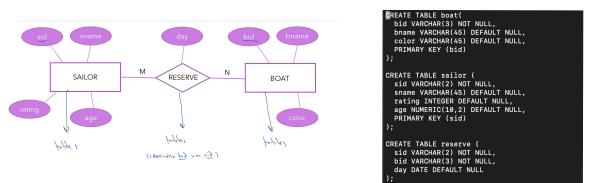


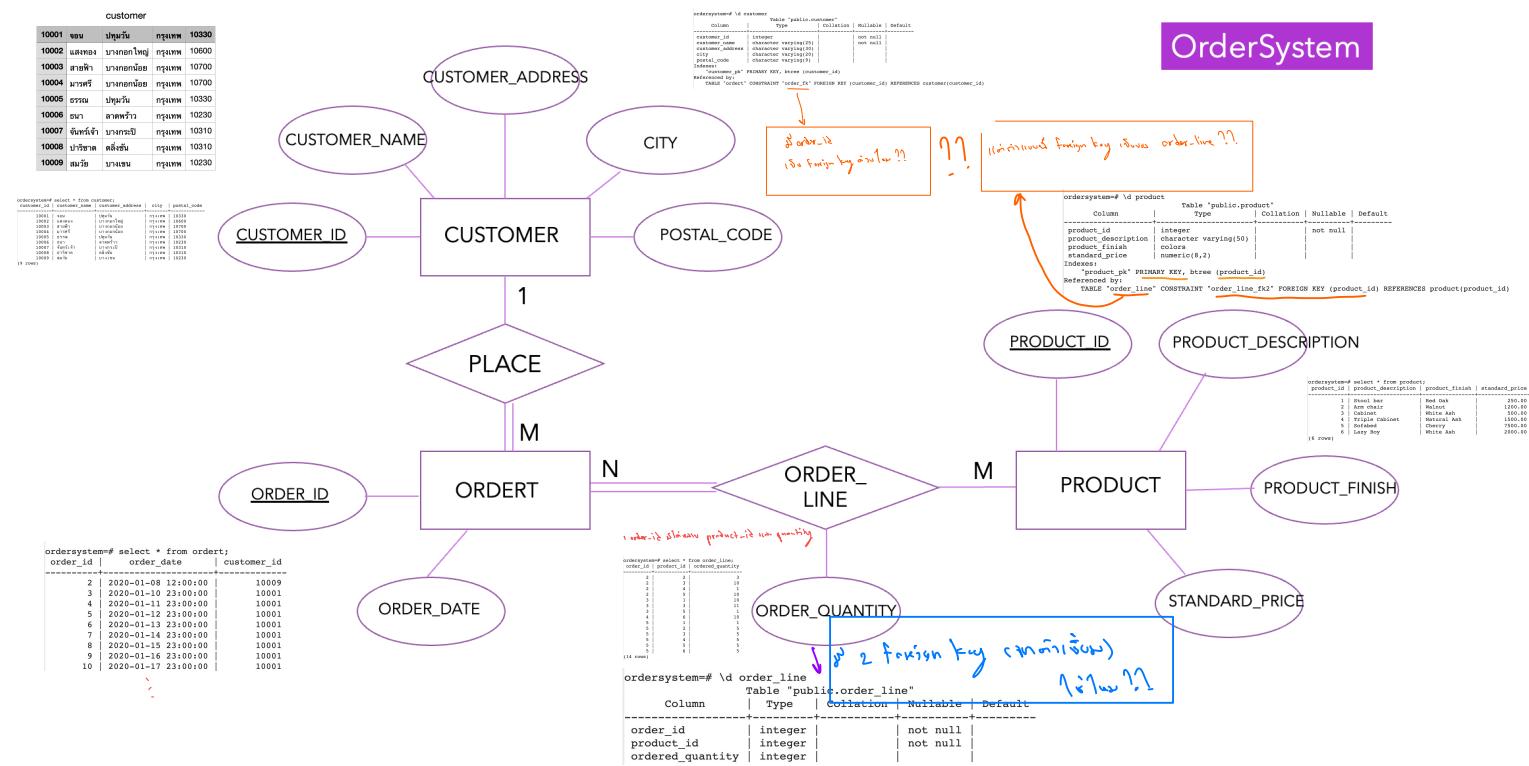
Prerequisite (step since beginning)

0. Install docker

1. Import csv file (data) you can use chatgpt or
search it oninternet



ตาราง CUSTOMER, ORDER, ORDER_LINE, PRODUCT มีความสัมพันธ์ดัง ER diagram ต่อไปนี้



- Pull git repo https://github.com/wichadak/2110322_DBSYS_2566_1 มาใหม่ เพราะมีไฟล์ข้อมูลเพิ่มเติม ✓
- จาก ER diagram ข้างต้นให้ทำการสร้างฐานข้อมูลชื่อ ordersystem และสร้างตาราง 4 (ตามตัวอย่างคำสั่งที่แสดงไว้ในไฟล์ PostgreSQL_ToTry_commands.sql ที่ pull มาจาก git repo ข้างต้น) ประกอบด้วย ตาราง customer, ordert, order_line, และตาราง product ✓
- หลังสร้างตารางเสร็จเรียบร้อยแล้ว ให้นำข้อมูลเข้าห้องสีตาราง โดยไฟล์ข้อมูลอยู่ใน folder ordersystem ที่ pull มาจาก git ✓

* consequence affects the import process, this cause error because of some constraints (please recheck) after you import all others table. The error is solved
- เมื่อได้ตารางข้อมูลเรียบร้อยแล้วให้เขียน SQL เพื่อตอบโจทย์ต่อไปนี้ โดยแต่ละข้อให้สร้าง 1 ไฟล์ เก็บคำสั่ง SQL

โดยกำหนดชื่อไฟล์เป็น 4_1.sql, 4_2.sql....

จากนั้นให้ zip ไฟล์ 4_1.sql..., 4_16.sql และ upload zip file เข้าใน mycourseville

**** ไม่ต้องส่งผล query แล้ว เอาแต่คำสั่งหรือชุดคำสั่ง SQL แยกตามข้อ ***

ลังเกตได้ว่า 2 table ที่เชื่อมกัน จะมี 1 col ที่เหมือนกัน
โดยที่table นึง จะเอา key ของอีก table นึงมาเป็น col
ตัวอย่างเช่น customer กับ ordert

customer_id	customer_name	customer_address	city	postal_code
10001	จอย	บ้านวัน	กรุงเทพฯ	10330
10002	แสวงทอง	บ้านคนใหญ่	กรุงเทพฯ	10600
10003	สุขุมพิทักษ์	บ้านคนน้อย	กรุงเทพฯ	10700
10004	มาราธี	บ้านคนน้อย	กรุงเทพฯ	10700
10005	ธรรม	บ้านวัน	กรุงเทพฯ	10330
10006	ธนา	ศาสพรร้า	กรุงเทพฯ	10230
10007	จันทร์รัชดา	บ้านกระปี	กรุงเทพฯ	10310
10008	ปราจิราศ	คุณชั้น	กรุงเทพฯ	10310
10009	สมรักษ์	บ้านเดอน	กรุงเทพฯ	10230

(9 rows)

primary key (unique)

order_id	order_date	customer_id
2	2020-01-08 12:00:00	10009
3	2020-01-10 23:00:00	10001
4	2020-01-11 23:00:00	10001
5	2020-01-12 23:00:00	10001
6	2020-01-13 23:00:00	10001
7	2020-01-14 23:00:00	10001
8	2020-01-15 23:00:00	10001
9	2020-01-16 23:00:00	10001
10	2020-01-17 23:00:00	10001

new foreign key (ผูกไว้)

