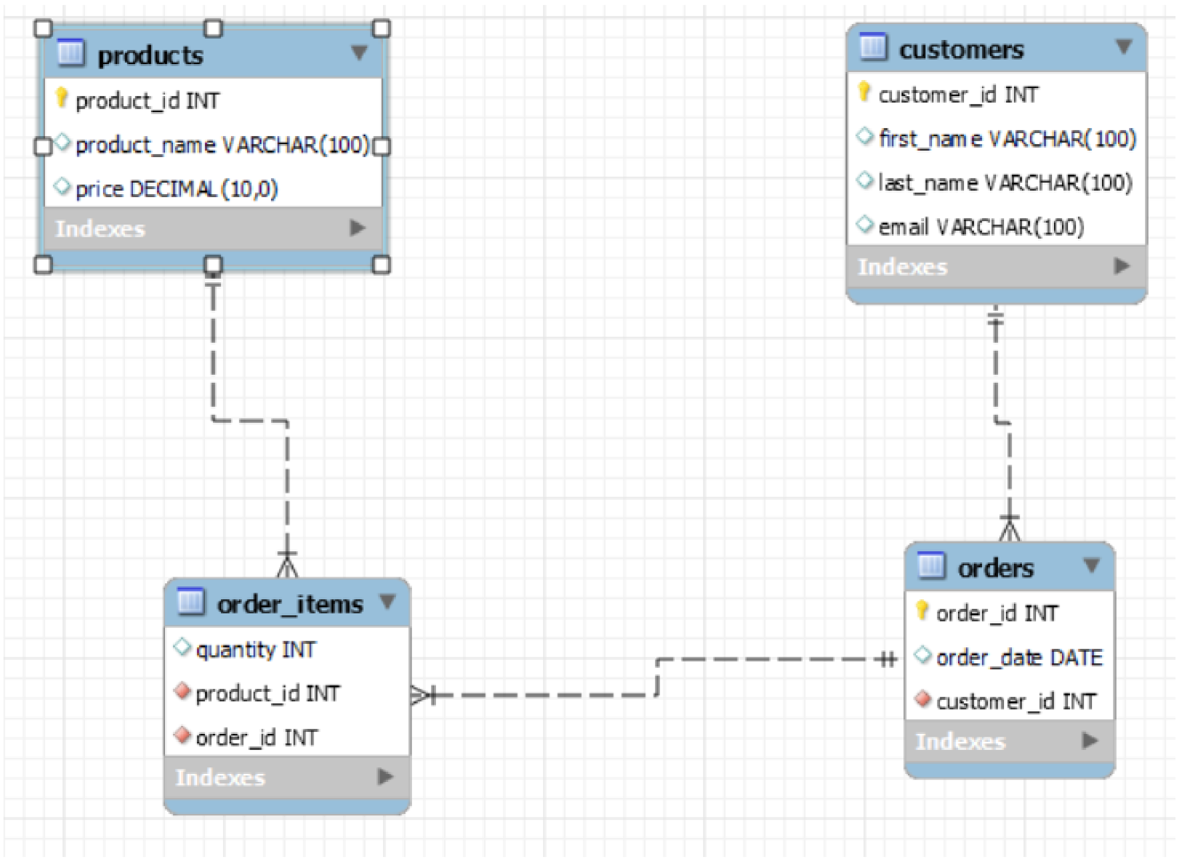
SQL CASE STUDY
DATA IN MOTION
TINY SHOP SALES
SHOP
OPEN
DATA IN MOTION

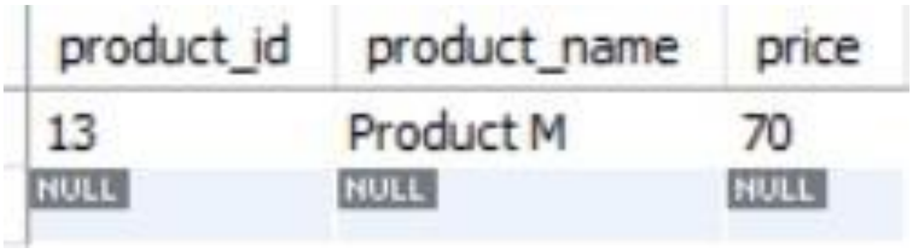
Database Schema



-- 1. Which product has the highest price? Only return a single row

```
SELECT * FROM products
WHERE price = (SELECT MAX (price) FROM products);
```

