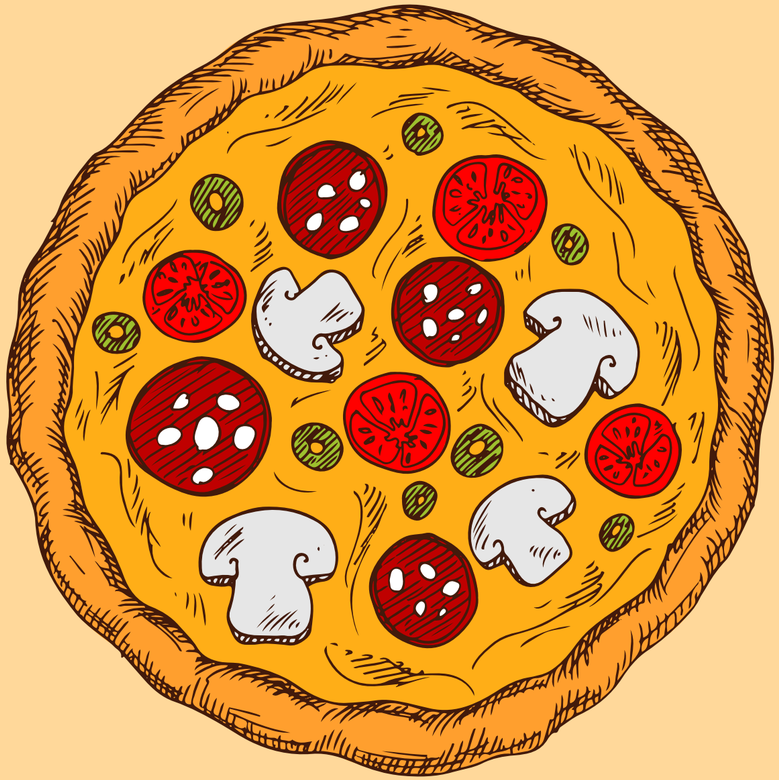


SQL PROJECT ON

PIZZA SALES

BY AHMAD ALI

SCHEMA OF DATASET

Order_details

	Field
1	order_details_id
2	order_id
3	pizza_id
4	quantity

Orders

	Field
1	order_id
2	order_date
3	order_time

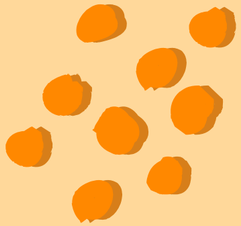
SCHEMA OF DATASET

Pizza_types

	Field
1	pizza_type_id
2	name
3	category
4	ingredients

Pizzas

	Field
1	pizza_id
2	pizza_type_id
3	size
4	price



EASY LEVEL QUERY

Q1: Retrieve the total number of orders placed.

```
3 •    SELECT
4          COUNT(order_id) AS total_orders
5      FROM
6          orders;
```

Result Grid

	total_orders
▶	21350

Q2: Calculate the total revenue generated from pizza sales.

```
3 •   SELECT
4         ROUND(SUM(order_details.quantity * pizzas.price),
5                 2) AS total_sales
6     FROM
7         order_details
8             JOIN
9         pizzas ON pizzas.pizza_id = order_details.pizza_id
```

Result Grid

total_sales
817860.05

Q3: Identify the highest-priced pizza.

```sql
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:

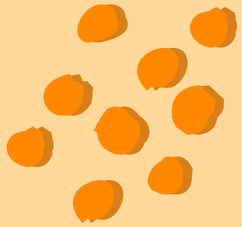
name	price
The Greek Pizza	35.95

Q4: Identify the most common pizza size ordered.

```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;
```

Result Grid | Filter Rows:

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



INTERMEDIATE LEVEL QUERY

Q5: Join the necessary tables to find the total quantity of each pizza category ordered.

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

Result Grid | Filter

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Q6: Group the orders by date and calculate the average number of pizzas ordered per day.

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

Result Grid | Filter Rows:

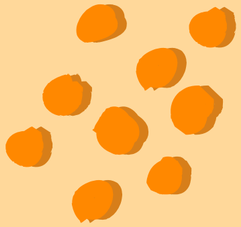
avg_pizza_ordered_per_day
138

Q7: Determine the top 3 most ordered pizza types based on revenue.

```sql
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:

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



ADVANCED LEVEL QUERY

Q8: Calculate the percentage contribution of each pizza type to total revenue.

```sql
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;
```

Result Grid | Filter

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

Q9: Analyze the cumulative revenue generated over time.

```
3  •   select order_date,
4      sum(revenue) over(order by order_date) as cum_revenue
5      from
6      (select orders.order_date,
7      sum(order_details.quantity * pizzas.price) as revenue
8      from order_details join pizzas
9      on order_details.pizza_id = pizzas.pizza_id
10     join orders
11     on orders.order_id = order_details.order_id
12     group by orders.order_date) as sales;
```

Result Grid | Filter Rows:

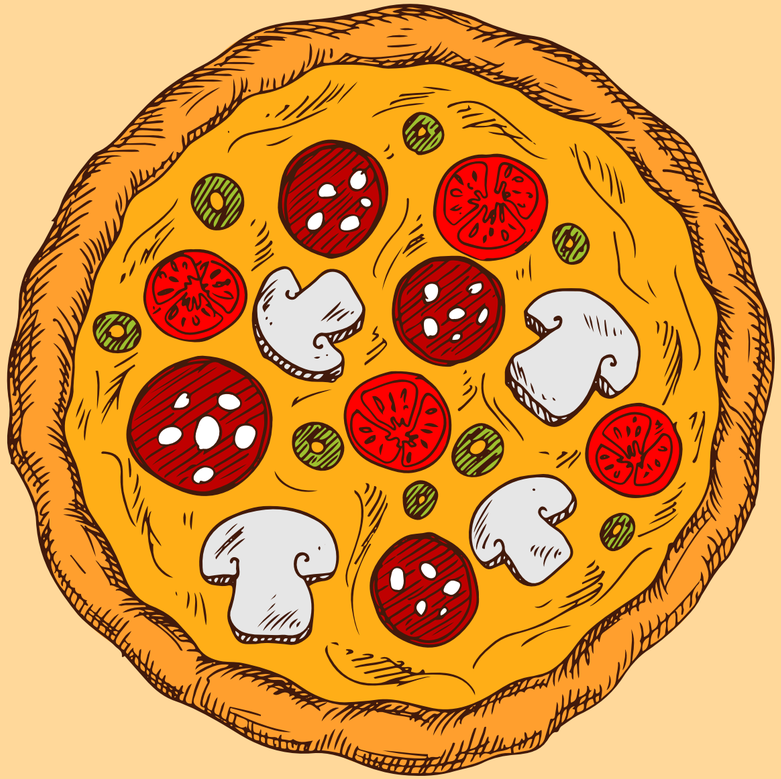
order_date	cum_revenue
2015-01-01	2713.850000000004
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55
2015-01-06	14358.5
2015-01-07	16560.7
2015-01-08	19399.05

Q10: Determine the top 3 most ordered pizza types based on revenue for each pizza category.

```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 a) as b
where rn <= 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 Classic Deluxe Pizza	38180.5
The Hawaiian Pizza	32273.25
The Pepperoni Pizza	30161.75



THANK YOU