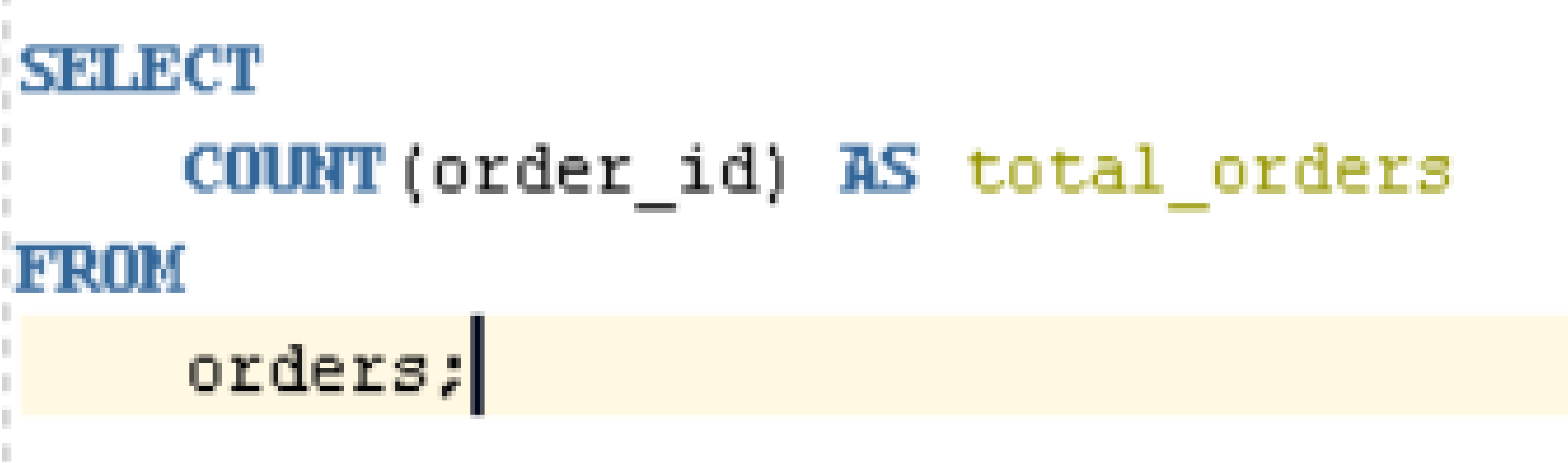
PIZZA SALES SQL PROJECT

BY VEDANT GAIKWAD

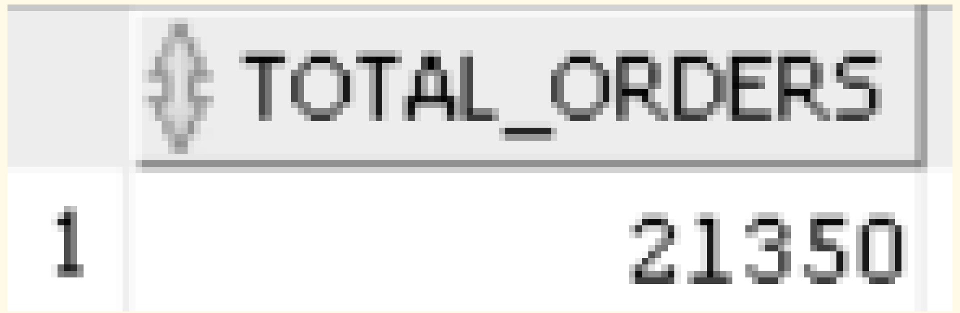
Here is where your presentation begins

--BASIC QUESTIONS--

Q1.Retrive the total number of orders placed.



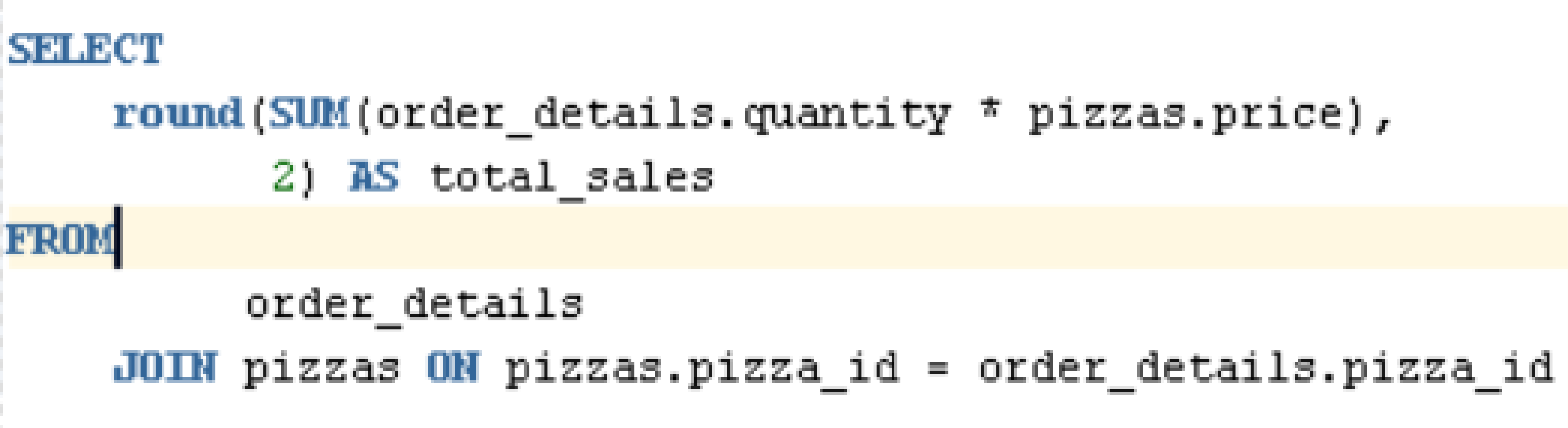
```
SELECT
    COUNT(order_id) AS total_orders
FROM
    orders;
```



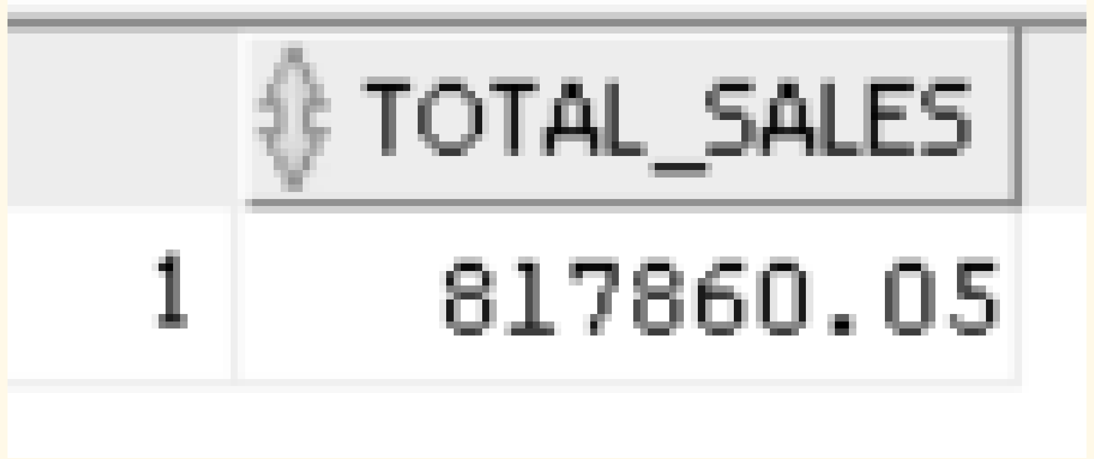
```
     TOTAL_ORDERS
1           21350
```

--BASIC QUESTIONS--

Q2.Calculate the total revenue from pizza sales.



