PIZZA SALE DATA ANALYSIS BY MYSQL

Pramod Kumar Sahoo

```sql
SELECT * FROM pizzahut.orders;
```

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

orders 1

```sql
SELECT * FROM pizzahut.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

order_details 1

```sql
SELECT * FROM pizzahut.pizza_types;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

pizza_type_id	name	category	ingredients
bbq_ckn	The Barbecue Chicken Pizza	Chicken	Barbecued Chicken, Red Peppers, Green Peppe...
cali_ckn	The California Chicken Pizza	Chicken	Chicken, Artichoke, Spinach, Garlic, Jalapeno P...
ckn_alfredo	The Chicken Alfredo Pizza	Chicken	Chicken, Red Onions, Red Peppers, Mushrooms...
ckn_pesto	The Chicken Pesto Pizza	Chicken	Chicken, Tomatoes, Red Peppers, Spinach, Garl...
southw_ckn	The Southwest Chicken Pizza	Chicken	Chicken, Tomatoes, Red Peppers, Red Onions, ...
thai_ckn	The Thai Chicken Pizza	Chicken	Chicken, Pineapple, Tomatoes, Red Peppers, T...

pizza_types 1

```sql
SELECT * FROM pizzahut.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

pizzas 1

```sql
-- Retrieve the total number of orders placed.

select * FROM orders;
select count(order_id) as total_orders from orders;
```

Result Grid

total_orders
21350

Function used ;

- SQL COMMAND
- AGGREGATE FUNCTION –COUNT
- ALLIAS FUNCTION

```sql
-- Calculate the total revenue generated from pizza sales
```

Open a script file in this editor

```sql
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
```

Result Grid

total_sales
817860.05

Function used ;

- SQL COMMAND
- AGGREGATE FUNCTION- SUM
- JOIN FUNCTION
- ALLIAS FUNCTION

```sql
-- Identify the highest-priced 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;
```

Result Grid | Filter Rows

name	price
The Greek Pizza	35.95

```sql
-- Identify the most common pizza size ordered.

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;
```

Result Grid | Filter Rows

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

Function used ;

- SQL COMMAND
- JOIN FUNCTION
- OPERATE FUNCTION- ORDERBY
- LIMIT FUNCTION

Function used ;

- SQL COMMAND
- OPERATER – ORDERBY,GROUP BY
- AGGREGATE FUNCTION- COUNT
- JOIN FUNCTION

```sql
-- List the top 5 most ordered pizza types along with their quantities.

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;
```

Result Grid | Filter Rows:

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

Function used ;

- SQL COMMAND
- AGGREGATE FUNCTION- SUM
- JOIN FUNCTION
- OPERATER – ORDERBY,GROUP BY
- LIMIT FUNCTION

```sql
-- 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

Function used ;

- SQL COMMAND
- AGGREGATE FUNCTION- SUM
- JOIN FUNCTION
- OPERATER – ORDERBY,GROUP BY

```sql
-- Determine the distribution of orders by hour of the day.

select hour(time) as hour , count(order_id) as order_count
from orders group by hour(time);
```

Result Grid | Filter Rows

hour	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920

Result 1

Function used ;

- SQL COMMAND
- TIME FUNCTION
- OPERATOR- GROUP BY
- AGGREGATE FUNCTION- COUNT

```sql
-- join relevant tables to find the category-wise distribution of pizzas

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

Result Grid | Filter Ro

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

Function used ;

- SQL COMMAND
- OPERATOR- GROUP BY
- AGGREGATE FUNCTION- COUNT

```sql
-- group the orders by date and calculate the average number of pizzas ordered per day.

select round( avg(quantity),0) as avg_pizza_ordered_per_day from
(select orders.date,sum(order_details.quantity) as quantity
from orders join order_details
on orders.order_id = order_details.order_id
group by orders.date) as order_quantity;
```

Result Grid | Filter Rows:

avg_pizza_ordered_per_day
138

Function used ;

- SQL COMMAND
- ALLIAS FUNCTION
- AGGREGATE FUNCTION- SUM
- JOIN FUNCTION
- OPERATOR – GROUP BY

```sql
select pizza_types.category,
(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 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;
```

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

Result Grid | Filter Rows:

category	revenue
Classic	26.90596025566967
Supreme	25.45631126009862
Chicken	23.955137556847287
Veggie	23.682590927384577

Function used ;

- AGGREGATE FUNCTION- SUM, ROUND
- JOIN FUNCTION
- OPERATOR- GROUPBY, ORDER BY

```sql
-- Determine the top 3 most ordered pizza types based on revenue.

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;
```

Result Grid | Filter Rows:

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

Function used ;

- SQL COMMAND
- AGGREGATE FUNCTION – SUM
- ALLIAS FUNCTION
- JOIN FUNCTION
- OPERATOR – GROUP BY

```sql
-- Determine the top 3 most ordered pizza types based on revenue for each pizza category.

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 ;
```

Result Grid | Filter Rows:

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

Result 1

Function used ;

- SQL COMMAND
- JOIN FUNCTION
- ALLIAS FUNCTION
- CLAUSE FUNCTION- WHERE
- RANK AND PARTITION FUNCTION