*CHARLES MAINA DATABASE WEEK 2 HOMEWORK, CLASS 13.*

*PART1.*

USE hyflesson2;

--Add a task with these attributes: title, description, created, updated, due\_date, status\_id, user\_id

INSERT INTO task

(title,

description,

created,

updated,

due\_date,

status\_id)

VALUES (

'chess',

'play with queen and pawns, bishop is defender',

'2020-05-15',

'2020-05-20',

'2020-05-29',

3);

--Change the title of a task

UPDATE task

SET title= 'cards'

WHERE title= 'chess';

--Change a task due date

UPDATE task

SET due\_date= '2020-06-01'

WHERE due\_date= '2020-05-29';

--Change a task status

UPDATE task

SET status\_id= 2

WHERE status\_id= 3;

--mark status as done

UPDATE task

SET status\_id= 3

WHERE status\_id= 2;

--Delete task.

DELETE task

WHERE title = 'cards';

*PART 2.*

*SCHOOL DATABASE.*

/\*Create a new database containing the following tables:

* **Class**: with the columns: id, name, begins (date), ends (date)
* **Student**: with the columns: id, name, email, phone, class\_id (foreign key)\*/

CREATE DATABASE schoolreg;

CREATE TABLE `CLASS` (

`class\_id` INT(10) unsigned NOT NULL AUTO\_INCREMENT PRIMARY KEY,

`name` VARCHAR (255) NOT NULL,

`begin\_date` DATETIME,

`ending\_date` DATETIME

);

CREATE TABLE `STUDENT` (

`student\_id` INT(10) unsigned NOT NULL AUTO\_INCREMENT PRIMARY KEY,

`name` VARCHAR (255) NOT NULL,

`email` VARCHAR(255) NOT NULL,

`Phone` NULL,

FOREIGN KEY `class\_id` INT (10)

);

--Create an index on the name column of the student table.

ALTER TABLE student ADD INDEX idx\_name (name);

--Or

CREATE INDEX idx\_name

ON student (name);

/\*Adding a new column to the class table named **status** which can only have the following values: not-started, ongoing, finished (hint: enumerations).\*/

ALTER TABLE class

ADD status

Values ENUM(‘not started’, ‘ongoing’, ‘finished’);

*PART3.*

--- Get all the tasks assigned to users whose email ends in @spotify.com

SELECT \*

FROM task

LEFT JOIN user

ON task.status\_id = user.id

WHERE email LIKE ‘%@spotify.com’;

--- Get all the tasks for 'Donald Duck' with status 'Not started'

--- first we get all Donald Duck’s tasks and his id,.,.His id is 11,.,.

SELECT \*

FROM user\_task

WHERE user\_id = 11;

SELECT title, status\_id

FROM task

WHERE id IN (2, 17, 26, 27, 32);

SELECT title

FROM task

INNER JOIN user\_task

ON task.id = user\_task.task\_id

WHERE user\_task.user\_id(

SELECT id

FROM user

WHERE name = ‘DONALD DUCK’

AND status.id =(SELECT id FROM status);

JOIN task on user.id = task.id;

);

/\*Get all the tasks for 'Maryrose Meadows' that were created in september (hint: month(created)=month\_number)\*/

--- first know Maryrose’s id,..,.

SELECT user.id, name

FROM user;

--- then know how many tasks maryrose has, her id is 6 , we get it from above query.

SELECT \*

FROM user\_task

WHERE user\_id = 6;

SELECT \*

FROM user\_task

WHERE user\_id = 6;

SELECT title, created DATETIME

FROM task

WHERE id IN (7, 10, 14, 27);

---Tasks where created in each month

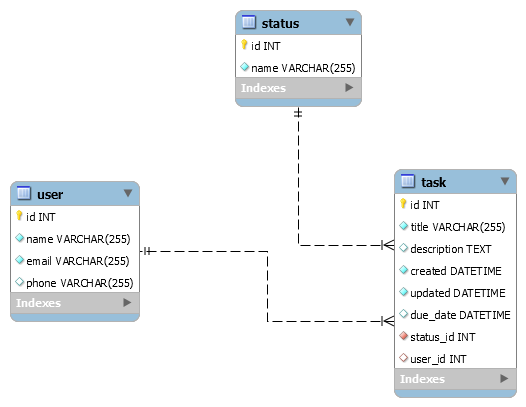
SELECT COUNT (task)

FROM task

ORDER BY month created

**Part 4: Creating a database**

/\*Your application must include at least one many-to-many relationship and any supporting tables (linking tables) that are needed. The entity relationship diagram must describe what tables you will need, the columns in these tables, which column is the primary key, and the relationships between tables.\*/



CREATE TABLE ´user´(

User\_ID ,INT,

name varchar(255),

email VARCHAR (255),

Phone varchar(255),

PRIMARY KEY (User\_ID)

);

CREATE TABLE `status`(

status\_ID, INT,

name varchar(255),

PRIMARY KEY (status\_id)

);

CREATE TABLE `task` (

Task\_ID, INT,

title, varchar(50),

To address, VARCHAR(255)

Description VARCHAR(255),

Created DATETIME

Updated DATETIME

Due\_date DATETIME

Status\_ID INT,

User\_ID

PRIMARY KEY (‘Task\_ID’)

FOREIGN KEY (‘status\_ID’, ‘user\_ID’)

);