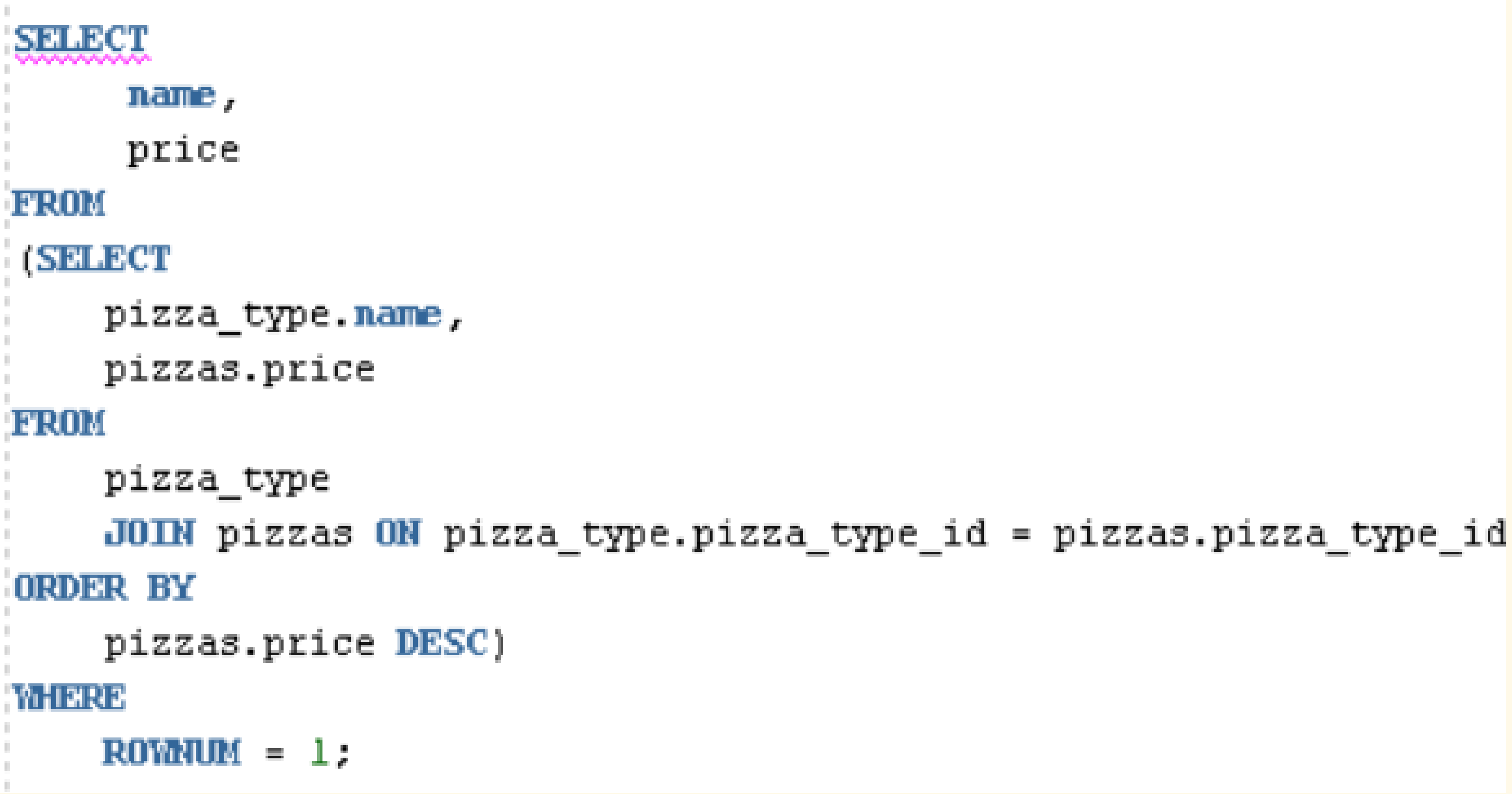
```
SELECT
    round(SUM(order_details.quantity * pizzas.price),
          2) AS total_sales
FROM
        order_details
    JOIN pizzas ON pizzas.pizza_id = order_details.pizza_id
```



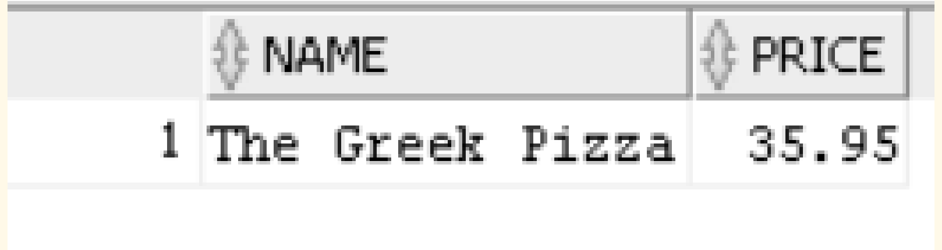
	TOTAL_SALES
1	817860.05

--BASIC QUESTIONS--

Q3.Identify the highest-priced pizza.



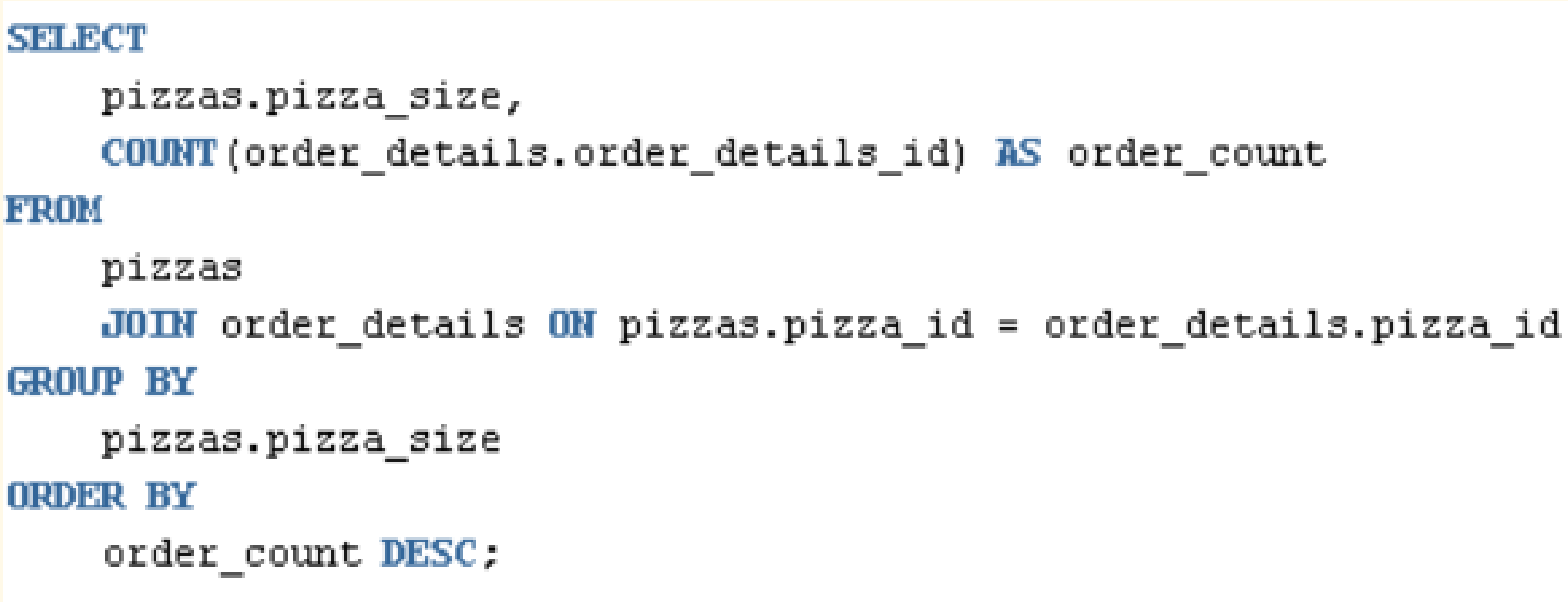
```
SELECT
      name,
      price
FROM
(SELECT
     pizza_type.name,
     pizzas.price
FROM
     pizza_type
     JOIN pizzas ON pizza_type.pizza_type_id = pizzas.pizza_type_id
ORDER BY
     pizzas.price DESC)
WHERE
     ROWNUM = 1;
```



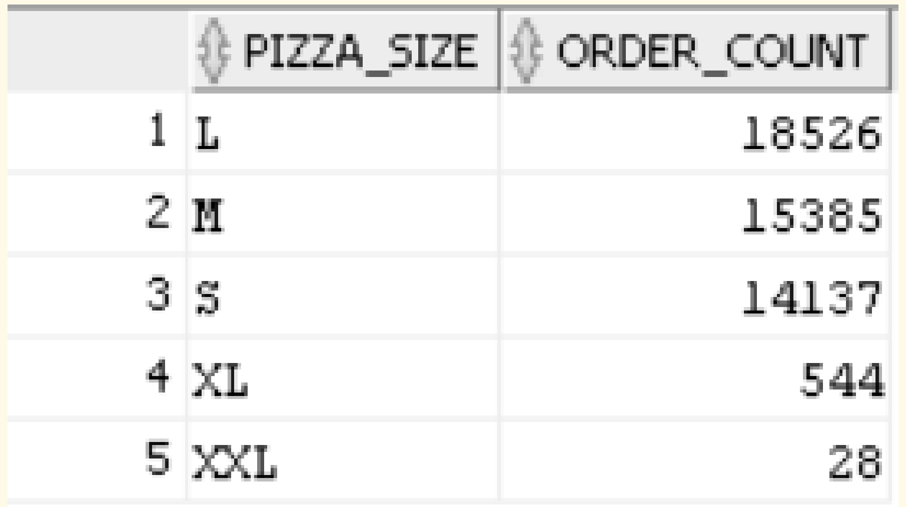
	NAME	PRICE
1	The Greek Pizza	35.95

--BASIC QUESTIONS--

Q4. Identify the most common pizza size ordered.



