**MySql\_Examination**

**Task- 1:**

Create two tables: users and orders.

Each user can have multiple orders.

Write a SQL query to fetch the names of users along with the total number of orders they have placed.

use hema;

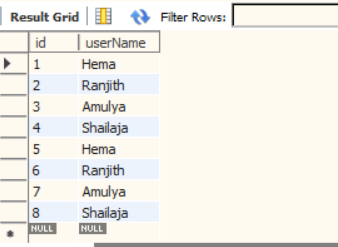
create table users(id int not null primary key auto\_increment, userName varchar(30));

show tables;

delete from users where id is NULL;

insert into users(userName) values('Hema'),('Ranjith'),('Amulya'),('Shailaja');

**select \* from users;**

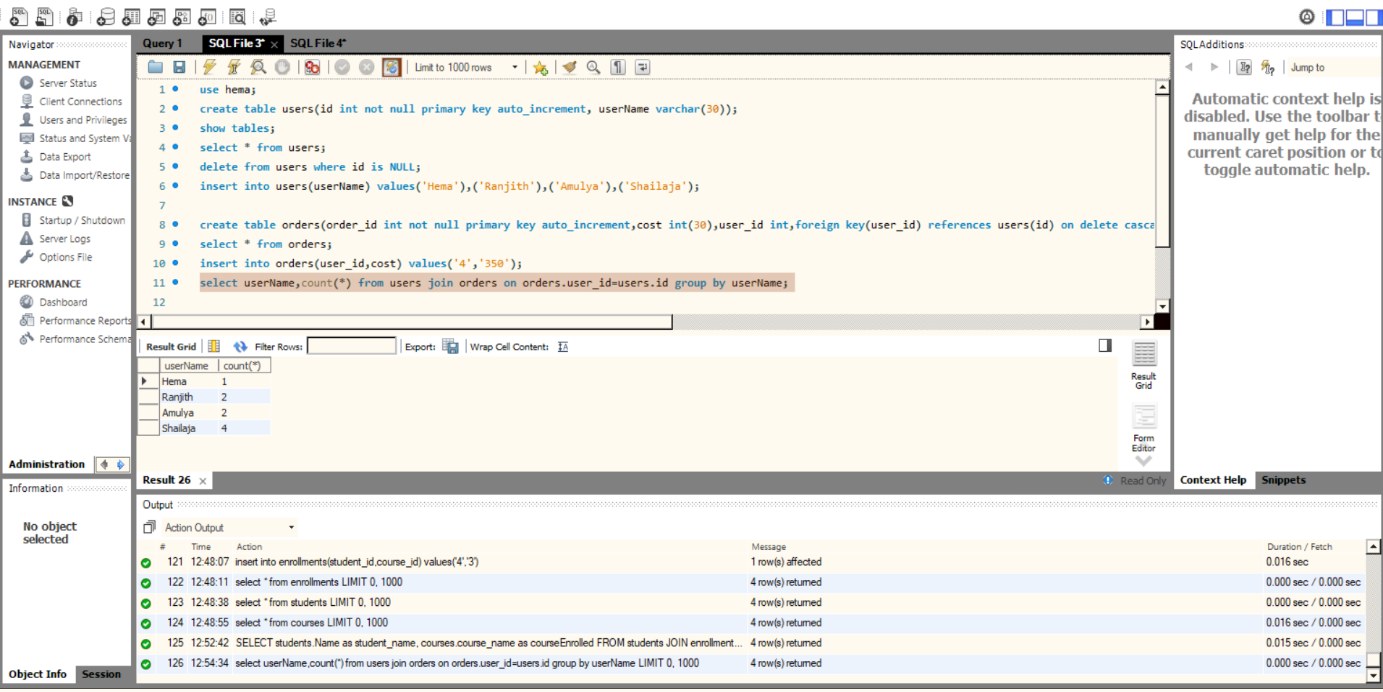


create table orders(order\_id int not null primary key auto\_increment,cost int(30),user\_id int,foreign key(user\_id) references users(id) on delete cascade);

select \* from orders;

insert into orders(user\_id,cost) values('4','350');

**select userName,count(\*) from users join orders on orders.user\_id=users.id group by userName;**



**Task 2**

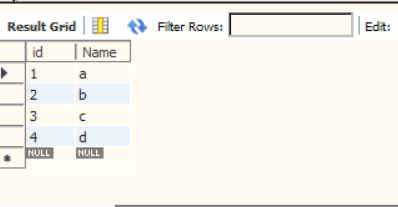
You are working with a database that stores information about students and their courses. There are three tables: students, courses, and enrollments.

