ตาราง CUSTOMER, ORDERT, ORDER\_LINE, PRODUCT มีความสัมพันธ์ดัง ER diagram ต่อไปนี้

Diagram

Description automatically generated

**จากข้อมูลข้างต้นขอให้นิสิตเขียนคำสั่ง SQL เพื่อตอบโจทย์ต่อไปนี้**

1. แสดงจำนวนลูกค้าแยกตาม postal\_code

SELECT postal\_code, COUNT(postal\_code)

FROM CUSTOMER

GROUP BY postal\_code;

+-------------+--------------------+

| postal\_code | count(postal\_code) |

+-------------+--------------------+

| 10230 | 1 |

| 10310 | 2 |

| 10330 | 2 |

| 10600 | 1 |

| 10700 | 2 |

+-------------+--------------------+

1. แสดงจำนวนลูกค้าแยกตาม postal\_code เฉพาะที่มีจำนวนลูกค้ามากกว่า 1

SELECT postal\_code, COUNT(postal\_code) as sumpost

FROM CUSTOMER

GROUP BY postal\_code

HAVING sumpost > 1;

+-------------+---------+

| postal\_code | sumpost |

+-------------+---------+

| 10310 | 2 |

| 10330 | 2 |

| 10700 | 2 |

+-------------+---------+

1. จงแสดงรายการออเดอร์ของลูกค้ารหัส 10001;

SELECT \*

FROM ORDERT

WHERE CUSTOMER\_ID = 10001;

+----------+---------------------+-------------+

| ORDER\_ID | ORDER\_DATE | CUSTOMER\_ID |

+----------+---------------------+-------------+

| 1 | 2021-10-09 14:16:03 | 10001 |

| 2 | 2021-10-09 14:16:03 | 10001 |

| 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 |

| 11 | 2020-01-18 23:00:00 | 10001 |

| 12 | 2020-01-19 23:00:00 | 10001 |

| 14 | 2020-01-20 23:00:00 | 10001 |

| 15 | 2020-01-21 23:00:00 | 10001 |

+----------+---------------------+-------------+

1. แสดงรหัสสินค้าและคำบรรยายสินค้าที่มีรายการสั่งซื้อมากสุด

CREATE VIEW ordered\_products AS

SELECT product\_id, count(order\_id) AS sum\_products

FROM order\_line

GROUP BY product\_id;

CREATE VIEW top\_product AS

SELECT product\_id

FROM ordered\_products

WHERE sum\_products IN (SELECT MAX(sum\_products) FROM ordered\_products);

SELECT product\_id, product\_description

FROM PRODUCT

WHERE product\_id IN (SELECT \* FROM top\_product);

+------------+---------------------+

| product\_id | product\_description |

+------------+---------------------+

| 1 | Table |

+------------+---------------------+

1. จงแสดงรหัสลูกค้า ชื่อลูกค้า และจำนวนออเดอร์ที่สั่งโดยลูกค้าแต่ละคน โดยเรียงลำดับข้อมูลตามจำนวนออเดอร์จากมากไปน้อย

SELECT C.customer\_id, C.customer\_name, count(order\_id)

FROM CUSTOMER C, ORDERT O

WHERE C.customer\_id = O.customer\_id

GROUP BY c.customer\_id;

+-------------+-----------------------+-----------------+

| customer\_id | customer\_name | count(order\_id) |

+-------------+-----------------------+-----------------+

| 10001 | จอน | 14 |

| 10002 | แสงทอง | 20 |

| 10003 | สายฟ้า | 5 |

| 10008 | ปาริชาต | 4 |

+-------------+-----------------------+-----------------+

1. จงแสดงรหัสลูกค้า และชื่อลูกค้า ที่มีจำนวนออร์เดอร์มากที่สุด

CREATE VIEW customer\_orders AS

SELECT C.customer\_id, C.customer\_name, count(order\_id) AS sum\_orders

FROM CUSTOMER C, ORDERT O

WHERE C.customer\_id = O.customer\_id

GROUP BY C.customer\_id;

