

By - Guruprasad Vishwakarma

# PIZZA STORE ANALYSIS

## Using SQL queries

Select \* From ppt  
WHERE  
Basic SQL Questions  
Intermediate SQL Questions  
Advanced SQL Question ;

# CSV Datasets

## **pizza\_types**

- pizza\_type\_id
- name
- category
- ingredients

## **order\_details**

- order\_detail\_id
- order\_id
- pizza\_id
- quantity

## **orders**

- order\_id
- order\_date
- order\_time

## **Pizza**

- pizza\_id
- pizza\_type\_id
- size
- price

# Basic SQL Questions

- Retrieve the total number of orders placed.

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

Result Grid	
	total_Orders
▶	21350

- Calculate the total revenue generated from pizza sales.

```
select round(sum(orders_details.quantity * pizzas.price),2) as Total_revenue  
from orders_details join pizzas  
on pizzas.pizza_id = orders_details.pizza_id
```

Result Grid	
	Total_revenue
▶	817860.05

# Basic SQL Questions

- Identify the highest-priced pizza.

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

Result Grid		Filter Rows:
	name	price
▶	The Greek Pizza	35.95

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

```
select pizza_types.name,  
sum(orders_details.quantity) as quantity  
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 quantity desc limit 5;
```

Result Grid		Filter Rows:
	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

# Intermediate SQL Questions

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

```
select pizza_types.category,  
sum(orders_details.quantity) as Total_quantity  
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_quantity desc;
```

Result Grid		
	category	Total_quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

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

```
select hour(order_time) as Hours, count(order_id) as Count_of_ID  
from orders  
group by hour(order_time) limit 10;
```

Result Grid		
	Hours	Count_of_ID
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336
	18	2399
	19	2009
	20	1642

# Intermediate SQL Questions

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

```
select avg(Quantity) as Average_quantity from
(select orders.order_date, sum(orders_details.quantity) as Quantity
from orders join orders_details
on orders.order_id = orders_details.order_id
group by orders.order_date) as orders
```

Result Grid |

Average_quantity
138.4749

# Advanced SQL Questions

- 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		
	name	revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5

# Advanced SQL Questions

- 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

# Advanced SQL Questions

- 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		
	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
	2015-01-06	14358.5