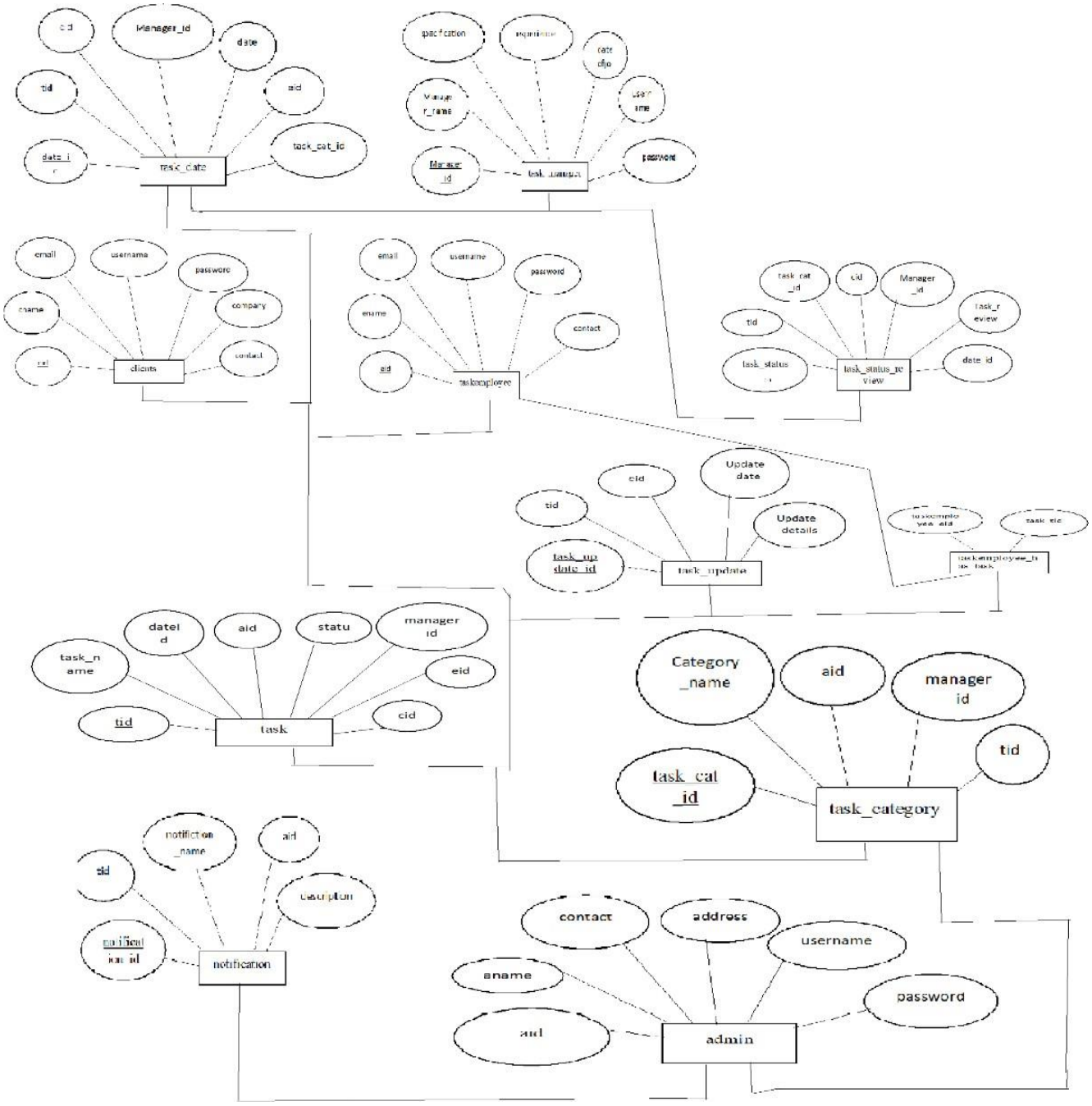

Fig (a)

ENHANCED ENTITY-RELATIONSHIP MODEL:

- The basic relationship between the entities includes one to many, many to many, and one-to-one.
- We are treating all these relationships within this model. If suppose, the task date is the primary key, the other entities and attributes are considered to be linked to it. All of the attributes also have a relationship that they share between each of them.
- 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. There is a one-to-many relationship that is experienced to the highest level among one-to-one and many-to-many relationships.

