



SQL PROJECT ON PIZZA SALES





PROJECT SUMMARY

"SQL-POWERED ANALYSIS OF
PIZZA SALES PERFORMANCE"

"Hello, I'm Sachin Ughade. In this project, I leveraged SQL queries to unlock insights from pizza sales data, answering key business questions and revealing patterns that drive smarter decisions."



TABLES

[Home](#)[About](#)[Contact](#)

Table- orders_details

```
CREATE TABLE `order_details` (
  `order_details_id` INT NOT NULL,
  `order_id` INT NULL,
  `pizza_id` VARCHAR(50) NULL,
  `quantity` INT NULL,
  PRIMARY KEY (`order_details_id`),
  CONSTRAINT `fk_order_details_order_id` FOREIGN
  KEY (`order_id`) REFERENCES `orders`(`order_id`),
  CONSTRAINT `fk_order_details_pizza_id` FOREIGN
  KEY (`pizza_id`) REFERENCES `pizzas`(`pizza_id`)
```



Table- orders

```
CREATE TABLE `orders` (
  `order_id` INT NOT NULL,
  `date` DATE NULL,
  `time` TIME NULL,
  PRIMARY KEY (`order_id`)
```



Table- pizza_types

```
CREATE TABLE `pizza_types` (
  `pizza_type_id` VARCHAR(50) NOT NULL,
  `name` VARCHAR(50) NULL,
  `category` VARCHAR(50) NULL,
  `ingredients` VARCHAR(100) NULL,
  PRIMARY KEY (`pizza_type_id`)
```



Table- pizzas

```
CREATE TABLE `pizzas` (
  `pizza_id` VARCHAR(50) NOT NULL,
  `pizza_type_id` VARCHAR(50) NULL,
  `size` VARCHAR(50) NULL,
  `price` DECIMAL(10,2) NULL,
  PRIMARY KEY (`pizza_id`),
  CONSTRAINT `fk_pizzas_pizza_type_id` FOREIGN KEY (`pizza_type_id`) REFERENCES `pizza_types`(`pizza_type_id`)
```



QUE 1.

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

Home

About

Contact

SELECT

COUNT(order_id) AS order_counts

FROM

orders;

OUTPUT

	order_counts
▶	21350



QUE 2:- CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```
SELECT  
    SUM(order_details.quantity * pizzas.price) AS total_sales  
FROM  
    order_details  
    JOIN  
    pizzas ON order_details.pizza_id = pizzas.pizza_id;
```

OUTPUT

	total_sales
▶	817860.0499999993



QUE 3:-

IDENTIFY THE HIGHEST-PRICED PIZZA

About

Contact

```
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY price DESC
LIMIT 1;
```

OUTPUT

	name	price
▶	The Greek Pizza	35.95



QUE 4:-

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

About

Contact

```
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS order_count
FROM
    pizzas
        JOIN
            order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC
LIMIT 1;
```

OUTPUT

	size	order_count
▶	L	18526



QUE 5 :- LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

```
SELECT
    pizza_types.name, SUM(order_details.quantity) AS quantities
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 quantities DESC
LIMIT 5;
```

OUTPUT

	name	quantities
▶	The Classic Deluxe Pizza	2453
	The Barbecue Chicken Pizza	2432
	The Hawaiian Pizza	2422
	The Pepperoni Pizza	2418
	The Thai Chicken Pizza	2371



QUE 6:- JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED

```
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS total_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 total_quantity DESC;
```

OUTPUT

	category	total_quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050



QUE 7:-

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

Contact

```
SELECT  
    HOUR(Order_time) AS hour, COUNT(Order_id) AS order_count  
FROM  
    Orders  
GROUP BY (Order_time);
```

OUTPUT

	hour	order_count
▶	11	2
	11	1
	12	1
	12	3
	12	1
	12	1
	12	1
	12	1



QUE 8:- GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.

```
SELECT avg(quantity) FROM
(SELECT orders.Order_date ,  SUM(order_details.quantity) as quantity
FROM orders
JOIN order_details
ON orders.Order_id=order_details.Order_id
GROUP BY orders.Order_date ) as order_quantity;
```

OUTPUT

	avg(quantity)
▶	138.4749



QUE 9:- DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

Contact

```
SELECT
    pizza_types.name,
    SUM(order_details.quantity * pizzas.price) 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.name
ORDER BY revenue DESC
LIMIT 3;
```

OUTPUT

	name	revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5



QUE 10 :-CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

[Home](#)[About](#)[Contact](#)

```
SELECT
    pizza_types.category,
    SUM(order_details.quantity * pizzas.price) / (SELECT
        SUM(order_details.quantity * pizzas.price) AS total_sales
    FROM
        order_details
        JOIN
            pizzas ON order_details.pizza_id = pizzas.pizza_id) * 100 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;
```

OUTPUT

	category	revenue
▶	Veggie	23.682590927384783
	Chicken	23.955137556847493
	Supreme	25.45631126009884
	Classic	26.905960255669903



Thank You!