```sql
-- Retrieve the total number of orders placed.

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

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

total_orders
21350

Result Grid

Result 1 ×

Read Only

```sql
-- Calculate the total revenue generated from pizza sales.

SELECT
    ROUND(SUM(order_details.quantity * pizzas.price), 2) AS total_revenue
FROM
    order_details
JOIN
    pizzas
ON
    pizzas.pizza_id = order_details.pizza_id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

total_revenue
817860.05

Result Grid

Result 2

Read Only

```sql
-- Identify the highest-priced pizza.

SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
JOIN
    pizzas
ON
    pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY
    pizzas.price DESC
LIMIT 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

name	price
The Greek Pizza	35.95

Result Grid

Result 3

Read Only

```sql
-- Identify the most common pizza size ordered.

SELECT
    pizzas.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.size
ORDER BY
    order_count DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

size	order_count
L	18526
M	15385
S	14137
XL	544
XXL	28

Result 4 | Read Only

```sql
-- List the top 5 most ordered pizza types along with their quantities.

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

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

name	quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

Result Grid

Result 5 | Read Only

```sql
-- Join the necessary tables to find the total quantity of each pizza category ordered.

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

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Result Grid

Result 6 ×

Read Only

```sql
-- Determine the distribution of orders by hour of the day.

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

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

HOUR	order_count
11	1231
12	2520
13	2455
14	1472
15	1468

Result 7 | Result Grid | Read Only

```sql
-- Join relevant tables to find the category-wise distribution of pizzas.

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

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

category	COUNT(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

Result 8 | Read Only

```sql
-- Group the orders by date and calculate the average number of pizzas ordered per day.

SELECT
    round(AVG(quantity), 0) AS avg_pizzas_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) AS order_quantity;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

avg_pizzas_ordered_per_day
138

Result 9

Read Only

```sql
-- Determine the top 3 most ordered pizza types based on revenue.

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

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

Result 10 ×

Read Only

```sql
-- Calculate the percentage contribution of each pizza type to total revenue.

SELECT
    pizza_types.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_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY revenue DESC;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Result 11 | Read Only

```sql
-- Analyze the cumulative revenue generated over time.

SELECT
    order_date,
    SUM(revenue) OVER(ORDER BY order_date) AS cum_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) AS sales;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

order_date	cum_revenue
2015-01-01	2713.8500000000004
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55

Result 12 | Result Grid | Read Only

```sql
-- Determine the top 3 most ordered pizza types based on revenue for each pizza category.
SELECT
    name, revenue
FROM
    (SELECT
        category, name, revenue,
        RANK() OVER(PARTITION BY category ORDER BY revenue DESC) AS rn
    FROM
        (SELECT
            pizza_types.category, pizza_types.name,
            SUM((order_details.quantity) * pizzas.price) AS revenue
        FROM
            pizza_types
        JOIN
            pizzas
        ON
            pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
            order_details
        ON
            order_details.pizza_id = pizzas.pizza_id
        GROUP BY
            pizza_types.category, pizza_types.name) AS a) AS b
WHERE rn <= 3;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5
The Classic Deluxe Pizza	38180.5
The Hawaiian Pizza	32273.25

Result 13 | Result Grid | Read Only