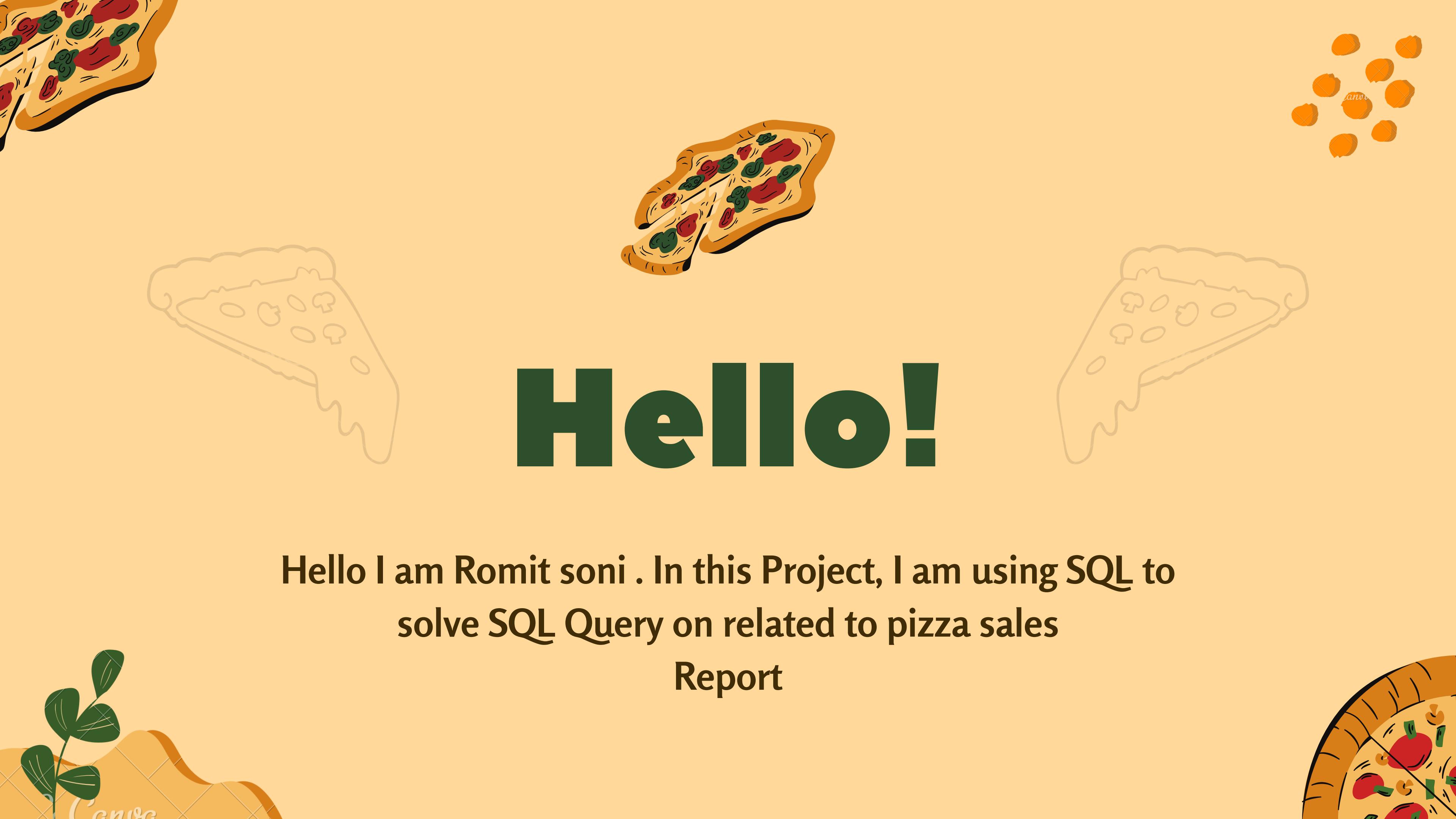


Delicious Pizza for Everyone!

PIZZA SALES REPORT

@Romit Soni



Hello!

Hello I am Romit soni . In this Project, I am using SQL to
solve SQL Query on related to pizza sales
Report



Project's Aim

This **SQL** project on sales pizza reports aims to leverage data analysis techniques to extract valuable insights from a database, enabling stakeholders to make informed decisions and drive business growth in the competitive pizza industry.



