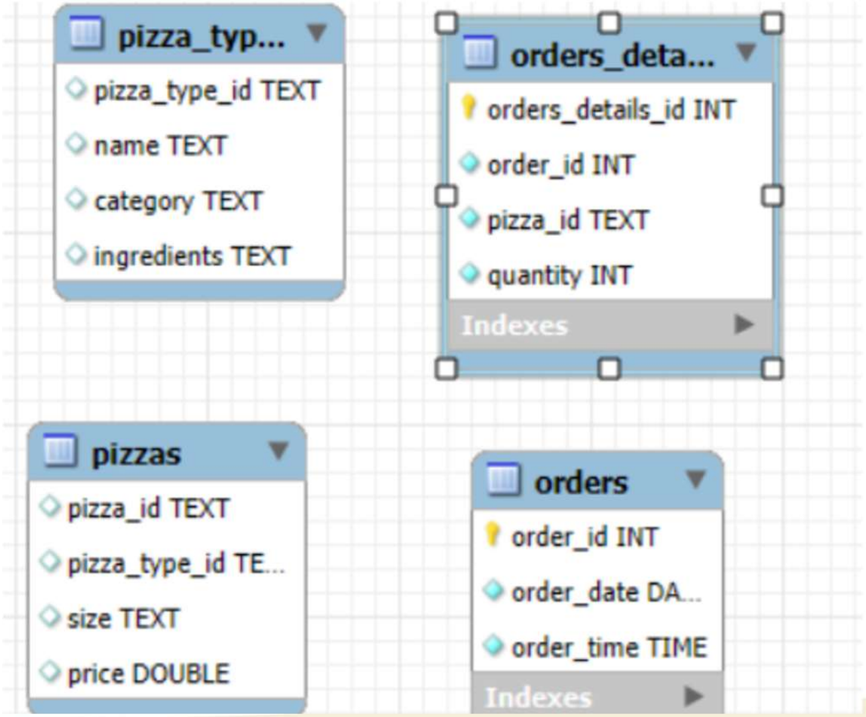
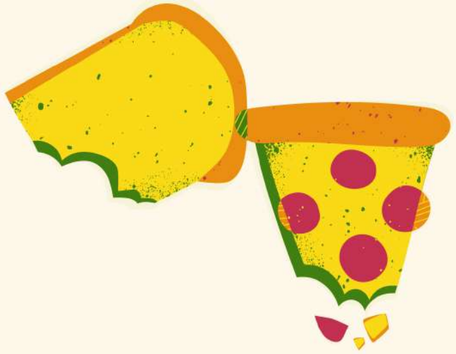
SQL PROJECT FOR PIZZA SALES

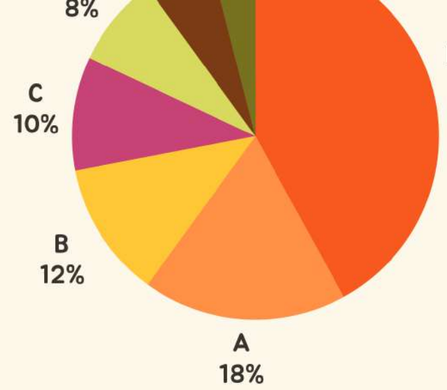
My name is Mukesh Bhatt

We have utilize SQL to answer some important question in Pizza sales





Retrieve the total number of orders placed.




```sql
SELECT
    COUNT(*) AS Total_orders
FROM
    orders;
```

Result Grid

	Total_orders
▶	21350

Calculate the total revenue generated from pizza sales.

```sql
SELECT
    ROUND(SUM(orders_details.quantity * pizzas.price),
            2) AS Total_sale
FROM
    orders_details
        JOIN
    pizzas ON pizzas.pizza_id = orders_details.pizza_id;
```

Result Grid

	Total_sale
▶	817860.05

Identify the highest-priced pizza.

```sql
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;
```

Result Grid | Filter Rows:

name	price
The Greek Pizza	35.95

Identify the most common pizza size ordered.

```sql
SELECT
    pizzas.size, COUNT(orders_details.quantity)
FROM
    pizzas
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizzas.size
ORDER BY COUNT(orders_details.quantity) DESC
LIMIT 1;
```

Result Grid | Filter Rows:

	size	COUNT(orders_details.quantity)
▶	L	18526

List the top 5 most ordered pizza types along with their quantities.

```sql
SELECT
    pizza_types.name, SUM(orders_details.quantity) AS most_fav
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY most_fav DESC
LIMIT 5;
```

Result Grid | Filter Rows:

name	most_fav
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.

```sql
SELECT
    pizza_types.category,
    SUM(orders_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity DESC;
```

Result Grid | Filter Rows:

