**SQL TASK**

|  |
| --- |
| **The Following are the tables has to be in your database & model deisgn:** |
|  |

|  |
| --- |
| users |

|  |
| --- |
| codekata |

|  |
| --- |
| Attendance |

|  |
| --- |
| topics |

|  |
| --- |
| tasks |

|  |
| --- |
| company\_drives |

|  |
| --- |
| mentors |

|  |
| --- |
| students\_activated\_courses |

|  |
| --- |
| courses |

|  |
| --- |
|  |

|  |
| --- |
| **The following are the queries need to be executed** |
|  |

|  |
| --- |
| 1. Create tables for the above list given |
| Ans: |

|  |
| --- |
| CREATE TABLE **users** (  user\_id int auto\_increment,  user\_name varchar(100),  user\_email varchar(100),  user\_gender varchar(20),  primary key(user\_id));  CREATE TABLE **codeKata** (  user\_id int not null,  total\_prob int not null,  solved\_count int,  earned\_points varchar(20),  primary key(user\_id));  CREATE TABLE **attendance** (  user\_id int not null,  total\_class int,  leave\_count int,  primary key(user\_id));  CREATE TABLE **topics** (  topic\_id int auto\_increment,  course\_id int not null,  topic varchar(200),  class\_required int,  primary key(topic\_id));  CREATE TABLE **company\_drives** (  company\_id varchar(100) ,  company\_name varchar(200),  total\_rounds int,  user\_attended int);  CREATE TABLE **mentors** (  mentor\_id int auto\_increment,  mentor\_name varchar(100),  mentor\_email varchar(100),  mentor\_gender varchar(20),  user\_id int,  primary key(mentor\_id));  CREATE TABLE **students\_activated\_courses** (  course\_id int,  student\_id int,  activated boolean);  CREATE TABLE **courses** (  course\_id int,  course\_name varchar(200),  primary key(course\_id));  2. insert at least 5 rows of values in each table |

|  |
| --- |
| Ans:  INSERT INTO **users** (user\_name,user\_email,user\_gender)  VALUES ('Ashley', 'ashley@mail.com','female'),  ('Ebin', 'ebin@mail.com','male'),  ('Clement', 'clement@gmail.com','male'),  ('Mark', 'mark@email.com','male'),  ('Merlin', 'merlin@yahoo.com','female');  INSERT INTO **codeKata** (user\_id,total\_prob,solved\_count,earned\_points)  VALUES (1,100,90,'900'),  (2,100,89,'890'),  (3,100,57,'570'),  (4,100,33,'330'),  (5,100,50,'500');  INSERT INTO **attendance** (user\_id,total\_class,leave\_count)  VALUES (1,100,1),  (2,100,0),  (3,100,4),  (4,100,2),  (5,100,10);  INSERT INTO **topics** (course\_id,topic,class\_required)  VALUES (101,"Joins",2),  (101,"DDL&DML statements",10),  (102,"Variable & datatypes",4),  (102,"DOM manipulation",3),  (103,"Components",2);  INSERT INTO **company\_drives** (company\_id,company\_name,total\_rounds,user\_attended)  VALUES ('D1',"Digital Techonologies",4,1),  ('E10',"Elite Techs",5,1),  ('K3',"Google",9,2),  ('Z14',"Zoho",5,4),  ('M8',"Microsoft",4,5);  INSERT INTO **mentors** (mentor\_name,mentor\_email,mentor\_gender,user\_id)  VALUES ("Nirmala","nirmala@gmail.com","female",4),  ("Seetha","setha@gmail.com","female",5),  ("Ram","ram@yahoo.com","male",3),  ("Sai","sai@gnail.com","male",1),  ('Amit',"amit@mail.com","male",2);  INSERT INTO **students\_activated\_courses** (course\_id,student\_id,activated)  VALUES (101,1,true), (101,2,false), (101,3,true), (101,4,false),  (101,5,true), (102,2,true), (103,3,false), (104,4,true),  (102,1,true),(105,5,false);  INSERT INTO **courses** (course\_id,course\_name)  VALUES (101,"DataBase"),  (102,"JS"),  (103,"React JS"),  (104,"Node JS"),  (105,"HTML");  3. get number problems solved in codekata by combining the users  Ans:  select u.user\_id,u.user\_name,c.solved\_count  from codekata c  inner join users u  on u.user\_id = c.user\_id |

|  |
| --- |
| 4. display the no of company drives attended by a user |
| Ans:  select c.user\_attended as user\_id, count(c.company\_id) as drives\_attended  from company\_drives c  where c.user\_attended =1 |
|  |

|  |
| --- |
| 5. combine and display students\_activated\_courses and courses for a specific user groping them based on the course  Ans:  select a.course\_id,c.course\_name,count(a.course\_id) from students\_activated\_courses a  inner join courses c on c.course\_id = a.course\_id  group by a.course\_id |

|  |
| --- |
| 6. list all the mentors |
| Ans:  select mentor\_name,mentor\_email,mentor\_gender from mentors; |

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

Ans:

select mentor\_id, mentor\_name, count(user\_id)

from mentors

group by mentor\_id