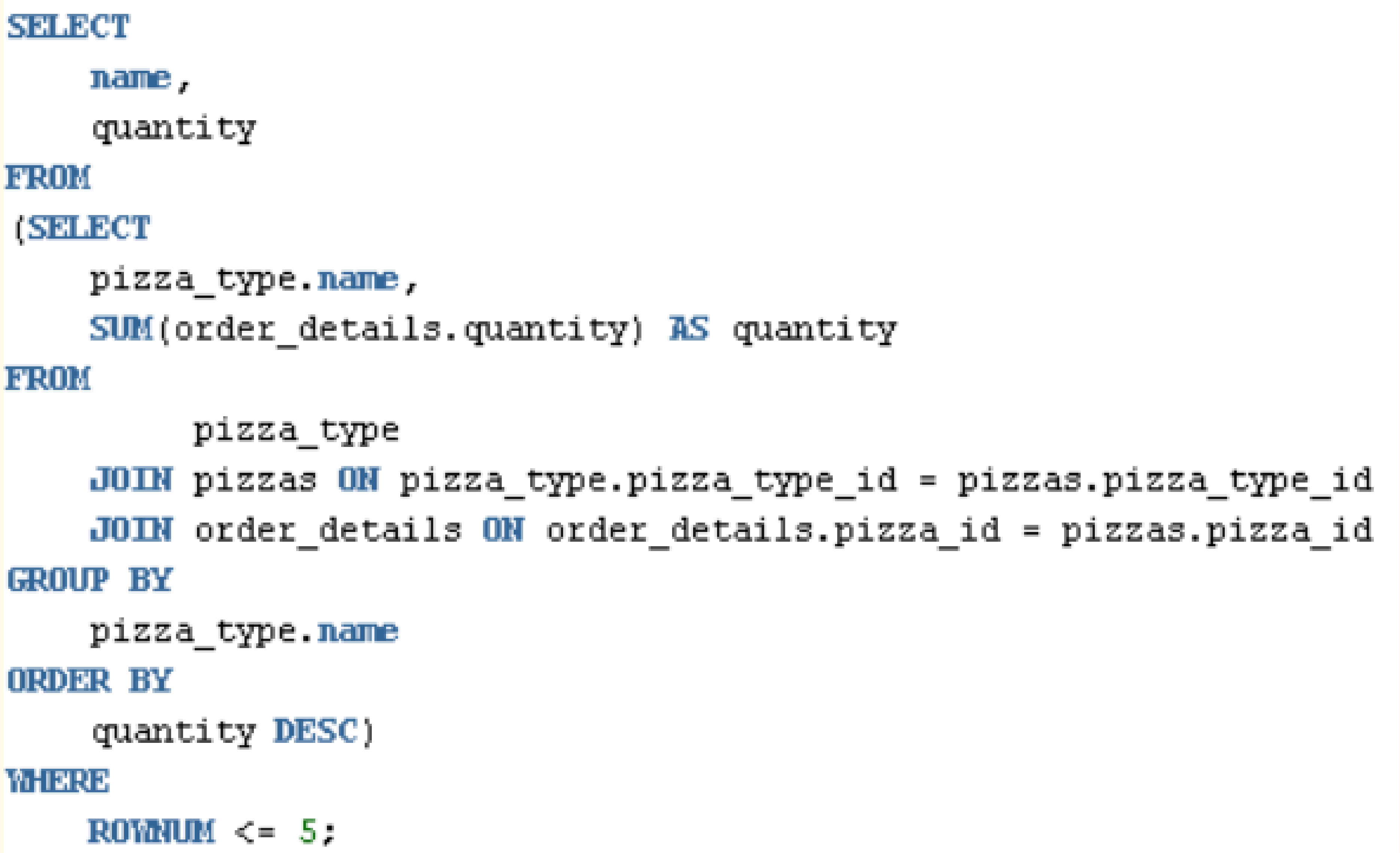
```
SELECT
    pizzas.pizza_size,
    COUNT(order_details.order_details_id) AS order_count
FROM
    pizzas
    JOIN order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY
    pizzas.pizza_size
ORDER BY
    order_count DESC;
```



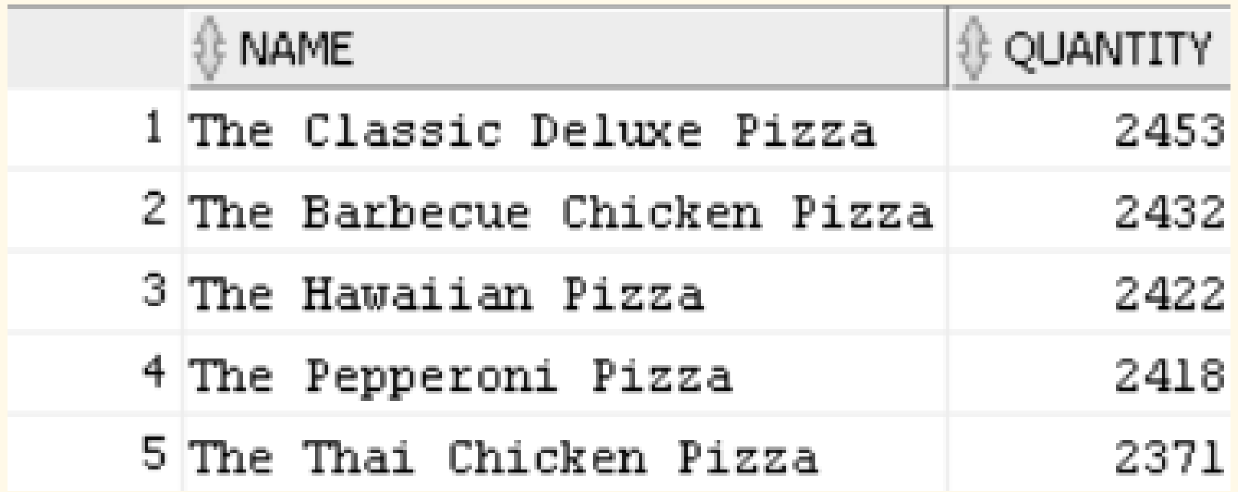
	PIZZA_SIZE	ORDER_COUNT
1	L	18526
2	M	15385
3	S	14137
4	XL	544
5	XXL	28

--BASIC QUESTIONS--

Q5.List top 5 most ordered pizza types along with their quantities.



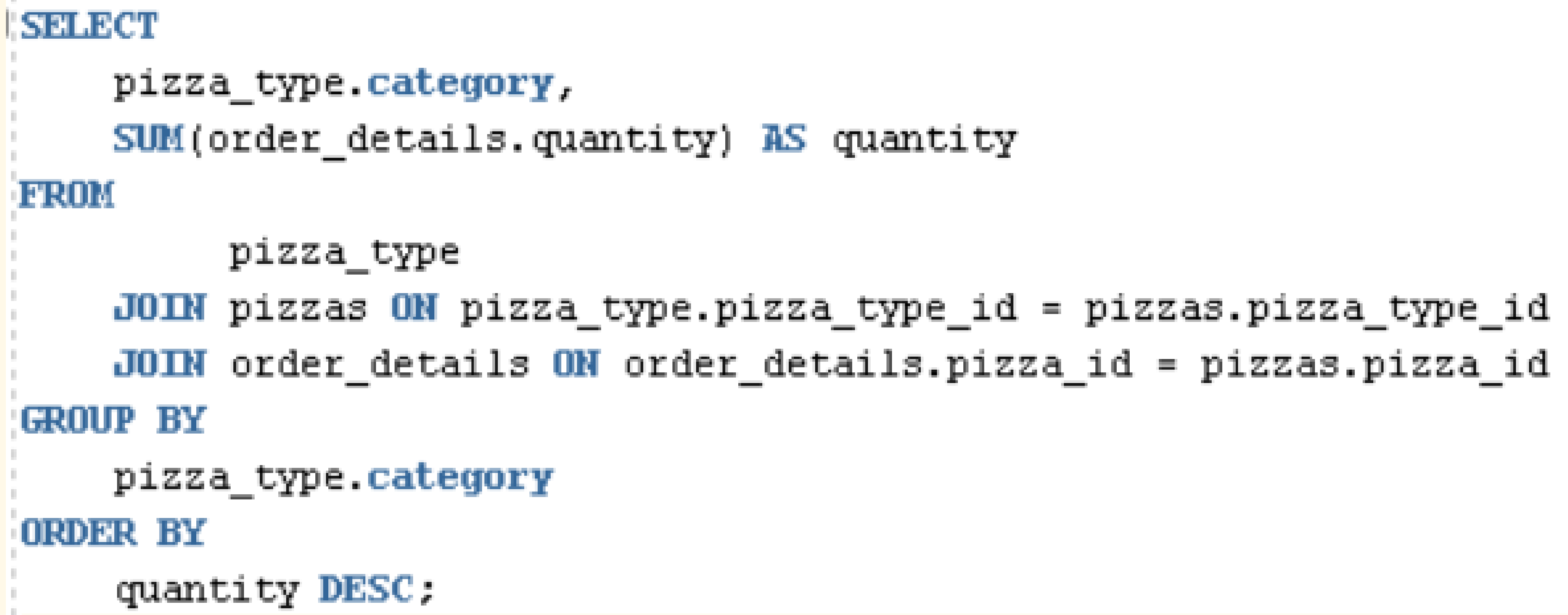
```
SELECT
    name,
    quantity
FROM
(SELECT
    pizza_type.name,
    SUM(order_details.quantity) AS quantity
FROM
        pizza_type
    JOIN pizzas ON pizza_type.pizza_type_id = pizzas.pizza_type_id
    JOIN order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY
    pizza_type.name
ORDER BY
    quantity DESC)
WHERE
    ROWNUM <= 5;
```



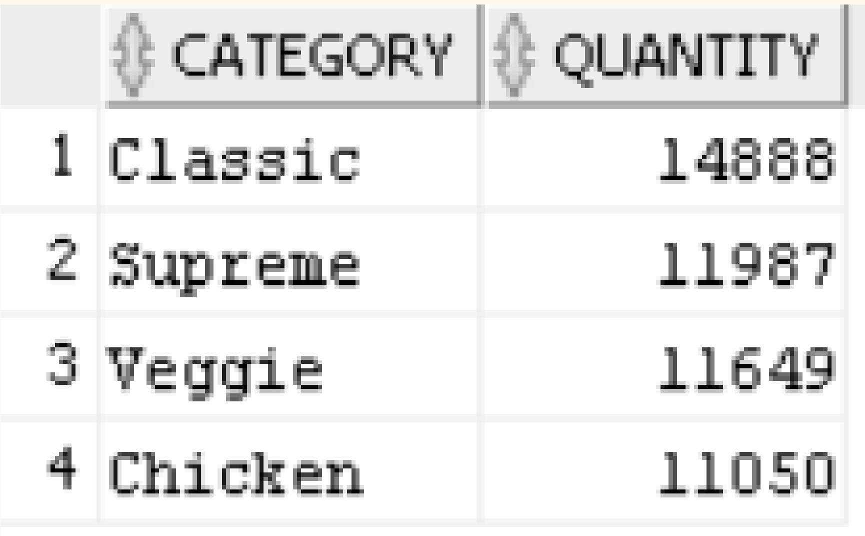
	NAME	QUANTITY
1	The Classic Deluxe Pizza	2453
2	The Barbecue Chicken Pizza	2432
3	The Hawaiian Pizza	2422
4	The Pepperoni Pizza	2418
5	The Thai Chicken Pizza	2371

INTERMEDIATE QUESTIONS

Q1.Join the necessary tables to find the total quantity of each pizza category.



