

# TechShop, An electronic gadgets shop

## Task:1. Database Design:

### 1. Create the database named "TechShop"

```
create database techshop;  
use techshop;
```

```
1 • create database techshop;  
2 • use techshop;
```


### 2. Define the schema for the Customers, Products, Orders, OrderDetails and Inventory tables based on the provided schema.


### 4. Create appropriate Primary Key and Foreign Key constraints for referential integrity.


#### #1 CUSTOMERS TABLE

```
create table Customers(  
CustomerID int not null auto_increment,  
FirstName varchar(20) not null,  
LastName varchar(20) not null,  
Email varchar(30),  
Phone int,  
Address varchar(100),  
primary key (CustomerID),  
unique(Email,Phone)  
);
```

Result Grid


Filter Rows:

Export:


Wrap Cell Content:


	Field	Type	Null	Key	Default	Extra
▶	CustomerID	int	NO	PRI	NULL	auto_increment
	FirstName	varchar(20)	NO		NULL	
	LastName	varchar(20)	NO		NULL	
	Email	varchar(30)	YES	MUL	NULL	
	Phone	int	YES		NULL	
	Address	varchar(100)	YES		NULL	

## #2 PRODUCTS TABLE

```
create table Products(  
ProductID int not null auto_increment,  
ProductName varchar(20) not null,  
Description varchar(30),  
Price int not null,  
primary key (ProductID)  
);
```

Result Grid

Filter Rows:

Export:


Wrap Cell Content:


	Field	Type	Null	Key	Default	Extra
▶	ProductID	int	NO	PRI	NULL	auto_increment
	ProductName	varchar(20)	NO		NULL	
	Description	varchar(30)	YES		NULL	
	Price	int	NO		NULL	


## #3 ORDERS TABLE

```
create table Orders(  
OrderID int not null auto_increment,  
CustomerID int not null,  
OrderDate date,  
TotalAmount decimal(7,2),  
primary key (OrderID),  
foreign key (CustomerID) references Customers(CustomerID) on delete cascade  
);
```

Result Grid


Filter Rows:

Export:


Wrap Cell Content:


	Field	Type	Null	Key	Default	Extra
▶	OrderID	int	NO	PRI	NULL	auto_increment
	CustomerID	int	NO	MUL	NULL	
	OrderDate	date	YES		NULL	
	TotalAmount	decimal(7,2)	YES		NULL	

## #4 ORDER DETAILS TABLE

```
create table OrderDetails(  
OrderDetailID int not null auto_increment,  
OrderID int not null,  
ProductID int not null,  
Quantity int not null,  
primary key (OrderDetailID),  
foreign key (OrderID) references Orders(OrderID) on delete cascade,
```

foreign key(ProductID) references Products(ProductID) on delete cascade  
);

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	Field	Type	Null	Key	Default	Extra
▶	orderdetailid	int	NO	PRI	NULL	auto_increment
	orderid	int	NO	MUL	NULL	
	productid	int	NO	MUL	NULL	
	quantity	int	NO		NULL	

## #5 INVENTORY TABLE

```
create table Inventory(
InventoryID int not null auto_increment,
ProductID int not null,
QuantityInStock int,
LastStockUpdate datetime,
primary key (InventoryID),
foreign key(ProductID) references Products(ProductID) on delete cascade
);
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	Field	Type	Null	Key	Default	Extra
▶	InventoryID	int	NO	PRI	NULL	auto_increment
	ProductID	int	NO	MUL	NULL	
	QuantityInStock	int	YES		NULL	
	LastStockUpdate	datetime	YES		NULL	

## 5. Insert at least 10 sample records into each of the following tables.

### a) Customers