ใช้ query ตรวจสอบว่า ตัวอะไร foreign key
ในตัวนี้

```
select *
from product join order_line on product.product_id = order_line.product_id
```

	product_id integer	product_description character varying (50)	product_finish colors	standard_price numeric (8,2)	order_id integer	product_id integer	ordered_quantity integer
1	2	Arm chair	Walnut	1200.00	2	2	3
2	3	Cabinet	White Ash	500.00	2	3	10
3	4	Triple Cabinet	Natural Ash	1500.00	2	4	1
4	5	Sofabed	Cherry	7500.00	2	5	10
5	1	Stool bar	Red Oak	250.00	3	1	10
6	3	Cabinet	White Ash	500.00	3	3	11
7	5	Sofabed	Cherry	7500.00	3	5	1
8	6	Lazy Boy	White Ash	2000.00	4	6	10
9	1	Stool bar	Red Oak	250.00	5	1	1
10	2	Arm chair	Walnut	1200.00	5	2	5
11	3	Cabinet	White Ash	500.00	5	3	5
12	4	Triple Cabinet	Natural Ash	1500.00	5	4	5
13	5	Sofabed	Cherry	7500.00	5	5	5
14	6	Lazy Boy	White Ash	2000.00	5	6	5

primary key
select * from product

	product_id [PK] integer	product_description character varying (50)	product_finish colors	standard_price numeric (8,2)
1	1	Stool bar	Red Oak	250.00
2	2	Arm chair	Walnut	1200.00
3	3	Cabinet	White Ash	500.00
4	4	Triple Cabinet	Natural Ash	1500.00
5	5	Sofabed	Cherry	7500.00
6	6	Lazy Boy	White Ash	2000.00

foreign key
(constraint)

constraint

select * from order_line

	order_id [PK] integer	product_id [PK] integer	ordered_quantity integer
1	2	2	3
2	2	3	10
3	2	4	1
4	2	5	10
5	3	1	10
6	3	3	11
7	3	5	1
8	4	6	10
9	5	1	1
10	5	2	5
11	5	3	5
12	5	4	5
13	5	5	5
14	5	6	5

** ถ้ามันยากก็ค่อยๆทำ ค่อยๆแก้ ตอนแรกให้เลือก select * แล้วมาดูว่าผลจากการ join หรืออะไรพวนนี้เป็นไป group by ก็ไว้ทีหลังได้ ลองดูน่าจะเห็นภาพขึ้น

4.1 แสดงจำนวนลูกค้ายกตาม postal_code โดยเรียงตามจำนวน customers จากมากไปน้อย

Expected output

postal_code	customer_numbers
10700	2
10330	2
10230	2
10310	2
10600	1

customer_id	customer_name	customer_address	city	postal_code
10001	สมชาย	บ้านที่อยู่	กรุงเทพ	10330
10002	สมหญิง	บ้านที่อยู่	กรุงเทพ	10600
10003	สมศักดิ์	บ้านที่อยู่	กรุงเทพ	10700
10004	สมพร	บ้านที่อยู่	กรุงเทพ	10700
10005	สมราย	บ้านที่อยู่	กรุงเทพ	10330
10006	สมนา	บ้านที่อยู่	กรุงเทพ	10230
10007	สมพงษ์	บ้านที่อยู่	กรุงเทพ	10310
10008	สมรักษ์	บ้านที่อยู่	กรุงเทพ	10310
10009	สมอุ๊	บ้านที่อยู่	กรุงเทพ	10230

```
1 select *
2 from customer
3
```

customer_id	postal_code
10001	10330
10002	10600
10003	10700
10004	10700
10005	10330
10006	10230
10007	10310
10008	10310
10009	10230

```
1 select count(customer.customer_id), customer.postal_code
2 from customer
3
```

แบบนี้คุณต้องใช้ group by

```
1 select count(customer.customer_id), customer.postal_code
2 from customer
```

ERROR: column "customer.postal_code" must appear in the GROUP BY clause or be used in an aggregate function
LINE 1: select count(customer.customer_id), customer.postal_code ^

SQL state: 42803
Character: 37

This query would indeed result in an error because you are selecting a non-aggregated column (`customer.postal_code`) along with an aggregate function

The `COUNT` function can be used without a `GROUP BY` clause when you're not selecting other columns that aren't part of an aggregate function. When you only want to know the total number of rows, or count of distinct values in a column, across the entire table or a subset of it (defined by a `WHERE` clause), you don't need to use `GROUP BY`.

```
1 select customer.postal_code, count(customer.customer_id) as customer_numbers
2 from customer
3 group by customer.postal_code
```

postal_code	customer_numbers
10700	2
10600	1
10330	2
10230	2
10310	2

```
1 select customer.postal_code, count(customer.customer_id) as customer_numbers
2 from customer
3 group by customer.postal_code
4 order by count(customer.customer_id) desc
```

postal_code	customer_numbers
10330	2
10310	2
10230	2
10700	2
10600	1

4.2 แสดงจำนวนลูกค้าแยกตาม postal_code โดยเรียงตามจำนวน customers จากมากไปน้อย

โดยแสดงเฉพาะ postal_code ที่มีจำนวนลูกค้ามากกว่า 1

Expected output

postal_code	customer_numbers
10700	2
10330	2
10230	2
10310	2

10700 4.1

เพิ่ม having

```
1 select customer.postal_code, count(customer.customer_id) as customer_numbers
2 from customer
3 group by customer.postal_code
4 having count(customer.customer_id) > 1
5 order by count(customer.customer_id) desc
```

Data Output Messages Notifications

postal_code	customer_numbers
1	10700
2	10330
3	10230
4	10310

4.3 จงแสดงรายการของเดอร์ของลูกค้าที่รหัส 10001

Expected output

```
orderstable select * from order1
order_id | order_date | customer_id
-----+-----+-----+
2 | 2020-01-09 23:00:00 | 10009
3 | 2020-01-10 23:00:00 | 10001
4 | 2020-01-11 23:00:00 | 10001
5 | 2020-01-12 23:00:00 | 10001
6 | 2020-01-13 23:00:00 | 10001
7 | 2020-01-14 23:00:00 | 10001
8 | 2020-01-15 23:00:00 | 10001
9 | 2020-01-16 23:00:00 | 10001
10 | 2020-01-17 23:00:00 |
11 | 2020-01-18 23:00:00 |
12 | 2020-01-19 23:00:00 |
14 | 2020-01-20 23:00:00 |
15 | 2020-01-21 23:00:00 |
```

```
15 select order_id, order_date
16 from ordert
17 WHERE customer_id = 10001
18
```

Data Output		Messages	Notifications
order_id	order_date		
[PK] integer	timestamp without time zone		
1	3 2020-01-10 23:00:00		
2	4 2020-01-11 23:00:00		
3	5 2020-01-12 23:00:00		
4	6 2020-01-13 23:00:00		
5	7 2020-01-14 23:00:00		
6	8 2020-01-15 23:00:00		
7	9 2020-01-16 23:00:00		

** ถ้ามันยากก็ค่อยๆทำ ค่อยๆแก้ ตอนแรกให้เลือก select * แล้วมาดูว่าผลจากการ join หรืออะไรพอกันนี้เป็นไป group by ก็ไว้ทีหลังได้ ลองข้อ 4 ดูน่าจะเห็นภาพขึ้น

```

1 SELECT
2   product.product_id,
3   product.product_description,
4   sum(order_line.order_qty)
5   FROM order_line
6   JOIN product ON order_line.product_id = product.product_id
7

