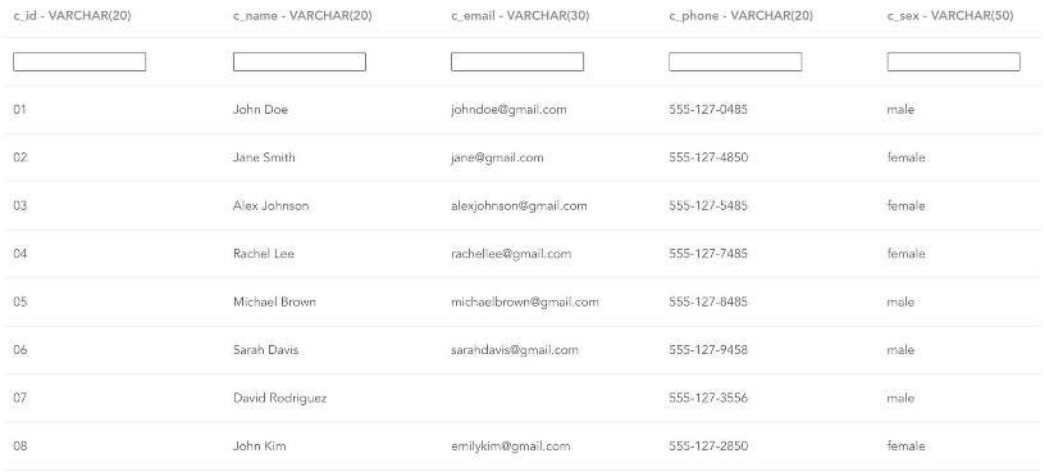
Online practice answer - mock exam

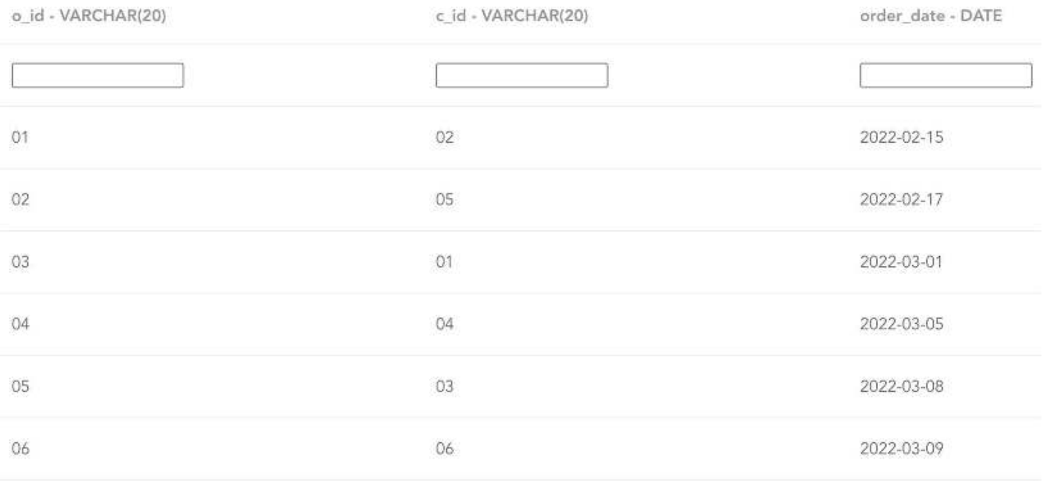
Database Systems (City University of Hong Kong)

