super delicious

PIZZA

Canva

ORDER NOW

Hello!

- I'm Neha Reddy,
- I've leveraged SQL queries to conduct an in-depth analysis of pizza sales data.

Database Brief

Database Name: Dominos

order details table

orders table

pizza types

pizzas

CASE STUDY

Basic

Retrieve the total number of orders placed.

Calculate the total revenue generated from pizza sales.

Identify the highest-priced pizza.

Identify the most common pizza size ordered.

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

Intermediate:

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

Determine the distribution of orders by hour of the day.

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

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

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

Advanced:

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

Analyze the cumulative revenue generated over time.

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

QL File 8* | order_details | orders | pizza_types | pizzas

Limit to 1000 rows

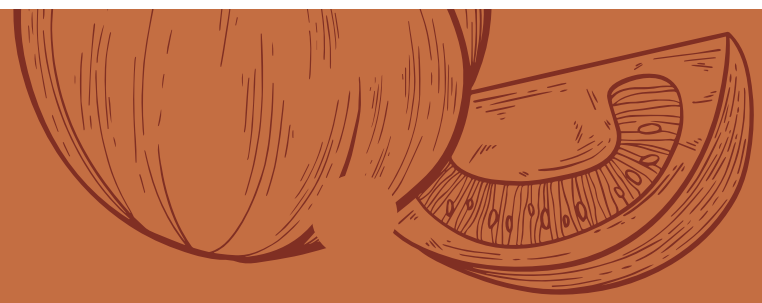
```sql
create table order_details (
order_details_id int Primary key,
order_id int not null,
pizza_id text not null,
quantity int not null
);

-- Retrieve the total number of orders placed.
select
count(order_id) as Total_orders
from
dominos.orders;

```

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

Total_orders
21350



SQL File 8* | order_details | pizzas | orders | pizza_types | SQL File 9*

Limit to 1000 rows

```sql
-- Calculate the total revenue generated from pizza sales.
select
round (sum((order_details.quantity*pizzas.price)) ,2) as total_Revenue
from
order_details
join pizzas
on
order_details.pizza_id=pizzas.pizza_id;
```

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

total_Revenue
817860.05

Result 6

SQL File 8* | order_details | pizzas | orders | pizza_types | SQL File 9* | SQL File 10*

Limit to 1000 rows

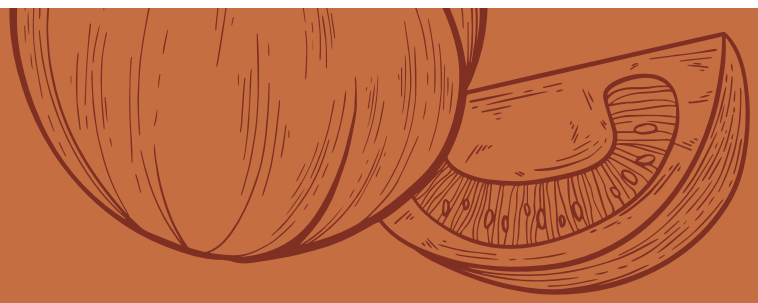
```sql
-- Identify the highest-priced pizza.--

select
pizzas.price,
pizza_types.name
from pizzas
join
pizza_types
on
pizzas.pizza_type_id=pizza_types.pizza_type_id
order by pizzas.price desc limit 1 ;

```

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

price	name
35.95	The Greek Pizza



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

select
COUNT( order_details.order_id) AS order_count,
pizzas.size
from order_details
join pizzas
on
order_details.pizza_id=pizzas.pizza_id
GROUP BY pizzas.size;
```

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

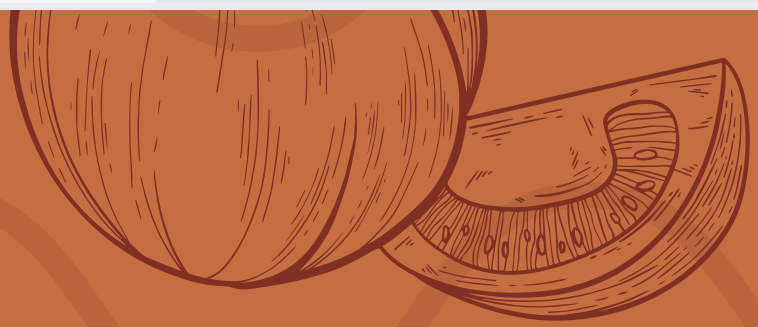
order_count	size
18526	L
15385	M
14137	S
544	XL
28	XXL

```sql
-- List the top 5 most ordered pizza types along with their quantities.
select
sum((order_details.quantity)) as quantity,
pizza_types.name
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: | Export: | Wrap Cell Content: | Fetch rows:

quantity	name
2453	The Classic Deluxe Pizza
2432	The Barbecue Chicken Pizza
2422	The Hawaiian Pizza
2418	The Pepperoni Pizza
2371	The Thai Chicken Pizza

Result 12



order_details | pizza_types | pizzas | orders | SQL File 8* | SQL File 9* | SQL File 10* | SQL File 11* | SQL File