```
insert into customers(firstname,lastname,email,phone,address) values
('Robert','Deniro','robden@gmail.com',1234567890,'houston'),
('al','pacino','alpac@gmail.com',1234567891,'madrid'),
('christopher','nolan','chrisnol@gmail.com',1234567892,'london'),
('martin','scorsese','marscor@gmail.com',1234567893,'new york'),
('cilian','murphy','cilmurph@gmail.com',1234567894,'belfast'),
('denzel','washington','denwash@gmail.com',1234567895,'DC'),
('hugh','jackman','hugjack@gmail.com',1234567896,'sydney'),
('dev','patel','patel@gmail.com',1234567897,'delhi'),
('ryan','reynolds','ryanrey@gmail.com',1234567898,'toronto'),
('brad','pitt','bpit@gmail.com',1234567899,'las vegas');
desc customers;
select * from customers;
```

	CustomerID	FirstName	LastName	Email	Phone	Address
▶	1	Robert	Deniro	robden@gmail.com	1234567890	houston
	2	al	pacino	alpac@gmail.com	1234567891	madrid
	3	christopher	nolan	chrisnol@gmail.com	1234567892	london
	4	martin	scorsese	mar scor@gmail.com	1234567893	new york
	5	cilian	murphy	cilmurph@gmail.com	1234567894	belfast
	6	denzel	washington	denwash@gmail.com	1234567895	DC
	7	hugh	jackman	hugjack@gmail.com	1234567896	sydney
	8	dev	patel	patel@gmail.com	1234567897	delhi
	9	ryan	reynolds	ryanrey@gmail.com	1234567898	toronto
	10	brad	pitt	bpit@gmail.com	1234567899	las vegas
*	NULL	NULL	NULL	NULL	NULL	NULL

## b) Products

insert into Products(productname,description,price) values

('laptops','gaming laptops',50000),  
('antivirus','safe and secure',2000),  
('ethernet','high speed internet',1000),  
('tablet','handy and portable',20000),  
('stylus','draw imagination',3000),  
('GPU','visual rendering',40000),  
('CPU','high speed process',40000),  
('RAM','quick multitasking',8000),  
('ROM','big storage access',8000),  
('monitor','HD display monitor',5000);

desc products;

select \* from products;

	ProductID	ProductName	Description	Price
▶	1	laptops	gaming laptops	50000
	2	antivirus	safe and secure	2000
	3	ethernet	high speed internet	1000
	4	tablet	handy and portable	20000
	5	stylus	draw imagination	3000
	6	GPU	visual rendering	40000
	7	CPU	high speed process	40000
	8	RAM	quick multitasking	8000
	9	ROM	big storage access	8000
	10	monitor	HD display monitor	5000
*	NULL	NULL	NULL	NULL

## c) Orders

insert into orders(customerid,orderdate,totalamount) values

(1,'2025-03-01',50000.00), #1-laptop  
(2,'2025-03-03',2000.00), #1-antivirus

```

(3,'2025-03-05',3000.00), #3-ethernet
(4,'2025-03-07',16000.00), #2-ram
(5,'2025-03-08',15000.00), #3-monitor
(6,'2025-03-09',80000.00), #2-cpu
(7,'2025-03-11',16000.00), #2-rom
(8,'2025-03-14',40000.00), #1-gpu
(9,'2025-03-16',40000.00), #2-tablet
(10,'2025-03-17',9000.00); #3-stylus
desc orders;
select * from orders;

```

	OrderID	CustomerID	OrderDate	TotalAmount
▶	1	1	2025-03-01	50000.00
	2	2	2025-03-03	2000.00
	3	3	2025-03-05	3000.00
	4	4	2025-03-07	16000.00
	5	5	2025-03-08	15000.00
	6	6	2025-03-09	80000.00
	7	7	2025-03-11	16000.00
	8	8	2025-03-14	40000.00
	9	9	2025-03-16	40000.00
	10	10	2025-03-17	9000.00
*	NULL	NULL	NULL	NULL

#### d) OrderDetails