Customers



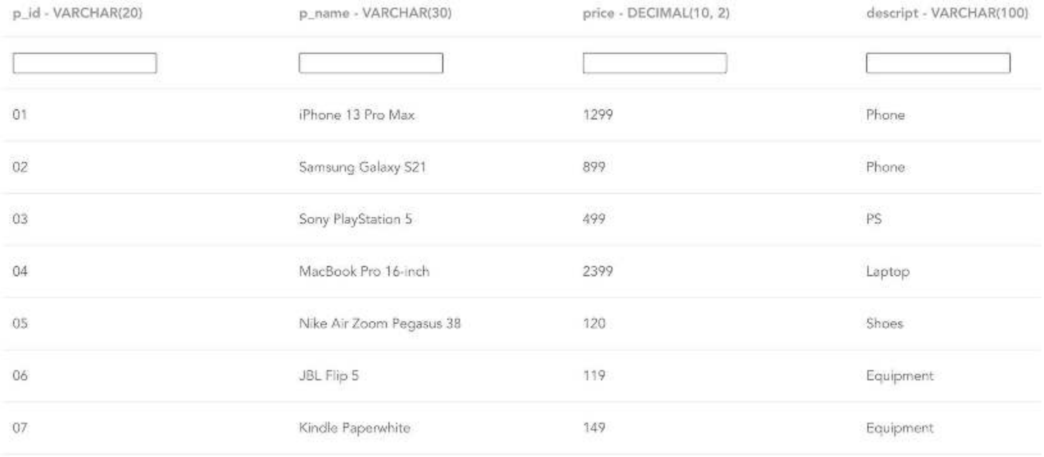
c_id - VARCHAR(20)	c_name - VARCHAR(20)	c_email - VARCHAR(30)	c_phone - VARCHAR(20)	c_sex - VARCHAR(50)
01	John Doe	johndoe@gmail.com	555-127-0485	male
02	Jane Smith	jane@gmail.com	555-127-4850	female
03	Alex Johnson	alexjohnson@gmail.com	555-127-5485	female
04	Rachel Lee	rachellee@gmail.com	555-127-7485	female
05	Michael Brown	michaelbrown@gmail.com	555-127-8485	male
06	Sarah Davis	sarahdavis@gmail.com	555-127-9458	male
07	David Rodriguez		555-127-3556	male
08	John Kim	emilykim@gmail.com	555-127-2850	female

Orders



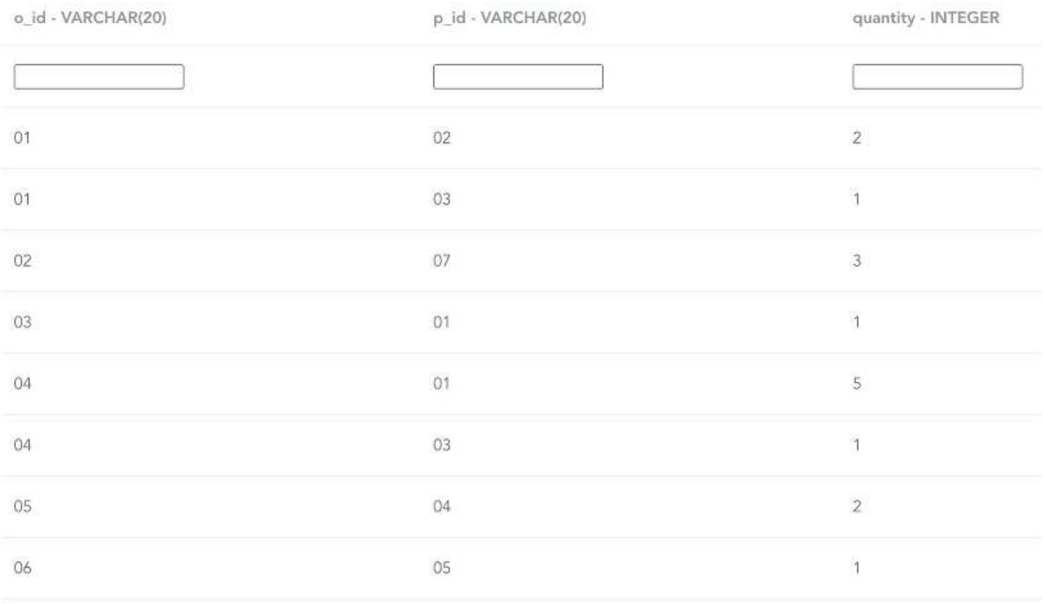
o_id - VARCHAR(20)	c_id - VARCHAR(20)	order_date - DATE
01	02	2022-02-15
02	05	2022-02-17
03	01	2022-03-01
04	04	2022-03-05
05	03	2022-03-08
06	06	2022-03-09