Limit to 1000 rows

```sql
-- Join the necessary tables to find the total quantity of each pizza category ordered.
select
sum(order_details.quantity),
pizza_types.category
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;
```

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

sum(order_details.quantity)	category
14888	Classic
11649	Veggie
11987	Supreme
11050	Chicken

orders | order_details | pizzas | pizza_types | SQL File 8* | SQL File 9* | SQL File 10* | SQL File 11* | SQL File 12* | S

Limit to 500 rows

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

select * from orders;

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

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

Hours	order_count
9	1
10	8
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009

Result 4

pizza_types | orders | order_details | pizzas | SQL File 8* | SQL File 9* | SQL File 10* | SQL File 11* | SQL File 12* | SQL File 13* | SQL File 14

Limit to 500 rows

```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 Rows: | Export: | Wrap Cell Content:

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

s orders order_details pizzas SQL File 8* SQL File 9* SQL File 10* SQL File 11* SQL File 12* SQL File 13* SQL F

Limit to 500 rows

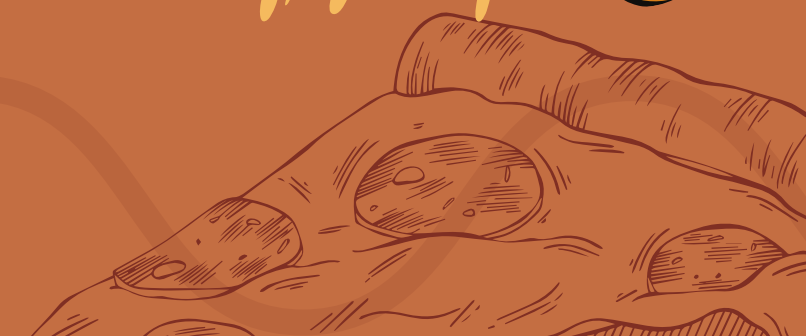
```sql
-- Group the orders by date and calculate the average number of pizzas ordered per day.

select round( Avg (quantity),0) as avg_pizza_perday from
(select
orders.order_date,
round (sum(order_details.quantity),2) as quantity
from orders
join order_details
on orders.order_id=order_details.order_id
group by orders.order_date) as order_quantity;
```

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

avg_pizza_perday
138

Result 16



SQL File 11* | SQL File 12* | SQL File 13* | SQL File 10* | SQL File 14* | SQL File 15* | SQL File 16* | pizza_types | pizzas | order_details | orders | SQL File 17*

Limit to 500 rows

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


select
pizza_types.name as pizza_name,
sum((pizzas.price*order_details.quantity)) as sales
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 sales desc limit 3;
```

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

pizza_name	sales
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

Result 8

SQL File 12* | SQL File 13* | SQL File 10* | SQL File 14* | SQL File 15* | SQL File 16* | SQL File 17* | pizza_types | pizzas | order_details | orders | SQL File 18*

Limit to 500 rows

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

select
pizza_types.name as pizza_name,
sum((pizzas.price*order_details.quantity)/817860.05)*100 as sales
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;
```

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

pizza_name	sales
The Hawaiian Pizza	3.9460602092007817
The Classic Deluxe Pizza	4.668341484585189
The Five Cheese Pizza	3.1871589766489445
The Italian Supreme Pizza	4.093212524563356
The Mexicana Pizza	3.2744905439505936

Result 12 | Read Only

Output

SQL File 13* | SQL File 10* | SQL File 14* | SQL File 15* | SQL File 16* | SQL File 17* | SQL File 18* | orders | pizzas | order_details | pizza_types | SQL File 19*

Limit to 500 rows

```sql
-- Analyze the cumulative revenue generated over time.

select
order_date,
sum(sales) over(order by order_date) as cum_sales
from
(select
orders.order_date,
sum(order_details.quantity*pizzas.price) as sales
from
orders join order_details
on orders.order_id=order_details.order_id
join pizzas
on order_details.pizza_id=pizzas.pizza_id
group by order_date) as day_sales;
```

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

order_date	cum_sales
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
2015-01-09	21526.4

Result Grid | Form Editor

Result 7 | Read Only

SQL File 10* | SQL File 14* | SQL File 15* | SQL File 16* | SQL File 17* | SQL File 18* | SQL File 19* | orders | pizzas | order_details | pizza_types | SQL File 20*

Limit to 500 rows

```sql
-- Determine the top 3 most ordered pizza types based on revenue for each pizza category.
select
category, name, revn
from
(select category, name, revn,
rank() over (partition by category order by revn desc) as Rnk
from
(select
pizza_types.category,
pizza_types.name,
sum((order_details.quantity)* pizzas.price) as revn
from pizzas
join pizza_types
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 table1) as table2
where rnk<=3;
```

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

category	name	revn
Chicken	The Thai Chicken Pizza	43434.25
Chicken	The Barbecue Chicken Pizza	42768
Chicken	The California Chicken Pizza	41409.5
Classic	The Classic Deluxe Pizza	38180.5
Classic	The Hawaiian Pizza	32273.25

Result Grid

Thankyou

Please Leave a Feedback @

nehareddy1004@gmail.com

Why was the pizza happy? It was feeling cheesy!