```

insert into OrderDetails(orderid,productid,quantity) values
(1,1,1), #1-laptop
(2,2,1), #1-antivirus
(3,3,3), #3-ethernet
(4,8,2), #2-ram
(5,10,3),#3-monitor
(6,7,2), #2-cpu
(7,9,2), #2-rom
(8,6,1), #1-gpu
(9,4,2), #2-tablet
(10,5,3); #3-stylus
desc orderdetails;
select * from orderdetails;

```

	orderdetailid	orderid	productid	quantity
▶	1	1	1	1
	2	2	2	1
	3	3	3	3
	4	4	8	2
	5	5	10	3
	6	6	7	2
	7	7	9	2
	8	8	6	1
	9	9	4	2
	10	10	5	3
*	NULL	NULL	NULL	NULL

### e) Inventory

insert into inventory(productid,quantityinstock,laststockupdate) values

(1,100,'2025-03-18 10:00:00'), #laptop

(2,200,'2025-03-18 10:00:00'), #antivirus

(3,50,'2025-03-18 10:00:00'), #ethernet

(4,80,'2025-03-18 10:00:00'), #tablet

(5,80,'2025-03-18 10:00:00'), #stylus

(6,50,'2025-03-18 10:00:00'), #GPU

(7,50,'2025-03-18 10:00:00'), #CPU

(8,100,'2025-03-18 10:00:00'), #RAM

(9,100,'2025-03-18 10:00:00'), #ROM

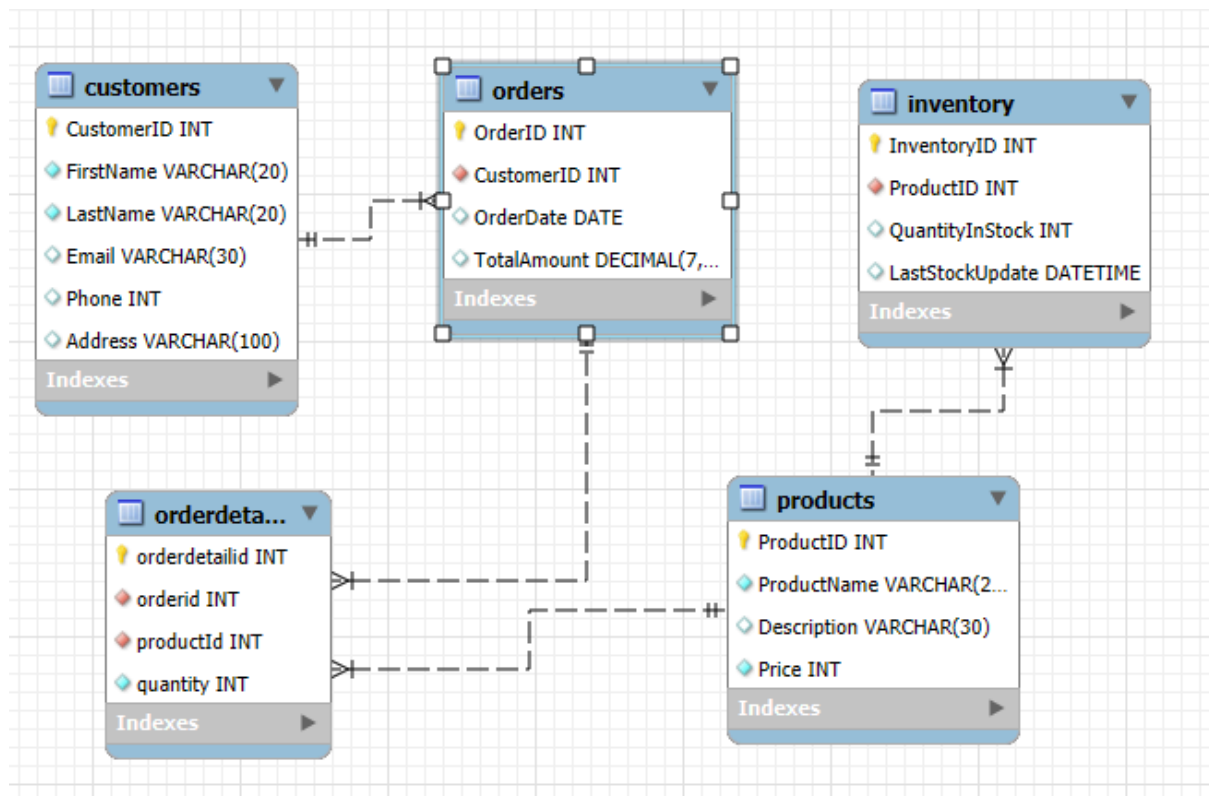
(10,50,'2025-03-18 10:00:00'); #monitor

desc inventory;

select \* from inventory;

	InventoryID	ProductID	QuantityInStock	LastStockUpdate
▶	1	1	100	2025-03-18 10:00:00
	2	2	200	2025-03-18 10:00:00
	3	3	50	2025-03-18 10:00:00
	4	4	80	2025-03-18 10:00:00
	5	5	80	2025-03-18 10:00:00
	6	6	50	2025-03-18 10:00:00
	7	7	50	2025-03-18 10:00:00
	8	8	100	2025-03-18 10:00:00
	9	9	100	2025-03-18 10:00:00
	10	10	50	2025-03-18 10:00:00
*	NULL	NULL	NULL	NULL

### 3. Create an ERD (Entity Relationship Diagram) for the database.



## Tasks 2: Select, Where, Between, AND, LIKE:

### 1. Write an SQL query to retrieve the names and emails of all customers.

#1

