

HELLO MY NAME IS
RISHABH AND IN THIS PROJECT
I HAVE UTILISED SQL QUERIES
TO SOLVE QUESTION RELATED
TO PIZZA SALES

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. 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. ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```
1 -- RETRIVE TOTAL NUMBER OF ORDER PLACED
2 • select count(order_id) as total_orders from orders;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	total_orders			
▶	21350			

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

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
total_sales	817860.05			

▶ 817860.05



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

Result Grid		Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:
	name	price			
▶	The Greek Pizza	35.95			



```
1      -- list top 5 most ordered pizzas types  
2      -- along with their quantity.
```

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

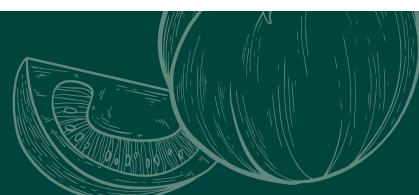
```
order_details ON order_details.pizza_id = pizzas.pizza_id
```

GROUP BY pizza_types.name

ORDER BY quantity **DESC**

LIMIT 5

	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



```
-- join the necessary table 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;
```

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050



-- determine the distribution of orders by hour of the day.

SELECT

HOUR(order_time), COUNT(order_id)

FROM

orders

GROUP BY HOUR(order_time);

Result Grid | Filter Rows:

	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

```
1      -- join relevant table to find the
2      -- category-wise distribution of pizzas
3  SELECT
4      category, COUNT(name)
5  FROM
6      pizza_types
7  GROUP BY category;
```

Result Grid | Filter Rows: | Export:

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

-- group the order by date and calculate the average
-- number of pizzas ordered per day

```
3   SELECT
4       ROUND(AVG(quantity), 0) as average_pizzas_ordered_per_day
5   FROM
6   (SELECT
7       orders.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 orders.order_date) AS order_quantity;
```

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

average_pizzas_ordered_per_day
138



```
1      -- determine the top 3 most ordered pizza types based on revenue
2
3 •  select pizza_types.name,
4      sum(order_details.quantity * pizzas.price) as revenue
5      from pizza_types join pizzas
6      on pizzas.pizza_type_id = pizza_types.pizza_type_id
7      join order_details
8      on order_details.pizza_id = pizzas.pizza_id
9      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



```
-- 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 |

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

-- 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
```

	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