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Determine the distribution of orders by hour of the day.

```sql
SELECT
    HOUR(order_time) AS total_orders_by_hours,
    COUNT(order_id) AS total_order
FROM
    orders
GROUP BY total_orders_by_hours
ORDER BY total_order DESC;
```

Result Grid | Filter Rows:

total_orders_by_hours	total_order
12	2520
13	2455
18	2399
17	2336
19	2009
16	1920
20	1642
14	1472
15	1468
11	1231
21	1198
22	663
23	28
10	8
9	1

Join relevant tables to find the category-wise distribution of pizzas.

```sql
SELECT
    category, COUNT(name) AS Total_pizzas
FROM
    pizza_types
GROUP BY category
ORDER BY Total_pizzas DESC;
```

Result Grid | Filter Rows:

category	Total_pizzas
Supreme	9
Veggie	9
Classic	8
Chicken	6

Group the orders by date and calculate the average number of pizzas ordered per day.

```sql
SELECT
    round(AVG(quantity),0) as avg_pizza_order_per_day
FROM
    (SELECT
        orders.order_date AS perDay_order,
            SUM(orders_details.quantity) AS quantity
    FROM
        orders
    JOIN orders_details ON orders_details.order_id = orders.order_id
    GROUP BY perDay_order) AS order_quanity;
```

Result Grid | Filter Rows:

avg_pizza_order_per_day
138

Determine the top 3 most ordered pizza types based on revenue.

```sql
SELECT
    pizza_types.name,
    SUM(orders_details.quantity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue DESC
LIMIT 3;
```

Result Grid | Filter 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.

```sql
SELECT
    pizza_types.category,
    (SUM(orders_details.quantity * pizzas.price) / (SELECT
            ROUND(SUM(orders_details.quantity * pizzas.price),
                    2)
        FROM
            orders_details
                JOIN
            pizzas ON pizzas.pizza_id = orders_details.pizza_id)) * 100 AS revenue_percentage
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY revenue_percentage;
```

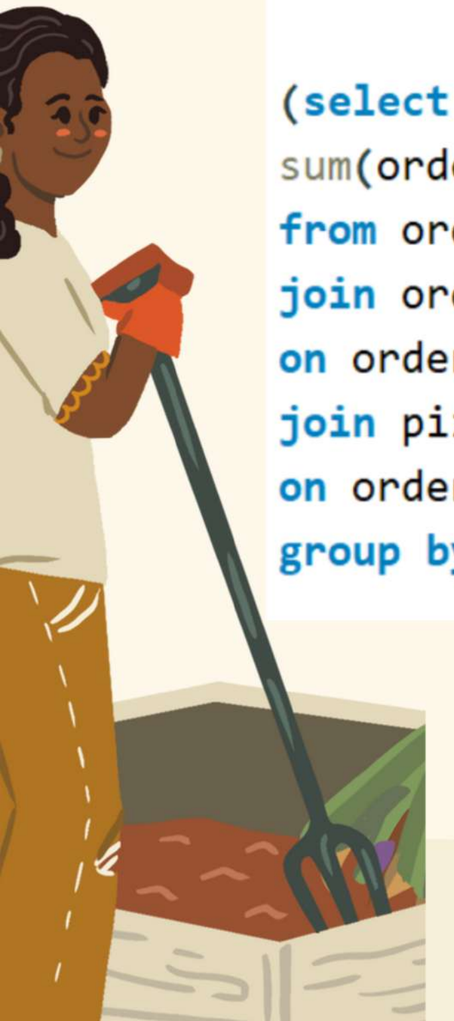
Result Grid | Filter Rows:

category	revenue_percentage
Veggie	23.682590927384577
Chicken	23.955137556847287
Supreme	25.45631126009862
Classic	26.90596025566967

Analyze the cumulative revenue generated over time.

```sql
select order_date,
sum(revenue)over(order by order_date) as cum_rev from

(select orders.order_date,
sum(orders_details.quantity * pizzas.price) as revenue
from orders
join orders_details
on orders.order_id = orders_details.order_id
join pizzas
on orders_details.pizza_id = pizzas.pizza_id
group by orders.order_date) as Sales;
```



Result Grid | Filter Rows:

order_date	cum_rev
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	23990.350000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.300000000003
2015-01-14	32358.700000000004

esult 4

Determine the top 3 most ordered pizza types based on revenue for each pizza category.

```sql
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(orders_details.quantity * pizzas.price ) as revenue
from pizza_types
join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join orders_details
on pizzas.pizza_id = orders_details.pizza_id
group by pizza_types.category, pizza_types.name)as a) as b
where rn <= 3 ;
```

Result Grid | Filter Rows:

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