Write a SQL query to display the names of students along with the courses they have enrolled in.

create table students(id int not null primary key auto\_increment, name varchar(30));

insert into students(name) values('d');

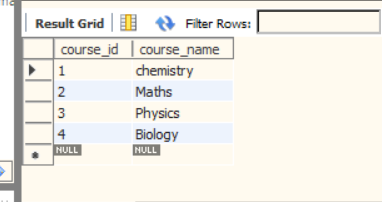
select \* from students;



create table courses (course\_id int not null primary key auto\_increment,course\_name varchar(30));

insert into courses(course\_name) values('Biology');

select \* from courses;



create table enrollments(student\_id int,course\_id int,foreign key(course\_id) references courses(course\_id));

select \* from enrollments;

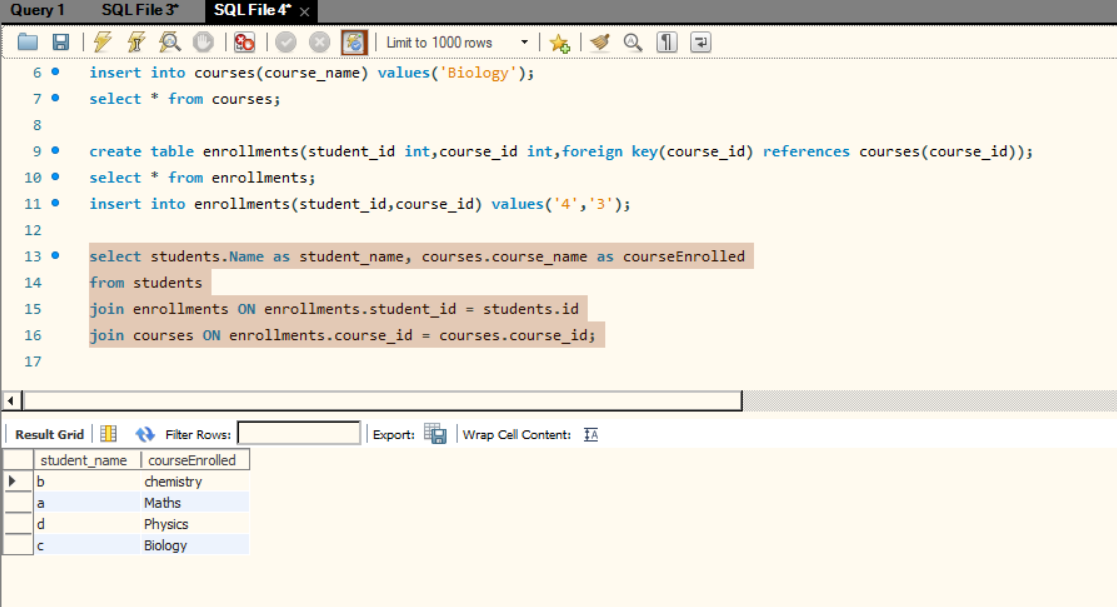
insert into enrollments(student\_id,course\_id) values('4','3');

select students.Name as student\_name, courses.course\_name as courseEnrolled

from students

join enrollments ON enrollments.student\_id = students.id

join courses ON enrollments.course\_id = courses.course\_id;



**Task-3:**

You need to retrieve data from a database that tracks product sales. There are tables for products, sales, and customers.

Write a SQL query to show the total sales amount for each product category.

select category, sum(price) from products group by category;

|  | **category** | **sum(price)** |
| --- | --- | --- |
|  | Electronic | 12000 |
|  | stationary | 60 |
|  | Plastic | 100 |
|  | Essentials | 1550 |
|  | Grocery | 100 |
|  | Clothing | 400 |

**Task-4:**

You have a database containing information about employees in a company.

