

HELLO, MY NAME IS VISHAL TOLANI. IN THIS PROJECT, I HAVE UTILISED SQL QUERIES TO SOLVE QUESTIONS RELATED TO PIZZA SALES.



Retrieve the total number of orders placed.

```
SELECT  
    COUNT(order_id) AS total_orders  
FROM  
    orders;
```

| | total_orders |
|---|--------------|
| ▶ | 21350 |

Calculate the total revenue generated from pizza sales

```
SELECT COUNT(order_details.quantity * pizzas.price) as Total_sales  
FROM  
    order_details  
    JOIN  
    pizzas ON order_details.pizza_id = pizzas.pizza_id
```

| Total_sales |
|-------------|
| 48620 |

IDENTIFY THE HIGHEST-PRICED PIZZA.

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

| | name | price |
|---|-----------------|-------|
| ▶ | The Greek Pizza | 35.95 |

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.TEXT

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

| | size | order_count |
|---|------|-------------|
| ▶ | L | 18526 |
| | M | 15385 |
| | S | 14137 |
| | XL | 544 |
| | XXL | 28 |

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

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

| | name | total_order |
|---|----------------------------|-------------|
| ▶ | 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 TABLES 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) AS hour, COUNT(order_id) AS order_count  
FROM  
    orders  
GROUP BY HOUR(order_time);
```

| | hour | order_count |
|---|------|-------------|
| ▶ | 9 | 1 |
| | 10 | 8 |
| | 11 | 1231 |
| | 12 | 2520 |
| | 13 | 2455 |
| | 14 | 1472 |
| | 15 | 1468 |
| | 16 | 1920 |

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

```
select category, count(name) from pizza_types  
group by category;
```

| | 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.

```
SELECT
    ROUND(AVG(quantity), 0) as avearage_pizza_per_day
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;
```

| | avearage_pizza_per_day |
|---|------------------------|
| ▶ | 138 |

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

```
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 |
```

| | 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)
    2) AS revenue
FROM
    pizza_types
    JOIN
        pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
    JOIN
```

| | 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
   group by orders.order_date) as sales;
```

| | order_date | cum_revenue |
|---|------------|-------------------|
| ▶ | 2015-01-01 | 2713.850000000004 |
| | 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 | 23990.35000000002 |
| | 2015-01-11 | 25862.65 |
| | 2015-01-12 | 27781.7 |
| | 2015-01-13 | 29831.30000000003 |

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

```
(select name, revenue from
(select category, name, revenue,
rank() over (partition by category order by revenue desc) as rn
from
(select pizza_types.category, 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.category, pizza_types.name) as a) as b
where rn <=3);
```

| | 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 |