



PIZZA
STORE

**THE PROJECT IS ABOUT THE
SALES ANALYSIS OF A PIZZA
STORE**



PIZZA
STORE

DATA SETS

- **TABLE1 - ORDER DETAILS** : ORDER IDS, ORDER DETAIL IDS, PIZZA IDS, QUANTITY
- **TABLE2 - ORDERS** : ORDER ID, DATE, TIME
- **TABLE3 - PIZZA TYPE** : PIZZA TYPE ID, NAME, CATEGORY, INGREDIENTS
- **TABLE4 - PIZZAS** : PIZZA IS, PIZZA TYPE ID, SIZE, PRICE



PIZZA
STORE

TOOLS USED

- **SQL** - TO FIND THE ANSWERS TO THE QUESTIONS DIFFERENT STAKEHOLDERS ASKED
- **POWER BI** - TO HAVE A VISUAL REPRESENTATION OF THE DATA



PIZZA
STORE

SQL BASIC QUESTIONS



PIZZA
STORE

Q1. RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

```
SELECT DISTINCT count(order_id) AS total_orders  
FROM Pizza_Store.orders;
```

total_orders

21350



PIZZA
STORE

Q2. CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```
SELECT round(sum(o.quantity*p.price),2) AS total_rev
  FROM order_details AS o
LEFT JOIN pizzas AS p ON p.pizza_id = o.pizza_id;
```

total_rev

817860.05



PIZZA
STORE

Q3. IDENTIFY THE HIGHEST-PRICED PIZZA.

```
SELECT t.name,p.price
  FROM pizzas AS p
LEFT JOIN pizza_type2 AS t ON t.pizza_type_id = p.pizza_type_id
 ORDER BY price DESC
 LIMIT 1;
```

name	price
The Greek Pizza	35.95



PIZZA
STORE

Q4. IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED

```
SELECT p.size , SUM(o.quantity) as orderss
  FROM order_details AS o
LEFT JOIN pizzas AS p ON p.pizza_id = o.pizza_id
 GROUP BY 1
ORDER BY orderss DESC
LIMIT 1;
```

size	orderss
L	18956



PIZZA
STORE

Q5. LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES

```
SELECT t.name,p.pizza_type_id , SUM(o.quantity) AS orderss
  FROM order_details AS o
  LEFT JOIN pizzas AS p ON p.pizza_id = o.pizza_id
  LEFT JOIN pizza_type2 AS t ON t.pizza_type_id = p.pizza_type_id
 GROUP BY 1,2
 ORDER BY orderss DESC
 LIMIT 5;
```

name	pizza_type_id	orderss
The Classic Deluxe Pizza	classic_dlx	2453
The Barbecue Chicken Pizza	bbq_ckn	2432
The Hawaiian Pizza	hawaiian	2422

The Pepperoni Pizza	pepperoni	2418
The Thai Chicken Pizza	thai_ckn	2371



PIZZA
STORE

SQL INTERMEDIATE QUESTIONS



PIZZA
STORE

Q1. JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

```
SELECT t.category , SUM(o.quantity) AS orderss
FROM order_details AS o
LEFT JOIN pizzas AS p ON p.pizza_id = o.pizza_id
LEFT JOIN pizza_type2 AS t ON t.pizza_type_id = p.pizza_type_id
GROUP BY 1;
```

category	orderss
Veggie	11649
Supreme	11987
Chicken	11050
Classic	14888



PIZZA
STORE

Q2. DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```
SELECT DISTINCT HOUR(time) as hour,  
    COUNT(order_id) AS orderss  
FROM orders  
GROUP BY 1  
ORDER BY orderss DESC;
```

hour	orderss
12	2520
13	2455
18	2399
17	2336

SO ON



PIZZA
STORE

Q3. JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.

```
SELECT category , COUNT(category) AS count
FROM pizza_type2
GROUP BY 1;
```

category	count
Veggie	9
Classic	8
Chicken	6
Supreme	9



PIZZA
STORE

Q4. GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.

```
WITH
    cal AS (SELECT DISTINCT o.date, SUM(quantity) AS ordersss
FROM orders AS o
    LEFT JOIN order_details AS d ON d.order_id = o.order_id
GROUP BY 1)

SELECT round(avg(ordersss)) AS avg_order
FROM cal;
```

avg_order
138



PIZZA
STORE

Q5. DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

```
SELECT t.name,p.pizza_type_id,  
       round(SUM(o.quantity*p.price),2) AS total_rev  
  FROM order_details AS o  
LEFT JOIN pizzas AS p ON p.pizza_id = o.pizza_id  
LEFT JOIN pizza_type2 AS t ON t.pizza_type_id = p.pizza_type_id  
 GROUP BY 1,2  
ORDER BY total_rev DESC  
LIMIT 3;
```

name	pizza_type_id	total_rev
The Thai Chicken Pizza	thai_ckn	43434.25
The Barbecue Chicken Pizza	bbq_ckn	42768
The California Chicken Pizza	cali_ckn	41409.5



PIZZA
STORE

SQL ADVANCE QUESTIONS



PIZZA
STORE

Q1. CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

```
SELECT t.name,
       round((SUM(o.quantity*p.price)/
              (SELECT round(SUM(o.quantity*p.price),2) AS total_rev
               FROM order_details AS o
               LEFT JOIN pizzas AS p ON p.pizza_id = o.pizza_id)
             )*100,2) AS perc
  FROM order_details AS o
  LEFT JOIN pizzas AS p ON p.pizza_id = o.pizza_id
  LEFT JOIN pizza_type2 AS t ON t.pizza_type_id = p.pizza_type_id
 GROUP BY 1;
```

name	perc
The Spinach Pesto Pizza	1.91
The Barbecue Chicken Pizza	5.23

SO ON



PIZZA
STORE

Q2. ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME

```
WITH
    cal AS
        (SELECT o.date, SUM(d.quantity*p.price) AS rev
         FROM orders AS o
         LEFT JOIN order_details AS d ON d.order_id = o.order_id
         LEFT JOIN pizzas AS p ON p.pizza_id = d.pizza_id
         GROUP BY 1)
SELECT date,
       SUM(rev) OVER(order by date) as cum_rev
FROM cal;
```

date	cum_rev
2015-01-01	2713.850...
2015-01-02	5445.75

SO ON



Q3. DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

```
SELECT category, name, rev, ranks
FROM
(WITH
    cal AS (SELECT t.category, t.name, SUM(o.quantity*p.price) AS rev
        FROM order_details AS o
        LEFT JOIN pizzas AS p ON p.pizza_id = o.pizza_id
        LEFT JOIN pizza_type2 AS t ON t.pizza_type_id = p.pizza_type_id
        GROUP BY 1,2 )
SELECT category, name, rev,
rank() over(PARTITION BY category ORDER BY rev DESC) AS ranks
FROM cal
) AS d
WHERE ranks <= 3
```

category	name	rev	ranks
Classic	The Classic Deluxe Pizza	38180.5	1
Classic	The Hawaiian Pizza	32273.25	2
Classic	The Pepperoni Pizza	30161.75	3
Chicken	The Thai Chicken Pizza	43434.25	1
Chicken	The Barbecue Chicken Pizza	42768	2
Chicken	The California Chicken Pizza	41409.5	3