Write a SQL query to list the names of employees along with their respective managers' names.

alter table employees add manager varchar (30);

select \* from employees;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | a | f | 22 | traniee |  |
| 2 | b | f | 22 | traniee |  |
| 3 | c | f | 22 | traniee |  |
| 4 | d | f | 22 | traniee |  |
| 5 | e | f | 22 | traniee |  |
| 6 | f | f | 22 | traniee |  |
| 7 | g | f | 22 | traniee |  |
| 8 | h | f | 22 | traniee |  |
| 9 | i | f | 22 | traniee |  |
| 10 | j | f | 22 | traniee |  |

update employees set manager='anuhya';

select \* from employees;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | a | f | 22 | traniee | anuhya |
| 2 | b | f | 22 | traniee | anuhya |
| 3 | c | f | 22 | traniee | anuhya |
| 4 | d | f | 22 | traniee | anuhya |
| 5 | e | f | 22 | traniee | anuhya |
| 6 | f | f | 22 | traniee | anuhya |
| 7 | g | f | 22 | traniee | anuhya |
| 8 | h | f | 22 | traniee | anuhya |
| 9 | i | f | 22 | traniee | anuhya |
| 10 | j | f | 22 | traniee | anuhya |

update employees set manager='Durga' where employee\_id in (1,2,5,8);

select \* from employees;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | a | f | 22 | traniee | Durga |
| 2 | b | f | 22 | traniee | Durga |
| 3 | c | f | 22 | traniee | anuhya |
| 4 | d | f | 22 | traniee | anuhya |
| 5 | e | f | 22 | traniee | Durga |
| 6 | f | f | 22 | traniee | anuhya |
| 7 | g | f | 22 | traniee | anuhya |
| 8 | h | f | 22 | traniee | Durga |
| 9 | i | f | 22 | traniee | anuhya |
| 10 | j | f | 22 | traniee | anuhya |

select employee\_name, manager from employees;

|  |  |
| --- | --- |
| a | Durga |
| b | Durga |
| c | anuhya |
| d | anuhya |
| e | Durga |
| f | anuhya |
| g | anuhya |
| h | Durga |
| i | anuhya |
| j | anuhya |

**Task-5:**

You are managing a database for an online store.

Write a query to retrieve the top 10 bestselling products based on the total number of units sold.

alter table products add unitsSold int;

update products set unitsSold='5' where product\_id="1";

update products set unitsSold='1' where product\_id="2";

update products set unitsSold='4' where product\_id="3";

update products set unitsSold='4' where product\_id="4";

update products set unitsSold='2' where product\_id="5";

update products set unitsSold='1' where product\_id="6";

update products set unitsSold='2' where product\_id="7";

update products set unitsSold='2' where product\_id="8";

update products set unitsSold='2' where product\_id="9";

update products set unitsSold='1' where product\_id="10";

select product\_name,unitsSold from products order by unitsSold desc limit 10;

product\_name|unitsSold

|  |  |
| --- | --- |
| realme | 5 |
| botle | 4 |
| watch | 4 |
| college bag | 2 |
| soap | 2 |
| pulses | 2 |
| pencil | 2 |
| book | 1 |
| Footwear | 1 |
| Pant | 1 |

**Task-6:**

You have tables for students, courses, and grades.

Write a SQL query to display the average grade for each student.

select \* from students;  
id|name

|  |  |
| --- | --- |
| 1 | a |
| 2 | b |
| 3 | c |
| 4 | d |
|  |  |

select \* from courses;  
course\_id, course\_name

|  |  |
| --- | --- |
| 1 | chemistry |
| 2 | Maths |
| 3 | Physics |
| 4 | Biology |
|  |  |

create table grades (student\_id int, course\_id int, grade int);

insert into grades values ('1','1','10'), ('1','2','10'), ('1','3','9'), ('1','4','8'), ('2','1','8'), ('2','2','10'), ('2','3','10'), ('2','4','7'),

('3','1','10'), ('3','2','10'), ('3','3','9'), ('3','4','5'), ('4','1','6'), ('4','2','10'), ('4','3','5'), ('4','4','6');

select \* from grades;

student\_id|course\_id|grade

|  |  |  |
| --- | --- | --- |
| 1 | 1 | 10 |
| 1 | 2 | 10 |
| 1 | 3 | 9 |
| 1 | 4 | 8 |
| 2 | 1 | 8 |
| 2 | 2 | 10 |
| 2 | 3 | 10 |
| 2 | 4 | 7 |
| 3 | 1 | 10 |
| 3 | 2 | 10 |
| 3 | 3 | 9 |
| 3 | 4 | 5 |
| 4 | 1 | 6 |
| 4 | 2 | 10 |
| 4 | 3 | 5 |
| 4 | 4 | 6 |

select students.name, AVG (grades.grade)as avg\_grade from grades join courses on courses.course\_id=grades.course\_id join students on students.id=grades.student\_id group by students.name;

name|avg\_grade

|  |  |
| --- | --- |
| a | 9.2500 |
| b | 8.7500 |
| c | 8.5000 |
| d | 6.7500 |

**Task-7:**

You are working with a database for a social media platform.