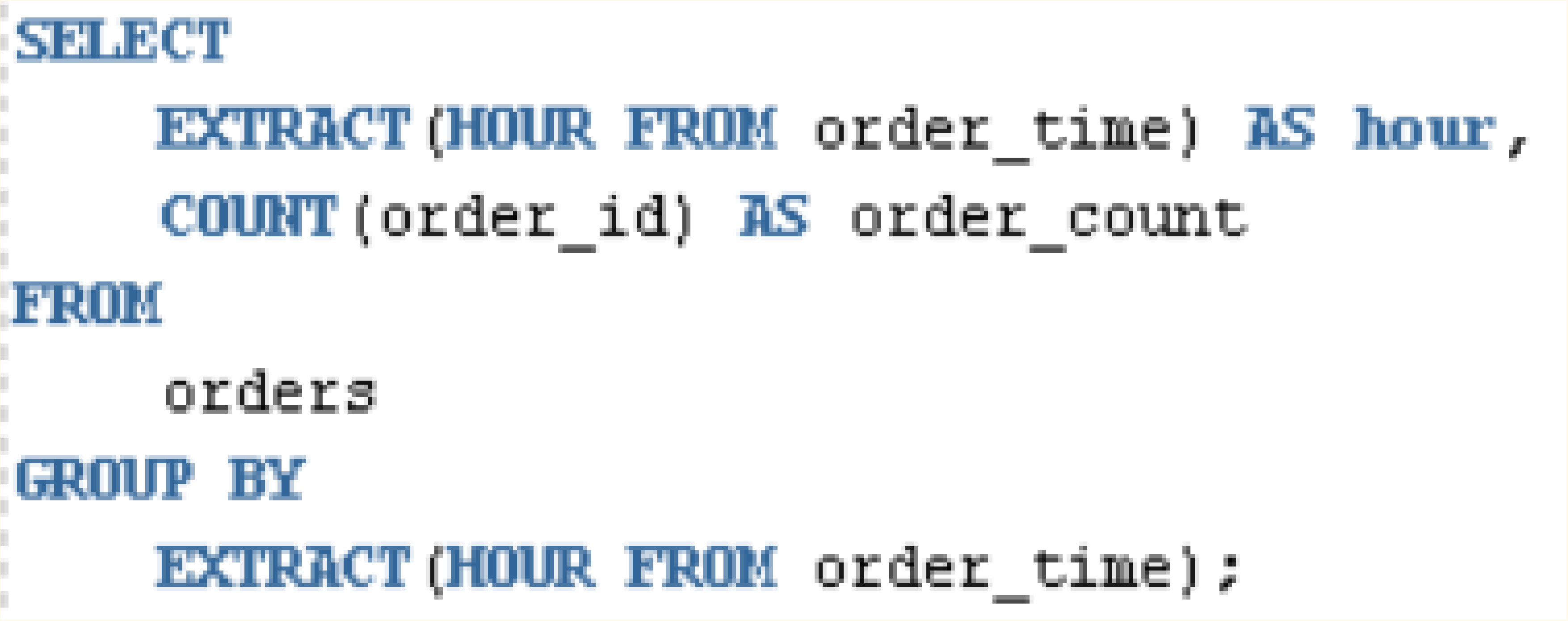
```
SELECT
    pizza_type.category,
    SUM(order_details.quantity) AS quantity
FROM
        pizza_type
    JOIN pizzas ON pizza_type.pizza_type_id = pizzas.pizza_type_id
    JOIN order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY
    pizza_type.category
ORDER BY
    quantity DESC;
```



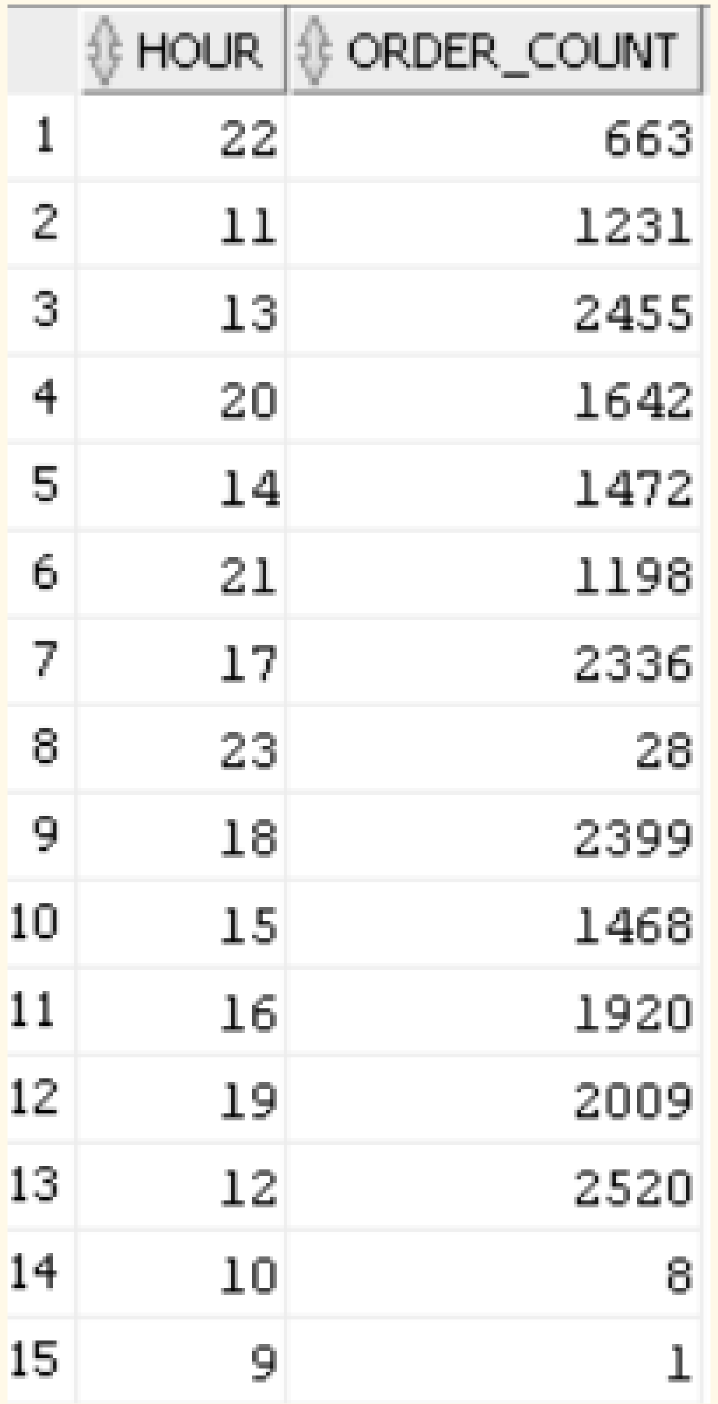
	CATEGORY	QUANTITY
1	Classic	14888
2	Supreme	11987
3	Veggie	11649
4	Chicken	11050

INTERMEDIATE QUESTIONS

Q2.Determine the distribution of orders by hour of the day.



