SQL Project for Data Analysis: PizzaHut Database

Introduction

Hello, I'm Aqsa, a data science enthusiast . As a statistical data analyst, I thrive on extracting meaningful insights from data. In this project, I've applied my expertise to analyze data from a fictional pizza chain, "Pizzahut," using MySQL.

Through this project, I aim to showcase my ability to ,perform complex SQL queries, and derive valuable business insights...

About database

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

Pizza Types

pizza_type_id	name	category	ingredients
bbq_ckn	The Barbecue Chicken Pizza	Chicken	Barbecued Chicken, R
cali_ckn	The California Chicken Pizza	Chicken	Chicken, Artichoke, Sp
ckn_alfredo	The Chicken Alfredo Pizza	Chicken	Chicken, Red Onions,

orders

order_id	order_date	order_time
1	2015-01-01	11:38:36
2	2015-01-01	11:57:40
3	2015-01-01	12:12:28

order_details

order_details_id	order_id	pizza_id	quantity
675	298	cali_ckn_m	1
676	298	cali_ckn_s	1
677	298	ckn_alfredo_m	1

Create Tables

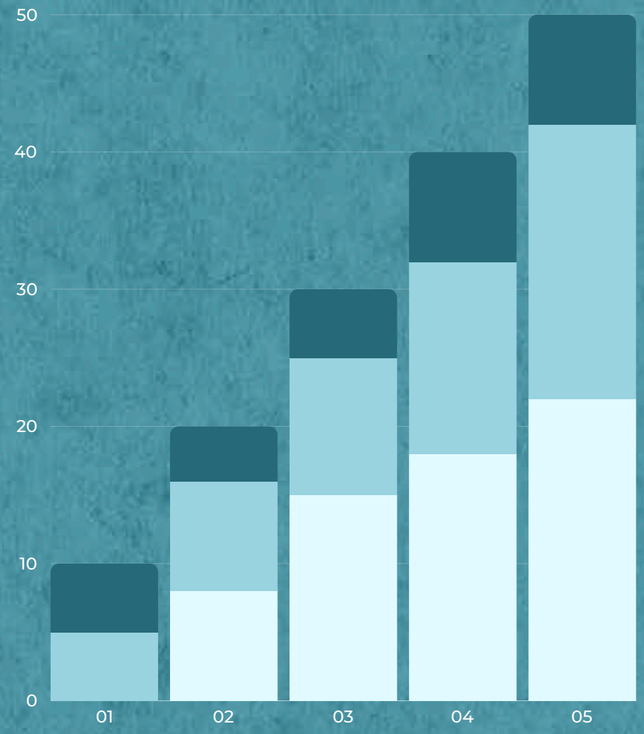
```sql
create database pizzahut;
-- now import csv file pizzas
-- right click on table and import wizard then browse file from folder
-- import pizzas--
-- import pizza_types--
-- for importing orders.csv we need time datatype ,
--  which is not availible so we need to creat this table
create table orders(
    order_id   int  not null,
    order_date  date not null,
    order_time  time not null,
    primary key(order_id)
);
```

```sql
-- 4th column
create table order_details(
    order_details_id int not null,
    order_id   int  not null,
    pizza_id  text  not null,
    quantity  int not null ,
    primary key(order_details_id)
);

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

Questions

```sql
35    -- 1 -- Select the total number of orders place
36 •  select count(order_id) as total_numbers from orders ;
37    -- 21350
38
39    -- 2- select the total revenue
40 •  SELECT
41        round(SUM(order_details.quantity * pizzas.price), 2) as total_sales
42        -- Assuming you want to calculate total price
43    FROM
44        order_details
45    JOIN
46        pizzas ON pizzas.pizza_id = order_details.pizza_id;
47     -- 817860.04999999993 alter round -- 817860.05
48
```



Questions

```sql
-- 3-- highest prise pizza
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;
-- the Greek pizza  35.95--

-- 4-- identify the most common pizza size orders

select  pizzas.size,  count(order_details.order_details_id) as total_orders
-- If you are using count or any other aggregate function, you need to use group by along with it
from pizzas
join order_details on pizzas.pizza_id = order_details.pizza_id
group by  pizzas.size order by total_orders desc;

```

Questions