Products



p_id - VARCHAR(20)	p_name - VARCHAR(30)	price - DECIMAL(10, 2)	descript - VARCHAR(100)
01	iPhone 13 Pro Max	1299	Phone
02	Samsung Galaxy S21	899	Phone
03	Sony PlayStation 5	499	PS
04	MacBook Pro 16-inch	2399	Laptop
05	Nike Air Zoom Pegasus 38	120	Shoes
06	JBL Flip 5	119	Equipment
07	Kindle Paperwhite	149	Equipment

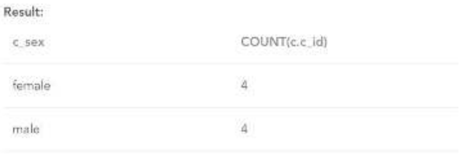
Orderitems



o_id - VARCHAR(20)	p_id - VARCHAR(20)	quantity - INTEGER
01	02	2
01	03	1
02	07	3
03	01	1
04	01	5
04	03	1
05	04	2
06	05	1

1. Find the number of male and female customers.

```
SELECT c.c_sex, COUNT(c.c_id)
FROM Customers c
GROUP BY c.c_sex;
```

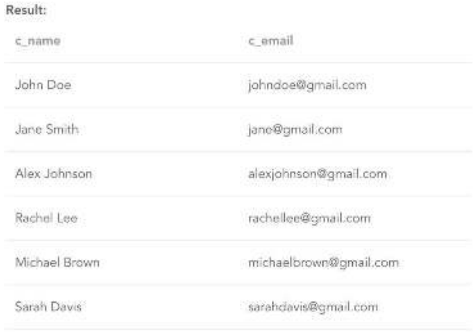


Result:

c_sex	COUNT(c.c_id)
female	4
male	4

2. Retrieve the names and email addresses of all customers who have ever placed an order

```
SELECT c_name, c_email
FROM Customers
WHERE c_id IN (SELECT c_id FROM Orders);
```

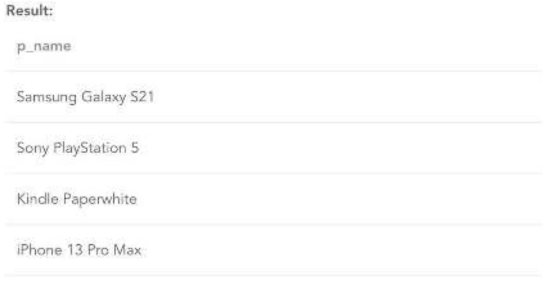


Result:

c_name	c_email
John Doe	johndoe@gmail.com
Jane Smith	jane@gmail.com
Alex Johnson	alexjohnson@gmail.com
Rachel Lee	rachellee@gmail.com
Michael Brown	michaelbrown@gmail.com
Sarah Davis	sarahdavis@gmail.com

3. Query all product names sold between 2022-02-15 and 2022-03-02.

SELECT p_name
FROM Products
INNER JOIN Orderitems ON Products.p_id = Orderitems.p_id
INNER JOIN Orders ON Orderitems.o_id = Orders.o_id
WHERE Orders.order_date BETWEEN DATE('2022-02-15') AND DATE('2022-03-02');

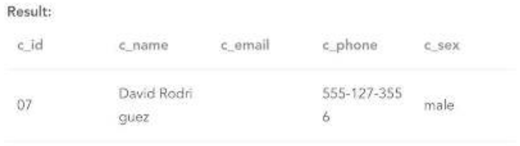


Result:

p_name
Samsung Galaxy S21
Sony PlayStation 5
Kindle Paperwhite
iPhone 13 Pro Max

4. Query the information of customers whose email is NULL.

SELECT *
FROM Customers
WHERE c_email IS NULL;



