





# PIZZA SALES ANALYSIS

CREATED BY DILIP KUMAR



Q1. Retrieve the total number of orders placed.

```
1  -- SQL Basic Question --
2
3  -- 1) Retrieve the total number of orders placed.
4
5 • select count(order_id) as total_orders from orders;
```

<	
Result Grid   Filter Rows: <input type="text"/>   Export:    Wrap Cell Content: 	
	total_orders
▶	21350

## Q2. Calculate the total revenue generated from pizza sales.

```
1  -- SQL Basic Question --
2
3  -- 2) Calculate the total revenue generated from pizza sales.
4
5  • SELECT
6      ROUND(SUM(pizza_world.order_details.quantity * pizza_world.pizzas.price),2) AS total_sales
7  FROM
8      order_details
9      JOIN
10     pizzas ON pizzas.pizza_id = order_details.pizza_id;
```

<	
Result Grid	Filter Rows: <input type="text"/> Export:  Wrap Cell Content:
	total_sales
▶	817860.05



### Q3. Identify the highest-priced pizza.

```
1  -- SQL Basic Question --
2  -- 3) Identify the highest-priced pizza.
3
4  • SELECT
5      pizza_types.name, pizzas.price
6  FROM
7      pizza_types
8      JOIN
9      pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
10 ORDER BY pizzas.price DESC
11 LIMIT 1;
```

<		
Result Grid		
Filter Rows: <input type="text"/>		
Export: 		
Wrap Cell Content: 		
Fetch rows: 		
	name	price
▶	The Greek Pizza	35.95

## Q4. Identify the most common pizza size ordered.

```
1  -- SQL Basic Question --
2  -- 4) Identify the most common pizza size ordered.
3
4  • SELECT
5      pizzas.size,
6      COUNT(order_details.order_details_id) AS order_count
7  FROM
8      pizzas
9      JOIN
10     order_details ON pizzas.pizza_id = order_details.pizza_id
11 GROUP BY pizzas.size
12 ORDER BY order_count DESC;
```

<		
Result Grid		
Filter Rows: <input type="text"/>		
Export:  Wrap Cell Content: 		
	size	order_count
▶	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28

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

```
1  -- SQL Basic Question --
2  -- List the top 5 most ordered pizza types along with their quantities.
3  * SELECT
4      pizza_types.name, SUM(order_details.quantity) AS quantity
5  FROM
6      pizza_types
7      JOIN
8      pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
9      JOIN
10     order_details ON order_details.pizza_id = pizzas.pizza_id
11 GROUP BY pizza_types.name
12 ORDER BY quantity DESC
13 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



Q6. Join the necessary tables to find the total quantity of each pizza category ordered.

```
1  -- SQL Intermediate Question --
2  -- 6)Join the necessary tables to find the total quantity of each pizza category ordered.
3  • SELECT
4      pizza_types.category,
5      SUM(order_details.quantity) AS quantity
6  FROM
7      pizza_types
8      JOIN
9      pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
10     JOIN
11     order_details ON order_details.pizza_id = pizzas.pizza_id
12 GROUP BY pizza_types.category
13 ORDER BY quantity DESC;
```

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

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

## Q7. Determine the distribution of orders by hour of the day.



```
1  -- SQL Intermediate Question --
2  -- 7)Determine the distribution of orders by hour of the day.
3  • SELECT
4      HOUR(time) AS hours, COUNT(order_id) AS order_count
5  FROM
6      orders
7  GROUP BY hours order by hours;
8
```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	hours	order_count			
▶	9	1			
	10	8			
	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			



## Q8. Join relevant tables to find the category-wise distribution of pizzas.

```
1  -- SQL Intermediate Question --
2  -- 8)Join relevant tables to find the category-wise distribution of pizzas.
3  * SELECT
4      category, COUNT(name) AS pizza_category_count
5  FROM
6      pizza_types
7  GROUP BY category;
```

<		
Result Grid		
Filter Rows: <input type="text"/>		
Export: 		
Wrap Cell Content: 		
	category	pizza_category_count
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

Q9. Group the orders by date and calculate the average number of pizzas ordered per day.

