PIZZAHUT SALES

USING WITH SQL

VIVEK VISHWAKARMA

VIVEK VISHWAKARMA

BASIC QUESTIONS

01 Retrieve the total number of orders placed.

02 Calculate the total revenue generated from pizza sales.

03 Identify the highest-priced pizza.

04 Identify the most common pizza size ordered

05 List the top 5 most ordered pizza types along with their quantities.
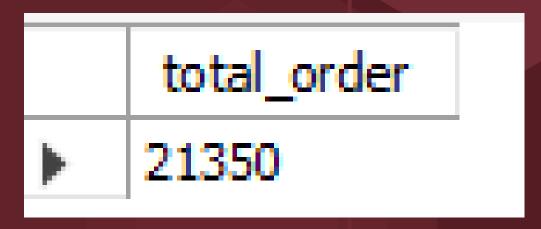
INTERMEDIATE QUESTIONS

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

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

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

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

05 Determine the top 3 most ordered pizza types based on revenue.
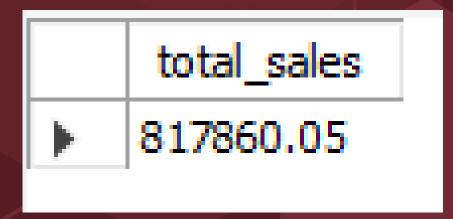
VIVEK VISHWAKARMA

ADVANCED QUESTIONS

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

02 Analyze the cumulative revenue generated over time.

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

01 Retrieve the total number of orders placed.

```
1    -- Retrieve the total number of orders placed.
2 ●  select count(Order_id) as total_order from orders;
```

RESULT

total_order
21350

02 Calculate the total revenue generated from pizza sales.

```sql
1        -- Calculate the total revenue generated from pizza sales.
2  ●   SELECT
3          ROUND(SUM(order_details.quantity * pizzas.price),
4                  2) AS total_sales
5      FROM
6          order_details
7              JOIN
8          pizzas ON pizzas.pizza_id = order_details.pizza_id
```

RESULT

total_sales
817860.05

03 Identify the highest-priced pizza.

```sql
1    -- Identify the highest-priced pizza.
2 •  SELECT
3        pizza_types.name, pizzas.price
4    FROM
5        pizza_types
6            JOIN
7        pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
8    ORDER BY pizzas.price DESC
9    LIMIT 1;
```

RESULT

name	price
The Greek Pizza	35.95

04 Identify the most common pizza size ordered

```sql
1    -- Identify the most common pizza size ordered.
2  • SELECT
3        pizzas.size,
4        COUNT(order_details.order_details_id) AS order_count
5    FROM
6        pizzas
7            JOIN
8        order_details ON pizzas.pizza_id = order_details.pizza_id
9    GROUP BY pizzas.size
10   ORDER BY order_count DESC;
```

RESULT

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

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

```sql
1       -- List the top 5 most ordered pizza types
2       -- along with their quantities.
3       SELECT
4           pizza_types.name, SUM(order_details.quantity) AS quantity
5       FROM
6           pizza_types
7               JOIN
8           pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
9               JOIN
10          order_details ON order_details.pizza_id = pizzas.pizza_id
11      GROUP BY pizza_types.name
12      ORDER BY quantity DESC
13      LIMIT 5;
```

RESULT

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

```sql
1    -- Join the necessary tables to find the
2    -- total quantity of each pizza category ordered.
3    SELECT
4        pizza_types.category,
5        SUM(order_details.quantity) AS quantity
6    FROM
7        pizza_types
8            JOIN
9        pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
10           JOIN
11       order_details ON order_details.pizza_id = pizzas.pizza_id
12   GROUP BY pizza_types.category
13   ORDER BY quantity DESC;
```

RESULT

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

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

```sql
1      -- Determine the distribution of orders by hour of the day.
2  •   select hour(order_time) as hour, count(order_id)as order_count
3      from orders
4      group by hour(order_time);
```

RESULT

hour	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

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

```sql
1        -- Join relevant tables to find the
2        -- category-wise distribution of pizzas.
3   ●    select category, count(name) from pizza_types
4        group by category;
```

RESULT

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

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

```sql
1        -- Group the orders by date and calculate the average number of pizzas ordered per day.
2  ●     SELECT
3            ROUND(AVG(quantity), 0) as avg_pizza_per_day
4        FROM
5            (SELECT
6                orders.order_date, SUM(order_details.quantity) AS quantity
7            FROM
8                orders
9            JOIN order_details ON orders.Order_id = order_details.Order_id
10           GROUP BY orders.order_date) AS oder_quantity;
```

RESULT

avg_pizza_per_day
▶ 138

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

```sql
1     -- Determine the top 3 most ordered pizza types based on revenue.
2  ●  SELECT
3         pizza_types.name,
4         SUM(order_details.quantity * pizzas.price) AS revenue
5     FROM
6         pizza_types
7             JOIN
8         pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
9             JOIN
10        order_details ON order_details.pizza_id = pizzas.pizza_id
11    GROUP BY pizza_types.name
12    ORDER BY revenue DESC
13    LIMIT 3;
```

RESULT

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

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

```sql
-- 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;
```

RESULT

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

02 Analyze the cumulative revenue generated over time.

```sql
1    -- Analyze the cumulative revenue generated over time.
2    select  order_date,
3    sum(revenue) over(order by order_date) as cum_revenue
4    from
5    (select orders.order_date,
6    sum(order_details.quantity*pizzas.price) as revenue
7    from order_details join pizzas
8    on order_details.pizza_id = pizzas.pizza_id
9    join orders
10   on orders.Order_id = order_details.Order_id
11   group by orders.order_date) as sales;
```

RESULT

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
2015-01-07	16560.7
2015-01-08	19399.05

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

```sql
1       -- Determine the top 3 most ordered pizza types based on revenue for each pizza category.
2
3  •    select name, revenue from
4       (select category, name, revenue,
5       rank() over (partition by category order by revenue desc) as rn
6       from
7       (select pizza_types.category, pizza_types.name,
8       sum(order_details.quantity*pizzas.price) as revenue
9       from pizza_types join pizzas
10      on pizza_types.pizza_type_id = pizzas.pizza_type_id
11      join order_details
12      on order_details.pizza_id = pizzas.pizza_id
13      group by pizza_types.category, pizza_types.name) as a) as b
14      where rn<=3;
```

RESULT

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
The Spicy Italian Pizza	34831.25
The Italian Supreme Pizza	33476.75

THANKS

Thank you so much for taking the time to visit my portfolio. I truly appreciate your interest in my work. Your feedback and support mean a lot to me as I continue to grow and develop my skills.

Please feel free to reach out if you have any questions or if there's anything specific you'd like to discuss. I look forward to staying connected.

Best regards,

VIVEK VISHWAKAMA