

SQL - ER diagram Customer Reviews

What i did...

- ในฐานะ Analyst ต้องช่วยธุรกิจในการหาข้อมูลเชิงผ่านสิ่งที่ลูกค้ารีวิว จากนั้นนำสิ่งนั้นมาปรับปรุงเพื่อให้ธุรกิจเดินต่อไปตามความต้องการลูกค้า
 - ตั้งคำถามเพื่อตอบโจทย์ธุรกิจ
 - หาเรตติ้งเฉลี่ยของสินค้าแต่ละรายการ
 - ดึงข้อมูลลูกค้าที่ให้เรตติ้งสูงเขียนไขคือลูกค้าให้เรตติ้งมากกว่า 4
 - นับจำนวนครั้งที่ลูกค้าแต่ละคนเข้ามาเขียนรีวิว
 - สินค้าที่ได้เรตติ้งเท่ากับหรือมากกว่า 3.5 เป็นสินค้าชนิดใดบ้าง ?
 - นับจำนวนข้อความที่เป็นรีวิวด้านลบจะมีคำว่า "disappointed"
 - ลูกค้าคนไหนบ้างที่เคยเขียนข้อความรีวิวในด้านบวกให้คะแนนเรตติ้งสูง (≥ 4)
 - หาค่าเฉลี่ยเรตติ้งในแต่ละวันในสัปดาห์
 - ลูกค้าแต่ละคนมักเขียนรีวิวโดยเฉลี่ยกี่คำ
 - สร้าง ER Diagram เพื่อทำให้การวิเคราะห์ในงานง่ายขึ้น รวดเร็วขึ้น
 - ดึงข้อมูลเพื่อวิเคราะห์

Query Data

```
# หาค่าเฉลี่ย rating ของสินค้าแต่ละรายการ
dbGetQuery(con,
"select
  pr.product_id AS product_id,
  pr.product_name AS product_name,
  ROUND(AVG(re.rating),2) AS average_rating
from products AS pr
INNER JOIN reviews AS re
  ON pr.product_id = re.product_id
GROUP BY 1,2"
)
```

- Product ที่ได้คะแนนรีวิวมากที่สุดคือ Product C 3.88 ส่วน Product B ได้ค่าเฉลี่ยเรตติ้งน้อยที่สุดต้องเข้าไปดูต่อว่าเกิดอะไรขึ้นและปรับปรุง

product_id	product_name	average_rating
1	Product A	3.67
2	Product B	3
3	Product C	3.88

```
# ดึงข้อมูลลูกค้าที่ให้ Rating มากกว่า 4
dbGetQuery(con,
"select
  DISTINCT cu.name,
  cu.contact_info
FROM customers AS cu
INNER JOIN reviews AS re
  ON cu.customer_id = re.customer_id
WHERE rating > 4
```

```
ORDER BY rating DESC"
)
```

- ลูกค้าที่ให้คะแนนเรตติ้งมากกว่า 4 คือคุณ John Doe

name	contact_info
John Doe	john.doe@example.com
Bob Johnson	bob.johnson@example.com
Emma Davis	emma.davis@example.com
Daniel Miller	daniel.miller@example.com
David Lee	david.lee@example.com

```
#นับจำนวนครั้งลูกค้าแต่ละคนเข้ามาเขียนรีวิว
dbGetQuery(con,
"select
  cu.name AS name,
  COUNT(re.customer_id) AS count_review
from customers AS cu
INNER JOIN reviews AS re
  ON cu.customer_id = re.customer_id
GROUP BY 1
ORDER BY count_review DESC"
)
```

- คุณ John Doe คือคนที่เขียนรีวิวมากที่สุดในกลุ่มของลูกค้า

name	count_review
John Doe	5
Alice Smith	4
Sarah Wilson	2
Olivia Taylor	2
Michael Brown	2
Emma Davis	2
Emily White	2
David Lee	2
Daniel Miller	2
Bob Johnson	2

```
#สินค้าที่ได้เรตติ้ง >= 3.5 เป็นสินค้าชนิดใดบ้าง
dbGetQuery(con,
"SELECT
  pr.product_id AS product_id,
  pr.product_name AS product_name,
  AVG(re.rating) AS average_rating
FROM products AS pr
INNER JOIN reviews AS re
  ON pr.product_id = re.product_id
GROUP BY 1,2
```

```
HAVING AVG(re.rating) >= 3.5"
)
```

