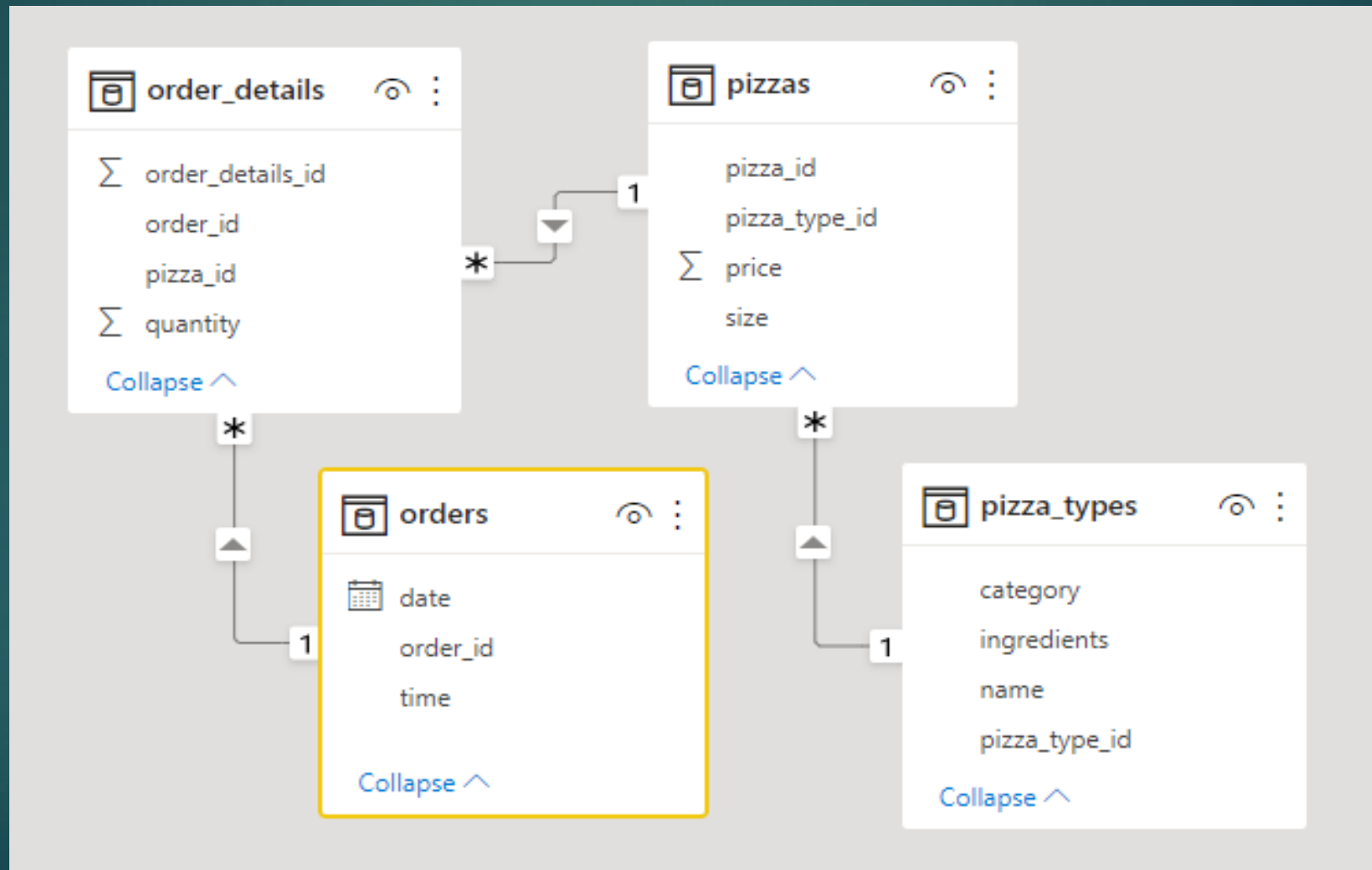# PIZZA SALES REPORT

HELLO , I AM PRAJAKTA AWALE

IN THIS PROJECT I HAVE UTILIZED SQL QUERIES TO SOLVE THE

QUESTIONS THAT WERE BASED ON PIZZA SALES.

THIS PROJECT COVERS BASIC, INTERMEDIATE AND ADVANCE LEVEL QUESTIONS RELATED TO PIZZA SALES.

Data set includes 4 files below is the schematic representation of relationship between them.

PIZZA SALES REPORT USING SQL

Solution for below questions

▶ *Basic:*

- Retrieve the total number of orders placed.

- Calculate the total revenue generated from pizza sales.

- Identify the highest-priced pizza.

- Identify the most common pizza size ordered.

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

▶ *Intermediate*:

- Join the necessary tables to find the total quantity of each pizza category ordered.

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

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

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

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

▶ *Advanced*:

- Calculate the percentage contribution of each pizza type to total revenue.

- Analyze the cumulative revenue generated over time.

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

Q1

-- Retrieve the total number of orders placed.

SELECT COUNT(order_id) AS total_orders

FROM   orders


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

	total_orders
▶	21350

Q2

- -- Calculate the total revenue generated from pizza sales.

SELECT

ROUND(SUM(order_details.quantity * pizzas.price), 2) AS total_revenue

FROM   order_details   JOIN pizzas

ON order_details.pizza_id = pizzas.pizza_id

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

total_revenue
817860.05

Q3

- -- Identify the highest-priced pizza

SELECT   (pizzas.price), pizza_types.name

FROM   pizzas   JOIN   pizza_types

ON pizzas.pizza_type_id = pizza_types.pizza_type_id

ORDER BY price DESC

LIMIT 1

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

	price	name
▶	35.95	The Greek Pizza

Q4

- -- Identify the most common pizza size ordered

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

```sql
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

Q5

-- 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

```sql
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

Q6

-- 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 pizzas.pizza_id = order_details.pizza_id

GROUP BY pizza_types.category

ORDER BY quantity DESC

```sql
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 pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity DESC
```

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Q7

-- 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)

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

hour	order_count
11	1231
12	2520
13	2455
14	1472
15	1468

Q8

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

select category, count(name)

from pizza_types

group by category

order by count(name) desc

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

category	count(name)
Supreme	9
Veggie	9
Classic	8
Chicken	6

Q9

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

SELECT   ROUND(AVG(quantity), 0) AS average_no_of_pizzas

FROM

(SELECTDATE(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

```sql
SELECT
    ROUND(AVG(quantity), 0) AS average_no_of_pizzas
FROM
    (SELECT
        DATE(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
```

average_no_of_pizzas
▶ 138

Q10

-- 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


```
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

Q11

-- 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 pizzason 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 ;

```
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

Q12

-- Analyze the cumulative revenue generated over time.

select order_date,

sum(revenue) over (order by order_date) as cumulative_revenuefrom

(

SELECT orders.order_date,sum(order_details.quantity*pizzas.price) as revenue

from pizzas

join order_details

on order_details.pizza_id=pizzas.pizza_id

join orders

on order_details.order_id=orders.order_id

group by orders.order_date

) as sales

```sql
select order_date,
sum(revenue) over (order by order_date) as cumulative_revenue
from
(SELECT orders.order_date,sum(order_details.quantity*pizzas.price) as revenue
from pizzas
join order_details
on order_details.pizza_id=pizzas.pizza_id
join orders
on order_details.order_id=orders.order_id
group by orders.order_date) as sales
```

order_date	cumulative_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

Q13

-- 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 abc) as pnq

where rn <=3

```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(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 abc) as pnq
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