SQL Queries:

1. 52 Weeks of Sales History: “select count of orders grouped by week”

(grouped\_by\_week.sql)

1. SELECT EXTRACT(WEEK FROM time) AS week\_number, COUNT(\*) AS order\_count FROM orders GROUP BY week\_number ORDER BY week\_number;

1. Realistic Sales History: “select count of orders, sum of order total grouped by hour”

(grouped\_by\_hour.sql)

2. SELECT EXTRACT(HOUR FROM time) AS hour\_of\_day, COUNT(\*) AS order\_count, SUM(price) AS total\_order\_amount FROM orders GROUP BY hour\_of\_day ORDER BY hour\_of\_day;

1. 2 Peak Days: “select top 10 sums of order total grouped by day in descending order”

(top\_10\_sums\_of\_orders.sql)

3.  SELECT

EXTRACT(DAY FROM time) AS day\_of\_month,

SUM(price) AS total\_order\_amount

FROM orders

GROUP BY day\_of\_month

ORDER BY total\_order\_amount DESC

LIMIT 10;

1. 20 Items in Inventory: “select row count from inventory”

(row\_count.sql)

4.  SELECT COUNT(\*) AS total\_items

FROM stock;

1. “retrieve all items”

(retrieve\_items.sql)

5. SELECT \* FROM items;

1. “retrieve all merch”

(retrieve\_merch.sql)

6. SELECT \* FROM merch;

1. “retrieve last 10 orders”

(retrieve\_last10\_orders.sql)

7. SELECT \* FROM orders ORDER BY time DESC LIMIT 10;

1. “calculate average cost for all orders”

(cost\_avg\_orders.sql)

1. SELECT AVG(price) AS average\_cost FROM orders;

1. “Get items will stock less than 10”

(less\_than\_10\_stock.sql)

9. SELECT \* FROM stock WHERE amount < 10;

1. “items in stock”

(items\_in\_stock.sql)

1. SELECT stockname, amount

2. FROM stock

3. WHERE amount > 0;