



# Pizza Store Sales Analysis

A deep analysis of pizza store data  
with SQL

WELCOME

# Hello!

I'm Gopal Kholade a data analyst. I will be showcasing my findings on data.



**Know  
About  
Data**

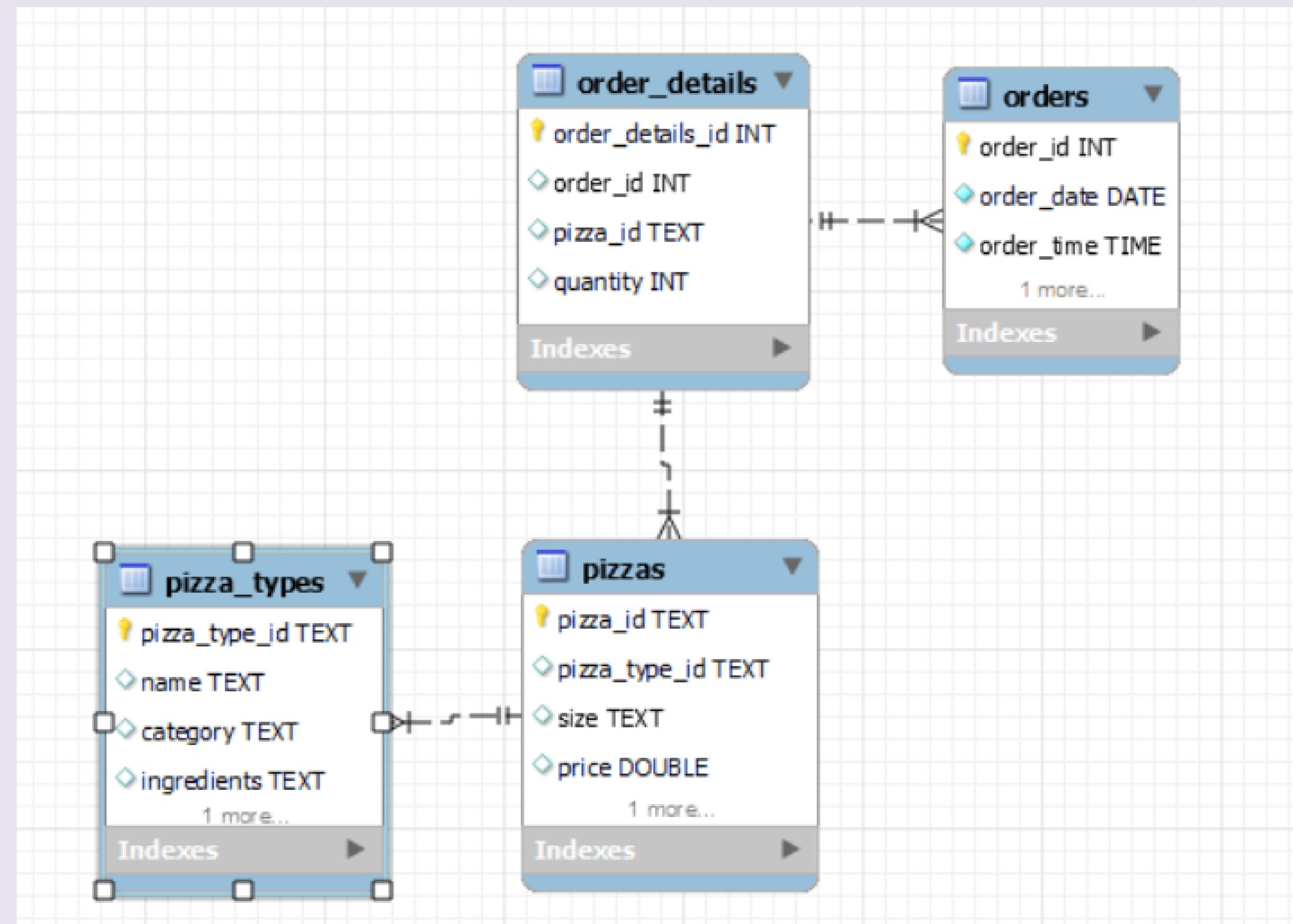
# TABLES



**There are four tables  
in our database**

- Order Details
- Orders
- Pizza
- Pizza Types

# ER Diagram





# FINDINGS

Showcased findings on next few pages

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

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

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
        2) AS sum_of_sales
FROM
    order_details
    LEFT JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id;
```

IDENTIFY THE HIGHEST-PRICED PIZZA.

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

## IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```
SELECT
    size, COUNT(order_id) AS cnt
FROM
    pizza_types
        LEFT JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        RIGHT JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY size
ORDER BY cnt DESC
LIMIT 1;
```

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

```
SELECT
    pizza_types.name,
    SUM(quantity),
    COUNT(order_id) AS cnt
FROM
    order_details
        LEFT JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id
        LEFT JOIN
    pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
GROUP BY pizza_types.name
ORDER BY cnt DESC
LIMIT 5;
```

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

```
SELECT
    category, SUM(quantity) AS total_quantity
FROM
    order_details
        LEFT JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id
        LEFT JOIN
    pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
GROUP BY pizza_types.category;
```

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```
SELECT  
    HOUR(orders.order_time) AS hour,  
    COUNT(order_id) AS No_of_orders  
FROM  
    orders  
GROUP BY hour;
```

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

```
SELECT
    category, COUNT(order_id)
FROM
    order_details
        LEFT JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id
        LEFT JOIN
    pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
GROUP BY category;
```

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

```
SELECT
    order_date, ROUND(AVG(order_id)) AS Average_orders
FROM
    orders
GROUP BY order_date;

-- Determine the top 3 most ordered pizza types based on revenue.
SELECT
    name, SUM(quantity * price) AS rev
FROM
    order_details
        LEFT JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id
        LEFT JOIN
    pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
GROUP BY pizza_types.name
ORDER BY rev DESC
LIMIT 3;
```

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```
select order_time, sum(quantity*price) over(order by order_time) as cum_sum
from order_details left join orders on order_details.order_id=orders.order_id left join pizzas on order_details.pizza_id=pizzas.pizza_id;
```

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

```
select category, name, rev, ranks
from
(select category, name, rev, rank() over(partition by category order by rev desc) as ranks
from
(select category, name, sum(quantity*price) as rev
from order_details left join pizzas on order_details.pizza_id=pizzas.pizza_id
left join pizza_types on pizzas.pizza_type_id=pizza_types.pizza_type_id
group by pizza_types.category, pizza_types.name) as tb) as tr
where ranks <= 3;
```

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER DATE.

```
select order_date, sum(sum_r) over (order by order_date) as cum_rev
from
(select order_date, sum(quantity*price) as sum_r
from order_details left join orders on order_details.order_id=orders.order_id
left join pizzas on order_details.pizza_id=pizzas.pizza_id
group by order_date) as sum_rev;
```

**Thanks  
FOR READING**