```
select concat(firstname,'_',lastname) as Fullname,email from customers;
```

	Fullname	email
▶	Robert_Deniro	robden@gmail.com
	al_pacino	alpac@gmail.com
	christopher_nolan	chrisnol@gmail.com
	martin_scorsese	marscor@gmail.com
	cilian_murphy	cilmurph@gmail.com
	denzel_washington	denwash@gmail.com
	hugh_jackman	hugjack@gmail.com
	dev_patel	patel@gmail.com
	ryan_reynolds	ryanrey@gmail.com
	brad_pitt	bpit@gmail.com

### 2. Write an SQL query to list all orders with their order dates and corresponding customer names.

#2

select

orders.orderid,orders.orderdate,concat(customers.firstname,'\_',customers.lastname)  
as Fullname

from orders join customers on orders.customerid = customers.customerid;

	orderid	orderdate	Fullname
▶	1	2025-03-01	Robert_Deniro
	2	2025-03-03	al_pacino
	3	2025-03-05	christopher_nolan
	4	2025-03-07	martin_scorsese
	5	2025-03-08	cilian_murphy
	6	2025-03-09	denzel_washington
	7	2025-03-11	hugh_jackman
	8	2025-03-14	dev_patel
	9	2025-03-16	ryan_reynolds
	10	2025-03-17	brad_pitt

**3. Write an SQL query to insert a new customer record into the "Customers" table. Include customer information such as name, email, and address.**

#3

insert into customers(firstname,lastname,email,address)

values('chris','evans','eva@gmail.com','chicago');

select \* from customers;

	CustomerID	FirstName	LastName	Email	Phone	Address
▶	1	Robert	Deniro	robden@gmail.com	1234567890	houston
	2	al	pacino	alpac@gmail.com	1234567891	madrid
	3	christopher	nolan	chrisnol@gmail.com	1234567892	london
	4	martin	scorsese	marscor@gmail.com	1234567893	new york
	5	cilian	murphy	cilmurph@gmail.com	1234567894	belfast
	6	denzel	washington	denwash@gmail.com	1234567895	DC
	7	hugh	jackman	hugjack@gmail.com	1234567896	sydney
	8	dev	patel	patel@gmail.com	1234567897	delhi
	9	ryan	reynolds	ryanrey@gmail.com	1234567898	toronto
	10	brad	pitt	bpit@gmail.com	1234567899	las vegas
	11	chris	evans	eva@gmail.com	NULL	chicago
*	NULL	NULL	NULL	NULL	NULL	NULL

**4. Write an SQL query to update the prices of all electronic gadgets in the "Products" table by increasing them by 10%.**

#4

update products

set price=price\*1.1;



select \* from products;

	ProductID	ProductName	Description	Price
▶	1	laptops	gaming laptops	55000
	2	antivirus	safe and secure	2200
	3	ethernet	high speed internet	1100
	4	tablet	handy and portable	22000
	5	stylus	draw imagination	3300
	6	GPU	visual rendering	44000
	7	CPU	high speed process	44000
	8	RAM	quick multitasking	8800
	9	ROM	big storage access	8800
	10	monitor	HD display monitor	5500
*	NULL	NULL	NULL	NULL

**5. Write an SQL query to delete a specific order and its associated order details from the "Orders" and "OrderDetails" tables. Allow users to input the order ID as a parameter.**

#5

delete from customers where customerid=1;

select \* from customers;

	CustomerID	FirstName	LastName	Email	Phone	Address
▶	2	al	pacino	alpac@gmail.com	1234567891	madrid
	3	christopher	nolan	chrisnol@gmail.com	1234567892	london
	4	martin	scorsese	mar scor@gmail.com	1234567893	new york
	5	cilian	murphy	cilmurph@gmail.com	1234567894	belfast
	6	denzel	washington	denwash@gmail.com	1234567895	DC
	7	hugh	jackman	hugjack@gmail.com	1234567896	sydney
	8	dev	patel	patel@gmail.com	1234567897	delhi
	9	ryan	reynolds	ryanrey@gmail.com	1234567898	toronto
	10	brad	pitt	bpit@gmail.com	1234567899	las vegas
	11	chris	evans	eva@gmail.com	NULL	chicago
*	NULL	NULL	NULL	NULL	NULL	NULL

**6. Write an SQL query to insert a new order into the "Orders" table. Include the customer ID, order date, and any other necessary information.**

#6

insert into customers(firstname,lastname,email,phone,address) values

('henry','cavill','cavil@gmail.com','1234567880','manchester');

insert into orders(customerid,orderdate,totalamount) values

(last\_insert\_id(),'2025-03-18',11000.00);

#5 antivirus bought

select \* from orders;

	OrderID	CustomerID	OrderDate	TotalAmount
▶	2	2	2025-03-03	2000.00
	3	3	2025-03-05	3000.00
	4	4	2025-03-07	16000.00
	5	5	2025-03-08	15000.00
	6	6	2025-03-09	80000.00
	7	7	2025-03-11	16000.00
	8	8	2025-03-14	40000.00
	9	9	2025-03-16	40000.00
	10	10	2025-03-17	9000.00
	11	12	2025-03-18	11000.00
*	NULL	NULL	NULL	NULL

**7. Write an SQL query to update the contact information (e.g., email and address) of a specific customer in the "Customers" table. Allow users to input the customer ID and new contact information.**

#7