```

```

1 SELECT
2   product.product_id,
3   product.product_description
4   FROM order_line
5   JOIN product ON order_line.product_id = product.product_id
6   GROUP BY product.product_id -- but have to delete order_id column

```

```

1 SELECT
2   product.product_id,
3   product.product_description,
4   COUNT(order_line.order_id) -- 2.) add count
5   FROM order_line
6   JOIN product ON order_line.product_id = product.product_id
7   GROUP BY product.product_id -- 1.) but have to delete order_id column

```

```

1 SELECT
2   product.product_id,
3   product.product_description,
4   COUNT(order_line.order_id) -- 2.) count for each product_id
5   FROM order_line
6   JOIN product ON order_line.product_id = product.product_id
7   GROUP BY product.product_id -- 1.) but have to delete order_id column
8   ORDER BY COUNT(order_line.order_id)

```

1.4 แสดงรหัสสินค้าและคำบรรยายสินค้าที่มีรายการสั่งซื้อมากสุด

product_id	product_description
3	Cabinet
5	Sofabed

```
ordersystem# select * from order_line;
```

```
order_id product_id ordered_quantity
```

```

2 | 2 | 3 | 10
2 | 2 | 4 | 1
2 | 2 | 5 | 10
3 | 3 | 1 | 10
3 | 3 | 3 | 11
3 | 3 | 5 | 1
4 | 4 | 6 | 10
5 | 5 | 1 | 1
5 | 5 | 2 | 5
5 | 5 | 3 | 5
5 | 5 | 4 | 5
5 | 5 | 5 | 5
5 | 5 | 6 | 5

```

(14 rows)

```
ordersystem# select * from product;
```

product_id	product_description	product_finish	standard_price
1	Stool bar	Red Oak	250.00
2	Arm chair	Walnut	1200.00
3	Cabinet	Wh. Ash	500.00
4	Triple Cabinet	Natural Ash	1500.00
5	Sofabed	Cherry	7500.00
6	Lazy Boy	White Ash	2000.00

(6 rows)

- * 1. อาย่าพึงลบ column count
- 2. ต้องดูว่ามันมี max กี่อันแล้วจึงจะใส่ limit

```
17) select product.product_id, product.product_description, order_line.order_qty
18) From order_line join product on order_line.product_id = product.product_id
19) order by order_qty desc
20) limit 1
```

product_id product_description order_qty

```

1 | Cabinet | 11
2 | Arm chair | 10
3 | Cabinet | 5
4 | Triple Cabinet | 3
5 | Sofabed | 2
6 | Lazy Boy | 1
7 | Stool bar | 1
8 | Cabinet | 1
9 | Cabinet | 1
10 | Cabinet | 1
11 | Cabinet | 1
12 | Cabinet | 1
13 | Cabinet | 1
14 | Cabinet | 1
15 | Cabinet | 1
16 | Cabinet | 1
17 | Cabinet | 1
18 | Cabinet | 1
19 | Cabinet | 1
20 | Cabinet | 1

```

```
21) select product.product_id, product.product_description, order_line.order_qty
22) From order_line join product on order_line.product_id = product.product_id
23) order by order_qty desc
24) limit 1
```

product_id product_description order_qty

```

1 | Cabinet | 11

```

The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Contains the SQL query: "select product.product_id, product.product_description, order_line.order_qty From order_line join product on order_line.product_id = product.product_id order by order_qty desc limit 1".
- Results:** Shows the output of the query, which is a single row with product_id 1 and product_description 'Cabinet'.
- Properties:** Shows the data type and other properties for the columns.
- Structure:** Shows the table structure for the product and order_line tables.
- Session Properties:** Shows session variables like auto_increment, character_set_client, character_set_connection, etc.

✓ ✓

4.5 จงแสดงรหัสลูกค้า ชื่อลูกค้า และจำนวนของเดอร์ที่สั่งโดยลูกค้าแต่ละคน โดยเรียงลำดับข้อมูลตามจำนวน
อเดอร์จากมากไปน้อย

Expected output

customer_id	customer_name	number_of_orders
10002	แสวงทอง	20
10001	จอน	12
10003	สายฟ้า	5
10008	ปาริชาต	4
10009	สม瓦យ	1

ห้ามดู

ห้ามใช้คำ

customer_id	customer_name	customer_address	city	postal_code
10001	สมวัย	บ้านกัน	กรุงเทพฯ	10330
10002	เสนาณรงค์	บ้านกอกไทร	กรุงเทพฯ	10600
10003	สายฟ้า	บ้านกอกน้อย	กรุงเทพฯ	10700
10004	นาฬิกา	บ้านกอกน้อย	กรุงเทพฯ	10700
10005	แมว	บ้านกอก	กรุงเทพฯ	10330
10006	ฟาร์ม	บ้านกอกวัว	กรุงเทพฯ	10230
10007	จันทร์-จีน	บ้านกระปี	กรุงเทพฯ	10310
10008	ปาริชาต	บ้านลี้ซึ้ง	กรุงเทพฯ	10310
10009	แมว	บ้านเงย	กรุงเทพฯ	10230

order_id	order_date	customer_id
2	2020-01-08 12:00:00	10009
3	2020-01-10 23:00:00	10001
4	2020-01-11 23:00:00	10001
5	2020-01-12 23:00:00	10001
6	2020-01-13 23:00:00	10001
7	2020-01-14 23:00:00	10001
8	2020-01-15 23:00:00	10001
9	2020-01-16 23:00:00	10001
10	2020-01-17 23:00:00	10001

customer_id	customer_name	customer_address	city	postal_code	order_id	order_date	customer_id
10009	เสนาณรงค์	บ้านกัน	กรุงเทพฯ	10330	10009	2020-01-08 12:00:00	10009
2	เสนาณรงค์	บ้านกอกไทร	กรุงเทพฯ	10600	3	2020-01-10 23:00:00	10001
3	เสนาณรงค์	บ้านกอกน้อย	กรุงเทพฯ	10700	4	2020-01-11 23:00:00	10001
4	เสนาณรงค์	บ้านกอกน้อย	กรุงเทพฯ	10700	5	2020-01-12 23:00:00	10001
5	เสนาณรงค์	บ้านกอก	กรุงเทพฯ	10330	6	2020-01-13 23:00:00	10001
6	เสนาณรงค์	บ้านกอกวัว	กรุงเทพฯ	10230	7	2020-01-14 23:00:00	10001
7	เสนาณรงค์	บ้านลี้ซึ้ง	กรุงเทพฯ	10310	8	2020-01-15 23:00:00	10001
8	เสนาณรงค์	บ้านกระปี	กรุงเทพฯ	10310	9	2020-01-16 23:00:00	10001
9	เสนาณรงค์	บ้านเงย	กรุงเทพฯ	10230	10	2020-01-17 23:00:00	10001

customer_id	customer_name
10009	เสนาณรงค์
10002	แสวงทอง
10001	จอน
10008	ปาริชาต
10003	สายฟ้า

ก็ต้อง order.order_id วน unique

customer_id	customer_name	order_id
10009	เสนาณรงค์	2
2	เสนาณรงค์	3
3	เสนาณรงค์	4
4	เสนาณรงค์	5
5	เสนาณรงค์	6
6	เสนาณรงค์	7

customer.customer_id customer_name order.order_id

... group by order_id ตอน select

customer_id	customer_name	count
10009	เสนาณรงค์	1
10002	แสวงทอง	20
10001	จอน	12
10008	ปาริชาต	4
10003	สายฟ้า	5

count row นับต่อ order.order_id

customer_id	customer_name	count(order_id)
10001	จอน	20
2	เสนาณรงค์	12
3	เสนาณรงค์	4
4	เสนาณรงค์	5