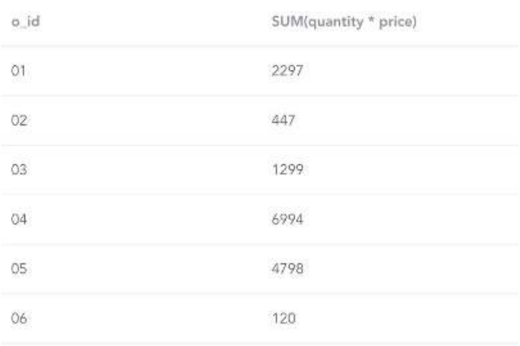
Result:

c_id	c_name	c_email	c_phone	c_sex
07	David Rodriguez		555-127-3556	male

5. Query the order id and total amount of each order, which is equal to the sum of the quantity of each product multiplied by its price.

SELECT o_id, SUM(quantity * price)
FROM Orderitems

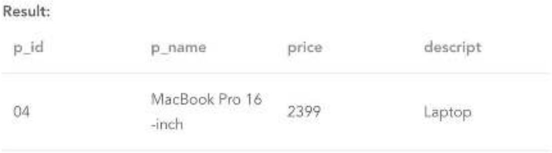
```
INNER JOIN Products ON Orderitems.p_id = Products.p_id
GROUP BY o_id;
```



o_id	SUM(quantity * price)
01	2297
02	447
03	1299
04	6994
05	4798
06	120

6. Query the information of the most expensive product.

```
SELECT *
FROM Products
WHERE price = (SELECT MAX(price) FROM Products);
```

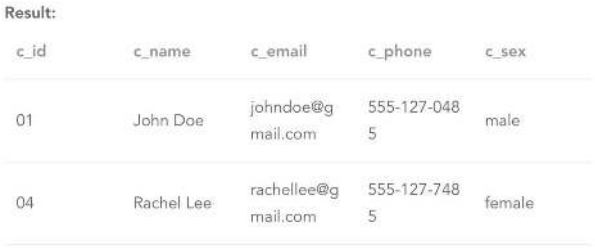


Result:

p_id	p_name	price	descript
04	MacBook Pro 16 -inch	2399	Laptop

7. Find the information of the customers who have bought ‘iPhone 13 Pro Max’ product.

```
SELECT *
FROM Customers
WHERE c_id IN (SELECT c_id FROM Orders o
INNER JOIN Orderitems oi ON o.o_id = oi.o_id
INNER JOIN Products p ON oi.p_id = p.p_id
WHERE p.p_name = 'iPhone 13 Pro Max');
```

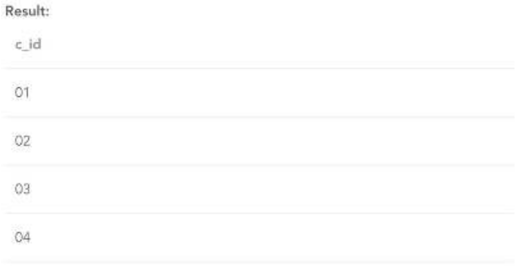


Result:

c_id	c_name	c_email	c_phone	c_sex
01	John Doe	johndoe@gmail.com	555-127-0485	male
04	Rachel Lee	rachellee@gmail.com	555-127-7485	female

8. Retrieve IDs of customers who have historically spent more than 1000 in buying products.

```
SELECT c.c_id
FROM Customers c
INNER JOIN Orders o ON c.c_id = o.c_id
INNER JOIN Orderitems oi ON o.o_id = oi.o_id
INNER JOIN Products p ON oi.p_id = p.p_id
GROUP BY c.c_id
HAVING SUM(p.price * oi.quantity) > 1000;
```

