

Task 1:

```
create table if not exists customers (  
  cust_ID int primary key auto_increment,  
  title varchar(20) not null,  
  first_name varchar(20) not null,  
  middle_name varchar(20),  
  last_name varchar(20) not null,  
  suffix varchar(4),  
  email varchar(20),  
  company varchar(20),  
  display_name varchar(20),  
  print_on_check_as varchar(20),  
  billing_street varchar(20) not null,  
  billing_city varchar(20) not null,  
  billing_state varchar(20) not null,  
  billing_zip int not null,  
  billing_country varchar(20) not null,  
  shipping_street varchar(20),  
  shipping_city varchar(20),  
  shipping_state varchar(20),  
  shipping_zip int,  
  shipping_country varchar(20) -- if shipping info null, then same as billing  
);
```

```
create table if not exists orders(  
  order_ID int primary key auto_increment,  
  cust_ID int not null,  
  invoice_creation_date date,  
  delivery_due_date date,  
  payment_due_date date,  
  custom_message varchar(20),  
  foreign key (cust_ID) references customers(cust_ID)  
);
```

```
create table if not exists products (  
  product_ID_PK int primary key auto_increment,  
  name varchar(20) not null,  
  description varchar(100),  
  price double not null  
);
```

```
create table if not exists orderproducts (  
  order_product_ID_PK int primary key auto_increment,  
  product_ID int auto_increment,
```

```
quantity int not null,  
foreign key (order_ID) references orders(order_ID),  
foreign key (product_ID) references products(product_ID_PK)  
);
```

Task 2:

```
insert into customers  
values
```

```
('customer1', 'McKenzie', 'Treasure', 'Church', 'Ms.', 'tchurch1@my.westga.edu',  
null, 'tchurch1', null, 'Maple St.', 'Carrollton', 'Georgia', 30117, 'United States'),  
( 'customer2', 'Raihan', null, 'Ahmed', 'Mr.', 'rahmed@westga.edu', 'UWG', 'rahmed',  
null, 'Maple St.', 'Carrollton', 'GA', 30117, 'United States'),  
( 'customer3', 'Provolone', 'the Cat', 'Church', 'Sir', null, null, 'provolonethekitty',  
'Sir Provolone', 'Lovvorn Rd', 'Carrollton', 'Georgia', 30117, 'U.S.A');
```

```
insert into products
```

```
values
```

```
('large box', 'large beautiful box for cat to sit in.', 50.00),  
( 'small box', 'small box for human to buy for cat to sit in.', 100.99),  
( 'socks', 'socks for human to use, but cat steals anyways', 20.50);
```

```
insert into orders
```

```
values
```

```
(1, date(now()), 2023-02-03, 2023-02-03, null),  
(1, 2022-12-25, 2022-12-31, 2022-12-25, 'merry christmas'),  
(2, 2023-01-20, 2023-02-01, 2023-01-20, 'I will love this box.');
```

```
insert into orderproducts
```

```
values
```

```
(1, 3, 2),  
(1, 1, 1),  
(3, 3, 1);
```

Task 3:

```
-- Find the payment due dates for all the orders where the due date is less than the  
-- current date to find out overdue payments. You have to show the invoice number  
-- and customer's full name
```

```
select orders.payment_due_date as 'Payment Due',  
orders.invoice_creation_date as 'Invoice Num.',  
concat(customers.first_name, ' ', customers.middle_name, ' ', customers.last_name)  
from orders  
join customers  
on orders.custID_FK=customers.cust_ID_PK
```

```
where orders.payment_due_date < date(now());
```

```
-- Find the products bought by a customer with the first name John. You must show  
-- the product names
```

```
select products.name as 'Product',  
customers.first_name as 'First Name'  
from products  
join orderproducts  
on products.product_ID=orderproducts.product_ID  
join orders  
on customers.cust_ID_PK=orders.cust_ID_FK  
where customers.first_name='John';
```

```
-- Find the products (only name) sold in the month of February
```

```
select products.name as 'Product'  
from products  
join orderproducts  
on orderproducts.product_ID_FK=products.product_ID  
join orders  
on orders.order_ID=orderproducts.order_ID  
where orderproducts.order_ID < 2023-02-01 and orderproducts.order_ID > 2023-02-28;
```

```
-- Find the order total for order id 3
```

```
select sum(products.price) as 'Total Price'  
from products  
join orderproducts  
on products.product_ID=orderproducts.product_ID_FK  
where orderproducts.order_ID=3;
```

```
-- Find the best selling product in the year 2022
```

```
select max(orderproducts.quantity) as 'Amount', products.name as 'Product'  
from orderproducts  
join products  
on orderproducts.product_ID_FK=products.product_ID;
```