```
66
67    -- 5-- top 5 most ordered pizzas types along with their quantities/
68    -- no common column in both tables , so need to use third column'
69    -- pizza_types has pizza_type_id which is common in pizzas,
70    --  pizzas has pizza_id which is common in order_details
71
72 •  select pizza_types.name , sum(order_details.quantity) as total
73    from pizza_types join pizzas on pizza_types.pizza_type_id = pizzas.pizza_type_id
74    join order_details on order_details.pizza_id = pizzas.pizza_id
75    group by pizza_types.name order by total desc ;
76    -- suggesting to create a table here that contains the name and the total.
77
```

Questions

```sql
-- 1--Join the necessary tables to find the quantity of each pizza Category ordered

select  pizza_types.category, sum(order_details.quantity) as total_orders
from order_details join pizzas on order_details.pizza_id = pizzas.pizza_id
join pizza_types on pizza_types.pizza_type_id = pizzas.pizza_type_id
group by pizza_types.category order by  total_orders desc;

-- 2-- determine the distribution of the orders by the hour of the day
select hour(order_time), count(order_id) from orders
group by hour(order_time) order by order_id; -- order by na b kry tou output aye gi


-- 3-- join relevant table to find category wise distribution of pizzas
select pizza_types.category , count(name) from pizza_types
group by  pizza_types.category;

```

Questions

```
98    -- 4-- Group the orders by date and calculate the average number of pizzas ordered per day--
99    select orders.order_date, count(order_details.quantity) as detail
100   from orders join order_details on order_details.order_id = orders.order_id
101   group by orders.order_date;
102
103   -- Now need to check for one day
104   select round(avg( detail), 0) as average_pizza_ordered
105    from (select orders.order_date, count(order_details.quantity) as detail
106   from orders join order_details on order_details.order_id = orders.order_id
107   group by orders.order_date) as order_quantity;
108
109
```

Questions

```
110     -- 5-- top 3 most ordered pizza types based on revenue
111     -- revenue is quantity * price
112
113 •   select pizza_types.name , sum(order_details.quantity * pizzas.price) as revenue
114     from pizza_types join pizzas on pizza_types.pizza_type_id = pizzas.pizza_type_id
115     join order_details on order_details.pizza_id = pizzas.pizza_id
116     group by  pizza_types.name order by revenue desc limit 3 ;
117
```

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

Questions

```sql
-- -- 1-- contribution of each pizza type in total revenue in percentage
SELECT
    pizza_types.category,
    round(SUM(order_details.quantity * pizzas.price) / (SELECT
        SUM(order_details.quantity * pizzas.price)
        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) * 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;
```

Answer

	category	revenue
▶	Veggie	23.68
	Chicken	23.96
	Supreme	25.46
	Classic	26.91

Question

```sql
-- 2-- Analyze the  cumulative  revenue generated over time
-- income of on day | cumulative income
-- 200                          200
-- 300                          200+300
-- 400                          200+300+400
SELECT
    sales.order_date,  -- order_date b just likh skty
    SUM(sales.revenue) OVER (ORDER BY sales.order_date) AS cumulative
FROM
    (SELECT
        orders.order_date,
        SUM(order_details.quantity * pizzas.price) AS revenue
     FROM
        order_details
     JOIN pizzas ON order_details.pizza_id = pizzas.pizza_id
     JOIN orders ON order_details.order_id = orders.order_id
     GROUP BY orders.order_date
    ) AS sales;
```

Answer

order_date	cumulative
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

Question

```sql
-- 3-- determine top 3 most ordered pizza types based on revenue for each pizza category
-- revenue for each pizza category and inside category each type

select category, name, revenue ,
rank()  over(partition by category order by revenue desc) as b
from
(select pizza_types.category, 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.category, pizza_types.name) a ;
```

Question

```sql
-- for Conditiion b<=3

select name , revenue from
(select category, name, revenue ,
rank()  over(partition by category order by revenue desc) as b
from
(select pizza_types.category, 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.category, pizza_types.name) a) as c
where b<=3  ;
```

Answer

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

Contact

aqsasamreen92@gmail.com

https://www.linkedin.com/in/aqsasumreen/

Thank you!