Write a query to show the users who have the most friends.

create table socialMedia (id int primary key not null auto\_increment,name varchar(30),friends int);

insert into socialMedia (name,friends) values('hema',500),('sandhya',200),('ranjith',1000),('raj',50),('meo',60);

select name, friends from socialMedia where friends>100 order by friends desc;

name|friends

|  |  |
| --- | --- |
| ranjith | 1000 |
| hema | 500 |
| sandhya | 200 |

**Task-8:**

You have tables for employees and departments.

Write a query to display the department names along with the total number of employees in each department.

create table departments (id int primary key not null auto\_increment,department\_name varchar(30));

insert into departments(department\_name) values ('cse'),('ece');

select \* from departments;

alter table employees add department\_id int, add constraint fk\_department foreign key(department\_id) references departments(id);

update employees set department\_id=1;

update employees set department\_id=2 where employee\_id in (2,5,6,8);

select \* from employees;

select department\_name,count(\*) as no\_of\_employees from departments join employees on departments.id=employees.department\_id group by employees.department\_id;

department\_name|no\_of\_employees

|  |  |
| --- | --- |
| cse | 6 |
| ece | 4 |

**Task-9:**

You need to retrieve data from a database tracking product inventory.

Write a query to display products with low stock (less than 10 units).

update clothing set In\_stock='No' where cloth\_id in (4,6);

select cloth\_id, cloth\_name from clothing where In\_stock='No';

cloth\_id|cloth\_name

|  |  |
| --- | --- |
| 1 | t-shirt |
| 4 | pant |
| 6 | kurthi |
| 9 | t-shirt |

**Task-10:**

You have tables for customers and orders.

Write a query to show the average order value for each customer.

select userName, Avg(cost) from orders join users on users.id=orders.user\_id group by orders.user\_id;

|  |  |
| --- | --- |
| Hema | 200.0000 |
| Shailaja | 350.0000 |
| Ranjith | 130.0000 |
| Amulya | 225.0000 |

**Task-11:**

In a database storing movie information,

Write a query to show the top 5 highest-rated movies by users.

create table ratings (id int not null primary key auto\_increment, movie\_name varchar(30),rating int,

user\_id int,foreign key(user\_id) references users(id));

insert into ratings(movie\_name, rating,user\_id) values('Bahubali',10,1),('Patas',9,2),('Mr Perfect',9,3),('Agadu',4,4),('100% Love',8,5),('RRR',10,2);

select userName,movie\_name,rating from ratings join users on ratings.user\_id=users.id order by rating desc limit 5;

|  |  |  |
| --- | --- | --- |
| Hema | Bahubali | 10 |
| Ranjith | RRR | 10 |
| Ranjith | Patas | 9 |
| Amulya | Mr Perfect | 9 |
| Hema | 100% Love | 8 |

**Task-12:**

You have tables for invoices and payments.

Write a query to show the unpaid invoices and their total amount.

create table invoices (invoice\_id int primary key not null auto\_increment,Total\_amount int);

insert into invoices (Total\_amount) values (100),(400),(600),(400),(250),(350);

create table payments (user\_id int,invoice\_id int,foreign key(invoice\_id) references invoices(invoice\_id), payment\_status varchar(20));

insert into payments values(1,1,'No'), (1,2,'Yes'), (2,3,'No'), (3,4,'No'), (3,5,'No'), (4,6,'yes');

select user\_id, invoices.invoice\_id, payment\_status, Total\_amount from payments join invoices on payments.invoice\_id=invoices.invoice\_id where payment\_status='No';

user\_id|invoice\_id|payment\_status|Total\_amount

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 1 | No | 100 |
| 2 | 3 | No | 600 |
| 3 | 4 | No | 400 |
| 3 | 5 | No | 250 |