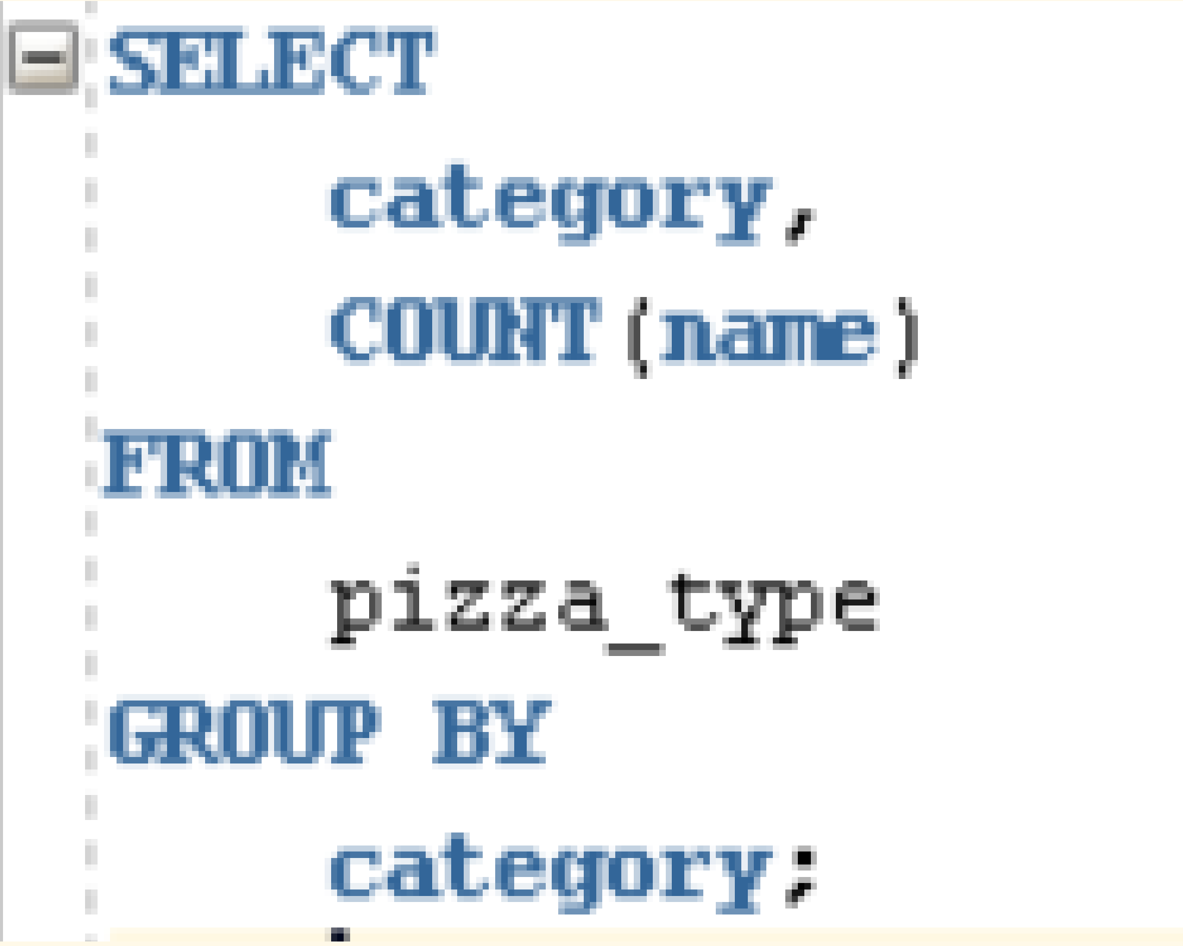
```
SELECT
    EXTRACT(HOUR FROM order_time) AS hour,
    COUNT(order_id) AS order_count
FROM
    orders
GROUP BY
    EXTRACT(HOUR FROM order_time);
```



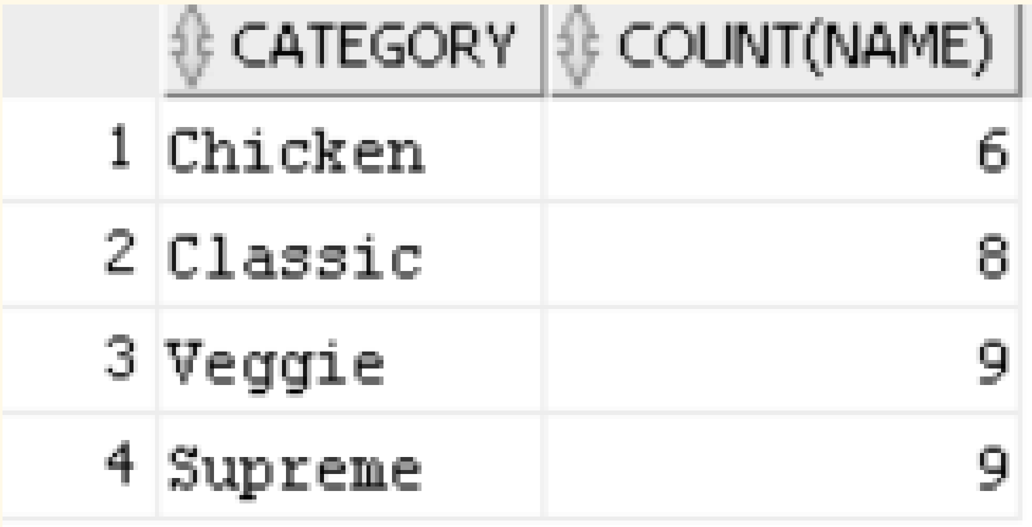
	HOUR	ORDER_COUNT
1	22	663
2	11	1231
3	13	2455
4	20	1642
5	14	1472
6	21	1198
7	17	2336
8	23	28
9	18	2399
10	15	1468
11	16	1920
12	19	2009
13	12	2520
14	10	8
15	9	1

INTERMEDIATE QUESTIONS

Q3.Join revelent tables to find the category wise distribution of pizzas.



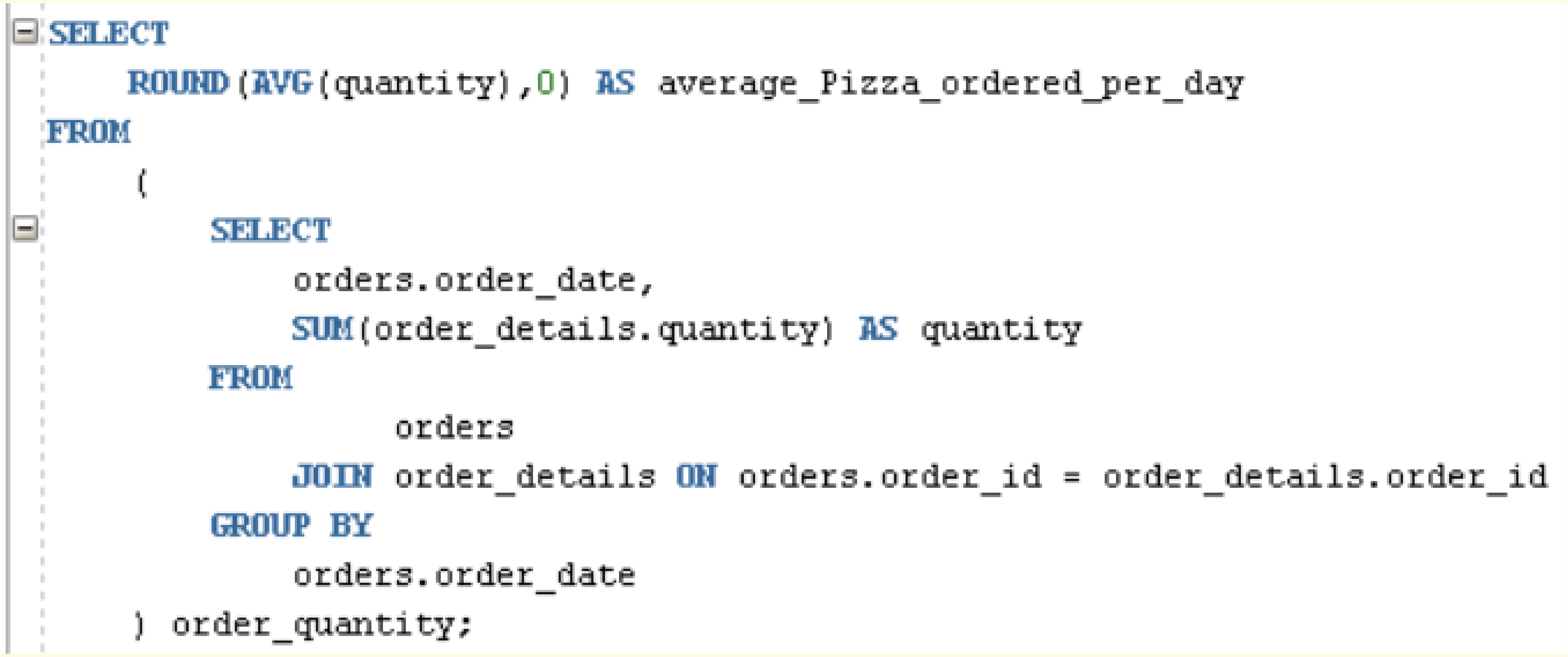
```
SELECT
    category,
    COUNT(name)
FROM
    pizza_type
GROUP BY
    category;
```



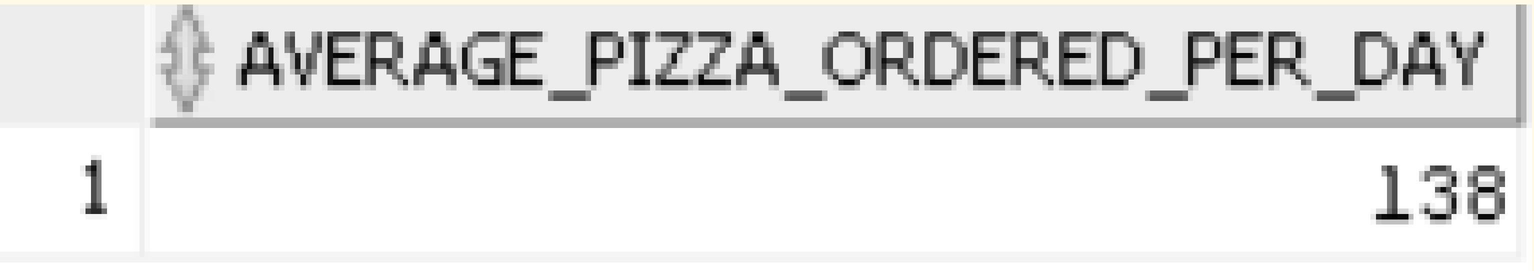
	CATEGORY	COUNT(NAME)
1	Chicken	6
2	Classic	8
3	Veggie	9
4	Supreme	9

INTERMEDIATE QUESTIONS

Q4. Group the orders by date and calculate the average number of pizza ordered per day.