OUTPUT :-



product_id	product_name	price
13	Product M	70
NULL	NULL	NULL

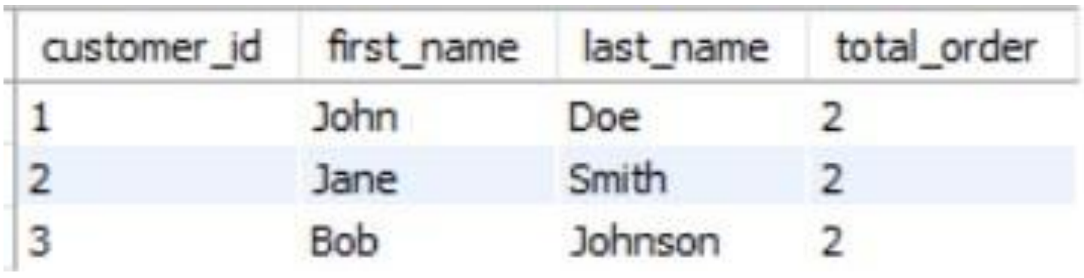
-- 2. Which customer has made the most orders?

```
SELECT
        c.customer_id, c.first_name, c.last_name,
        count(o.order_id) as total_order
FROM customers c
JOIN orders o
ON c.customer_id = o.customer_id
GROUP BY o.customer_id LIMIT 3;
```

-- Using CTE

```
WITH cte AS (
SELECT c.customer_id, c.first_name, c.last_name,
          count(o.order_id) as total_order
FROM customers c
JOIN orders o
ON c.customer_id = o.customer_id
GROUP BY o.customer_id),
cte1 AS(
SELECT *, dense_rank() OVER (ORDER BY total_order DESC) AS rk FROM cte)
SELECT customer_id, first_name, last_name, total_order FROM cte1 WHERE rk = 1;
```

OUTPUT :-

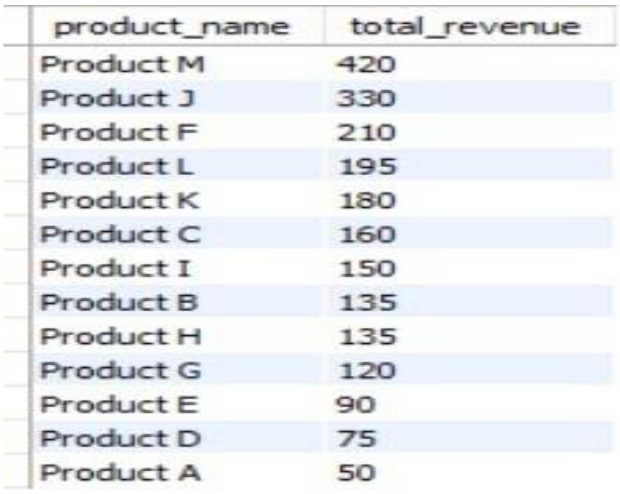


customer_id	first_name	last_name	total_order
1	John	Doe	2
2	Jane	Smith	2
3	Bob	Johnson	2

-- 3. What’s the total revenue per product?

```
WITH cte AS(
SELECT p.product_name, p.price,
            o.quantity FROM products p
JOIN order_items o ON p.product_id = o.product_id),
cte1 AS
(SELECT *, (price*quantity) AS total FROM cte)
SELECT product_name, SUM(total) AS total_revenue FROM cte1
GROUP BY product_name ORDER BY total_revenue DESC;
```

OUTPUT :-

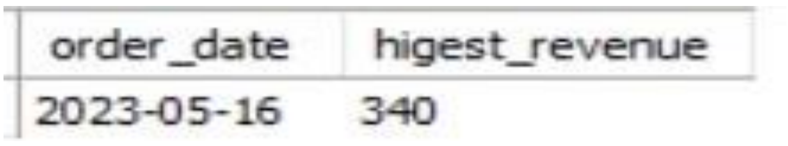


product_name	total_revenue
Product M	420
Product J	330
Product F	210
Product L	195
Product K	180
Product C	160
Product I	150
Product B	135
Product H	135
Product G	120
Product E	90
Product D	75
Product A	50

-- 4. Find the day with the highest revenue.

```
WITH cte AS(
SELECT o.order_date, p.price, ot.quantity
FROM products p
JOIN order_items ot ON p.product_id = ot.product_id
JOIN orders o ON ot.order_id = o.order_id),
cte1 AS ( SELECT *, (price*quantity) AS total FROM cte)
SELECT order_date, SUM(total) AS higest_revenue FROM cte1
GROUP BY order_date ORDER BY higest_revenue DESC LIMIT 1;
```