ការសម្រេចទូទាត់តាមលក្ខណៈ និង ការរកស្នើសុំព័ត៌មាន

4.6 ຈក់សេចក្តី និង សេចក្តី ដែលមានគុណភាព និង តម្លៃ ស្ថិតិយវត្ថុ

Expected output

```
customer_id | customer_name
-----+-----
10002 | ផែនខោ
```

```
SELECT *
FROM customer
JOIN ordert ON customer.customer_id = ordert.customer_id;
```

customer_id	customer_name	customer_address	city	country	postal_code	order_id	order_date	order_date without time zone	customer_id
10001	ស៊ុខ សាស្ត្រ	បឹងកេងកង ផ្ទះលេខ ៩៧	ភ្នំពេញ	កម្ពុជា	123456	1	2020-01-08 12:00:00	2020-01-08 12:00:00	10001
10001	ស៊ុខ សាស្ត្រ	បឹងកេងកង ផ្ទះលេខ ៩៧	ភ្នំពេញ	កម្ពុជា	123456	2	2020-01-19 23:00:00	2020-01-19 23:00:00	10001
10001	ស៊ុខ សាស្ត្រ	បឹងកេងកង ផ្ទះលេខ ៩៧	ភ្នំពេញ	កម្ពុជា	123456	3	2020-01-12 23:00:00	2020-01-12 23:00:00	10001
10001	ស៊ុខ សាស្ត្រ	បឹងកេងកង ផ្ទះលេខ ៩៧	ភ្នំពេញ	កម្ពុជា	123456	4	2020-01-14 23:00:00	2020-01-14 23:00:00	10001
10001	ស៊ុខ សាស្ត្រ	បឹងកេងកង ផ្ទះលេខ ៩៧	ភ្នំពេញ	កម្ពុជា	123456	5	2020-01-15 23:00:00	2020-01-15 23:00:00	10001
10001	ស៊ុខ សាស្ត្រ	បឹងកេងកង ផ្ទះលេខ ៩៧	ភ្នំពេញ	កម្ពុជា	123456	6	2020-01-16 23:00:00	2020-01-16 23:00:00	10001
10001	ស៊ុខ សាស្ត្រ	បឹងកេងកង ផ្ទះលេខ ៩៧	ភ្នំពេញ	កម្ពុជា	123456	7	2020-01-17 23:00:00	2020-01-17 23:00:00	10001

```
31 select customer.customer_id, customer.customer_name, ordert.order_id
32 from customer join ordert on customer.customer_id = ordert.customer_id
```

customer_id	customer_name	order_id
1	10009 ស៊ុខ	2
2	10001 សោរ	3
3	10001 សោរ	4
4	10001 សោរ	5
5	10001 សោរ	6
6	10001 សោរ	7

⑥ នូវ ordert.order_id នេះ

```
31 select customer.customer_id, customer.customer_name
32 from customer join ordert on customer.customer_id = ordert.customer_id
33 group by customer.customer_id
34 order by customer.customer_id
```

customer_id	customer_name
1	10009 ស៊ុខ
2	10002 ផែនខោ
3	10001 សោរ
4	10008 បានចាន់
5	10003 ម៉ោង

```
31 select customer.customer_id, customer.customer_name, count(ordert.order_id)
32 from customer join ordert on customer.customer_id = ordert.customer_id
33 group by customer.customer_id
34 order by count(ordert.order_id) desc
35 limit 1
```

customer_id	customer_name	count
1	10002 ផែនខោ	20

4.7 แสดงรหัสสินค้าและคำบรรยายสินค้าที่มีจำนวนการสั่งซื้อมากสุด

Expected output

product_id	product_description
3	Cabinet

```

39 select *
40 from order_line join product on order_line.product_id = product.product_id
41
Data Output Messages Notifications

```

order_id	product_id	ordered_quantity	product_id	product_description	product_finish	standard_price
1	2	2	2	Arm chair	Walnut	1200.00
2	2	3	10	3 Cabinet	White Ash	500.00
3	2	4	1	4 Triple Cabinet	Natural Ash	1500.00
4	2	5	10	5 Sofabed	Cherry	7500.00
5	3	1	10	1 Stool bar	Red Oak	250.00
6	3	3	11	3 Cabinet	White Ash	500.00

```

39 select order_line.order_id, product.product_id, product.product_description
40 from order_line join product on order_line.product_id = product.product_id
41
Data Output Messages Notifications

```

order_id	product_id	product_description
1	2	Arm chair
2	2	Cabinet
3	2	Triple Cabinet
4	2	Sofabed
5	3	Stool bar
6	3	Cabinet

group by order_line.product_id select count(*) from
 ① in order_line.product_id on count(*)

```

39 select product.product_id, product.product_description
40 from order_line join product on order_line.product_id = product.product_id
41 group by product.product_id ①
42
Data Output Messages Notifications

```

product_id	product_description
1	Stool bar
2	Arm chair
3	Cabinet
4	Triple Cabinet
5	Sofabed
6	Lazy Boy

```

39 select product.product_id, product.product_description
40 from order_line join product on order_line.product_id = product.product_id
41 group by product.product_id
42 order by count(order_line.order_id) desc
43 limit 1
44

```

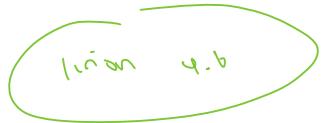
Data Output Messages Notifications

product_id	product_description
3	Cabinet

4.8 จงแสดงรายชื่อลูกค้าที่มีจำนวนออเดอร์มากสุด 3 อันดับแรกและจำนวนออร์дерรวมของพากษา

Expected output

customer_id	customer_name	order_count
10002	แสงทอง	20
10001	ジョン	12
10003	สายฟ้า	5



(เก็ง (ปรับ limit))

```
45 select customer.customer_id, customer.customer_name, count(ordert.order_id)
46 from customer join ordert on customer.customer_id = ordert.customer_id
47 group by customer.customer_id
48 order by count(ordert.order_id) desc
49 limit 3
```

Data Output Messages Notifications

	customer_id	customer_name	count
1	10002	แสงทอง	20
2	10001	ジョン	12
3	10003	สายฟ้า	5

4.9 จงแสดงรหัสและรายชื่อลูกค้า (ไม่ซ้ำ) ที่มีการสั่งสินค้าตั้งแต่วันที่ 2020-01-10 ถึง 2020-01-15

Expected output

customer_id | customer_name

customer_id	customer_name
10001	جون
10002	แสวงทอง

```
53 select *
54 from customer join ordert on customer.customer_id = ordert.customer_id
55
```

Data Output Messages Notifications

	customer_id	customer_name	customer_address	city	postal_code	order_id	order_date	customer_id
1	10009	สมนึก	บางเขน	กรุงเทพ	10230	2	2020-01-08 12:00:00	10009
2	10001	สอน	บปนรัตน์	กรุงเทพ	10330	3	2020-01-10 23:00:00	10001
3	10001	สอน	บปนรัตน์	กรุงเทพ	10330	4	2020-01-11 23:00:00	10001
4	10001	สอน	บปนรัตน์	กรุงเทพ	10330	5	2020-01-12 23:00:00	10001
5	10001	สอน	บปนรัตน์	กรุงเทพ	10330	6	2020-01-13 23:00:00	10001
6	10001	สอน	บปนรัตน์	กรุงเทพ	10330	7	2020-01-14 23:00:00	10001

```
53 select customer.customer_id, customer.customer_name, ordert.order_date
54 from customer join ordert on customer.customer_id = ordert.customer_id
55
```