```
update customers
set email='jackman@gmail.com',address='perth'
where customerid=7;
select * from customers;
```

	CustomerID	FirstName	LastName	Email	Phone	Address
▶	2	al	pacino	alpac@gmail.com	1234567891	madrid
	3	christopher	nolan	chrisnol@gmail.com	1234567892	london
	4	martin	scorsese	marscor@gmail.com	1234567893	new york
	5	cilian	murphy	cilmurph@gmail.com	1234567894	belfast
	6	denzel	washington	denwash@gmail.com	1234567895	DC
	7	hugh	jackman	jackman@gmail.com	1234567896	perth
	8	dev	patel	patel@gmail.com	1234567897	delhi
	9	ryan	reynolds	ryanrey@gmail.com	1234567898	toronto
	10	brad	pitt	bpit@gmail.com	1234567899	las vegas
	11	chris	evans	eva@gmail.com	NULL	chicago
	12	henry	cavill	cavil@gmail.com	1234567880	manches...
*	NULL	NULL	NULL	NULL	NULL	NULL

**8. Write an SQL query to recalculate and update the total cost of each order in the "Orders" table based on the prices and quantities in the "OrderDetails" table.**

#8

```
update orders o
join(
  select od.orderid, sum(p.price * od.quantity) as newtotal
  from orderdetails od
  join products p on od.productid = p.productid
```

```

group by od.orderid
)
as calculated_totals on o.orderid = calculated_totals.orderid
set o.totalamount = calculated_totals.newtotal;
select * from orderdetails;

```

	orderdetailid	orderid	productid	quantity
▶	2	2	2	1
	3	3	3	3
	4	4	8	2
	5	5	10	3
	6	6	7	2
	7	7	9	2
	8	8	6	1
	9	9	4	2
	10	10	5	3
*	NULL	NULL	NULL	NULL

**9. Write an SQL query to delete all orders and their associated order details for a specific customer from the "Orders" and "OrderDetails" tables. Allow users to input the customer ID as a parameter.**

```

#9
delete from orderdetails
where orderid in(select orderid from orders where customerid = 2);
delete from orders
where orderid=2;

```

	orderdetailid	orderid	productid	quantity
▶	3	3	3	3
	4	4	8	2
	5	5	10	3
	6	6	7	2
	7	7	9	2
	8	8	6	1
	9	9	4	2
	10	10	5	3
*	NULL	NULL	NULL	NULL

**10. Write an SQL query to insert a new electronic gadget product into the "Products" table, including product name, category, price, and any other relevant details.**

