

# Pizza Sales Project MySQL

## Questions solved in this project with MySQL

### Basic:

1. Retrieve the total number of orders placed.
2. Calculate the total revenue generated from pizza sales.
3. Identify the highest-priced pizza.
4. Identify the most common pizza size ordered.
5. List the top 5 most ordered pizza types along with their quantities.

### Intermediate:



1. Join the necessary tables to find the total quantity of each pizza category ordered.
2. Determine the distribution of orders by hour of the day.
3. Join relevant tables to find the category-wise distribution of pizzas.
4. Group the orders by date and calculate the average number of pizzas ordered per day.
5. Determine the top 3 most ordered pizza types based on revenue.

### Advanced:

1. Calculate the percentage contribution of each pizza type to total revenue.
2. Analyse the cumulative revenue generated over time.
3. Determine the top 3 most ordered pizza types based on revenue for each pizza category.



Q1) Retrieve the total number of orders placed.

```
1  -- Retrieve the total number of orders placed.
2
3 • SELECT
4      COUNT(order_id) AS total_orders
5  FROM
6      orders;
```

Result Grid			
	total_orders		
▶	21350		

Q2) Calculate the total revenue generated from pizza sales.

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

Result Grid			
	total_sales		
▶	817860.05		

Q3) Identify the highest-priced pizza.

```

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

```



Result Grid     Filter Rows:		
	name	price
▶	The Greek Pizza	35.95

Q4) Identify the most common pizza size ordered.

```

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

```

Result Grid     Filter Rows:		
	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  -- List the top 5 most ordered pizza types
2  -- along with their quantities.
3
4  • SELECT
5      pizza_types.name, 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.name
13 ORDER BY quantity DESC
14 LIMIT 5;

```



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

```

Result Grid   Filter Rows:		
	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

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

```

1  -- Determine the distribution of orders by hour of the day.
2
3 • SELECT
4     HOUR(order_time) AS hour, COUNT(order_id) AS order_count
5 FROM
6     orders
7 GROUP BY HOUR(order_time);

```

Result Grid			Filter Rows:
	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	

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

```

1  -- join relevant tables to find the category-wise distribution of pizzas
2
3 • select category, count(name) from pizza_types
4  group by category

```

Result Grid			Filter Rows:
	category	count(name)	
▶	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  -- Group the orders by date and
2  -- calculate the average number of pizzas ordered per day.
3
4 • SELECT
5     ROUND(AVG(quantity), 0) AS average_pizzas_ordered_per_day
6  FROM
7     (SELECT
8         orders.order_date, SUM(order_details.quantity) AS quantity
9     FROM
10        orders
11     JOIN order_details ON orders.order_id = order_details.order_id
12     GROUP BY orders.order_date) AS order_quantity;

```

Result Grid		Filter Rows:
	average_pizzas_ordered_per_day	
▶	138	

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

```

1  -- Determine the top 3 most ordered pizza types based on revenue.
2
3 • SELECT
4      pizza_types.name,
5      SUM(order_details.quantity * pizzas.price) 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 pizzas.pizza_id = order_details.pizza_id
12 GROUP BY pizza_types.name
13 ORDER BY revenue DESC
14 LIMIT 3;

```

Result Grid			Filter 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  -- Calculate the percentage contribution of each pizza type to total revenue.
2
3 • SELECT
4      pizza_types.category,
5      (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 pizzas.pizza_id = order_details.pizza_id)) * 100 AS revenue
12 FROM
13     pizza_types
14     JOIN
15     pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
16     JOIN
17     order_details ON order_details.pizza_id = pizzas.pizza_id
18 GROUP BY pizza_types.category
19 ORDER BY revenue DESC;

```

Result Grid			Filter Rows:
	category	revenue	
▶	Classic	26.90596025566967	
	Supreme	25.45631126009862	
	Chicken	23.955137556847287	
	Veggie	23.682590927384577	

Q12) Analyse the cumulative revenue generated over time.

```

1  -- Analyze the cumulative revenue generated over time.
2
3 • select order_date,
4      sum(revenue) over (order by order_date) as cum_revenue
5  from
6  (select orders.order_date ,
7      sum(order_details.quantity * pizzas.price) as revenue
8  from order_details join pizzas
9  on order_details.pizza_id = pizzas.pizza_id
10 join orders
11 on orders.order_id = order_details.order_id
12 group by orders.order_date) as sales;

```

Result Grid			Filter Rows:
	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	34343.500000000001	
	2015-01-16	36937.650000000001	
	2015-01-17	39001.750000000001	
	2015-01-18	40978.600000000006	
	2015-01-19	43365.750000000001	
	2015-01-20	45763.650000000001	
	2015-01-21	47804.200000000001	
	2015-01-22	50300.900000000001	
	2015-01-23	52724.600000000006	
	2015-01-24	55013.850000000006	

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

```
1  -- Determine the top 3 most ordered pizza types
2  -- based on revenue for each pizza category.
3
4 • select name, revenue from
5  (select category, name, revenue, rank() over(partition by category order by revenue desc) as rn
6   from
7   (select pizza_types.category, pizza_types.name,
8    sum((order_details.quantity) * pizzas.price) as revenue
9    from pizza_types join pizzas
10   on pizza_types.pizza_type_id = pizzas.pizza_type_id
11   join order_details
12   on order_details.pizza_id = pizzas.pizza_id
13   group by pizza_types.category, pizza_types.name) as a) as b
14  where rn<=3;
```

Result Grid			Filter Rows:
	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	
	The Pepperoni Pizza	30161.75	
	The Spicy Italian Pizza	34831.25	
	The Italian Supreme Pizza	33476.75	
	The Sicilian Pizza	30940.5	
	The Four Cheese Pizza	32265.70000000065	
	The Mexicana Pizza	26780.75	
	The Five Cheese Pizza	26066.5	

Thank you!