

Ques 1:- Retrieve the total number of orders placed ?

- ```
select count(order_id) as total_orders
from orders
```

Ans –

| Result Grid |              |
|-------------|--------------|
|             | total_orders |
| ▶           | 21350        |

Ques 2 :- Calculate the total revenue generated from pizza sales ?

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

Ans –

Result Grid	
	total_sales
▶	817860.05

Ques 3:- 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 price DESC
LIMIT 1
```

Ans –

| Result Grid |                 |       |
|-------------|-----------------|-------|
|             | name            | price |
| ▶           | The Greek Pizza | 35.95 |

Ques 4:-Identify the most common pizza size ordered ?

- ```
SELECT
    pizzas.size,
    COUNT((order_details.pizza_id * order_details.quantity)) AS total_order
FROM
    pizzas
    JOIN
        order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY size
order by total_order desc
limit 1;
```

Ans-

Result Grid	Filter R
size	total_order
L	18526

Ques. 5:- List the top 5 most ordered pizza types along with their quantities ?

- ```
SELECT
 pizza_types.name,
 SUM(order_details.quantity) AS total_quantity
FROM
 pizzas
 JOIN
 pizza_types 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 total_quantity DESC
LIMIT 5;
```

Ans –

| Result Grid                | Filter Rows:   | Export: |
|----------------------------|----------------|---------|
| name                       | total_quantity |         |
| The Classic Deluxe Pizza   | 2453           |         |
| The Barbecue Chicken Pizza | 2432           |         |
| The Hawaiian Pizza         | 2422           |         |
| The Pepperoni Pizza        | 2418           |         |
| The Thai Chicken Pizza     | 2371           |         |

Ques. 6:- Join the necessary tables to find the total quantity of each pizza category ordered ?

- **SELECT**

```
pizza_types.category |,
SUM(order_details.quantity) AS total_quantity
FROM
 pizzas
 JOIN
 pizza_types 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.category
ORDER BY total_quantity DESC
```

Ans -

| Result Grid | Filter Rows:   |
|-------------|----------------|
| category    | total_quantity |
| Classic     | 14888          |
| Supreme     | 11987          |
| Veggie      | 11649          |
| Chicken     | 11050          |

Ques. 7:- 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
```

Ans-

| Result Grid | Filter      |
|-------------|-------------|
| hour        | order_count |
| 11          | 1231        |
| 12          | 2520        |
| 13          | 2455        |
| 14          | 1472        |
| 15          | 1468        |
| 16          | 1920        |
| 17          | 2336        |
| 18          | 2399        |
| 19          | 2009        |
| 20          | 1642        |
| 21          | 1198        |
| 22          | 663         |
| 23          | 28          |
| 10          | 8           |
| 9           | 1           |

Ques. 8:- Join relevant tables to find the category-wise distribution of pizzas ?

- **SELECT**  
    category, COUNT(name)  
**FROM**  
    pizza\_types  
**GROUP BY** category;

Ans -

|   | category | count(name) |
|---|----------|-------------|
| ▶ | Chicken  | 6           |
|   | Classic  | 8           |
|   | Supreme  | 9           |
|   | Veggie   | 9           |

Ques. 9:- Group the orders by date and calculate the average number of pizzas ordered per day ?

- **SELECT**  
    ROUND(AVG(total\_orders), 0) AS avg\_daily\_orders  
**FROM**  
    (SELECT  
        (orders.order\_date) AS date,  
        SUM(order\_details.quantity) AS total\_orders  
**FROM**  
        orders  
**JOIN** order\_details **ON** order\_details.order\_id = orders.order\_id  
**GROUP BY** date) AS date\_wise

Ans -

|   | avg_daily_orders |
|---|------------------|
| ▶ | 138              |

Ques. 10:- Determine the top 3 most ordered pizza types based on revenue ?

```

• SELECT
 name , round(sum(order_details.quantity * pizzas.price),2) as revenue
FROM
 pizza_types
 JOIN
 pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
 JOIN
 order_details ON pizzas.pizza_id = order_details.pizza_id
group by name
order by revenue desc
limit 3