OUTPUT :-

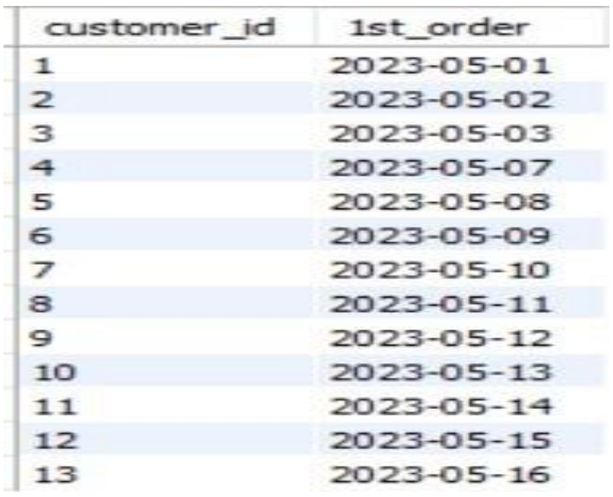


order_date	higest_revenue
2023-05-16	340

-- 5. Find the first order (by date) for each customer

```
SELECT customer_id, MIN(order_date) AS 1st_order FROM orders

GROUP BY customer_id;
```

OUTPUT :-



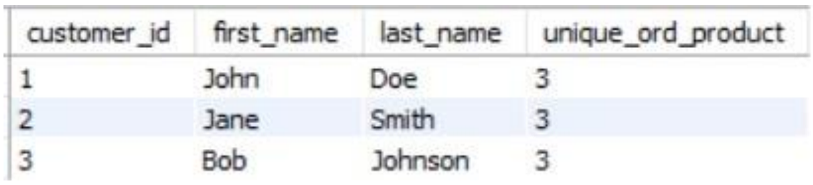
customer_id	1st_order
1	2023-05-01
2	2023-05-02
3	2023-05-03
4	2023-05-07
5	2023-05-08
6	2023-05-09
7	2023-05-10
8	2023-05-11
9	2023-05-12
10	2023-05-13
11	2023-05-14
12	2023-05-15
13	2023-05-16

-- 6. Find the top 3 customers who have ordered the most distinct products

-- Using CTE

```
WITH cte AS

(SELECT c.customer_id, c.first_name, c.last_name, ot.product_id

FROM customers c

JOIN orders o ON c.customer_id = o.customer_id

JOIN order_items ot ON ot.order_id = o.order_id)

SELECT customer_id, first_name, last_name, COUNT(DISTINCT product_id) AS
unique_ord_product

FROM cte

GROUP BY c.customer_id,c.first_name, c.last_name

LIMIT 3;
```

OUTPUT :-

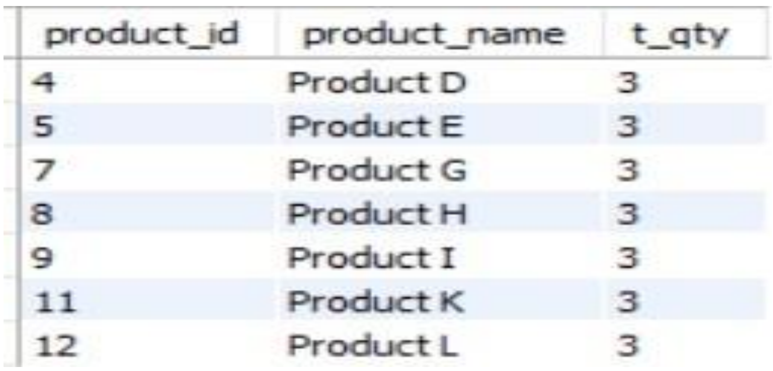


customer_id	first_name	last_name	unique_ord_product
1	John	Doe	3
2	Jane	Smith	3
3	Bob	Johnson	3

-- 7. Which product has been bought the least in terms of quantity?

```
WITH cte AS
(SELECT p.product_id, p.product_name, ot.quantity
FROM products p
JOIN order_items ot ON p.product_id = ot.product_id),
cte1 AS
(SELECT product_id, product_name, SUM(quantity) AS t_qty FROM cte
GROUP BY product_id, product_name ORDER BY t_qty ASC),
cte2 AS
(SELECT *, DENSE_RANK() OVER(ORDER BY t_qty) AS rk FROM cte1)
SELECT product_id, product_name, t_qty FROM cte2 WHERE rk = 1;
```

