

PIZZA SALES REPORT--SQL

BY__KEDAR_KUMBHAR





LARANA PIZZA

WELCOME TO

PIZZA SALES REPORT--SQL

The Pizza Sales Report provides a comprehensive analysis of sales trends, customer preferences, and business performance. Using MySQL, the report leverages structured query language to extract, analyze, and present data from the pizza shop's database.



CREATE DATABASE & TABLES

DATABASE QUERY

```
3 • create database pizza_sales;  
4  
5 • ⏹ CREATE TABLE orders (  
6     order_id INT NOT NULL,  
7     order_date DATE NOT NULL,  
8     order_time TIME NOT NULL,  
9     PRIMARY KEY (order_id)  
10    );  
11   • ⏹ CREATE TABLE order_details (  
12     order_details_id INT NOT NULL,  
13     order_id INT NOT NULL,  
14     pizza_id TEXT NOT NULL,  
15     quantity INT NOT NULL,  
16     PRIMARY KEY (order_details_id)  
17    );
```



RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

```
3 •   SELECT  
4       COUNT(order_id) AS Total_orders  
5   FROM  
6       orders;
```

Result Grid

	Total_orders
▶	21350

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES



```
3 •   SELECT
4   ⚑     ROUND(SUM(order_details.quantity * pizzas.price),
5           2) AS total_sales
6
7   FROM
8
9   order_details
10
11   JOIN
12
13   pizzas ON order_details.pizza_id = pizzas.pizza_id;
```

Result Grid	
	total_revenue
▶	12852.55



IDENTIFY THE HIGHEST-PRICED PIZZA

```
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 2 DESC
10  LIMIT 1;
```

Result Grid | Filter Rows:

	name	price
▶	The Greek Pizza	35.95



IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED



```
4 •   SELECT
5       pizzas.size, COUNT(order_details.quantity) AS order_count
6   FROM
7       pizzas
8   JOIN
9       order_details ON pizzas.pizza_id = order_details.pizza_id
10  GROUP BY 1
11  ORDER BY 2 DESC;
```

Result Grid |   Filter Rows: _____

	size	order_count
▶	L	311
	M	233
	S	207
	XL	6

LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES

```
3 •   SELECT
4     pizza_types.name,
5     SUM(order_details.quantity) AS order_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 pizzas.pizza_id = order_details.pizza_id
12  GROUP BY 1
13 ORDER BY 2 DESC
14 LIMIT 5;
```

Result Grid | Filter Rows:

name	order_quantity
The Classic Deluxe Pizza	43
The Italian Supreme Pizza	41
The Thai Chicken Pizza	40
The Barbecue Chicken Pizza	40
The California Chicken Pizza	38

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 1
13 ORDER BY 2 DESC;
```

Result Grid

	category	quantity
▶	Classic	232
	Supreme	186
	Chicken	179
	Veggie	175

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY

```
3 •   SELECT  
4       HOUR(order_time), COUNT(order_id) AS count_order_id  
5   FROM  
6       orders  
7   GROUP BY 1;
```

	HOUR(order_time)	count_order_id
▶	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

JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS

```
3 •   SELECT  
4       category, COUNT(name)  
5   FROM  
6   pizza_types  
7   GROUP BY 1;
```

Result Grid | Filter Rows:

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

GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY

```
3 • SELECT
4     ROUND(AVG(quantity), 0) AS average_pizza_per_day
5 FROM
6     (SELECT
7         order_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 1) AS order_quantity;
```

Result Grid | Filter Rows:

	average_pizza_per_day
▶	129

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE

```
3 •   SELECT
4     pizza_types.name, SUM(order_details.quantity * pizzas.price) as revenue
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 1
12 ORDER BY 2 DESC
13 LIMIT 3;
```

Result Grid | Filter Rows:

	name	revenue
▶	The Italian Supreme Pizza	732.75
	The Thai Chicken Pizza	730
	The Barbecue Chicken Pizza	726

CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE

```
2 • SELECT
3     pizza_types.category,
4     ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
5             SUM(order_details.quantity * pizzas.price) AS total_sales
6         FROM
7             order_details
8             JOIN
9                 pizzas ON order_details.pizza_id = pizzas.pizza_id)
10        2) AS revenue_percentage
11    FROM
12        pizza_types
13        JOIN
14            pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
15        JOIN
16            order_details ON order_details.pizza_id = pizzas.pizza_id
17    GROUP BY 1
18    ORDER BY 2 DESC;
```

Result Grid | Filter Rows:

	category	revenue_percentage
▶	Classic	26.46
	Supreme	25.37
	Chicken	25.01
	Veggie	23.16

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME

```
3 •   select order_date,round(sum(revenue) over(order by order_date),2) as cum_revenue from
4   (SELECT orders.order_date,SUM(order_details.quantity * pizzas.price) AS revenue
5    FROM order_details
6    JOIN pizzas
7    ON order_details.pizza_id = pizzas.pizza_id
8    join orders
9    on order_details.order_id = orders.order_id
10   group by 1) as sales;
```

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

	order_date	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	12852.55

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY

```
3 • select name ,revenue from
4   (select category,name,revenue,rank() over(partition by category order by revenue desc) as rn from
5   (select pizza_types.category,pizza_types.name,sum(order_details.quantity * pizzas.price) AS revenue
6   FROM order_details
7   JOIN pizzas
8   ON order_details.pizza_id = pizzas.pizza_id
9   join pizza_types
10  on pizza_types.pizza_type_id = pizzas.pizza_type_id
11  group by pizza_types.category,pizza_types.name) as a) as b
12  where rn<=3;
```

	name	revenue
▶	The Thai Chicken Pizza	730
	The Barbecue Chicken Pizza	726
	The California Chicken Pizza	676.5
T	The California Chicken Pizza	5
	The Hawaiian Pizza	478.5
	The Pepperoni Pizza	450
	The Italian Supreme Pizza	732.75
	The Spicy Italian Pizza	593
	The Sicilian Pizza	447
	The Five Cheese Pizza	573.5
	The Four Cheese Pizza	467.39...
	The Mexicana Pizza	435.5



LARANA PIZZA

OUR CONTACT

📞 7219292348

🌐 www.linkedin.com/in/kedar-kumbhar-357286222

📍 kumbharkedar5256@gmail.com



LARANA PIZZA

THANK YOU!

