HELLO

My self-Harish in this project i used SQL query to analyze the pizza sales.

1. RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

SQL File 7* | SQL File 8* | SQL File 9* | SQL File 10* | SQL File 11*

Limit to 1000 rows

```
1   -- Retrive the total number of orders placed
2 • Select count(order_id) as total_orders from orders
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

total_orders
21350

2. CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

SQL File 7* | SQL File 8* | SQL File 9* | SQL File 10* | SQL File 11* | SQL File 12* | SQL Fi

Limit to 1000 rows

```sql
-- 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 pizzas.pizza_id = order_details.pizza_id
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

total_revenue
817860.05

3. IDENTIFY THE HIGHEST-PRICED PIZZA.

SQL File 7* | SQL File 8* | SQL File 9* | SQL File 10* | SQL File 11* | SQL File 12* | SQL File 13*

Limit to 1000 rows

```
1   -- Identify the highest-priced pizza.
2   select pizza_types.name,pizzas.price
3   from pizza_types join pizzas
4   on pizza_types.pizza_type_id = pizzas.pizza_type_id
5   order by pizzas.price desc limit 1
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

name	price
The Greek Pizza	35.95

4. IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

SQL File 7* | SQL File 8* | SQL File 9* | SQL File 10* | SQL File 11* | SQL File 12* | SQL File 13*

Limit to 1000 rows

```
1  -- Identify the most common pizza size ordered.
2  select pizzas.size, count(order_details.order_details_id) as order_count
3  from pizzas join order_details
4  on pizzas.pizza_id = order_details.pizza_id
5  group by pizzas.size order by order_count desc
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

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

5. LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

SQL File 7* | SQL File 8* | SQL File 9* | SQL File 10* | SQL File 11* | SQL File 12* | SQL File 13*

Limit to 1000 rows

```sql
-- 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
```

6. JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

SQL File 7* | SQL File 8* | SQL File 9* | SQL File 10* | SQL File 11* | SQL File 12* | SQL File 13*

Limit to 1000 rows

```
3 •  select sum(order_details.quantity) as total_quantity , pizza_types.category
4    from pizza_types join pizzas
5    on pizza_types.pizza_type_id = pizzas.pizza_type_id
6    join order_details
7    on order_details.pizza_id = pizzas.pizza_id
8    group by pizza_types.category order by total_quantity
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

total_quantity	category
11050	Chicken
11649	Veggie
11987	Supreme
14888	Classic

7. DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

SQL File 7* | SQL File 8* | SQL File 9* | SQL File 10* | SQL File 11* | SQL File 12* | SQL File 13* | SQL File 14*

Limit to 1000 rows

```sql
-- Intermediate
-- Determine the distribution of orders by hour of the day.
select hour(order_time) as hour, count(order_id) as oder_count from orders
group by hour(order_time);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

hour	oder_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 File 7* | SQL File 8* | SQL File 9* | SQL File 10* | SQL File 11* | SQL File 12* | SQL File 13* | SQL File 14*

Limit to 1000 rows

```
1   -- intermediate
2   -- Join relevant tables to find the category-wise distribution of pizzas
3   select category, count(name) from pizza_types
4   group by category
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

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 File 7* SQL File 8* SQL File 9* SQL File 10* SQL File 11* SQL File 12* SQL File 13* SQL File 14* SQL File 15*

Limit to 1000 rows

```
1   -- Group the orders by date and calculate the average number of pizzas ordered per day.
2
3
4   select round(avg(quantity),0) as avg_pizza_ordered_per_day from
5   (select orders.order_date, sum(order_details.quantity) as quantity
6   from orders join order_details
7   on orders.order_id = order_details.order_id
8   group by orders.order_date) as order_quantity
9
10
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

avg_pizza_ordered_per_day
138

10. DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

SQL File 7* | SQL File 8* | SQL File 9* | SQL File 10* | SQL File 11* | SQL File 12* | SQL File 13* | SQL File 14* | SQL File 15* | SQL File 16*

Limit to 1000 rows

```sql
-- 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 pizzas.pizza_type_id = pizza_types.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
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch 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 File 7* | SQL File 8* | SQL File 9* | SQL File 10* | SQL File 11* | SQL File 12* | SQL File 13* | SQL File 14* | SQL File 15* | SQL File 16* | SQL File 17*

Limit to 1000 rows

```sql
-- Calculate the percentage contribution of each pizza type to total revenue.
select pizza_types.category,
(sum(order_details.quantity*pizzas.price) / (select
round(sum(order_details.quantity*pizzas.price),2) as total_revenue
from order_details join pizzas
on pizzas.pizza_id = order_details.pizza_id))*100 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
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

category	revenue
Classic	26.90596025566967
Supreme	25.45631126009862
Chicken	23.955137556847287
Veggie	23.682590927384577

12. ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

SQL File 8* | SQL File 9* | SQL File 10* | SQL File 11* | SQL File 12* | SQL File 13* | SQL File 14* | SQL File 15* | SQL File 16* | SQL File 17* | SQL File 18*

Limit to 1000 rows

```sql
-- 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
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

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
2015-01-11	25862.65
2015-01-12	27781.7

Result Grid | Form Editor | Field Types | Read Only

Result 1

13. DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

SQL File 9* | SQL File 10* | SQL File 11* | SQL File 12* | SQL File 13* | SQL File 14* | SQL File 15* | SQL File 16* | SQL File 17* | SQL File 18* | SQL File 19*

Limit to 1000 rows

```
6   rank() over(partition by category order by revenue desc) as rn
7   from
8   (select pizza_types.category, pizza_types.name,
9   sum((order_details.quantity)*pizzas.price) as revenue
10  from pizza_types join pizzas
11  on pizza_types.pizza_type_id = pizzas.pizza_type_id
12  join order_details
13  on order_details.pizza_id = pizzas.pizza_id
14  group by pizza_types.category, pizza_types.name) as a) as b
15  where rn<=3
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

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

Result Grid | Form Editor | Field Types

HARISH R

ASPIRING DATA ANALYST

THANK YOU