Data Output Messages Notifications

	customer_id	customer_name	order_date
1	10009	สมนึก	2020-01-09 12:00:00
2	10001	สอน	2020-01-10 23:00:00
3	10001	สอน	2020-01-11 23:00:00
4	10001	สอน	2020-01-12 23:00:00
5	10001	สอน	2020-01-13 23:00:00
6	10001	สอน	2020-01-14 23:00:00

ค่าคง정 assume ว่า

customer_id 1-1 กับ customer_name

```
53 select customer.customer_id, customer.customer_name
54 from customer join ordert on customer.customer_id = ordert.customer_id
55 group by customer.customer_id
56
```

Data Output Messages Notifications

	customer_id	customer_name
1	10009	สมนึก
2	10002	แสวงทอง
3	10001	สอน
4	10008	บาริชาต
5	10003	สามพิพ

```
53 select customer.customer_id, customer.customer_name
54 from customer join ordert on customer.customer_id = ordert.customer_id
55 where ordert.order_date between '2020-01-10 00:00:00' and '2020-01-15 23:59:59'
56 group by customer.customer_id
57
```

Data Output Messages Notifications

	customer_id	customer_name
1	10001	สอน
2	10002	แสวงทอง

where

| ๔๐๐๗๔ | ๖๖๖๖๖๖

4.10 จงแสดงรายการที่สินค้า คำอธิบายสินค้า และ product_finish ทั้งหมดที่มีสีขาวประกอบ

Expected output

product_id	product_description	product_finish
3	Cabinet	White Ash
6	Lazy Boy	White Ash

```
select product.product_id, product.product_description, product.product_finish
from product
WHERE product.product_finish::text LIKE '%White%'
```



4.11 จงแสดงรายการรหัสและชื่อลูกค้าทั้งหมดที่ยังไม่เคยมีรายการสั่งซื้อของเลย

Expected output

customer_id	customer_name
10004	มาร์ศรี
10005	ธีรดา
10006	ธน่า
10007	จันทร์เจ้า

```

60 select *
61 from customer join ordert on customer.customer_id = ordert.customer_id
62
Data Output Messages Notifications

```

customer_id	customer_name	customer_address	city	postal_code	order_id	order_date	customer_id
1	สมวิช	บางเขน	กรุงเทพ	10230	2	2020-01-12 00:00:00	10009
2	อนัน	ปทุมธานี	กรุงเทพ	10330	3	2020-01-10 23:00:00	10001
3	อนัน	ปทุมธานี	กรุงเทพ	10330	4	2020-01-11 23:00:00	10001
4	อนัน	ปทุมธานี	กรุงเทพ	10330	5	2020-01-12 23:00:00	10001
5	อนัน	ปทุมธานี	กรุงเทพ	10330	6	2020-01-13 23:00:00	10001
6	อนัน	ปทุมธานี	กรุงเทพ	10330	7	2020-01-14 23:00:00	10001

```

60 select *
61 from customer left join ordert on customer.customer_id = ordert.customer_id
62
Data Output Messages Notifications

```

customer_id	customer_name	customer_address	city	postal_code	order_id	order_date	customer_id
39	บัวลดา	ตเล็ชัย	กรุงเทพ	10310	41	2020-01-29 23:00:00	10008
40	บัวลดา	ตเล็ชัย	กรุงเทพ	10310	42	2020-01-27 23:00:00	10008
41	บัวลดา	ตเล็ชัย	กรุงเทพ	10310	43	2020-01-28 23:00:00	10008
42	บัวลดา	ตเล็ชัย	กรุงเทพ	10310	44	2020-01-29 23:00:00	10008
43	ธรรม	ปทุมธานี	กรุงเทพ	10330	[null]	[null]	[null]
44	จันทร์เจ้า	บางกะปี	กรุงเทพ	10310	[null]	[null]	[null]

```

60 select customer.customer_id, customer.customer_name, ordert.customer_id
61 from customer left join ordert on customer.customer_id = ordert.customer_id
62
Data Output Messages Notifications

```

customer_id	customer_name	customer_id
39	บัวลดา	10008
40	บัวลดา	10008
41	บัวลดา	10008
42	บัวลดา	10008
43	ธรรม	[null]
44	จันทร์เจ้า	[null]

ที่นี่คือ 10004, 10006 แล้ว??

```

60 select customer.customer_id, customer.customer_name
61 from customer left join ordert on customer.customer_id = ordert.customer_id
62 group by customer.customer_id
63
Data Output Messages Notifications

```

customer_id	customer_name
1	สมวิช
2	แสงทอง
3	อนัน
4	จันทร์เจ้า
5	ธน่า
6	ธรรม
7	มาร์ศรี
8	บัวลดา
9	ສอาด้า

ไม่ใช่ group by แล้วซักท่าครับ!!

```

60 select customer.customer_id, customer.customer_name
61 from customer left join ordert on customer.customer_id = ordert.customer_id
62 group by customer.customer_id
63 having count(ordert.order_id) = 0
64
Data Output Messages Notifications

```

customer_id	customer_name
1	จันทร์เจ้า
2	อนัน
3	ธรรม
4	มาร์ศรี

ใช่ having

4.12 จงแสดงจำนวนออเดอร์ที่เกิดขึ้นระหว่างวันที่ 2020-01-10 ถึง 2020-01-15

Expected output

total_orders

12

select count()

from —

where

```
88 select count(ordert.order_id) as total_orders
89 from ordert
90 where ordert.order_date BETWEEN '2020-01-10 00:00:00' AND '2020-01-15 23:59:59'
```

Data Output Messages Notifications



total_orders	bigint
1	12

4.13 จงแสดงค่า total payment ของออเดอร์หมายเลข 3

Expected output

total_payment

15500.00

```
select *  
from order_line join product on order_line.product_id = product.product_id
```

Data Output Messages Notifications

order_id	product_id	ordered_quantity	product_id	product_description	product_finish	standard_price
integer	integer	integer	integer	character varying (50)	colors	numeric (8,2)
1	2	2	3	Arm chair	Walnut	1200.00
2	2	3	10	3 Cabinet	White Ash	500.00
3	2	4	1	4 Triple Cabinet	Natural Ash	1500.00
4	2	5	10	5 Sofabed	Cherry	7500.00
5	3	1	10	1 Stool bar	Red Oak	250.00
6	3	3	11	3 Cabinet	White Ash	500.00
7	3	5	1	5 Sofabed	Cherry	7500.00
8	4	6	10	6 Lazy Boy	White Ash	2000.00
9	5	1	1	1 Stool bar	Red Oak	250.00

```
65 select *  
66 from order_line join product on order_line.product_id = product.product_id  
67 where order_line.order_id = 3
```

Data Output Messages Notifications

order_id	product_id	ordered_quantity	product_id	product_description	product_finish	standard_price
integer	integer	integer	integer	character varying (50)	colors	numeric (8,2)
1	3	1	10	1 Stool bar	Red Oak	250.00
2	3	3	11	3 Cabinet	White Ash	500.00
3	3	5	1	5 Sofabed	Cherry	7500.00

select sum(order_line.ordered_quantity * product.standard_price) as total_payment
from order_line join product on order_line.product_id = product.product_id
where order_line.order_id = 3

