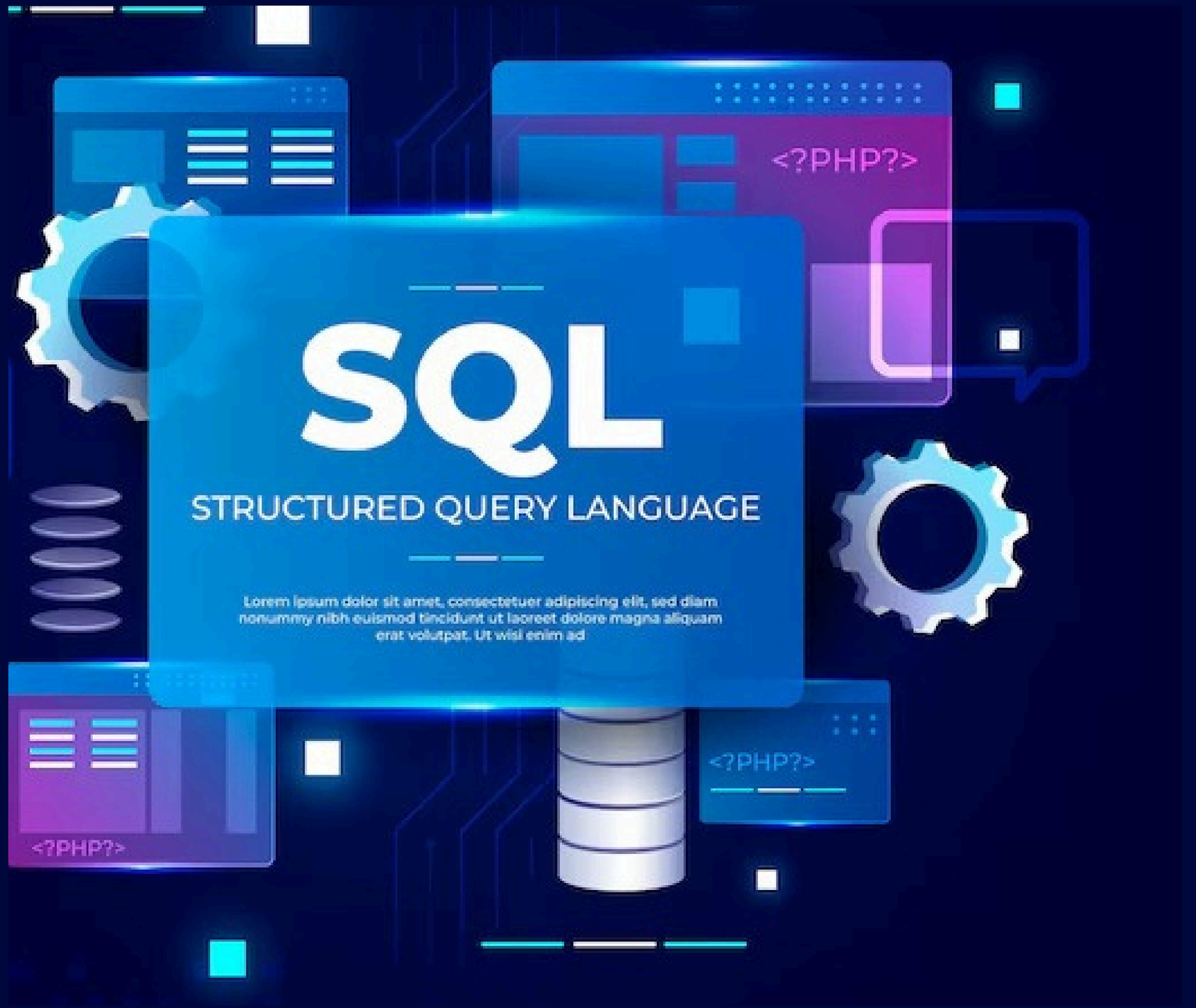
PIZZA SALES PROJECT REPORT

NIKITA VISHWAKARMA



THESE ARE THE QUESTION WE SOLVED THIS PROJECT

1. Retrieve the total number of product sales.
2. Calculate the total sales from pizza.
3. . Identify the top 3 highest price pizzas.
4. Identify the most common pizza size ordered.
5. List the top 5 most ordered pizza along with their quantities.
6. Joint the necessary table to find the total quantity of each pizza category.
7. Determine the distribution of order hour of the day.
8. Join Relevant table to find the category wise distribution of pizzas.
9. Group the ordered by date and calculate the average number of pizzas order per day.
10 . Determine the top 3 most ordered pizza types based on revenue.
11. Calculate the percentage contribution of each pizzas type to total revenue.
12. Analyze the cumulative revenue generate over time.
13. Determine the top 3 most ordered pizza type based on revenue for each pizza category.

FIRSTLY WE WILL UPLOADE ALL DATA WITH TABLE

Order_detail

Orders

Pizza_types

Pizzas

1. Retrieve the total number of product sales.

```sql
SELECT
    COUNT(order_detail.quantity) AS total_product_sales
FROM
    order_detail;
```

total_product_sales
48620

2. Calculate the total sales from pizza.

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

	total_sales
▶	817860.05

3. Identify the top 3 highest price pizzas.

```sql
SELECT
    pizza_types.name, pizzas.price, pizzas.size
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 3;
```

name	price	size
The Greek Pizza	35.95	XXL
The Greek Pizza	25.5	XL
The Brie Carre Pizza	23.65	S

4. Identify the most common pizza size ordered.

```
SELECT
    pizzas.size,
    COUNT(order_detail.order_detail_id) AS order_product
FROM
    pizzas
        JOIN
    order_detail ON pizzas.pizza_id = order_detail.pizza_id
GROUP BY pizzas.size
ORDER BY order_product DESC;
```

size	order_product
L	18526
M	15385
S	14137
XL	544
XXL	28

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

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

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

6. Joint the necessary table to find the total quantity of each pizza category.

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

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

