SQL views and queries

- ## Views

#Question 1. Retrieve the total number of orders placed.

SELECT * FROM total_number_of_orders;

#Question 2.Calculate the total revenue generated from pizza sales

SELECT * FROM total_revenue_generated_from_pizza_sales;

#Question 3.Identify the most common pizza size ordered

SELECT * FROM most_common_pizza_size ;

#Question 4. List the top 5 most ordered pizza types along with their quantities

SELECT * FROM top_5_most_ordered_pizzas;

#Question 5. Determine the distribution of orders by hour of the day.

SELECT * FROM orders_by_hour;

#Question 6. Find the category-wise distribution of pizzas.

SELECT * FROM category_wise_distribution_of_pizzas;

#Question 7. Group the orders by date and calculate the average number of pizzas ordered per day

SELECT * FROM avg_number_of_pizza_order_per_day;

#Question 8. Determine the top 3 most ordered pizza types based on revenue

SELECT * FROM top_3_by_revenue;

#Question 9. Calculate the percentage contribution of each pizza type to total revenue

SELECT * FROM percent_revenue_of_pizza_by_total_revenue;

#Question 10. Determine the top 3 most ordered pizza types based on revenue for each pizza category

SELECT * FROM top_3_by_revenue_category;

- ## Queries

1.CREATE VIEW total_number_of_orders AS

```
  SELECT
  COUNT(order_id) AS total_orders
  FROM
  orders;
2.CREATE VIEW total_revenue_generated_from_pizza_sales AS
  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;
3.CREATE VIEW most_common_pizza_size AS
  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;


4.CREATE VIEW top_5_most_ordered_pizzas AS
  SELECT
  pizza_types.name, SUM(order_details.quantity) AS qauntity
  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 qauntity DESC
  LIMIT 5;
5. CREATE VIEW orders_by_hour AS
  SELECT
  HOUR(order_time), COUNT(order_id)
  FROM
  orders
  GROUP BY HOUR(order_time);

6.CREATE VIEW category_wise_distribution_of_pizzas AS
  SELECT
  category, COUNT(name)
  FROM
  pizza_types
  GROUP BY category;
7.CREATE VIEW avg_number_of_pizza_order_per_day AS
  SELECT
  AVG(quantity)
  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;

8.CREATE VIEW top_3_by_revenue AS
  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;
9.CREATE VIEW percent_revenue_of_pizza_by_total_revenue AS
  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;
10.create view top_3_by_revenue_category as
  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;
```