Data Output Messages Notifications

total_payment
15500.00

จำนวน

4.14 เพิ่มข้อมูลสินค้าใหม่ต่อไปนี้เข้าตาราง product

PRODUCT_ID, PRODUCT_DESCRIPTION, PRODUCT_FINISH, STANDARD_PRICE

7,kitchen cabinet,Cherry,1500.00

8,table,Red Oak,550.00

และหลังเพิ่มแล้วให้ทำการ select รายการทั้งหมดของ product ออกรายงานด้วย



Syntax:

```
INSERT INTO table(column1, column2, ...)
VALUES
    (value1, value2, ...);
```

IN PostgreSQL, the INSERT statement is used to add new rows to a database table. As the name suggests, it inserts new data into the table. PostgreSQL provides the INSERT statement to insert data into the database. Syntax: `INSERT INTO table(column1, column2, ...) VALUES (value1, value2, ...);`

(1)

(2)

```
182 INSERT INTO product(product_id, product_description, product_finish, standard_price)
183 VALUES
184 ('7', 'kitchen cabinet', 'Cherry', 1500.00),
185 ('8', 'table', 'Red Oak', 550.00)
186
187 select * from product
188
```

Data Output Messages Notifications

product_id	product_description	product_finish	standard_price
1	Steelbar	Steel	290.00
2	Am-cher	Walnut	1000.00
3	Am-ash	Walnut	1000.00
4	Tropic Cabinet	Natural Ash	1500.00
5	Sofabed	Cherry	7500.00
6	Lacy Roy	White Ash	2000.00
7	kitchen cabinet	Cherry	1500.00
8	table	Red Oak	550.00

(3)

๙๘๗

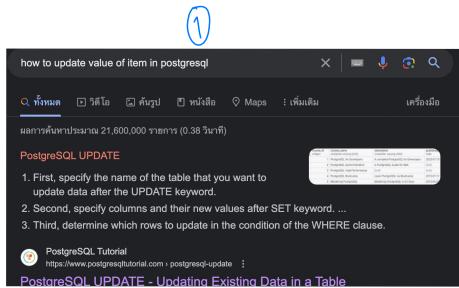
- 4.15 ลบรายการทั้งหมดของ customer_id 10001 ออกจากตาราง customer และข้อมูลทั้งหมดของ
customer 10001 ในตารางทั้งหมดที่เกี่ยวข้อง จากนั้นทำการ select * from ตาราง customer,
ordert, และ order_line ออกมาแสดง

4.16 ทำการ update ราคา product Sofabed จาก 7500.00 เหลือ 5400.00 และทำการ select * from

product มาแสดง

Expected output

product_id	product_description	product_finish	standard_price
1	Stool bar	Red Oak	250.00
2	Arm chair	Walnut	1200.00
3	Cabinet	White Ash	500.00
4	Triple Cabinet	Natural Ash	1500.00
6	Lazy Boy	White Ash	2000.00
7	kitchen cabinet	Cherry	1500.00
8	table	Red Oak	550.00
5	Sofabed	Cherry	5400.00



```
117 update product
118 set standard_price = 5400.00
119 where product.product_description = 'Sofabed'
120
121 select * from product
```

ເຖິງບໍລິການ (ເກົ່າສະຫງົບສະໄໝ)

ວິທີ່
ວ.11 → ສັນຕະກຳ

ວ.16 → ຂົມໜຸນ

4.17 multiple tables with intermediate (cont'd database 1(w))

expected output

You want to retrieve a list of customers along with the details of their orders. Here's the SQL to create these tables and the query to join them:

```
20 select * from customers
21
```

Data Output Messages Notifications

	customer_id	name	email
1	1	John Doe	john.doe@example.com
2	2	Jane Smith	jane.smith@example.com

```
20 select * from orders
21
```

Data Output Messages Notifications

	order_id	order_date	customer_id
1	1	2023-01-01	1
2	2	2023-01-02	2

```
20 select * from order_details
21
```

Data Output Messages Notifications

	customer_id	name	detail_id	product_name	quantity
1	1	John Doe	1	Laptop	1
2	1	John Doe	2	Mouse	2
3	2	Jane Smith	3	Keyboard	1

```
20 select *
21 from customers
22 join orders on customers.customer_id = orders.customer_id
23 join order_details on orders.order_id = order_details.order_id
24 Data Output Messages Notifications
```

	customer_id	name	email	order_id	order_date	customer_id	detail_id	product_name	quantity
1	1	John Doe	john.doe@example.com	1	2023-01-01	1	1	Mouse	1
2	1	John Doe	john.doe@example.com	1	2023-01-01	1	2	Laptop	1
3	2	Jane Smith	jane.smith@example.com	2	2023-01-02	2	3	Keyboard	1

```
20 select customers.customer_id, customers.name, order_details.detail_id, order_details.product_name, order_details.quantity
21 from customers
22 join orders on customers.customer_id = orders.customer_id
23 join order_details on orders.order_id = order_details.order_id
24 where order_details.detail_id = 1
25 order by order_details.order_id
26 Data Output Messages Notifications
```

	customer_id	name	detail_id	product_name	quantity
1	1	John Doe	1	Laptop	1
2	1	John Doe	2	Mouse	1
3	2	Jane Smith	3	Keyboard	1

```
20 select customers.customer_id, customers.name, order_details.detail_id, order_details.product_name, order_details.quantity
21 from customers
22 join orders on customers.customer_id = orders.customer_id
23 join order_details on orders.order_id = order_details.order_id
24 where order_details.detail_id = 1
25 order by order_details.order_id
26 Data Output Messages Notifications
```

	customer_id	name	detail_id	product_name	quantity
1	1	John Doe	1	Laptop	1
2	1	John Doe	2	Mouse	1
3	2	Jane Smith	3	Keyboard	1

```
20 select *
21 from (
22 select c.customer_id, c.name, od.detail_id, od.product_name, od.quantity
23 from customers c
24 join orders o on c.customer_id = o.customer_id
25 join order_details od on o.order_id = od.order_id
26 join products p on od.product_name = p.product_name
27 group by c.customer_id, c.name, od.detail_id, od.product_name, od.quantity
28 ) q
29 where q.detail_id = 1
30 order by q.customer_id
31 Data Output Messages Notifications
```

	customer_id	name	detail_id	product_name	quantity
1	1	John Doe	1	Laptop	1
2	1	John Doe	2	Mouse	1
3	2	Jane Smith	3	Keyboard	1

```
20 select *
21 from (
22 select c.customer_id, c.name, od.detail_id, od.product_name, od.quantity
23 from customers c
24 join orders o on c.customer_id = o.customer_id
25 join order_details od on o.order_id = od.order_id
26 join products p on od.product_name = p.product_name
27 group by c.customer_id, c.name, od.detail_id, od.product_name, od.quantity
28 ) q
29 where q.detail_id = 1
30 order by q.customer_id
31 Data Output Messages Notifications
```

	customer_id	name	detail_id	product_name	quantity
1	1	John Doe	1	Laptop	1
2	1	John Doe	2	Mouse	1
3	2	Jane Smith	3	Keyboard	1

```
ordersystem# select * from customer;
customer_id | customer_name | customer_address | city | postal_code
-----+-----+-----+-----+-----+
10001 | กาน | ถนนสุขุมวิท 10330
10002 | ลักษณ์ | ถนนสุขุมวิท 10330
10003 | นิติ | ถนนสุขุมวิท 10700
10004 | นราพร | ถนนสุขุมวิท 10700
10005 | ธรรม | ถนนสุขุมวิท 10230
10006 | นิตา | ถนนสุขุมวิท 10230
10007 | จันทร์ | ถนนสุขุมวิท 10310
10008 | ปราสาท | ถนนสุขุมวิท 10310
10009 | อรุณ | ถนนสุขุมวิท 10310
(9 rows)
```