7. Determine the distribution of order hour of the day.

```sql
SELECT
    HOUR(order_time) as_hour, COUNT(order_id) AS order_count
FROM
    orders
GROUP BY HOUR(order_time);
```

as_hour	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642

20	1642
21	1198
22	663
23	28
10	8
9	1

8. Join Relevant table 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

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

```sql
SELECT
    ROUND(AVG(quantity), 0) AS avg_pizza_order_per_day
FROM
    (SELECT
        orders.order_date, SUM(order_detail.quantity) AS quantity
    FROM
        orders
    JOIN order_detail ON orders.order_id = order_detail.order_id
    GROUP BY orders.order_date) AS order_quantity;
```

avg_pizza_order_per_day
138

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

```sql
SELECT
    pizza_types.name,
    SUM(order_detail.quantity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_detail ON order_detail.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

11. Calculate the percentage contribution of each pizzas type to total revenue

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

category	revenue
Veggie	23.68
Chicken	23.96
Supreme	25.46
Classic	26.91

12. Analyze the cumulative revenue generate over time.

```
select order_date ,sum(revenue) over(order by order_date) as cum_revenue from
(select orders.order_date,
sum(order_detail.quantity * pizzas.price) as revenue
from order_detail join pizzas
on order_detail.pizza_id = pizzas.pizza_id
join orders
on orders.order_id = order_detail.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	23990.350000000002

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

```sql
select name, category, 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_detail.quantity*pizzas.price) as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join order_detail
on order_detail.pizza_id = pizzas.pizza_id
group by pizza_types.category , pizza_types.name) as p ) as r
where rn>3;
```

name	category	revenue
The Southwest Chicken Pizza	Chicken	34705.75
The Chicken Alfredo Pizza	Chicken	16900.25
The Chicken Pesto Pizza	Chicken	16701.75
The Greek Pizza	Classic	28454.100000
The Italian Capocollo Pizza	Classic	25094
The Napolitana Pizza	Classic	24087
The Big Meat Pizza	Classic	22968

THANK YOU

Nikita Vishwakarma