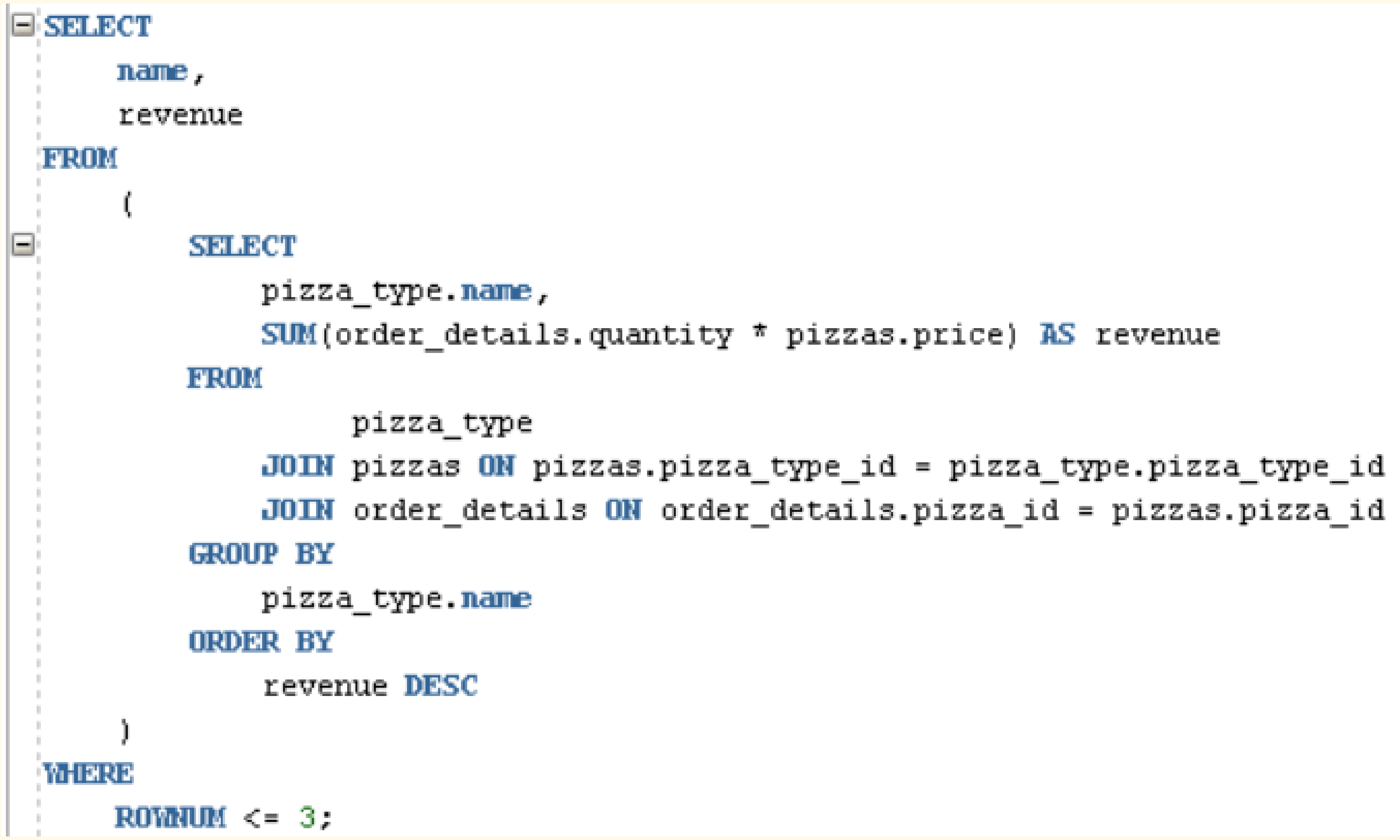
```
SELECT
    ROUND(AVG(quantity),0) AS average_Pizza_ordered_per_day
FROM
    (
        SELECT
            orders.order_date,
            SUM(order_details.quantity) AS quantity
        FROM
                orders
            JOIN order_details ON orders.order_id = order_details.order_id
        GROUP BY
            orders.order_date
    ) order_quantity;
```



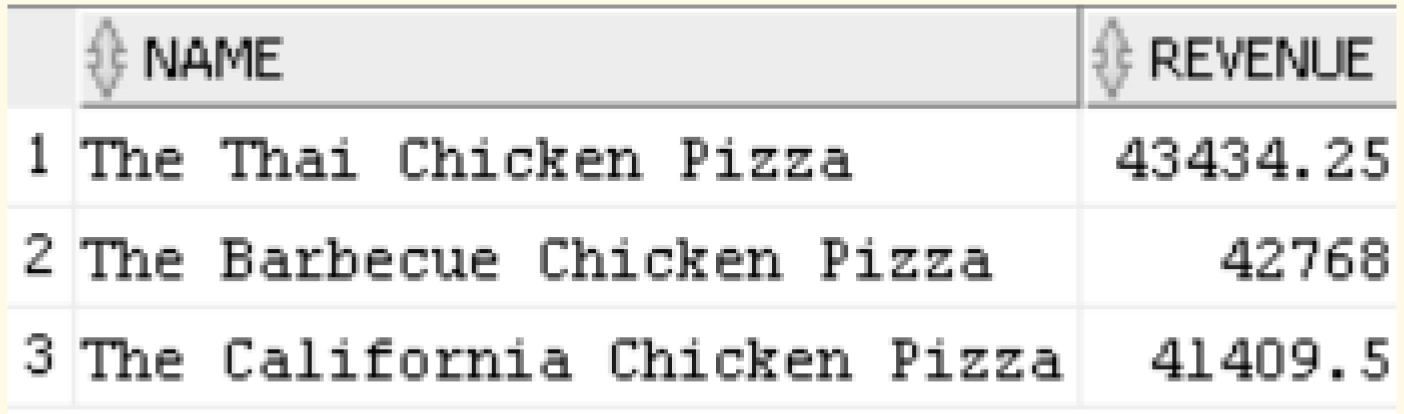
	AVERAGE_PIZZA_ORDERED_PER_DAY
1	138

INTERMEDIATE QUESTIONS

Q5. Determine top 3 most ordered pizza type based on revenue.



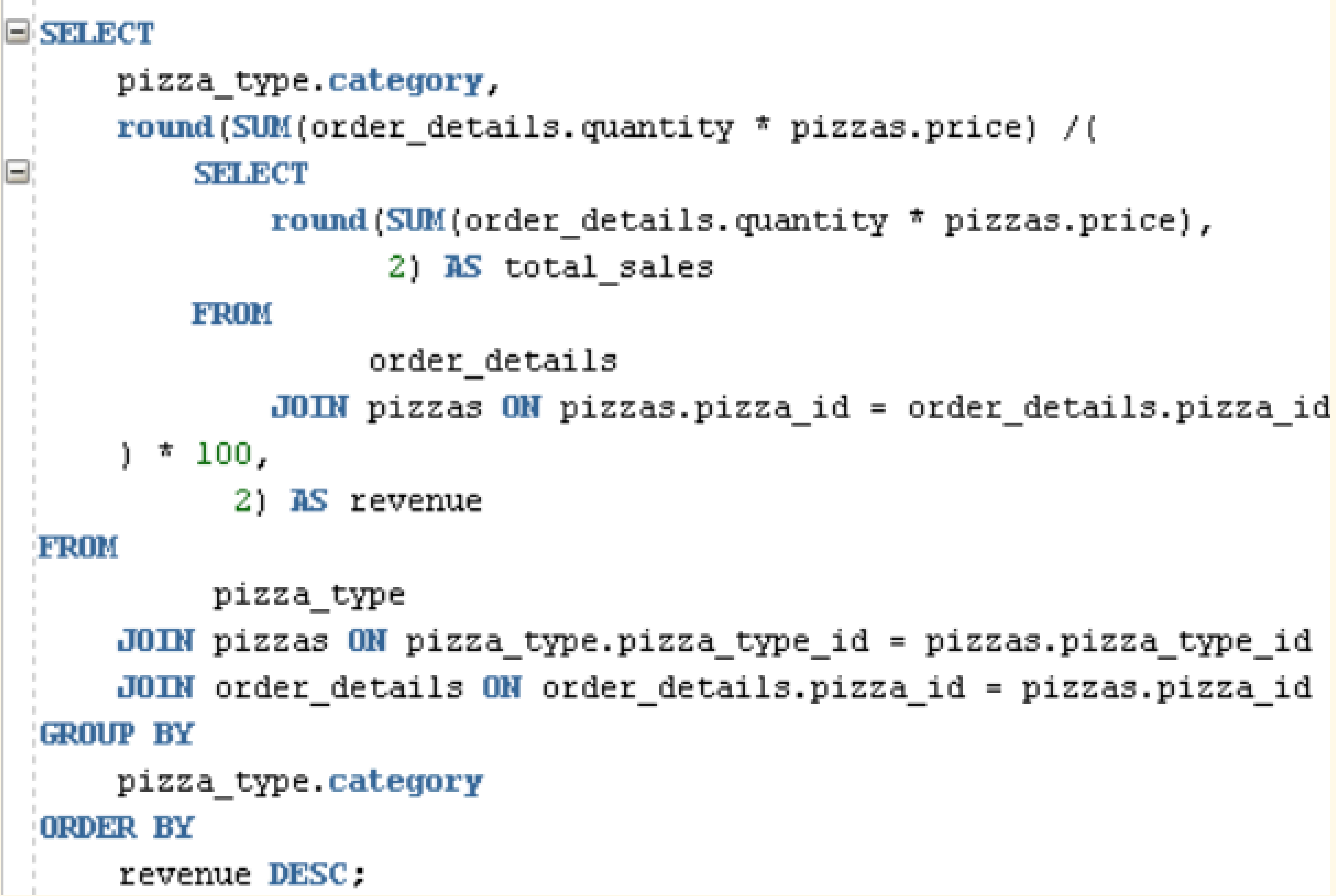
```
SELECT
    name,
    revenue
FROM
    (
        SELECT
            pizza_type.name,
            SUM(order_details.quantity * pizzas.price) AS revenue
        FROM
                pizza_type
            JOIN pizzas ON pizzas.pizza_type_id = pizza_type.pizza_type_id
            JOIN order_details ON order_details.pizza_id = pizzas.pizza_id
        GROUP BY
            pizza_type.name
        ORDER BY
            revenue DESC
    )
WHERE
    ROWNUM <= 3;
```



