PIZZA_SALES

SQL EXERCISES

INTRODUCTION:

This dataset simulates a pizza delivery business and is a hands-on exercise for practicing SQL in a business intelligence context. The slides cover SQL problems focused on **real-world scenarios** using JOINs, aggregations, window functions, and subqueries.

The goal was to enhance my skills in:

- Retrieving meaningful insights using various types of **JOINs**
- Performing data transformations and **aggregations**
- Ranking data with **RANK()**
- Writing clean, optimized SQL queries

DATASET OVERVIEW:

1.orders1

- Columns: order_id, date, time

2.order_details

- Columns: order_details_id, order_id, pizza_id, quantity

3. pizzas

- Columns: pizza_id, pizza_type_id, size, price

4.pizza_types

- Columns: pizza_type_id, name, category

1.ORDERS1

order_id	date	time
1	2015-01-01	11:38:36
2	2015-01-01	11:57:40
3	2015-01-01	12:12:28
4	2015-01-01	12:16:31
5	2015-01-01	12:21:30
6	2015-01-01	12:29:36
7	2015-01-01	12:50:37
8	2015-01-01	12:51:37
9	2015-01-01	12:52:01
10	2015-01-01	13:00:15
11	2015-01-01	13:02:59
12	2015-01-01	13:04:41
13	2015-01-01	13:11:55
14	2015-01-01	13:14:19
15	2015-01-01	13:33:00
16	2015-01-01	13:34:07

2.ORDER DETAILS

order_details_id	order_id	pizza_id	quantity
1	1	hawaiian_m	1
2	2	classic_dlx_m	1
3	2	five_cheese_l	1
4	2	ital_supr_l	1
5	2	mexicana_m	1
6	2	thai_ckn_l	1
7	3	ital_supr_m	1
8	3	prsc_argla_l	1
9	4	ital_supr_m	1
10	5	ital_supr_m	1
11	6	bbq_ckn_s	1
12	6	the_greek_s	1
13	7	spinach_supr_s	1
14	8	spinach_supr_s	1
15	9	classic_dlx_s	1
16	9	green_garde...	1

3.pizzas

pizza_id	pizza_type_id	size	price
bbq_ckn_s	bbq_ckn	S	12.75
bbq_ckn_m	bbq_ckn	M	16.75
bbq_ckn_l	bbq_ckn	L	20.75
cali_ckn_s	cali_ckn	S	12.75
cali_ckn_m	cali_ckn	M	16.75
cali_ckn_l	cali_ckn	L	20.75
ckn_alfredo_s	ckn_alfredo	S	12.75
ckn_alfredo_m	ckn_alfredo	M	16.75
ckn_alfredo_l	ckn_alfredo	L	20.75
ckn_pesto_s	ckn_pesto	S	12.75
ckn_pesto_m	ckn_pesto	M	16.75
ckn_pesto_l	ckn_pesto	L	20.75
southw_ckn_s	southw_ckn	S	12.75
southw_ckn_m	southw_ckn	M	16.75
southw_ckn_l	southw_ckn	L	20.75
thai_ckn_s	thai_ckn	S	12.75

4.PIZZA TYPES

pizza_type_id	name	category	ingredients
bbq_ckn	The Barbecue Chicken Pizza	Chicken	Barbecued Chicken, Red Peppers, Green Peppers, Toma
cali_ckn	The California Chicken Pizza	Chicken	Chicken, Artichoke, Spinach, Garlic, Jalapeno Peppers, F
ckn_alfredo	The Chicken Alfredo Pizza	Chicken	Chicken, Red Onions, Red Peppers, Mushrooms, Asiago
ckn_pesto	The Chicken Pesto Pizza	Chicken	Chicken, Tomatoes, Red Peppers, Spinach, Garlic, Pesto
southw_ckn	The Southwest Chicken Pizza	Chicken	Chicken, Tomatoes, Red Peppers, Red Onions, Jalapeno
thai_ckn	The Thai Chicken Pizza	Chicken	Chicken, Pineapple, Tomatoes, Red Peppers, Thai Swee
big_meat	The Big Meat Pizza	Classic	Bacon, Pepperoni, Italian Sausage, Chorizo Sausage
classic_dlx	The Classic Deluxe Pizza	Classic	Pepperoni, Mushrooms, Red Onions, Red Peppers, Baco
hawaiian	The Hawaiian Pizza	Classic	Sliced Ham, Pineapple, Mozzarella Cheese
ital_cpdlo	The Italian Capocollo Pizza	Classic	Capocollo, Red Peppers, Tomatoes, Goat Cheese, Garlic
napolitana	The Napolitana Pizza	Classic	Tomatoes, Anchovies, Green Olives, Red Onions, Garlic
pep_msh_pep	The Pepperoni, Mushroom, ...	Classic	Pepperoni, Mushrooms, Green Peppers