OUTPUT :-

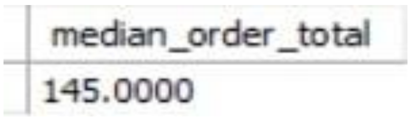


product_id	product_name	t_qty
4	Product D	3
5	Product E	3
7	Product G	3
8	Product H	3
9	Product I	3
11	Product K	3
12	Product L	3

-- 8. What is the median order total?

```
WITH cte1 AS (
  SELECT o.order_date, o.customer_id, i.quantity, p.price
  FROM order_items i
  JOIN products p ON i.product_id = p.product_id
  JOIN orders o ON o.order_id = i.order_id),
cte2 AS ( SELECT *, (quantity * price) AS total_revenue FROM cte1),
cte3 AS ( SELECT customer_id, SUM(total_revenue) AS total_revenue FROM cte2 GROUP BY
customer_id)
SELECT AVG(total_revenue) AS median_order_total
FROM ( SELECT total_revenue, ROW_NUMBER() OVER (ORDER BY total_revenue) AS
row_num,    COUNT(*) OVER () AS total_rows
  FROM cte3) AS subquery
WHERE row_num IN (FLOOR((total_rows + 1) / 2), CEIL((total_rows + 1) / 2));
```

OUTPUT :-

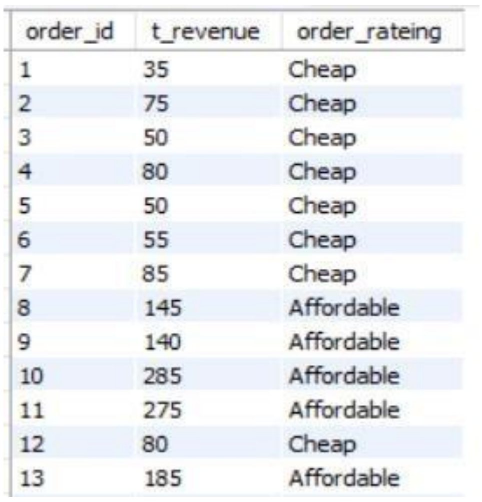


median_order_total
145.0000

-- 9. For each order, determine if it was ‘Expensive’ (total over 300), ‘Affordable’ (total over 100), or ‘Cheap’.

```
WITH cte AS (
        SELECT p.product_id,ot.order_id, p.price, ot.quantity FROM products p
        JOIN order_items ot ON p.product_id = ot.product_id        ),
cte1 AS (SELECT *, (price*quantity) AS revenue FROM cte),
cte2 AS (SELECT order_id, SUM(revenue) AS t_revenue FROM cte1     GROUP BY order_id)
SELECT order_id, t_revenue,
        CASE
        WHEN t_revenue>300 THEN "Expensive"
        WHEN t_revenue>100 THEN "Affordable"
        ELSE "Cheap"
        END AS order_rateing FROM cte2;
```

OUTPUT :-

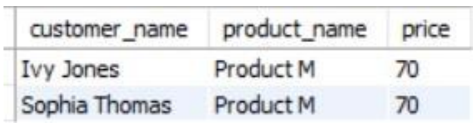


order_id	t_revenue	order_rateing
1	35	Cheap
2	75	Cheap
3	50	Cheap
4	80	Cheap
5	50	Cheap
6	55	Cheap
7	85	Cheap
8	145	Affordable
9	140	Affordable
10	285	Affordable
11	275	Affordable
12	80	Cheap
13	185	Affordable

-- 10. Find customers who have ordered the product with the highest price.

```
WITH cte AS (
        SELECT o.customer_id, CONCAT(c.first_name, " ", c.last_name) AS customer_name,
                        p.product_name, p.price
        FROM products p
        JOIN order_items ot ON p.product_id = ot.product_id
        JOIN orders o ON o.order_id = ot.order_id
        JOIN customers c ON o.customer_id = c.customer_id)
SELECT customer_name, product_name, price FROM cte
WHERE price = (SELECT MAX(price) FROM cte);
```

OUTPUT :-



customer_name	product_name	price
Ivy Jones	Product M	70
Sophia Thomas	Product M	70