Select, Groupby Count ()

ORDER BY

4.1 แสดงจำนวนลูกค้าแยกตาม postal_code โดยเรียงตามจำนวน customers มากไปน้อย

Expected output

postal_code	customer_numbers
10700	2
10330	2
10230	2
10310	2
10600	1

```
1 select postal_code, COUNT(customer_id) AS customer_numbers
2 From customer
3 GROUP BY postal_code
4 ORDER BY COUNT(customer_id) DESC
```

เมื่อ postal_code นำมารักษา

4.2 แสดงจำนวนลูกค้าแยกตาม postal_code โดยเรียงตามจำนวน customers มากไปน้อย โดยแสดงเฉพาะ postal_code ที่มีจำนวนลูกค้ามากกว่า 1

Expected output

postal_code	customer_numbers
10700	2
10330	2
10230	2
10310	2

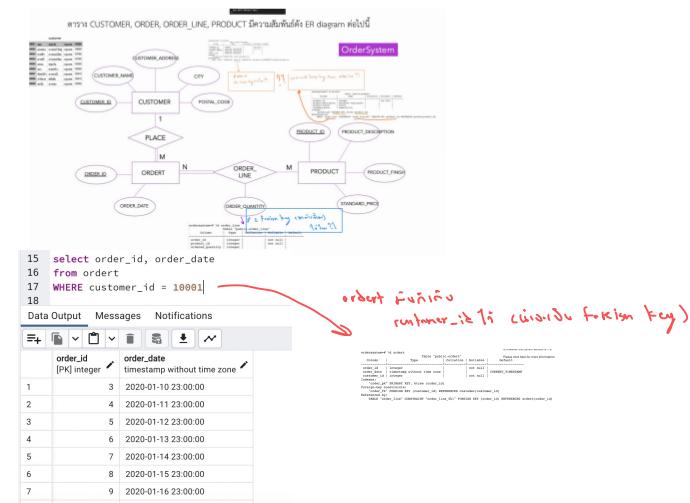
```
8 select postal_code, COUNT(customer_id) AS customer_numbers
9 From customer
10 GROUP BY postal_code
11 Having COUNT(customer_id) > 1
12 ORDER BY COUNT(customer_id) DESC
13
```

postal_code	customer_numbers
10700	2
10330	2
10230	2
10310	2

4.3 จงแสดงรายการของลูกค้ารหัส 10001

Expected output

order_id	order_date
3	2020-01-10 23:00:00
4	2020-01-11 23:00:00
5	2020-01-12 23:00:00
6	2020-01-13 23:00:00
7	2020-01-14 23:00:00
8	2020-01-15 23:00:00
9	2020-01-16 23:00:00
10	2020-01-17 23:00:00
11	2020-01-18 23:00:00
12	2020-01-19 23:00:00
14	2020-01-20 23:00:00
15	2020-01-21 23:00:00



select order_id, order_date
from order
where order.customer_id = 10001

เมื่อใน order_line.order_id

4.4 แสดงรหัสสินค้าและคำบรรยายสินค้าที่มีรายการสั่งซื้อจำนวนมากสุด

Expected output

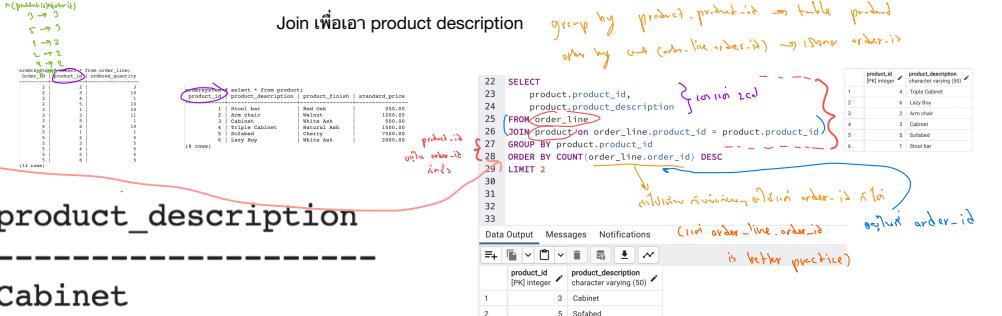
(select clear id).

(bus count)



or base on number of orders appear in
not total quantity

กรุณาปิดไฟล์นี้ก่อน



product id | product description

-----+
3 | Cabinet
5 | Sofabed

4.5 จงแสดงรหัสลูกค้า ชื่อลูกค้า และจำนวนของเครื่องที่สั่งโดยลูกค้าแต่ละคน โดยเรียงลำดับข้อมูลตามจำนวน
ของเครื่องมากไปน้อย

Expected output

customer_id	customer_name	number_of_orders
10002	แสงทอง	20
10001	جون	12
10003	สายฟ้า	5
10008	ปาริชาต	4
10009	สมวัย	1

4.6 จงแสดงรหัสลูกค้า และชื่อลูกค้า ที่มีจำนวนออร์เดอร์รวมมากที่สุด

Expected output

customer_id	customer_name	คำอ่านภาษาไทย
10002	แสงทอง	

4.7 แสดงรหัสสินค้าและคำบรรยายสินค้าที่มีจำนวนการสั่งซื้อมากสุด

Expected output

product_id	product_description
3	Cabinet

4.8 จงแสดงรายชื่อลูกค้าที่มีจำนวนออเดอร์มากสุด 3 อันดับแรกและจำนวนออร์เดอร์รวมของพวกเขา

Expected output

customer_id	customer_name	order_count
10002	แสงทอง	20
10001	جون	12
10003	สายฟ้า	5

4.9 จงแสดงรหัสและรายชื่อลูกค้า (ไม่ซ้ำ) ที่มีการสั่งสินค้าตั้งแต่วันที่ 2020-01-10 ถึง 2020-01-15

Expected output

```
70 select distinct on (customer.customer_id) customer.customer_id, customer.customer_name
71 from customer join ordert on customer.customer_id = ordert.customer_id
72 WHERE ordert.order_date BETWEEN '2020-01-10 00:00:00' AND '2020-01-15 23:59:59'
73 group by customer.customer_id
74
```

Data Output Messages Notifications

☰

	customer_id [PK] integer	customer_name character varying (25)
1	10001	ອອນ
2	10002	ອອນລາວ

customer_id	customer_name
10001	จอน
10002	แสงทอง

4.10 จงแสดงรายการหัศสินค้า คำอธิบายสินค้า และ product_finish ทั้งหมดที่มีสีขาวประกอบ

Expected output

product_id	product_description	product_finish
3	Cabinet	White Ash
6	Lazy Boy	White Ash

4.11 จงแสดงรายการรหัสและชื่อลูกค้าทั้งหมดที่ยังไม่เคยมีรายการสั่งซื้อของเลย

Expected output

<code>customer_id</code>	<code>customer_name</code>
10004	มาร์ค
10005	ธีรอน
10006	ธน่า
10007	จันทร์เจ้า