```
1  -- SQL Intermediate Question --
2  -- 9)Group the orders by date and calculate the average number of pizzas ordered per day.
3  • SELECT
4      ROUND(AVG(quantity), 0) AS avg_order_pizza_per_day
5  FROM
6      (SELECT
7          orders.date AS date, SUM(order_details.quantity) AS quantity
8      FROM
9          orders
10         JOIN order_details ON orders.order_id = order_details.order_id
11         GROUP BY date) AS order_qt;
12
```



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

	avg_order_pizza_per_day
--	-------------------------

▶	138
---	-----

Q10. Determine the top 3 most ordered pizza types based on revenue.

```
1  -- SQL Intermediate Question --
2  -- 10) Determine the top 3 most ordered pizza types based on revenue.
3  • SELECT
4      pizza_types.name,
5      ROUND(SUM(order_details.quantity * pizzas.price),2) AS revenue
6  FROM
7      pizza_types
8      JOIN
9      pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
10     JOIN
11     order_details ON order_details.pizza_id = pizzas.pizza_id
12 GROUP BY pizza_types.name
13 ORDER BY revenue DESC
14 LIMIT 3;
```

<	Result Grid		 Filter Rows: <input type="text"/>	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				



## Q11. Calculate the percentage contribution of each pizza type to total revenue.

```
1  -- SQL Advanced Question --
2  -- 11) Calculate the percentage contribution of each pizza type to total revenue.
3  • SELECT
4      pizza_types.category,
5      ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
6          ROUND(SUM(order_details.quantity * pizzas.price),
7              2) AS total_sales
8          FROM
9              order_details
10             JOIN
11                 pizzas ON order_details.pizza_id = pizzas.pizza_id) * 100,
12          2) AS revenue
13  FROM
14      pizza_types
15      JOIN
16      pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
17      JOIN
18      order_details ON order_details.pizza_id = pizzas.pizza_id
19  GROUP BY pizza_types.category
20  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

## Q12. Analyze the cumulative revenue generated over the time.

```
1  -- SQL Advanced Question --
2  -- 12)Analyze the cumulative revenue generated over the time.
3  • select dates , round(sum(revenue) over(order by dates),2) as cum_revenue from
4  • (select orders.date as dates , sum(order_details.quantity * pizzas.price) as revenue from order_details join orders
5  | on order_details.order_id = orders.order_id join pizzas on pizzas.pizza_id = order_details.pizza_id |
6  group by orders.date ) as sales;
```

	dates	cum_revenue
▶	2015-01-01	2713.85
	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	23990.35
	2015-01-11	25862.65
	2015-01-12	27781.7



## Q13. Determine the top 3 most ordered pizza types based on revenue for each pizza category.

```
1  -- SQL Advanced Question --
2  -- 13)Determine the top 3 most ordered pizza types based on revenue for each pizza category.
3  * select pizza_category, pizza_name, revenue, ra_nk
4  from
5  (select pizza_category, pizza_name, revenue , rank()
6   over(partition by pizza_category order by revenue desc) as ra_nk
7   from
8   (select pizza_types.category as pizza_category, pizza_types.name as pizza_name, round(sum(order_details.quantity * pizzas.price),2) as revenue from
9    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
10   group by pizza_types.category, pizza_types.name order by pizza_types.category ) as sales) as another_sales where ra_nk <= 3;
```

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

	pizza_category	pizza_name	revenue	ra_nk
▶	Chicken	The Thai Chicken Pizza	43434.25	1
	Chicken	The Barbecue Chicken Pizza	42768	2
	Chicken	The California Chicken Pizza	41409.5	3
	Classic	The Classic Deluxe Pizza	38180.5	1
	Classic	The Hawaiian Pizza	32273.25	2
	Classic	The Pepperoni Pizza	30161.75	3
	Supreme	The Spicy Italian Pizza	34831.25	1
	Supreme	The Italian Supreme Pizza	33476.75	2
	Supreme	The Sicilian Pizza	30940.5	3
	Veggie	The Four Cheese Pizza	32265.7	1
	Veggie	The Mexicana Pizza	26780.75	2
	Veggie	The Five Cheese Pizza	26066.5	3



THANK YOU FOR WATCHING

