PIZZA SALES REPORT



QUESTIONS

Basic:

- 1. Retrieve the total number of orders placed.
- 2. Calculate the total revenue generated from pizza sales.
- 3. Identify the highest-priced pizza.
- 4. Identify the most common pizza size ordered.
- 5. List the top 5 most ordered pizza types along with their quantities.

Intermediate:

- 1. Join the necessary tables to find the total quantity of each pizza category ordered.
- 2. Determine the distribution of orders by hour of the day.
- 3. Join relevant tables to find the category-wise distribution of pizzas.
- 4. Group the orders by date and calculate the average number of pizzas ordered per day.
- 5. Determine the top 3 most ordered pizza types based on revenue.

Advanced:

- 1. Calculate the percentage contribution of each pizza type to total revenue.
- 2. Analyze the cumulative revenue generated over time.
- 3. Determine the top 3 most ordered pizza types based on revenue for each pizza category.

```
-- Retrive the total number order placed .

SELECT

COUNT(order_id) AS total_orders

FROM

orders;
```

total_orders

21350

total_sales

817860.05

```
-- Identify the highest-priced pizza.

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

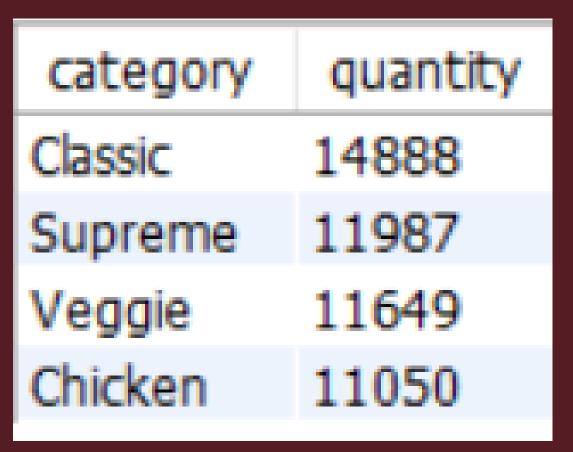
size count(order_details_id)
L 18526

```
-- List the top 5 most ordered pizza types along with their quantities.
SELECT
   pizza_types.name, 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.name
ORDER BY quantity DESC
LIMIT 5;
```

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

-- 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
limit 5;
```



```
-- Determine the distribution of orders by hour of the day.

SELECT

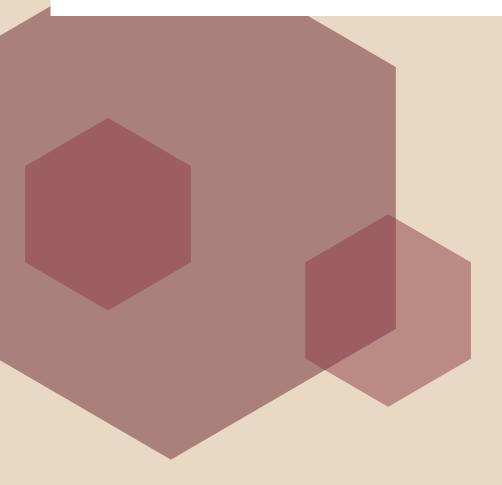
HOUR(order_time) AS hour, COUNT(order_id) AS orders

FROM

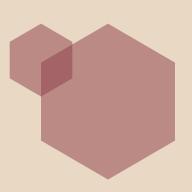
orders

GROUP BY HOUR(order_time)

ORDER BY COUNT(order_id) desc;
```



hour	orders
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
11. 6	



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

SELECT
category, COUNT(name) AS name_counts

FROM
pizza_types
GROUP BY category
ORDER BY name_counts;
```

category	name_counts
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 avg_order_per_day
FROM
    (SELECT
        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;
```

avg_order_per_day

138

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

name	revenue
The Thai Chicken Pizza	43434
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41410

```
-- 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) * 100,
           2) 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 DESC;
```

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	2713.8500000000004
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55

```
-- 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