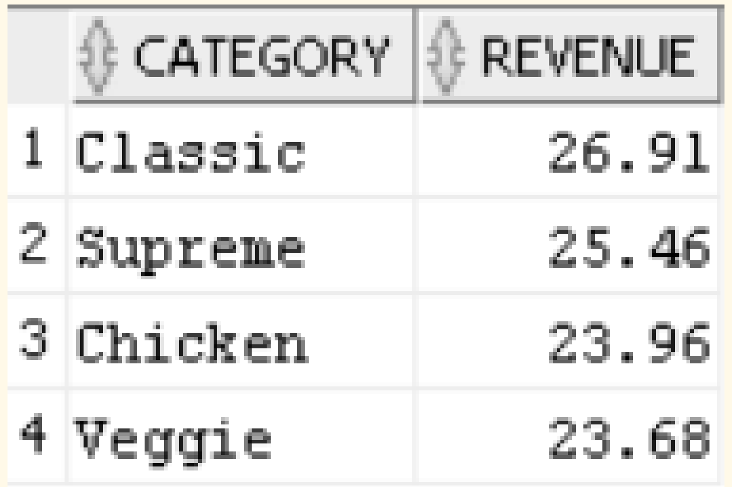
	NAME	REVENUE
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768
3	The California Chicken Pizza	41409.5

ADVANCED QUESTIONS

Q1.Calculate the percentage contribution of each pizza type to total revenue.



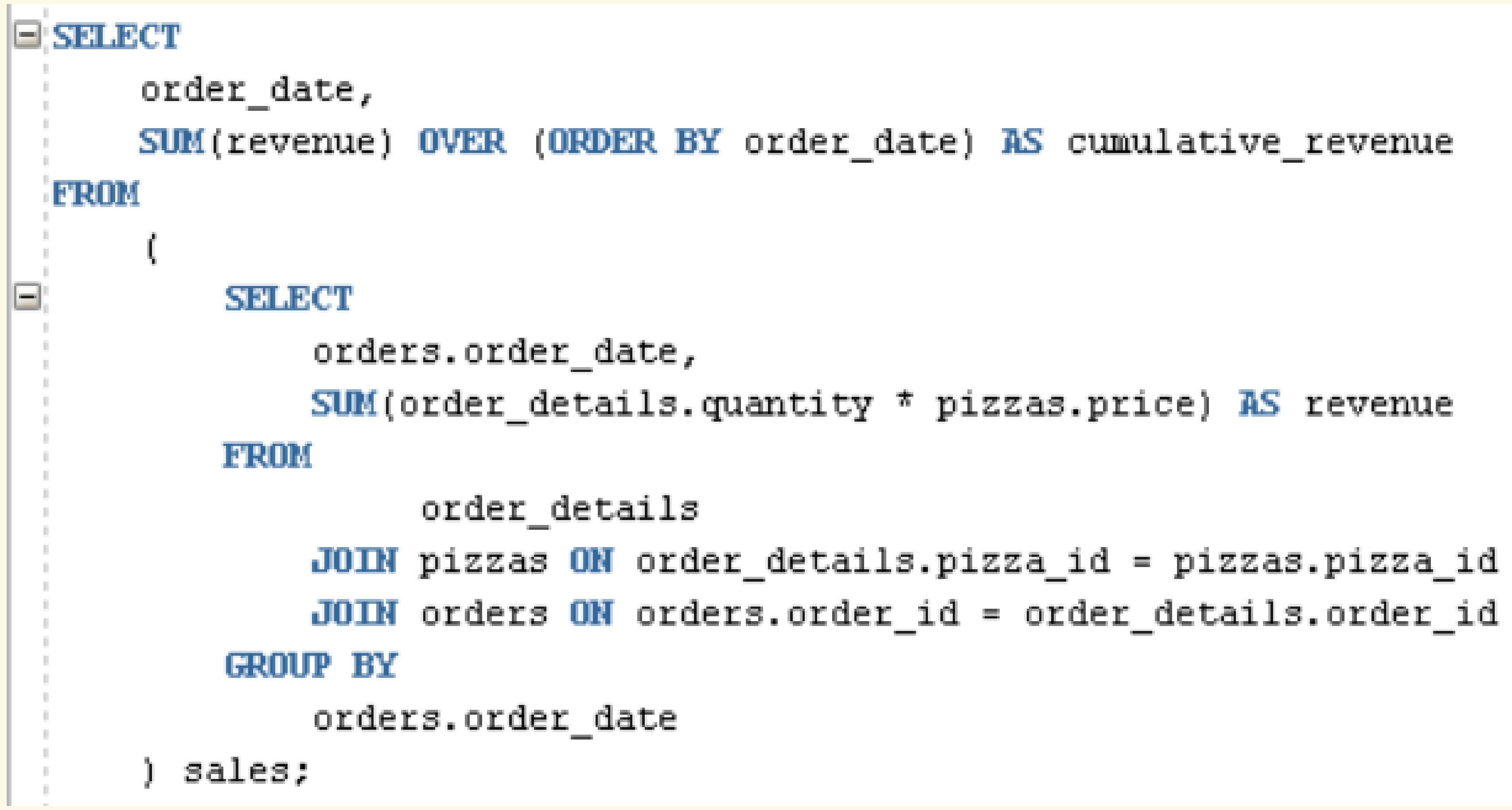
```
SELECT
    pizza_type.category,
    round(SUM(order_details.quantity * pizzas.price) /(
        SELECT
            round(SUM(order_details.quantity * pizzas.price),
                  2) AS total_sales
        FROM
                 order_details
            JOIN pizzas ON pizzas.pizza_id = order_details.pizza_id
    ) * 100,
          2) AS revenue
FROM
         pizza_type
    JOIN pizzas ON pizza_type.pizza_type_id = pizzas.pizza_type_id
    JOIN order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY
    pizza_type.category
ORDER BY
    revenue DESC;
```



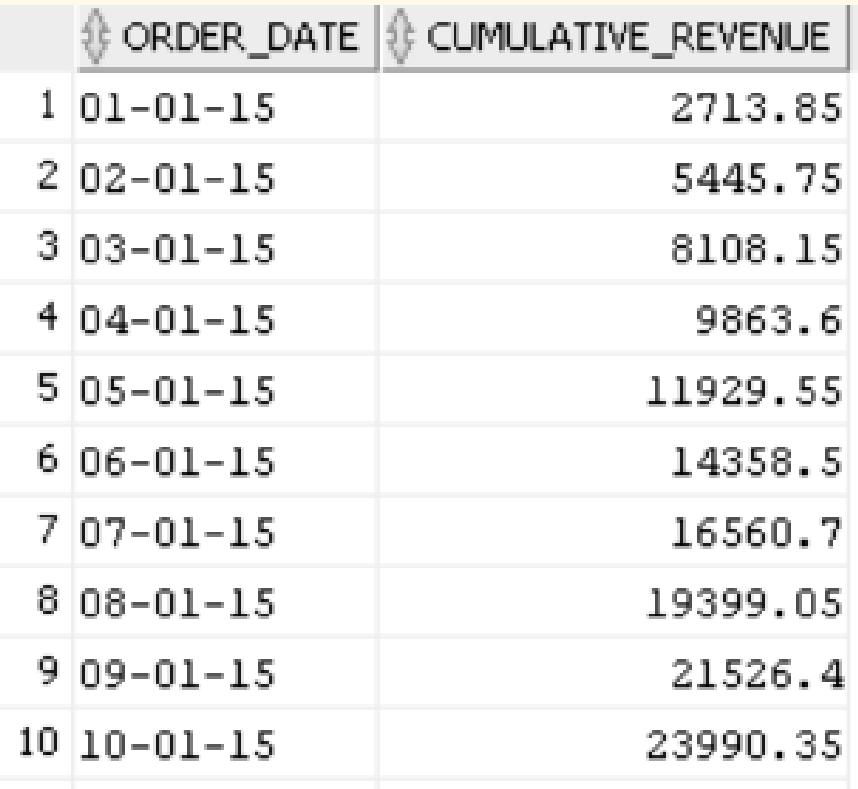
	CATEGORY	REVENUE
1	Classic	26.91
2	Supreme	25.46
3	Chicken	23.96
4	Veggie	23.68

ADVANCED QUESTIONS

Q2. Analyse the cumulative revenue generated over time.



