Customer Sales and Analysis

A diagram of a product

Description automatically generated

**-Products with empty stock in each store**

SELECT

s.store\_id,

s.store\_name,

p.product\_id,

p.product\_name,

st.quantity

FROM

stores s

JOIN

Stocks st

ON

s.store\_id = st.store\_id

JOIN

products p

ON

p.product\_id = st.product\_id

WHERE

st.quantity =0;

A screenshot of a table

Description automatically generated

A screenshot of a phone

Description automatically generated

**-Retrieve a list of all customers along with their order count:**

SELECT

c.customer\_id,

c.first\_name,

c.last\_name,

COUNT(o.order\_id) AS order\_count

FROM

customers c

LEFT JOIN

orders o ON c.customer\_id = o.customer\_id

GROUP BY

c.customer\_id, c.first\_name, c.last\_name

ORDER BY

c.customer\_id;

A screenshot of a table

Description automatically generated

**Find the top 5 best-selling products based and its total revenue**

SELECT

p.product\_id,

p.product\_name,

oi.list\_price,

SUM(oi.quantity) AS total\_units\_sold,

SUM(oi.quantity \* (oi.list\_price \* (1 - oi.discount))) AS total\_revenue

FROM

order\_items oi

INNER JOIN

products p ON oi.product\_id = p.product\_id

GROUP BY

p.product\_id,

p.product\_name

ORDER BY

total\_units\_sold DESC

LIMIT 5

A screenshot of a list

Description automatically generated

**-Rank stores based on their total revenues**

WITH StoreRevenue AS (

SELECT

s.store\_id,

s.store\_name,

SUM(oi.quantity \* oi.list\_price \* (1 - oi.discount)) AS total\_revenue

FROM

stores s

JOIN

orders o ON s.store\_id = o.store\_id

JOIN

order\_items oi ON o.order\_id = oi.order\_id

GROUP BY

s.store\_id, s.store\_name

)

SELECT

store\_id,

store\_name,

total\_revenue,

RANK() OVER (ORDER BY total\_revenue DESC) AS store\_rank

FROM

StoreRevenue;

A screenshot of a number

Description automatically generated

**- Average units sold for each month of the year, for each product category**

WITH AvgUnitsSold AS (

SELECT

strftime('%Y', order\_date) AS year,

strftime('%m', order\_date) AS month,

c.category\_name,

AVG(oi.quantity) AS avg\_units\_sold

FROM

order\_items oi

INNER JOIN

orders o ON oi.order\_id = o.order\_id

INNER JOIN

products p ON oi.product\_id = p.product\_id

INNER JOIN

categories c ON p.category\_id = c.category\_id

GROUP BY

strftime('%Y', order\_date),

strftime('%m', order\_date),

c.category\_name

)

SELECT

year,

month,

category\_name,

avg\_units\_sold

FROM

AvgUnitsSold

ORDER BY

year,

month,

category\_name;

A screenshot of a computer

Description automatically generated

- What city and state has the highest customer concentration?

SELECT

city,

state,

COUNT (DISTINCT customer\_id) AS total\_customers

FROM

customers

GROUP BY

city, state

ORDER BY

total\_customers DESC;