Q1. Retrieve the total number of orders placed.

```
select count(order_id) from orders
```

	count bigint	🔒
1	21350	

Q2. Calculate the total revenue generated from pizza sales

```
select sum(a.quantity*b.price) as revenue  
from order_details as a  
join  
pizzas as b  
on a.pizza_id = b.pizza_id
```

revenue	
double precision	
817860.049999993	

Q3. Identify the highest-priced pizza.

```
select a.name , b.price  
from pizza_types as a  
join pizzas as b  
on a.pizza_type_id = b.pizza_type_id  
order by b.price desc  
limit 1
```

name character varying	price double precision
The Greek Pizza	35.95

Q4. Identify the most common pizza size ordered.

```
select pizzas.size , count(order_details.order_details_id) as total_type  
from pizzas  
join order_details on  
pizzas.pizza_id = order_details.pizza_id  
group by pizzas.size  
order by total_type desc
```

size character varying	total_type bigint
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
pizzas.pizza_id = order_details.pizza_id
group by pizza_types.name
order by quantity desc
limit 5
```

	name character varying	quantity bigint
1	The Classic Deluxe Pizza	2453
2	The Barbecue Chicken Pizza	2432
3	The Hawaiian Pizza	2422
4	The Pepperoni Pizza	2418
5	The Thai Chicken Pizza	2371

Q6 .Join the necessary tables to find the total quantity of each pizza category ordered.

```
select sum(a.quantity) as quantity , b.category  
from order_details as a  
join pizzas on  
a.pizza_id = pizzas.pizza_id  
join pizza_types as b  
on pizzas.pizza_type_id = b.pizza_type_id  
group by b.category  
order by quantity desc
```

	quantity bigint	category character varying
1	14888	Classic
2	11987	Supreme
3	11649	Veggie
4	11050	Chicken

Q7. Join relevant tables to find the category-wise distribution of pizzas.

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

category character varying	distribution bigint
Supreme	9
Veggie	9
Classic	8
Chicken	6

Q8. Determine the top 3 most ordered pizza types based on revenue.

```
select pizza_types.name ,sum(a.quantity*b.price) as revenue  
from order_details as a  
join  
pizzas as b  
on a.pizza_id = b.pizza_id  
join pizza_types on  
b.pizza_type_id = pizza_types.pizza_type_id  
group by pizza_types.name  
order by revenue desc  
limit 3
```

	name character varying	revenue double precision
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768
3	The California Chicken Pizza	41409.5

Q9. Group the orders by date and calculate the average number of pizzas ordered per day.

```
select round(avg(avg_per_day),0) as average_per_day from  
(select a.date , round(sum(b.quantity),0) as avg_per_day  
from orders as a  
join order_details as b  
on a.order_id = b.order_id  
group by a.date  
order by date asc) as order_quantity
```

average_per_day
numeric
138

Q10. Determine the distribution of orders by hour of the day

```
select extract (hour from time)as hours ,  
count(order_id)  as per_hour_order  
from orders  
group by hours  
order by hours desc
```

	hours numeric	per_hour_order bigint
1	23	28
2	22	663
3	21	1198
4	20	1642
5	19	2009

Q11. 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 sum(a.quantity*c.price) as revenue , b.name , b.category  
from order_details as a  
join pizzas as c  
on a.pizza_id = c.pizza_id  
join pizza_types as b  
on c.pizza_type_id = b.pizza_type_id  
group by 2,3  
order by revenue desc) as y) as z  
where rn <= 3
```

name character varying	revenue double precision
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5
The Classic Deluxe Pizza	38180.5

Q12. Calculate the percentage contribution of each pizza type to total revenue.