```

#10
insert into products(productname,description,price) values
('wi-fi','wireless connectivity',1100);
select * from products;

```

	ProductID	ProductName	Description	Price
▶	1	laptops	gaming laptops	55000
	2	antivirus	safe and secure	2200
	3	ethernet	high speed internet	1100
	4	tablet	handy and portable	22000
	5	stylus	draw imagination	3300
	6	GPU	visual rendering	44000
	7	CPU	high speed process	44000
	8	RAM	quick multitasking	8800
	9	ROM	big storage access	8800
	10	monitor	HD display monitor	5500
	11	wi-fi	wireless connectivity	1100
*	NULL	NULL	NULL	NULL

**11. Write an SQL query to update the status of a specific order in the "Orders" table (e.g., from "Pending" to "Shipped"). Allow users to input the order ID and the new status.**

#11

alter table orders

add column orderstatus varchar(10) default 'pending';

update orders

set orderstatus='shipped'

where orderid=3;

	OrderID	CustomerID	OrderDate	TotalAmount	orderstatus
▶	3	3	2025-03-05	3300.00	shipped
	4	4	2025-03-07	17600.00	pending
	5	5	2025-03-08	16500.00	pending
	6	6	2025-03-09	88000.00	pending
	7	7	2025-03-11	17600.00	pending
	8	8	2025-03-14	44000.00	pending
	9	9	2025-03-16	44000.00	pending
	10	10	2025-03-17	9900.00	pending
	11	12	2025-03-18	11000.00	pending
*	NULL	NULL	NULL	NULL	NULL

**12. Write an SQL query to calculate and update the number of orders placed by each customer in the "Customers" table based on the data in the "Orders" table.**

#12

alter table customers

add column totalorders int default 0;

update customers

set totalorders=(

select count(\*)

from orders  
 where orders.customerid=customers.customerid  
 );

	CustomerID	FirstName	LastName	Email	Phone	Address	totalorders
▶	2	al	pacino	alpac@gmail.com	1234567891	madrid	0
	3	christopher	nolan	chrisnol@gmail.com	1234567892	london	1
	4	martin	scorsese	mar scor@gmail.com	1234567893	new york	1
	5	cilian	murphy	cilmurph@gmail.com	1234567894	belfast	1
	6	denzel	washington	denwash@gmail.com	1234567895	DC	1
	7	hugh	jackman	jackman@gmail.com	1234567896	perth	1
	8	dev	patel	patel@gmail.com	1234567897	delhi	1
	9	ryan	reynolds	ryanrey@gmail.com	1234567898	toronto	1
	10	brad	pitt	bpit@gmail.com	1234567899	las vegas	1
	11	chris	evans	eva@gmail.com	NULL	chicago	0
	12	henry	cavill	cavil@gmail.com	1234567880	manches...	1
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

### Task 3. Aggregate functions, Having, Order By, GroupBy and Joins:

1. Write an SQL query to retrieve a list of all orders along with customer information (e.g., customer name) for each order.

#1

```
select o.orderid, o.orderdate, o.totalamount, CONCAT(c.firstname, ' ', c.lastname) as
customername,
c.Email, c.Phone, c.Address from orders o
join customers c on o.customerid = c.customerid
order by o.orderdate;
```

	orderid	orderdate	totalamount	customername	Email	Phone	Address
▶	3	2025-03-05	3300.00	christopher nolan	chrisnol@gmail.com	1234567892	london
	4	2025-03-07	17600.00	martin scorsese	mar scor@gmail.com	1234567893	new york
	5	2025-03-08	16500.00	cilian murphy	cilmurph@gmail.com	1234567894	belfast
	6	2025-03-09	88000.00	denzel washington	denwash@gmail.com	1234567895	DC
	7	2025-03-11	17600.00	hugh jackman	jackman@gmail.com	1234567896	perth
	8	2025-03-14	44000.00	dev patel	patel@gmail.com	1234567897	delhi
	9	2025-03-16	44000.00	ryan reynolds	ryanrey@gmail.com	1234567898	toronto
	10	2025-03-17	9900.00	brad pitt	bpit@gmail.com	1234567899	las vegas
	11	2025-03-18	11000.00	henry cavill	cavil@gmail.com	1234567880	manchester

2. Write an SQL query to find the total revenue generated by each electronic gadget product. Include the product name and the total revenue.

