Pizza Sales Analysiss

This project analyzes pizza sales data using SQL, solving multiple real-world business problems across three difficulty levels: **Basic**, **Intermediate**, and **Advanced**. The project uses 4 tables:

1. pizzas
2. pizza_types
3. orders
4. order_details

```
create database pizza_data_analysis;

use pizza_data_analysis;

select * from pizzas;
select * from pizza_types;
select * from orders;
select * from order_details;
```

Question_1 : Retrieve the total number of orders placed.

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

Question_2 : Calculate the total revenue generated from pizza sales.

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

```
select pi.name, max(price) as highest_price from pizzas as p
inner join pizza_types pi on
p.pizza_type_id = pi.pizza_type_id
group by pi.name order by highest_price desc;
```

Question_3 : Identify the highest-priced pizza.

```
select top 1 pt.name, round(max(p.price),2) as highest_price
from pizzas as p inner join pizza_types as pt
on p.pizza_type_id = pt.pizza_type_id
group by pt.name order by highest_price desc;
```

Question_4 : Identify the most common pizza size ordered.

```
select top 1 p.size, count(od.quantity) as pizza_count
from pizzas as p inner join order_details as od
on p.pizza_id = od.pizza_id
group by p.size order by pizza_count desc;
```

Question_5 : List the top 5 most ordered pizza types along with their quantities.

```sql
select top 5 pt.name, sum(od.quantity) as quantity_count
from pizzas as p join pizza_types as pt
on p.pizza_type_id = pt.pizza_type_id
join order_details as od on p.pizza_id = od.pizza_id
group by pt.name, p.size order by quantity_count desc;
```

Question_6 : Join the necessary tables to find the total quantity of each pizza category ordered.

```sql
select pt.category as category, sum(od.quantity) as total_quantity
from order_details as od join pizzas as p
on
p.pizza_id = od.pizza_id join pizza_types as pt
on
p.pizza_type_id = pt.pizza_type_id
group by category order by total_quantity desc;
```

Question_7 : Determine the distribution of orders by hour of the day.

```sql
select  DATEPART(hour, time) as total_hour, count(order_id) as total_order  from orders
group by DATEPART(hour, time) order by total_order desc;
```

Question_8 : Join relevant tables to find the category-wise distribution of pizzas.

```sql
select pt.category, count(od.quantity) as total_count
from pizzas as p inner join order_details as od
on p.pizza_id = od.pizza_id inner join pizza_types as pt
on p.pizza_type_id = pt.pizza_type_id
group by pt.category order by total_count desc;
```

Question_9 : Group the orders by date and calculate the average number of pizzas ordered per day.

```sql
select  avg(total_pizza) as average_pizzas_ordered from
       (select o.date, sum(od.quantity) as total_pizza  from orders as o
inner join order_details as od on
o.order_id = od.order_id group by o.date) as a
```

Question_10 Determine the top 3 most ordered pizza types based on revenue.

```sql
select top 3 pt.name, round(sum(p.price*od.quantity),0) as revenue  from pizzas as p
inner join pizza_types as pt on p.pizza_type_id = pt.pizza_type_id
inner join order_details as od on
p.pizza_id = od.pizza_id group by pt.name order by revenue desc;
```

Question_11 : Calculate the percentage contribution of each pizza type to total revenue.

```sql
select category, round((revenue/pizza_count)*100,2) as precentage_revenue from
```

```
(select pt.category,sum(sum(od.quantity*p.price)) over() as pizza_count,
round(sum(p.price*od.quantity),0) as revenue
from pizzas as p inner join pizza_types as pt on
p.pizza_type_id = pt.pizza_type_id inner join order_details as od
on p.pizza_id = od.pizza_id group by pt.category) as precent order by precentage_revenue
desc;
```

Question_12 : Analyze the cumulative revenue generated over time.

```
select date, sum(revenue) over(order by date ) as cum_revenue from
       (select o.date, round(sum(p.price*od.quantity),2) revenue from pizzas as p
inner join order_details as od on
p.pizza_id = od.pizza_id inner join orders o on
o.order_id = od.order_id group by o.date) as cum_rev;
```

Question_13 : Determine the top 3 most ordered pizza types based on revenue for each pizza category.

```
select name,category,  revenue from
(select category, name, revenue, rank()
over(partition by category order by revenue desc)  as net_Revenue from
       (select pt.name, pt.category, round(sum(p.price*od.quantity),2) as revenue
from pizzas as p inner join pizza_types as pt on p.pizza_type_id = pt.pizza_type_id
inner join order_details as od on p.pizza_id = od.pizza_id
group by pt.category, pt.name) as top_3) as most_ordered
where net_Revenue<=3;
```