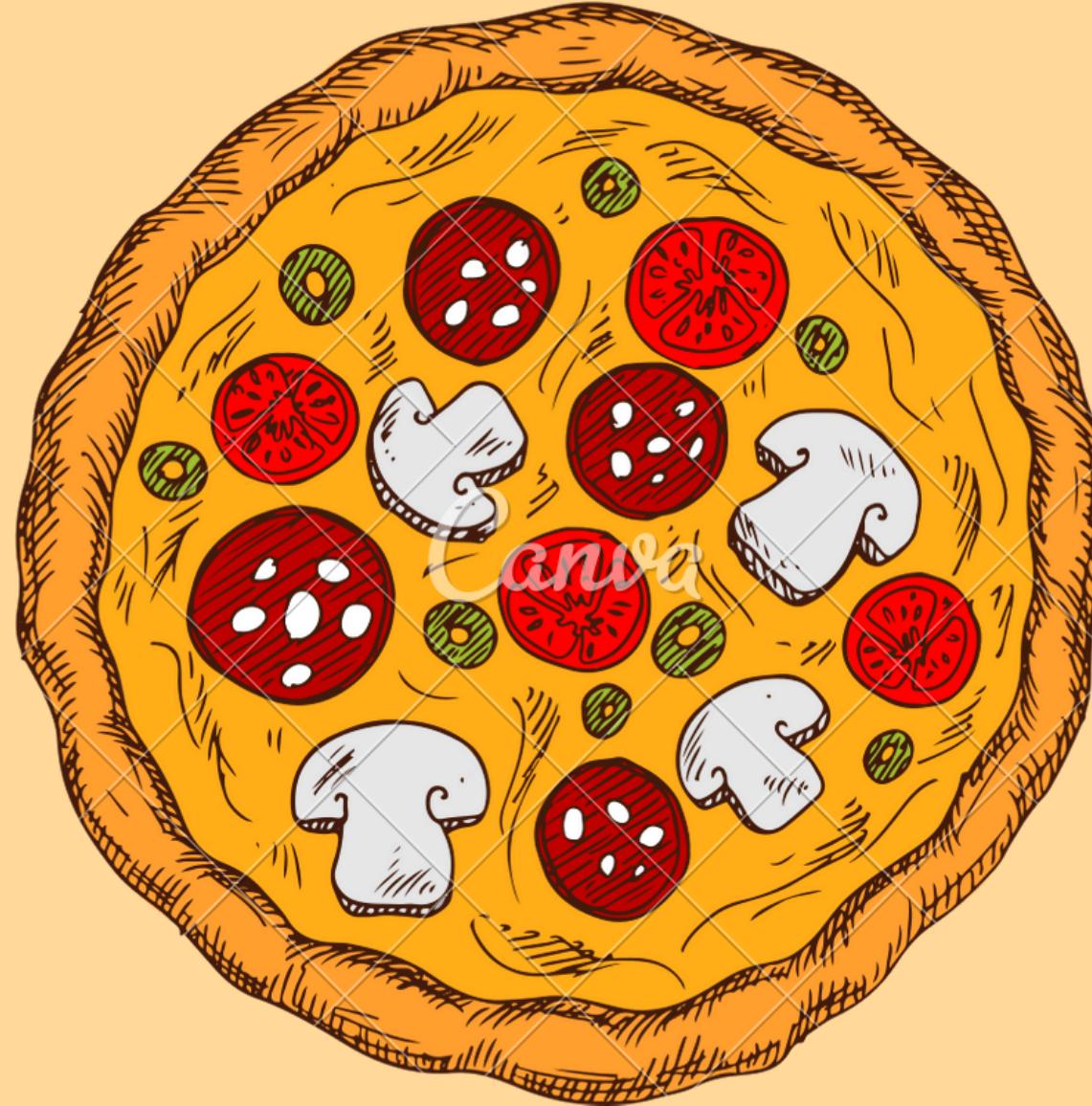
```
WITH total_revenue AS (
    SELECT SUM(a.quantity * b.price) AS total
    FROM order_details AS a
    JOIN pizzas AS b ON a.pizza_id = b.pizza_id)
SELECT c.category ,
    (sum(a.quantity*b.price) / (SELECT total FROM total_revenue)*100)
AS percentage_contribution
FROM order_details AS a
JOIN
    pizzas AS b
ON a.pizza_id = b.pizza_id
JOIN
    pizza_types AS c ON
    b.pizza_type_id = c.pizza_type_id
GROUP BY c.category
ORDER BY percentage_contribution DESC
```

category character varying	percentage_contribution double precision
Classic	26.905960255669903
Supreme	25.45631126009884
Chicken	23.955137556847493
Veggie	23.682590927384783

Q.13 Analyze the cumulative revenue generated over time.

```
select date ,  
       sum(revenue) over (order by date) as cum_revenue  
  from  
( select orders.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  
order_details.order_id = orders.order_id  
 group by orders.date  
 order by revenue desc ) as sales
```

date date	cum_revenue double precision
2015-01-01	2713.8500000000004
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6



Pizza Sales Report

**THANK
YOU**

romitsoni143@gmail.com