SQL PROJECT ON PIZZA SALES

Hello, I am Kiran Kumar. I have contributed to the GitHub project titled "Pizza Sales" by employing SQL queries in my work.

1.Retrieve the total number of orders placed.

```
select count(order_id) as Total_Orders from orders;
```

Result Grid

Total_Orders
21350

2. Calculate the total revenue generated from pizza sales.

```
select sum(orders_details.quantity * pizzas.price) as Total_Sales
from orders_details join pizzas
on pizzas.pizza_id = orders_details.pizza_id;
```

Result Grid | Filter

	Total_Sales
▶	817860.049999993

3.Identify the highest-priced pizza.

```
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:

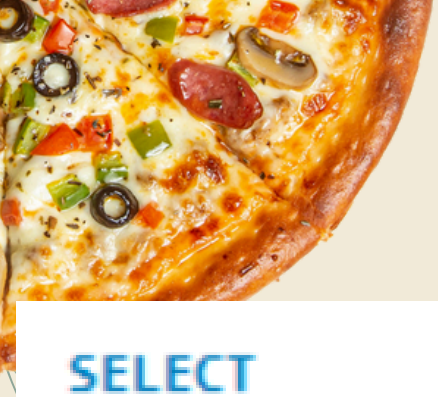
	name	price
▶	The Greek Pizza	35.95

4.Identify the most common pizza size ordered.

```sql
SELECT
    pizzas.size,
    COUNT(DISTINCT order_id) AS 'No. of Orders',
    SUM(quantity) AS 'Total Qty'
FROM
    orders_Details
        JOIN
    pizzas ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizzas.size
ORDER BY COUNT(DISTINCT order_id) DESC;
```

Result Grid | Filter Rows:

size	No. of Orders	Total Qty
L	12736	18956
M	11159	15635
S	10490	14403
XL	544	552
XXL	28	28

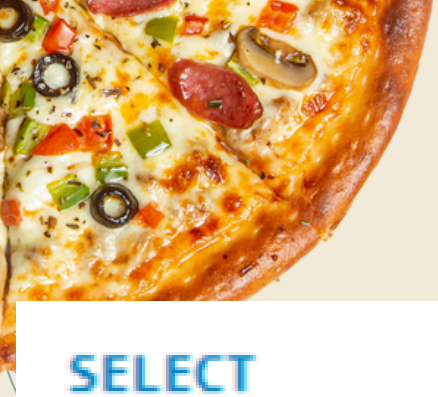


5.List the top 5 most ordered pizza types along with their quantities.

```sql
SELECT
    pizza_types.name, SUM(orders_details.quantity) AS Qty
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 Qty DESC
LIMIT 5;
```

Result Grid | Filter Rows:

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



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

```sql
SELECT
    pizza_types.name, SUM(orders_details.quantity) AS Qty
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 Qty DESC
LIMIT 5;
```

Result Grid | Filter Rows:

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

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

```sql
SELECT
    pizza_types.category AS 'Pizza type',
    SUM(orders_details.quantity) AS 'Total Qty'
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 'Total Qty' DESC;
```

Result Grid | Filter Rows:

Pizza type	Total Qty
Classic	14888
Veggie	11649
Supreme	11987
Chicken	11050

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

```sql
SELECT
    HOUR(time) AS Hour, COUNT(order_id) AS Order_Count
FROM
    orders
GROUP BY HOUR(time);
```

Result Grid | Filter Rows:

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

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

```sql
SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category;
```

Result Grid | Filter Row

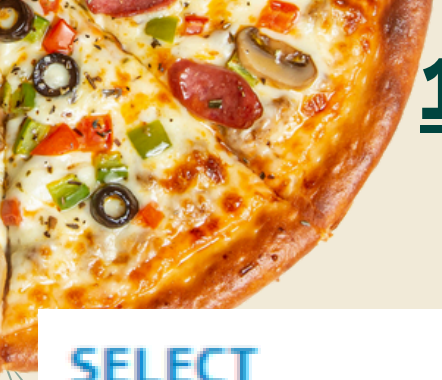
category	count(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

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

```sql
SELECT
    AVG(Qty)
FROM
    (SELECT
        orders.date, SUM(orders_details.quantity) AS Qty
    FROM
        orders
    JOIN orders_details ON orders.order_id = orders_details.order_id
    GROUP BY orders.date) AS Average_Qty;
```

Result Grid

	avg(Qty)
▶	138.4749



10.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:

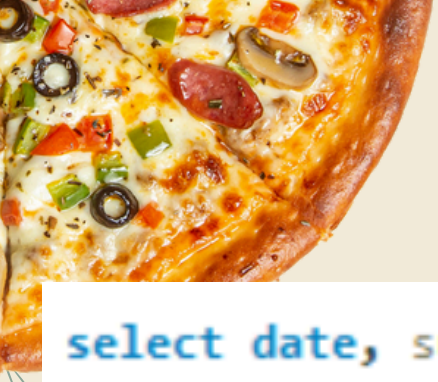
name	Revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

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

```sql
select pizza_types.category,
round( sum(orders_details.quantity*pizzas.price) / (select
round(sum(orders_details.quantity*pizzas.price), 2) as 'Total Sales'
from orders_details join pizzas on pizzas.pizza_id = orders_details.pizza_id) *100,2)
as 'Revenue'
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 desc;
```

Result Grid | Filter

category	Revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

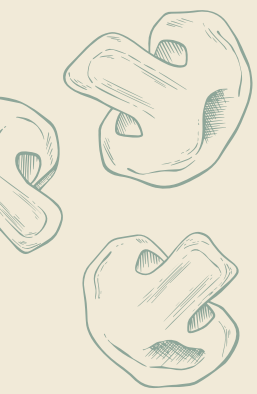


12.Analyze the cumulative revenue generated over time.

```sql
select date, sum(revenue) over (order by date) as 'Cumulative Revenue'
from
(select orders.date, sum(orders_details.quantity * pizzas.price) as 'revenue'
from orders_details join pizzas
on orders_details.pizza_id = pizzas.pizza_id
join orders on orders.order_id = orders_details.order_id
group by orders.date) as Sales;
```

Result Grid | Filter Rows:

	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
	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



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

```sql
select name, revenue from
(select category, name, Revenue,
Dense_Rank() over(partition by category order by Revenue desc) as 'Rank'
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 orders_details.pizza_id = pizzas.pizza_id
group by pizza_types.category, pizza_types.name) as a) as b
where 'Rank' <= 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 S	75
The Chicken Alfredo Pizza	16900.25
The Chicken Pesto Pizza	16701.75
The Classic Deluxe Pizza	38180.5
The Hawaiian Pizza	32273.25

The California Chicken Pizza