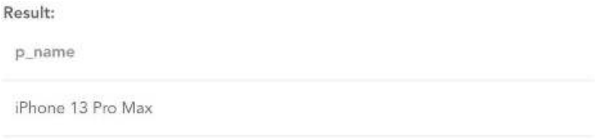


Result:

c_id
01
02
03
04

9. Retrieve product’s names whose historical quantity of sale is larger than 3

```
SELECT p.p_name
FROM Products p
INNER JOIN Orderitems oi ON p.p_id = oi.p_id
GROUP BY p.p_id, p.p_name
HAVING SUM(quantity) > 3;
```



Result:

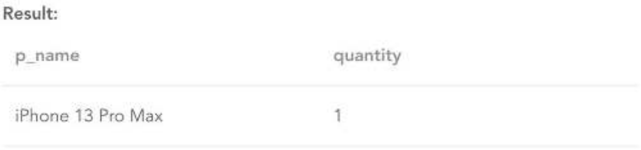
p_name
iPhone 13 Pro Max

10. Query the product’s names and quantity bought by the customers with the first name ‘John’.

```
SELECT p.p_name, oi.quantity
FROM Customers c
INNER JOIN Orders o ON c.c_id = o.c_id
INNER JOIN Orderitems oi ON o.o_id = oi.o_id
INNER JOIN Products p ON oi.p_id = p.p_id
WHERE c.name LIKE 'John %';
```

```
SELECT p.p_name, oi.quantity
FROM Customers c
INNER JOIN Orders o ON c.c_id = o.c_id
INNER JOIN Orderitems oi ON o.o_id = oi.o_id
INNER JOIN Products p ON oi.p_id = p.p_id
```

```
WHERE c.c_name LIKE 'John %';
```

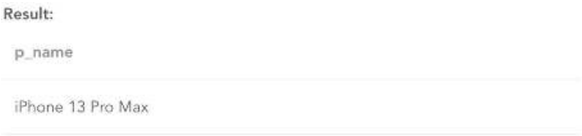


Result:

p_name	quantity
iPhone 13 Pro Max	1

11. Retrieve the name of the product that has been sold the most.

```
SELECT p.p_name
FROM Products p
INNER JOIN Orderitems oi ON p.p_id = oi.p_id
INNER JOIN Orders o ON oi.o_id = o.o_id
GROUP BY p.p_name
HAVING SUM(oi.quantity) = (
    SELECT MAX(total_quantity)
    FROM (
        SELECT SUM(oi.quantity) total_quantity
        FROM Orderitems oi
        INNER JOIN Orders o ON oi.o_id = o.o_id
        GROUP BY oi.p_id
    ) product_sales
);
```



Result:

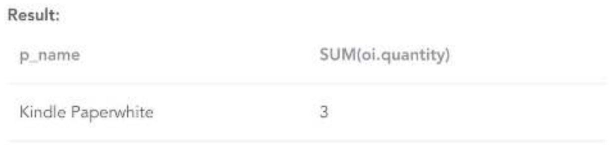
p_name
iPhone 13 Pro Max

12. List the products that have been bought by male customers before 2022-03-01, along with their total quantity of sales before 2022-03-01. The results should be sorted in the descending order of quantity of sale.

```
SELECT p.p_name, SUM(oi.quantity)
FROM Products p
INNER JOIN Orderitems oi ON p.p_id = oi.p_id
INNER JOIN Orders o ON oi.o_id = o.o_id
INNER JOIN Customers c ON o.c_id = c.c_id
WHERE c.gender = 'Male' AND o.order_date < Date('2022-03-01')
GROUP BY p.p_id, p.p_name
ORDER BY SUM(oi.quantity) DESC;
```

```
SELECT p.p_name, SUM(oi.quantity)
FROM Products p
```

```
INNER JOIN Orderitems oi ON p.p_id = oi.p_id
INNER JOIN Orders o ON oi.o_id = o.o_id
INNER JOIN Customers c ON o.c_id = c.c_id
WHERE c.c_sex = 'male' AND o.order_date < Date('2022-03-01')
GROUP BY p.p_id, p.p_name
ORDER BY SUM(oi.quantity) DESC;
```



Result:

p_name	SUM(oi.quantity)
Kindle Paperwhite	3