**1. Create tables for the below list given**

-- (users,

-- codekata,

-- attendance,

-- topics,

-- tasks,

-- company\_drives,

-- mentors,

-- students\_activated\_courses,

-- courses)

**// solution**

CREATE DATABASE students;

use students;

**# Users**

CREATE TABLE users(userid INTEGER AUTO\_INCREMENT PRIMARY KEY, username VARCHAR(100),useremail VARCHAR(100)

,mentorid INTEGER,

FOREIGN KEY (mentorid) REFERENCES mentors(mentorid));

**# codekata**

CREATE TABLE codekata(codekataid INTEGER AUTO\_INCREMENT PRIMARY KEY,userid INTEGER,number\_of\_problems INTEGER,

string\_problems INTEGER,

FOREIGN KEY (userid) REFERENCES users(userid) );

**# Attendance**

CREATE TABLE attendance(attendanceid INTEGER AUTO\_INCREMENT PRIMARY KEY, userid INTEGER,courseid INTEGER ,topicsid INTEGER, attended BOOLEAN,

FOREIGN KEY (userid) REFERENCES users(userid),

FOREIGN KEY (topicsid) REFERENCES topics(topicsid),

FOREIGN KEY (courseid) REFERENCES courses(courseid));

**# topics**

CREATE TABLE topics(topicsid INTEGER AUTO\_INCREMENT PRIMARY KEY,courseid INTEGER, topic VARCHAR(200),

FOREIGN KEY (courseid) REFERENCES courses(courseid));

**# tasks**

CREATE TABLE tasks(tasksid INTEGER AUTO\_INCREMENT PRIMARY KEY,courseid INTEGER,task VARCHAR(200),

FOREIGN KEY (courseid) REFERENCES courses(courseid));

**# Company drives**

CREATE TABLE company\_drives(drivesid INTEGER AUTO\_INCREMENT PRIMARY KEY,userid INTEGER,company VARCHAR(100),

FOREIGN KEY (userid) REFERENCES users(userid));

**# mentors**

CREATE TABLE mentors(mentorid INTEGER AUTO\_INCREMENT PRIMARY KEY,mentorname VARCHAR(100),mentoremail VARCHAR(100));

**# students\_activated\_courses**

CREATE TABLE students\_activated\_courses(id INTEGER AUTO\_INCREMENT PRIMARY KEY,userid INTEGER,courseid INTEGER,

FOREIGN KEY (userid) REFERENCES users(userid),

FOREIGN KEY (courseid) REFERENCES courses(courseid));

# **courses**

CREATE TABLE courses(courseid INTEGER AUTO\_INCREMENT PRIMARY KEY,coursename VARCHAR(100));

**2. Insert at least 5 rows of values in each table**

**// solution**

use students;

**# users**

INSERT INTO users(username,useremail,mentorid) VALUES("Hari","Hari@gmail.com",4),

("Subash","Subash@gmail.com",2),

("Rashmika","Rashmika@gmail.com",1),

("Ajith","Ajith@gmail.com",5),

("Steve","Steve@gmail.com",3);

**# codekata**

INSERT INTO codekata(number\_of\_problems,string\_problems,userid) VALUES(40,34,1),

(35,25,2),

(60,40,3),

(20,10,4),

(25,20,5);

**# attendence**

INSERT INTO attendance(userid,courseid,topicsid,attended) VALUES(2,5,5,true),

(5,1,1,true),

(1,3,3,false),

(3,4,4,ture),

(4,2,2,true);

**# topics**

INSERT INTO topics(topic,courseid) VALUES("HTML - Basics",1),

("CSS - Basics",2),

("Node - Basics",3),

("JavaScript - Basics",4),

("React - Basics",5);

**# tasks**

INSERT INTO tasks(task,courseid) VALUES ("HTML Task",1),

("CSS Task",2),

("Bootstrap Task",3),

("JavaScript Task",4),

("React Task",5);

**# company drives**

INSERT INTO company\_drives(userid,company) VALUES (3,"Apple"),

(2,"Amazon")

(4,"Flipkart")

(1,"Zomato");

(5,"TCS")

**# mentors**

INSERT INTO mentors(mentorname,mentoremail) VALUES ("Arun","Arun12@gmail.com"),

("Vijay","Vijay34@gmail.com"),

("Karthi","Karthi56@gmail.com"),

("Mahesh","Mahesh78@gmail.com"),

("Gokul","Gokul90@gmail.com");

**# students activated courses**

INSERT INTO students\_activated\_courses(userid,courseid) VALUES(1,1),(2,1),(3,5),(4,2),(5,3);

**# courses**

INSERT INTO courses(coursename) VALUES("HTML"),("CSS"),("JavaScript"),("ReactJS"),("NodeJS");

**3. Get number problems solved in codekata by combining the users**

use students;

SELECT users.userid, users.username,users.useremail, codekata.number\_of\_problems

FROM users

INNER JOIN codekata ON users.userid = codekata.userid;

**4. Display the no of company drives attended by a user**

use students;

SELECT userid ,COUNT(userid) FROM company\_drives GROUP BY userid;

**5. Combine and display students\_activated\_courses and courses for a specific user groping them based on the course**

use students;

SELECT students\_activated\_courses.userid,students\_activated\_courses.courseid,

COUNT(students\_activated\_courses.courseid) ,courses.coursename

FROM students\_activated\_courses

INNER JOIN courses

ON students\_activated\_courses.courseid=courses.courseid

WHERE students\_activated\_courses.userid=1

GROUP BY courses.courseid;

**6. List all the mentors**

use students;

SELECT mentorname FROM mentors;

**7. List the number of students that are assigned for a mentor**

use students;

SELECT mentors.mentorid,mentors.mentorname,COUNT(users.mentorid)

FROM mentors,users

WHERE mentors.mentorid=users.mentorid

GROUP BY mentors.mentorid;