SELECT customer\_id, customer\_name

FROM customer\_orders

WHERE sum\_orders = (SELECT MAX(sum\_orders) FROM customer\_orders);

+-------------+--------------------+

| customer\_id | customer\_name |

+-------------+--------------------+

| 10002 | แสงทอง |

+-------------+--------------------+

1. แสดงรหัสสินค้าและคำบรรยายสินค้าที่มีจำนวนการสั่งซื้อมากสุด

CREATE VIEW ordered\_product\_quantity AS

SELECT product\_id, sum(ORDERED\_QUANTITY) AS sum\_product\_quantity

FROM order\_line

GROUP BY product\_id;

CREATE VIEW top\_product\_quantity AS

SELECT product\_id

FROM ordered\_product\_quantity

WHERE sum\_product\_quantity IN (SELECT MAX(sum\_product\_quantity) FROM ordered\_product\_quantity);

SELECT product\_id, product\_description

FROM PRODUCT

WHERE product\_id IN (SELECT \* FROM top\_product\_quantity);

+------------+---------------------+

| product\_id | product\_description |

+------------+---------------------+

| 3 | Cabinet |

+------------+---------------------+

1. จงแสดงรายชื่อลูกค้าที่มีจำนวนออเดอร์มากสุด 3 อันดับแรกและจำนวนออร์เดอร์รวมของพวกเขา

SELECT C.customer\_id, C.customer\_name, count(order\_id) AS order\_count

FROM CUSTOMER C, ORDERT O

WHERE C.customer\_id = O.customer\_id

GROUP BY c.customer\_id

ORDER BY order\_count DESC LIMIT 3;

+-------------+--------------------+-------------+

| customer\_id | customer\_name | order\_count |

+-------------+--------------------+-------------+

| 10002 | แสงทอง | 20 |

| 10001 | จอน | 14 |

| 10003 | สายฟ้า | 5 |

+-------------+--------------------+-------------+

1. จงแสดงรหัสและรายชื่อลูกค้า (ไม่ซ้ำ) ที่มีการสั่งสินค้าตั้งแต่วันที่ 2020-01-10 ถึง 2020-01-15

SELECT distinct C.customer\_id, C.customer\_name

FROM customer C, ordert O

WHERE C.customer\_id = O.customer\_id AND

order\_date BETWEEN '2020-01-10' AND '2020-01-15';

+-------------+

| customer\_id |

+-------------+

| 10001 |

| 10002 |

+-------------+

1. จงแสดงรายการรหัสสินค้า คำอธิบายสินค้า และ product\_finish ทั้งหมดที่มีสีขาวประกอบ

SELECT product\_id, product\_description, product\_finish FROM product WHERE PRODUCT\_FINISH LIKE '%White%';

+------------+---------------------+

| product\_id | product\_description |

+------------+---------------------+

| 3 | Cabinet |

| 6 | Lazy Boy |

+------------+---------------------+

1. จงแสดงรายการรหัสและชื่อลูกค้าทั้งหมดที่ยังไม่เคยมีรายการสั่งซื้อของเลย
2. จงแสดงจำนวนออเดอร์ที่เกิดขึ้นระหว่างวันที่ 2020-01-10 ถึง 2020-01-15
3. จงแสดงค่า total payment ของออเดอร์หมายเลข 3
4. จงแสดงค่า total payment ของทุกออเดอร์ที่มีข้อมูลสินค้าที่สั่ง
5. จงแสดงรหัสและชื่อลูกค้าที่มี total payment ต่อออเดอร์สูงสุด
6. NATURAL JOIN
7. LEFT OUTER JOIN, RIGHT OUTER JOIN

ตารางเดิม เพิ่มข้อมูล

ตารางเดิมเพิ่ม columns

เปลี่ยนชื่อ column

SQL เพิ่มเติม แต่ใกล้เคียงกัน