- สินค้าที่ได้เรตติ้งมากกว่า 3.5 คือ Product A และ Product C

product_id	product_name	average_rating
1	Product A	3.67
3	Product C	3.88

```
#นับจำนวนข้อความที่เป็นรีวิวด้านลบจะมีคำว่า "disappointed"
dbGetQuery(con,
"SELECT
  COUNT(review_text) AS count_review
FROM reviews
WHERE review_text LIKE '%disappointed%'
")
```

- คำตอบคือมี 2 ข้อความ

```
#ลูกค้าคนไหนบ้างที่เคยเขียนข้อความรีวิวในด้านบวกให้คะแนนเรตติ้งสูง >= 4
dbGetQuery(con,
"SELECT
  cu.customer_id AS customer_id,
  cu.name AS name
FROM customers AS cu
INNER JOIN reviews AS re
  ON cu.customer_id = re.customer_id
WHERE review_text LIKE '%excellent%' OR review_text LIKE '%impressive%' OR
  review_text LIKE '%recommend%' and rating >= 4"
)
```

- มีอยู่ 3 ท่านที่รีวิวในด้านบวกและให้คะแนนเรตติ้งมากกว่า 4 คือคุณ John Doe, Bob Johnson และ Daniel Miller

customer_id	name
1	John Doe
3	Bob Johnson
9	Daniel Miller

```
# หาค่าเฉลี่ย rating ในแต่ละวันในสัปดาห์
dbGetQuery(con,
"SELECT
  CASE STRFTIME('%w', date)
    WHEN '0' THEN 'Sunday'
    WHEN '1' THEN 'Monday'
    WHEN '2' THEN 'Tuesday'
    WHEN '3' THEN 'Wednesday'
    WHEN '4' THEN 'Thursday'
    WHEN '5' THEN 'Friday'
    ELSE 'Saturday'
  END 'day_of_week',
  AVG(rating) AS avg_rate
```

```
FROM Reviews
GROUP BY 1" )
```

- วันที่ได้คะแนนเรตติ้งเฉลี่ยสูงคือวันศุกร์ รองลงมาคือวันจันทร์และวันพฤหัสบดี

day_of_week	avg_rate
Friday	4
Monday	4
Saturday	3
Sunday	3
Thursday	4
Tuesday	3
Wednesday	3.67

```
# ลูกค้าแต่ละคนมักเขียนรีวิวเฉลี่ยกี่คำ
dbGetQuery(con,
"SELECT
  cu.name,
  cu.contact_info,
  AVG(LENGTH(re.review_text)) AS length_text_avg
FROM Customers AS cu
INNER JOIN Reviews AS re ON cu.customer_id = re.customer_id
GROUP BY 1
ORDER BY length_text_avg DESC;" )
```

- ลูกค้าที่เขียนรีวิวยาวที่สุดคือคุณ Olivia Taylor

name	contact_info	length_text_avg
Olivia Taylor	olivia.taylor@example.com	46.5
Alice Smith	alice.smith@example.com	41.75
Emma Davis	emma.davis@example.com	38.5
David Lee	david.lee@example.com	37
Bob Johnson	bob.johnson@example.com	37
Emily White	emily.white@example.com	36.5
Sarah Wilson	sarah.wilson@example.com	36
Daniel Miller	daniel.miller@example.com	34.5
John Doe	john.doe@example.com	33.8
Michael Brown	michael.brown@example.c...	30.5

ER diagram

```
Table customers {
  customer_id int [primary key]
  name varchar
  contact_info varchar
}

Table products {
  product_id int [primary key]
  product_name varchar
}
```

```

Table reviews {
  review_id int [primary key]
  customer_id int
  product_id int
  rating int
  review_text varchar
  date datetime
}

// one - many relationship
ref a_relationship {
  customers.customer_id < reviews.customer_id
}

// one - many relationship
ref a_relationship {
  products.product_id < reviews.product_id
}

```