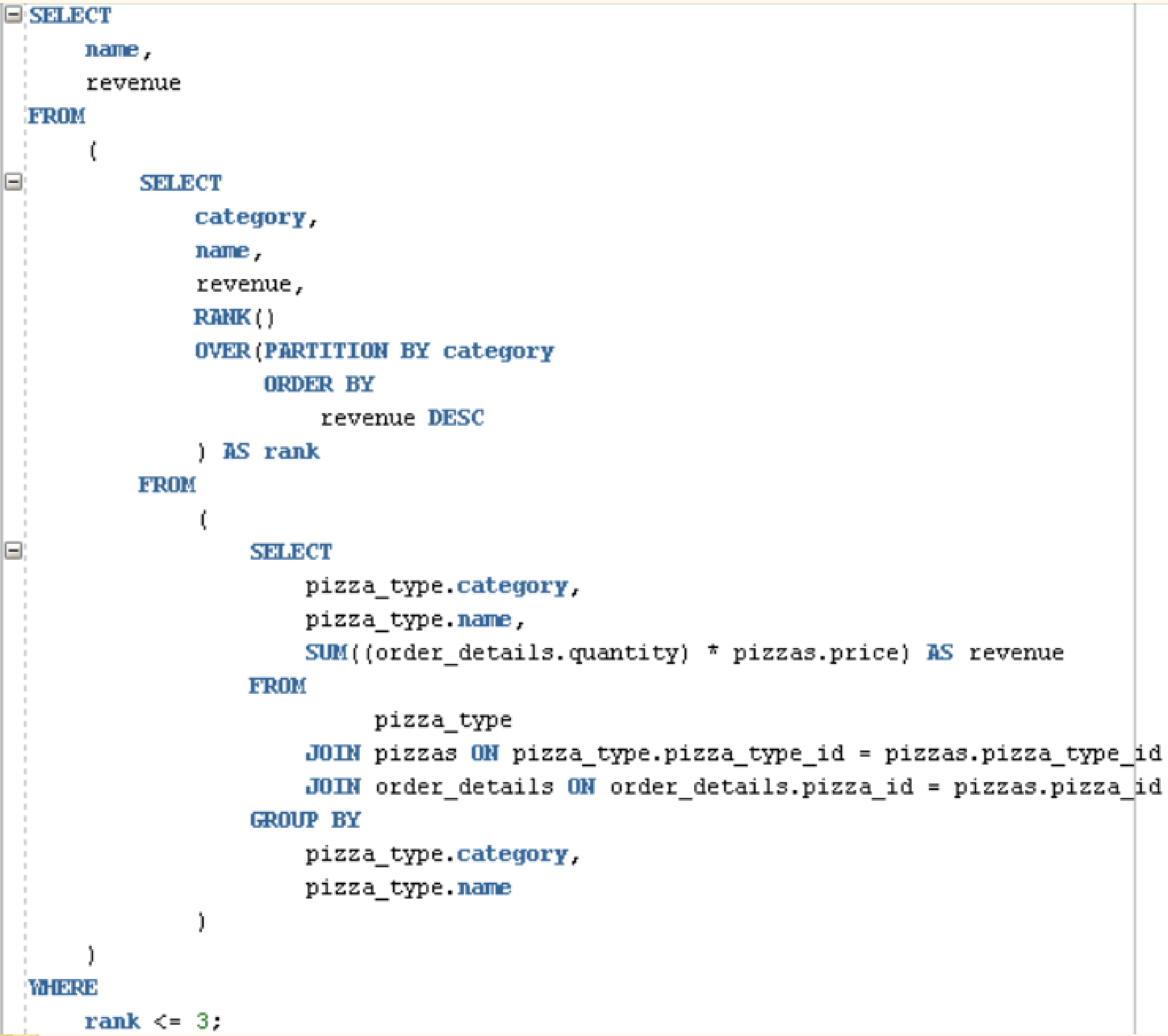
```
SELECT
    order_date,
    SUM(revenue) OVER (ORDER BY order_date) AS cumulative_revenue
FROM
    (
        SELECT
            orders.order_date,
            SUM(order_details.quantity * pizzas.price) AS revenue
        FROM
                order_details
            JOIN pizzas ON order_details.pizza_id = pizzas.pizza_id
            JOIN orders ON orders.order_id = order_details.order_id
        GROUP BY
            orders.order_date
    ) sales;
```



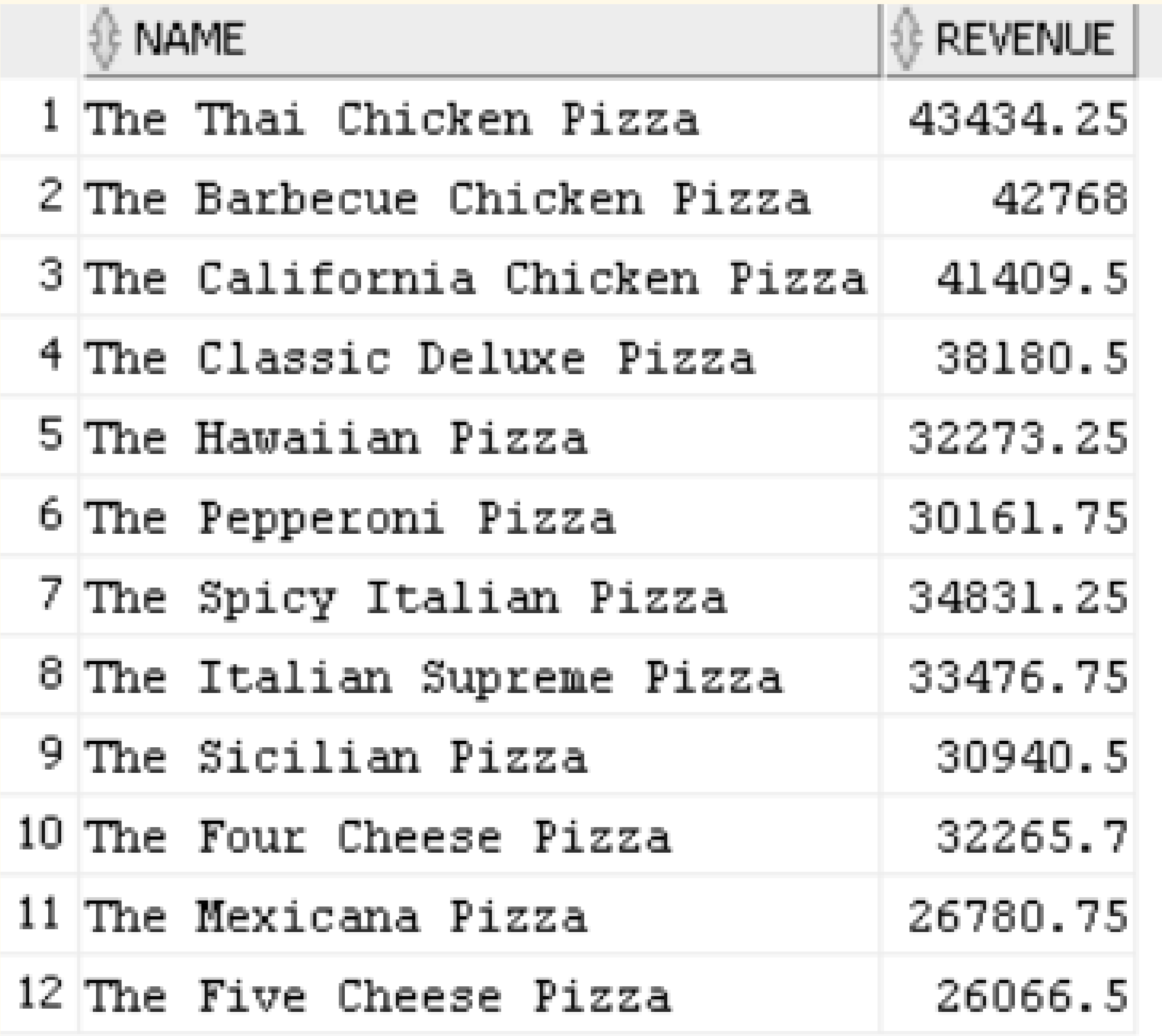
	ORDER_DATE	CUMULATIVE_REVENUE
1	01-01-15	2713.85
2	02-01-15	5445.75
3	03-01-15	8108.15
4	04-01-15	9863.6
5	05-01-15	11929.55
6	06-01-15	14358.5
7	07-01-15	16560.7
8	08-01-15	19399.05
9	09-01-15	21526.4
10	10-01-15	23990.35

ADVANCED QUESTIONS

Q3. Determine the top 3 most ordered pizza types based on revenue for each pizza category



```sql
SELECT
    name,
    revenue
FROM
    (
        SELECT
            category,
            name,
            revenue,
            RANK()
            OVER(PARTITION BY category
                ORDER BY
                    revenue DESC
            ) AS rank
        FROM
            (
                SELECT
                    pizza_type.category,
                    pizza_type.name,
                    SUM((order_details.quantity) * pizzas.price) AS revenue
                FROM
                         pizza_type
                    JOIN pizzas ON pizza_type.pizza_type_id = pizzas.pizza_type_id
                    JOIN order_details ON order_details.pizza_id = pizzas.pizza_id
                GROUP BY
                    pizza_type.category,
                    pizza_type.name
            )
    )
WHERE
    rank <= 3;
```



	NAME	REVENUE
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768
3	The California Chicken Pizza	41409.5
4	The Classic Deluxe Pizza	38180.5
5	The Hawaiian Pizza	32273.25
6	The Pepperoni Pizza	30161.75
7	The Spicy Italian Pizza	34831.25
8	The Italian Supreme Pizza	33476.75
9	The Sicilian Pizza	30940.5
10	The Four Cheese Pizza	32265.7
11	The Mexicana Pizza	26780.75
12	The Five Cheese Pizza	26066.5