```
create database sqltasks;
use sqltasks;
```

Taskl1:

```
create table users (
   user_id int primary key,
   user_name varchar(30)
);
insert into users (user_id, user_name) values (1,'leela'), (2,'hima'), (3,'jnani'),(4,'sun'),(5,'niru');
create table orders (
   sr_no int,
   user_id int,
   amount decimal,
   foreign key (user_id) references users(user_id)
);
insert into orders(sr_no,user_id,amount) values (1,1,200), (2,4,200), (3,2,150), (4,1,250),
(5,3,300),(6,2,100),(7,5,120),(8,3,190);
select u.user_name, count(o.sr_no) as total_orders
from users u
left join orders o on u.user_id = o.user_id
group by u.user_name;


create table friendships (
   user_id int,
   friend_id int,
   foreign key (user_id) references users(user_id),
   foreign key (friend_id) references users(user_id)
);
```

```
insert into friendships(user_id, friend_id) values (1, 2), (1, 3), (2, 3);


select u.user_name, count(f.friend_id) as total_friends
from users u
join friendships f on u.user_id = f.user_id
group by u.user_name
order by total_friends desc
limit 1;


select u.user_name, avg(o.amount) as avg_order_value
from users u
join orders o on u.user_id = o.user_id
group by u.user_name;
```

Task2:

```
use sqltasks;
create table students (
   stu_id int primary key,
   stu_name varchar(30)
);
insert into students (stu_id, stu_name) values (1,'eela'),(2,'priya'),(3,'jnani'),(4,'lil');


create table courses (
   cour_id int primary key,
   cour_name varchar(30)
);
insert into courses(cour_id, cour_name) values (1,'python'), (2,'java'),(3,'c'),(4,'r');
```

```
create table enrollments (
    stu_id int,
    cour_id int,
    foreign key (stu_id) references students(stu_id),
    foreign key (cour_id) references courses(cour_id)
);
insert into enrollments(stu_id, cour_id) values(1,3),(1,2),(3, 1),(4,3),(2,3),(3,4),(2,4);

select s.stu_name, c.cour_name
from students s
join enrollments e on s.stu_id = e.stu_id
join courses c on e.cour_id = c.cour_id;

create table grades (
    student_id int,
    course_id int,
    grade decimal,
    foreign key (student_id) references students(stu_id),
    foreign key (course_id) references courses(cour_id)
);
insert into grades(student_id,course_id,grade) values(1, 2, 80), (1, 1, 90), (2, 2, 85);

select s.stu_name, avg(g.grade) as avg_grade
from students s
join grades g on s.stu_id = g.student_id
group by s.stu_name;
```

Task3:

```
create table products (
   prod_id int primary key,
   prod_name varchar(50),
   category varchar(50)
);
insert into products (prod_id,prod_name,category) values(1,'mobile','electronics'),
(2, 'fruits', 'food'),
(3, 'tops', 'clothing');

create table customers (
   cus_id int primary key,
   cus_name varchar(50)
);
insert into customers (cus_id, cus_name) values(1,'leela'),(2, 'hima');

create table sales (
   sal_id int primary key,
   prod_id int,
   cus_id int,
   amount decimal,
   foreign key (prod_id) references products(prod_id),
   foreign key (cus_id) references customers(cus_id)
);
insert into sales(sal_id, prod_id, cus_id, amount) values(1,2,1,500),(2, 3, 2, 1000),
(3,3,2,300),
(4,1,1,600);
```

```
select p.category, sum(s.amount) as total_sales
from products p
join sales s on p.prod_id = s.prod_id
group by p.category;

create table inventory (
   product_id int primary key,
   stock int,
   foreign key (product_id) references products(prod_id)
);

insert into inventory (product_id, stock) values (1, 5), (2, 15);

select p.prod_name, i.stock
from products p
join inventory i on p.prod_id = i.product_id
where i.stock < 10;

create table invoices (
   id int primary key,
   customer_id int,
   amount decimal(10,2),
   status varchar(20),
   foreign key (customer_id) references users(user_id)
);

insert into invoices (id, customer_id, amount, status) values (1, 1, 500, 'unpaid'), (2, 2, 300, 'paid');
```

```
select id, amount from invoices where status = 'unpaid';
```

Task:

```
use sqltasks;
drop table employees;
create table employees (
    id int primary key,
    name varchar(50),
    manager_id int,
    foreign key (manager_id) references employees(id)
);

insert into employees (id, name, manager_id) values (1, 'leela', 1), (2, 'hima', 2), (3, 'jnani', 1);

select e.name as employee, m.name as manager
from employees e
left join employees m on e.manager_id = m.id;



create table departments (
    id int primary key,
    name varchar(50)
);
```

Task11:

```
create table movies (
 sr_no int,
 m_name varchar(20),
 re_date date,
 m_status varchar(10),
 rating float
);
insert into movies (sr_no, m_name, re_date, m_status, rating)
values
(1, 'bahubali', '2017-09-18', 'hit', 5.0),
(2, 'sir', '2023-12-19', 'hit', 4.0),
(3, 'mad', '2023-03-15', 'hit', 3.0),
(4, 'lila', '2025-04-20', 'flop', 2.0),
(5, 'kil', '2023-05-20', 'flop', 3.5),
(6, 'avatar', '2009-12-18', 'hit', 5.0),
(7, 'titanic', '2000-04-01', 'hit', 5.0),
(8, 'toli', '2019-10-04', 'flop', 3.5),
(9, 'jo', '1999-03-31', 'flop', 2.0),
(10, 'kalki', '2024-08-26', 'hit', 5.0);

select * from movies;

select *  from movies order by rating desc limit 5;


alter table employees add column department_id int;
alter table employees add foreign key (department_id) references departments(id);
```

insert into departments (id, name) values (1, 'hr'), (2, 'it');

```
insert into employees (id, name, department_id) values (4, 'arun', 1), (5, 'nikas', 2);

select d.name as department, count(e.id) as total_employees
from departments d
left join employees e on d.id = e.department_id
group by d.name;
```