```
select p.productname, SUM(od.quantity * p.price) as totalrevenue
from orderdetails od
```

```

join products p on od.productid = p.productid
group by p.productname
order by totalrevenue desc;

```

	productname	totalrevenue
▶	CPU	88000
	GPU	44000
	tablet	44000
	RAM	17600
	ROM	17600
	monitor	16500
	stylus	9900
	ethernet	3300

**3. Write an SQL query to list all customers who have made at least one purchase. Include their names and contact information.**

#3

```

select distinct c.customerid, CONCAT(c.firstname, ' ', c.lastname) as customername,
c.Email, c.Phone, c.Address from customers c
join orders o on c.customerid = o.customerid
order by customername;

```

	customerid	customername	Email	Phone	Address
▶	10	brad pitt	bpitt@gmail.com	1234567899	las vegas
	3	christopher nolan	chrisnol@gmail.com	1234567892	london
	5	cilian murphy	cilmurph@gmail.com	1234567894	belfast
	6	denzel washington	denwash@gmail.com	1234567895	DC
	8	dev patel	patel@gmail.com	1234567897	delhi
	12	henry cavill	cavil@gmail.com	1234567880	manchester
	7	hugh jackman	jackman@gmail.com	1234567896	perth
	4	martin scorsese	marscor@gmail.com	1234567893	new york
	9	ryan reynolds	ryanrey@gmail.com	1234567898	toronto

**4. Write an SQL query to find the most popular electronic gadget, which is the one with the highest total quantity ordered. Include the product name and the total quantity ordered.**

#4

```

select p.productname, SUM(od.quantity) as totalquantityordered
from orderdetails od
join products p on od.productid = p.productid
group by p.productid, p.productname
order by totalquantityordered desc
limit 1;

```

	productname	totalquantityordered
▶	ethernet	3

**5. Write an SQL query to retrieve a list of electronic gadgets along with their corresponding categories.**

```
alter table products add column category varchar(50);
update products
set category = 'Electronic Gadgets'
where productname in ('laptops', 'tablet', 'GPU', 'CPU', 'RAM', 'ROM', 'monitor',
'stylus', 'ethernet', 'wi-fi');
select p.productname, p.category
from products p
where p.category = 'Electronic Gadgets';
```

	productname	category
▶	laptops	Electronic Gadgets
	ethernet	Electronic Gadgets
	tablet	Electronic Gadgets
	stylus	Electronic Gadgets
	GPU	Electronic Gadgets
	CPU	Electronic Gadgets
	RAM	Electronic Gadgets
	ROM	Electronic Gadgets
	monitor	Electronic Gadgets
	wi-fi	Electronic Gadgets

**6. Write an SQL query to calculate the average order value for each customer. Include the customer's name and their average order value.**

#6

```
select c.customerid, CONCAT(c.firstname, ' ', c.lastname) as customername,
round(avg(o.totalamount)) as Averageordervalue
from orders o
join customers c on o.customerid = c.customerid
group by c.customerid, customername
order by customerid;
```

	customerid	customername	Averageordervalue
▶	3	christopher nolan	3300
	4	martin scorsese	17600
	5	cilian murphy	16500
	6	denzel washington	88000
	7	hugh jackman	17600
	8	dev patel	44000
	9	ryan reynolds	44000
	10	brad pitt	9900
	12	henry cavill	11000

**7. Write an SQL query to find the order with the highest total revenue. Include the order ID, customer information, and the total revenue.**

#7

```
select o.orderid,CONCAT(c.firstname, ' ', c.lastname) as customername,
c.Email,c.Phone,c.Address, o.totalamount AS totalrevenue
from orders o
join customers c on o.customerid = c.customerid
order by o.totalamount desc
limit 1;
```

	orderid	customername	Email	Phone	Address	totalrevenue
▶	6	denzel washington	denwash@gmail.com	1234567895	DC	88000.00

**8. Write an SQL query to list electronic gadgets and the number of times each product has been ordered.**

#8

```
select p.productname,COUNT(od.productid) as totalorders
from products p
left join orderdetails od on p.productid = od.productid
where p.category = 'Electronic Gadgets'
group by p.productname
order by totalorders desc;
```

	productname	totalorders
▶	ethernet	1
	tablet	1
	stylus	1
	GPU	1
	CPU	1
	RAM	1
	ROM	1
	monitor	1
	laptops	0
	wi-fi	0



**9. Write an SQL query to find customers who have purchased a specific electronic gadget product. Allow users to input the product name as a parameter.**

#9

```
select distinct c.customerid, CONCAT(c.firstname, ' ', c.lastname) as customername
from customers c
join orders o on c.customerid = o.customerid
join orderdetails od on o.orderid = od.orderid
join products p on od.productid = p.productid
where p.productname = 'monitor' and p.category = 'Electronic Gadgets';
```

	customerid	customername
▶	5	cilian murphy

**10. Write an SQL query to calculate the total revenue generated by all orders placed within a specific time period. Allow users to input the start and end dates as parameters.**

#10