```

Ans -

| Result Grid |                              |          |
|-------------|------------------------------|----------|
|             | name                         | revenue  |
| ▶           | The Thai Chicken Pizza       | 43434.25 |
|             | The Barbecue Chicken Pizza   | 42768    |
|             | The California Chicken Pizza | 41409.5  |

Ques. 11:- Calculate the percentage contribution of each pizza type to total revenue ?

```

• SELECT name,
 round((SUM(order_details.quantity * pizzas.price) /
 (SELECT SUM(order_details.quantity * pizzas.price) AS total_sales
 FROM order_details
 JOIN pizzas ON order_details.pizza_id = pizzas.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 pizzas.pizza_id = order_details.pizza_id
GROUP BY name
ORDER BY revenue desc
limit 7

```

Ans -

| Result Grid |                              |         |
|-------------|------------------------------|---------|
|             | name                         | revenue |
| ▶           | The Thai Chicken Pizza       | 5.31    |
|             | The Barbecue Chicken Pizza   | 5.23    |
|             | The California Chicken Pizza | 5.06    |
|             | The Classic Deluxe Pizza     | 4.67    |
|             | The Spicy Italian Pizza      | 4.26    |
|             | The Southwest Chicken Pizza  | 4.24    |
|             | The Italian Supreme Pizza    | 4.09    |

Ques. 12:- 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 orders join order_details
on orders.order_id = order_details.order_id
join pizzas
on pizzas.pizza_id = order_details.pizza_id
group by orders.order_date) as sales

```

Ans -

| 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           |
| 2015-01-06 | 14358.5            |
| 2015-01-07 | 16560.7            |
| 2015-01-08 | 19399.05           |
| 2015-01-09 | 21526.4            |
| 2015-01-10 | 23990.350000000002 |
| 2015-01-11 | 25862.65           |
| 2015-01-12 | 27781.7            |
| 2015-01-13 | 29831.300000000003 |
| 2015-01-14 | 32358.700000000004 |
| 2015-01-15 | 34343.50000000001  |

Ques. 13:- Calculate the percentage contribution of each pizza category to total revenue ?

```

SELECT category,
round((SUM(order_details.quantity * pizzas.price) /
(SELECT SUM(order_details.quantity * pizzas.price) AS total_sales
FROM order_details
JOIN pizzas ON order_details.pizza_id = pizzas.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 pizzas.pizza_id = order_details.pizza_id
GROUP BY category
ORDER BY revenue desc

```

Ans -

| category | revenue |
|----------|---------|
| Classic  | 26.91   |
| Supreme  | 25.46   |
| Chicken  | 23.96   |
| Veggie   | 23.68   |

Ques. 14:-Determine the top 3 most ordered pizza types based on revenue for each pizza category ?

```

select category, name, revenue from
(select category , name, revenue,
rank() over (partition by category order by revenue desc) as rnk 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 pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.category , pizza_types.name) as a) as b
where rnk <= 3;

```

Ans -

| category | name                         | revenue           |
|----------|------------------------------|-------------------|
| Chicken  | The Thai Chicken Pizza       | 43434.25          |
| Chicken  | The Barbecue Chicken Pizza   | 42768             |
| Chicken  | The California Chicken Pizza | 41409.5           |
| Classic  | The Classic Deluxe Pizza     | 38180.5           |
| Classic  | The Hawaiian Pizza           | 32273.25          |
| Classic  | The Pepperoni Pizza          | 30161.75          |
| Supreme  | The Spicy Italian Pizza      | 34831.25          |
| Supreme  | The Italian Supreme Pizza    | 33476.75          |
| Supreme  | The Sicilian Pizza           | 30940.5           |
| Veggie   | The Four Cheese Pizza        | 32265.70000000065 |
| Veggie   | The Mexicana Pizza           | 26780.75          |
| Veggie   | The Five Cheese Pizza        | 26066.5           |

## CONCLUSION

- Classic pizzas are the biggest contributor in revenue generation.
- The classic dulex pizza is the best seller but Thai chicken Pizza is biggest individual contributor to revenue generation
- In morning (12:00 – 14:00) are the busiest hours and in evening (16:00 – 19:00) are the busiest hours with maximum pizza sales
- Sale total to average daily sale of 138 pizzas
- ( L ) is the most orders size among all pizza sizes.