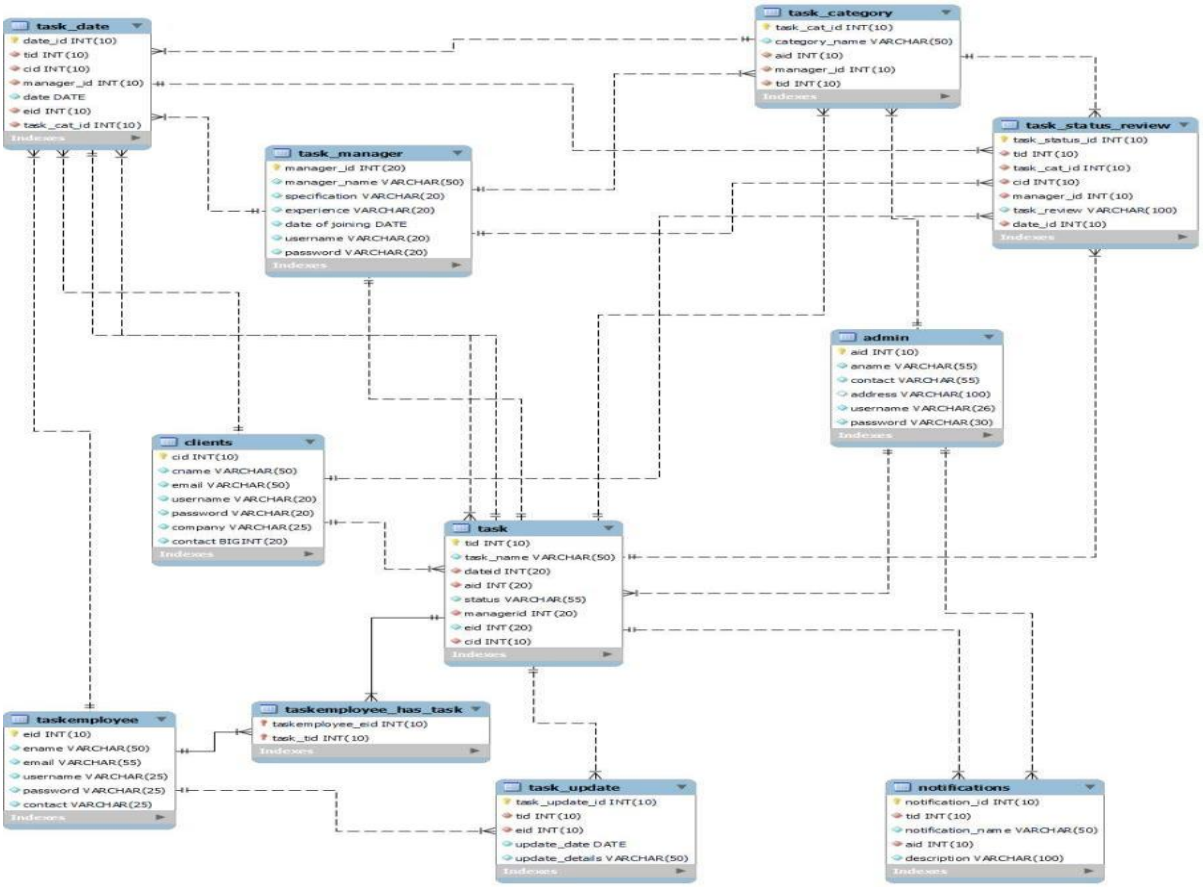

Fig (b)

DATABASE DEVELOPMENT

SQL code

ADMIN TABLE: -

```
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`));
```

- The above 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.

CLIENTS TABLE: -

```
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`));
```

- The above 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.

TASK_MANAGER TABLE: -

```
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`)) ;
```

- The above 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.

TASKEMPLOYEE TABLE: -

```
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`));
```

- The above 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.

TASK_CATEGORY TABLE: -

```
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);
```

- The above 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.

TASK_DATE TABLE: -

```
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);
```

- The above 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.

TASK TABLE: -

```
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);
```

- The above 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.

NOTIFICATION TABLE: -

```
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);
```

- The above 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.

TASK_STATUS _REVIEW TABLE: -

```
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);
```

- The above 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.

TASK_UPDATE TABLE: -

```
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)
```

- The above 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.

TASKEMPLOYEE_HAS_TASK TABLE: -

```
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)
```

- The above 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.

DESCRIPTION:

table	attribute	Primary key/foreign key
admin	Aid (int)	Primary key
	Aname(varchar)	
	Contact(varchar)	
	Address(varchar)	
	Username(varchar)	
	Password(varchar)	
Clients	Cid(int)	Primary key
	Cname(varchar)	
	email(varchar)	
	username(varchar)	
	password(varchar)	
	company(varchar)	
	Contact(int)	
Notification	Notification id(int)	Primary key
	Tid(int)	Foreign key

	Notification name(varchar)	
	Aid(int)	Foreign key
	description(varchar)	
task	Tid(int)	Primary key
	Task name(varchar)	
	Date id(int)	Foreign key
	Aid(int)	Foreign key
	status(varchar)	
	Managerid(int)	Foreign key
	Eid(int)	Foreign key
	Cid(int)	Foreign key
Task employee	Eid(int)	Primary key
	E name(varchar)	
	email(varchar)	
	username(varchar)	
	password(varchar)	
	contact(varchar)	
Task category	Task cat id(int)	Primary key

	Category name(varchar)	
	Aid(int)	Foreign key
	Manager id(int)	Foreign key
	Tid((int)	Foreign key
Task date	Date id(int)	Primary key
	Tid(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)	Primary key
	Manager name(varchar)	
	Specification(varchar)	
	Experience(varchar)	
	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)	

References:

- [1] Cheng, D. R., & South, M. (2020). Electronic task management system: a pediatric institution's experience. *Applied Clinical Informatics*, *11*(05), 839- 845.
- [2] https://www.consumervoice.org/wrike-review
- https://project-management.com/calendar-software/