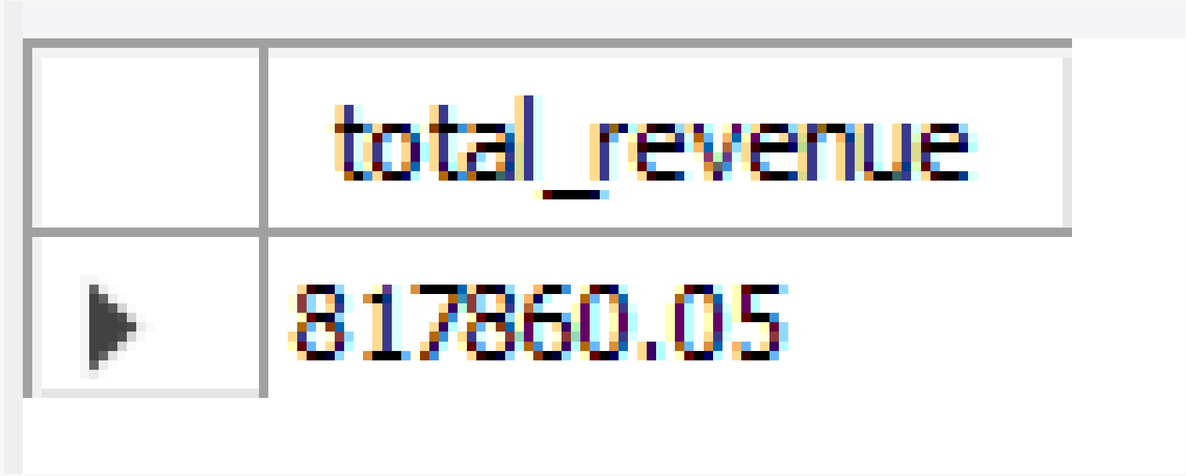
Q1:Retrieve the total number of orders placed.

```sql
select count(order_id) as total_orders
from orders1;
```

	total_orders
▶	21350

Q2:Calculate the total revenue generated from pizza sales.

```sql
select round(sum(order_details.quantity * pizzas.price),2) as total_revenue
from pizzas
join order_details
on pizzas.pizza_id = order_details.pizza_id
```



	total_revenue
▶	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;
```

name	price
The Greek Pizza	35.95

Q4:Identify the most common pizza size ordered.

```sql
SELECT sum(order_details.quantity) AS total_pizzas_sold, pizzas.size
FROM order_details
JOIN pizzas ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizzas.size
ORDER BY total_pizzas_sold DESC
limit 1;
```

total_pizzas_sold	size
18956	L

Q5:List the top 5 most ordered pizza types along with their quantities.

```sql
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;
```

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

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

```
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;
```

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Q7:Determine the distribution of orders by hour of the day.

```sql
SELECT
    EXTRACT(HOUR FROM time) AS hour_of_day,
    COUNT(order_id) AS order_count
FROM
    orders1
GROUP BY hour_of_day;
```

hour_of_day	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

Q8:Join relevant tables to find the category-wise distribution of pizzas.

```
select category,count(name) from pizza_types
group by category;
```

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

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

```sql
SELECT
    ROUND(AVG(pizzas_per_day), 0) AS avg_pizzas_per_day
FROM
    (SELECT
        orders1.date AS order_date,
            SUM(order_details.quantity) AS pizzas_per_day
    FROM
        orders1
    JOIN order_details ON orders1.order_id = order_details.order_id
    GROUP BY order_date) AS order_quantity;
```

avg_pizzas_per_day
138

Q10: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;
```

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

Q11:Calculate the percentage contribution of each pizza type to total revenue.

```sql
SELECT pizza_types.category, ROUND(SUM(order_details.quantity * pizzas.price) /
         (SELECT SUM(order_details.quantity * pizzas.price)
          FROM pizzas
          JOIN order_details ON pizzas.pizza_id = order_details.pizza_id) * 100, 2) AS revenue_percentage
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.category
ORDER BY revenue_percentage DESC;
```

category	revenue_percentage
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Q12:Analyze the cumulative revenue generated over time.

```sql
select order_date, sum(revenue) over(order by order_date) as cumulative_revenue
FROM
(SELECT orders1.date AS order_date, SUM(order_details.quantity*pizzas.price) AS revenue
FROM order_details
JOIN pizzas
ON order_details.pizza_id = pizzas.pizza_id
join orders1
on orders1.order_id = order_details.order_id
group by order_date) as sales
```

order_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

Q13:Determine the top 3 most ordered pizza types based on revenue for each pizza category.

```sql
select category,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 pizzas.pizza_id = order_details.pizza_id
group by pizza_types.category,pizza_types.name) as a) as b
where rn <=3;
```

Q13:Determine the top 3 most ordered pizza types based on revenue for each pizza category.

category	name	revenue
Chicken	The Thai Chicken Pizza	43434.25
Chicken	The Barbecue Chicken Pizza	42768
Chicken	The California Chicken Pizza	41409.5
Classic	The Classic Deluxe Pizza	38180.5
Classic	The Hawaiian Pizza	32273.25
Classic	The Pepperoni Pizza	30161.75
Supreme	The Spicy Italian Pizza	34831.25
Supreme	The Italian Supreme Pizza	33476.75
Supreme	The Sicilian Pizza	30940.5
Veggie	The Four Cheese Pizza	32265.70000000065
Veggie	The Mexicana Pizza	26780.75
Veggie	The Five Cheese Pizza	26066.5

Thank you