Expected output

select count()

total orders

From —

her

12

4.13 จงแสดงค่า total payment ของออเดอร์หมายเลข 3

Expected output

total_payment

15500.00

13

• sum of quantity = price

4.14 เพิ่มข้อมูลสินค้าใหม่ต่อไปในเข้าตาราง product

PRODUCT ID PRODUCT DESCRIPTION PRODUCT FINISH STANDARD PRICE

7 kitchen cabinet Cherry 1500.00

8 table Bed Oak 550.00

และหลังเพิ่มแล้วให้ทำการ select รายการทั้งหมดของ product ออกมาแสดงด้วย

In PostgreSQL, the `INSERT` statement is used to add new rows to a database table. As one can imagine, it is a database operation that must be atomic, so it is part of the ACID properties of a database. Syntax: `INSERT INTO table(column1, column2, ...) VALUES (value1, value2, ...)`

Syntax:

60

```
INSERT INTO employee (
    employee_id,
    first_name,
    last_name,
    manager_id
)
VALUES
    (1, 'Sandeep', 'Jain', NULL),
    (2, 'Abhishek', 'Kleinlein', 1),
    (3, 'Harsh', 'Aggarwal', 1),
    (4, 'Raju', 'Kumar', 2),
    (5, 'Nikhil', 'Aggarwal', 2),
    (6, 'Anshul', 'Aggarwal', 2),
    (7, 'Virat', 'Kohli', 3),
```

1

```
182 INSERT INTO product(product_id, product_description, product_finish, standard_price)
183 VALUES
184     ('T_1', 'Kitchen cabinet', 'Curry', 1500.00,
185      'W_1', 'Oak', 2000.00),
186     ('T_2', 'Bed Oak', 500.00);
187
188 select * from product
```

Data Output: Messages Notifications

	product_id	product_description	product_finish	standard_price
1	T_1	Kitchen cabinet	Curry	1500.00
2	T_2	Bed Oak	Red Oak	2000.00
3	A_1	Arm chair	Walnut	1200.00
4	A_2	Chair	Walnut	500.00
5	A_3	Triple Cabinet	Natural Ash	1500.00
6	S_1	Sofabed	Cherry	7500.00
7	L_1	Lazy Boy	White Ash	2000.00

run តារាងនិវាទ
សាស្ត្រ
នៅ run ពាណិជ្ជកម្ម
(unique primary key)

Expected output

product_id	product_description	product_finish	standard_price
1	Stool bar	Red Oak	250.00
2	Arm chair	Walnut	1200.00
3	Cabinet	White Ash	500.00
4	Triple Cabinet	Natural Ash	1500.00
5	Sofabed	Cherry	7500.00
6	Lazy Boy	White Ash	2000.00
7	kitchen cabinet	Cherry	1500.00
8	table	Red Oak	550.00

๙๙๙๖

4.15 ลบรายการทั้งหมดของ customer_id 10001 ออกจากตาราง customer และข้อมูลทั้งหมดของ customer 10001 ในตารางทั้งหมดที่เกี่ยวข้อง จากนั้นทำการ select * from ตาราง customer, ordert, และ order_line ออกรายงานแสดง

PostgreSQL - DELETE

Read Discuss Courses

To delete data from a table PostgreSQL has a DELETE statement.

Syntax:

```
DELETE FROM table
WHERE condition;
```

```
DELETE FROM employee
WHERE last_name = 'Kelenia';
```

```

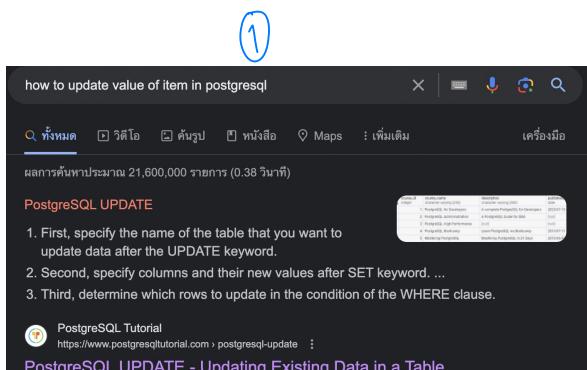
113 ALTER TABLE ordert
114 DROP CONSTRAINT IF EXISTS order_fk, -- Drop the existing constraint
115 ADD CONSTRAINT order_fk_new
116 FOREIGN KEY (customer_id)
117 REFERENCES customer(customer_id)
118 ON DELETE CASCADE;
119
120 ALTER TABLE order_line
121 DROP CONSTRAINT IF EXISTS order_line_fk1, -- Drop the existing constraint
122 ADD CONSTRAINT order_line_fk1_new
123 FOREIGN KEY (order_id)
124 REFERENCES ordert(order_id)
125 ON DELETE CASCADE;
126
127 DELETE FROM ordert
128 WHERE customer_id = 10001;
129
130 DELETE FROM customer
131 WHERE customer_id = 10001;
132
133 SELECT * FROM customer;
134 SELECT * FROM ordert;
135 SELECT * FROM order_line;
136
Data Output Messages Notifications
```

order_id	product_id	ordered_quantity
1	2	3
2	2	10
3	2	1
4	2	10

4.16 ทำการ update ราคา product Sofabed จาก 7500.00 เหลือ 5400.00 และทำการ select * from product มาแสดง

Expected output

product_id	product_description	product_finish	standard_price
1	Stool bar	Red Oak	250.00
2	Arm chair	Walnut	1200.00
3	Cabinet	White Ash	500.00
4	Triple Cabinet	Natural Ash	1500.00
6	Lazy Boy	White Ash	2000.00
7	kitchen cabinet	Cherry	1500.00
8	table	Red Oak	550.00
5	Sofabed	Cherry	5400.00



```
UPDATE table_name  
SET column1 = value1,  
    column2 = value2,  
    ...  
WHERE condition;
```

```
UPDATE courses  
SET published_date = '2020-08-01'  
WHERE course_id = 3;
```

```
117 update product
118 set standard_price = 5400.00
119 where product.product_description = 'Sofabed'
120
121 select * from product
```

Data Output Messages Notifications

≡+

	product_id [PK] integer	product_description character varying (50)	product_finish colors	standard_price numeric (8,2)
1	1	Stool bar	Red Oak	250.00
2	2	Arm chair	Walnut	1200.00
3	3	Cabinet	White Ash	500.00
4	4	Triple Cabinet	Natural Ash	1500.00
5	6	Lazy Boy	White Ash	2000.00
6	7	kitchen cabinet	Cherry	1500.00
7	8	table	Red Oak	550.00
8	5	Sofabed	Cherry	5400.00

Businesses

- e.g. (individuals customer-id numbers)

ans save command

The screenshot shows the pgAdmin interface. At the top, there's a status bar indicating the user is running version 7.5 of pgAdmin, while the current version is 7.8. Below the status bar is a toolbar with various icons. The main area has tabs for 'Query' (selected), 'Query History', and 'Save File'. The 'Query' tab contains the following SQL code:

```
1 select postal_code,
2 from customer
3 group by postal_code
4 order by count(customer)
```

The 'Save File' tab is open, showing a file browser window. The 'Save As' field is highlighted with a green circle and a circled number '6'. The browser lists three files: '4_1.sql' (modified Nov 1, 2023, size 126.0 B), 'banking' (modified Oct 11, 2023, size 126.0 B), and 'ordersystem' (modified Oct 29, 2023, size 126.0 B). Below the browser is a 'File Format' dropdown set to 'sql'.

At the bottom of the pgAdmin window, there's a 'Data Output' tab showing a table with two columns: 'postal_code' and 'customer_numbers'. The data is as follows:

	postal_code	customer_numbers
1	10700	2
2	10230	2
3	10310	2
4	10600	1
5	10330	1

Below the table, it says 'Total rows: 5 of 5' and 'Query complete 00:00:00.071'. In the bottom right corner, it says 'Ln 4, Col 10'.