```
select SUM(o.totalamount) as totalrevenue
from orders o
where o.orderdate between '2025-03-01' and '2025-03-30';
```

	totalrevenue
▶	251900.00

## Task 4. Subquery and its type:

**1. Write an SQL query to find out which customers have not placed any orders.**

#1

```
select c.customerid, c.firstname, c.lastname
from customers c
where not exists (
    select 1 from orders o where o.customerid = c.customerid
```

	customerid	firstname	lastname
▶	2	al	pacino
	11	chris	evans

**2. Write an SQL query to find the total number of products available for sale.**

#2

```
select (select COUNT(*) from inventory where QuantityInStock > 0) AS  
Totalavailableproducts;
```

	Totalavailableproducts
▶	10

**3. Write an SQL query to calculate the total revenue generated by TechShop.**

#3

```
select (select SUM(totalamount) from orders) as Totalrevenue;
```

	Totalrevenue
▶	251900.00

**4. Write an SQL query to calculate the average quantity ordered for products in a specific category. Allow users to input the category name as a parameter.**

#4

```
select (select avg(od.quantity)  
from orderdetails od  
join products p on od.Productid = p.productid  
where p.category = 'Electronic Gadgets') as avgquantityordered;
```

	avgquantityordered
▶	2.2500

**5. Write an SQL query to calculate the total revenue generated by a specific customer. Allow users to input the customer ID as a parameter.**

#5

```
select(select SUM(o.totalamount)  
from orders o  
where o.customerid = 5) as Totalrevenue;
```

	Totalrevenue
▶	16500.00

**6. Write an SQL query to find the customers who have placed the most orders. List their names and the number of orders they've placed.**

#6

```
select c.customerid, CONCAT(c.firstname, ' ', c.lastname) as Customername,  
COUNT(o.orderid) as Totalorders  
from customers c  
join orders o on c.customerid = o.customerid  
group by c.customerid  
having totalorders = (select MAX(ordercount) from (select customerid,  
COUNT(orderid) as ordercount  
from orders group by customerid) as sub);
```

	customerid	Customername	Totalorders
▶	3	christopher nolan	1
	4	martin scorsese	1
	5	cilian murphy	1
	6	denzel washington	1
	7	hugh jackman	1
	8	dev patel	1
	9	ryan reynolds	1
	10	brad pitt	1
	12	henry cavill	1

**7. Write an SQL query to find the most popular product category, which is the one with the highest total quantity ordered across all orders.**

#7

```
select category, Totalquantityordered from (  
select p.category, SUM(od.quantity) as Totalquantityordered  
from products p  
join orderdetails od on p.productid = od.productid  
group by p.category  
) as categorysales  
order by totalquantityordered desc  
limit 1;
```

	category	Totalquantityordered
▶	Electronic Gadgets	18

**8. Write an SQL query to find the customer who has spent the most money (highest total revenue) on electronic gadgets. List their name and total spending.**

#8

```
select CONCAT(c.FirstName, ' ', c.LastName) AS CustomerName, s.TotalSpending
from customers c join (
select o.customerid, SUM(od.quantity * p.price) as Totalspending
from orders o
join orderdetails od on o.orderid = od.orderid
join products p on od.productid = p.productid
where p.category = 'Electronic Gadgets'
group by o.customerid
order by Totalspending desc
limit 1
) s on c.customerid = s.customerid;
```

	CustomerName	TotalSpending
▶	denzel washington	88000

**9. Write an SQL query to calculate the average order value (total revenue divided by the number of orders) for all customers.**

#9

```
select (select SUM(Totalamount) from Orders) / (select COUNT(*) from orders)
as AverageOrderValue;
```

	AverageOrderValue
▶	27988.888889

**10. Write an SQL query to find the total number of orders placed by each customer and list their names along with the order count.**

```
select CONCAT(c.Firstname, ' ', c.Lastname) as Customerfullname,
(select COUNT(*) from orders o where o.customerid = c.customerid) as totalorders
from customers c;
```

	Customerfullname	totalorders
▶	al pacino	0
	christopher nolan	1
	martin scorsese	1
	cilian murphy	1
	denzel washington	1
	hugh jackman	1
	dev patel	1
	ryan reynolds	1
	brad pitt	1
	chris evans	0
	henry cavill	1