

# PIZZA SALES QUERY

--- create table ---

```
create table pizza_sales
(
  pizza_id int8 primary key,
  order_id int8 not null,
  pizza_name_id varchar(100),
  quantity int8 not null,
  order_date date,
  order_time time,
  unit_price decimal,
  total_price decimal,
  pizza_size varchar(100),
  pizza_category varchar(100),
  pizza_ingredients varchar(100),
  pizza_name varchar(100)
)
```

--- import data ---

```
copy pizza_sales (pizza_id, order_id, pizza_name_id, quantity, order_date, order_time, unit_price,
                  total_price, pizza_size, pizza_category, pizza_ingredients, pizza_name)
from 'D:\SKD\Data Analyst\3. SQL\4. Project\4. Pizza Sales\pizza_sales.csv'
delimiter ';'
csv header
```

--- size and details of dataset ---

```
select * from pizza_sales
select count(*) as rows from pizza_sales
select count(*) as columns from information_schema.columns where table_name = 'pizza_sales'
select column_name, data_type from information_schema.columns where table_name='pizza_sales'
```

--- problem statement ---

-- KPI's requirement (we need to analyze key indicators from our pizza sales data to gain insights into our  
-- business performance. specifically, we want to calculate the following metrics)

-- 1. total revenue - the sum of total price of all pizza orders

```
select
    sum(total_price) as total_revenue
from pizza_sales
```

-- 2. average order value - the average amount spent per order, calculated by dividing the total revenue by  
-- the total number of orders

```
select
    round(sum(total_price) / (select count (distinct order_id)), 2) as average_order_value
from pizza_sales
```

-- 3. total pizzas sold - the sum of quantities of all pizzas sold

```
select  
    sum(quantity) as total_pizzas_sold  
from pizza_sales
```

-- 4. total orders - total number of orders placed

```
select  
    count(distinct order_id) as total_orders  
from pizza_sales
```

-- 5. average pizzas per order - the average number of pizzas sold per order, calculated by dividing the total number of pizzas sold by the total number of orders

```
select  
    round(sum(quantity) / (select count(distinct order_id)), 2) as average_pizzas_per_order  
from pizza_sales
```

-- charts requirement - we would like to visualize various aspects of our pizza sales data to gain insights and understand key trends. we have identified the following requirements for creating charts, but first we will find the values in SQL

-- 1. daily trend for total orders - this will help us to identify any patterns or fluctuations in order volumes on daily basis

```
select  
    to_char(order_date, 'day') as order_day,  
    count(distinct order_id) as total_orders  
from pizza_sales  
group by order_day  
order by total_orders desc
```

-- 2. monthly trend for total orders - this will help us to identify any pattern or fluctuations in order volumes on monthly basis.

```
select  
    to_char(order_date, 'month') as order_month,  
    count (distinct order_id) as total_orders  
from pizza_sales  
group by order_month  
order by total_orders desc
```

-- 3. percentage of sales by pizza category - this will provide insights into the popularity of various pizza categories and their contribution to overall sales.

```
select  
    pizza_category,  
    round(sum(total_price) / (select sum(total_price) from pizza_sales) * 100, 2) as pct  
from pizza_sales  
group by pizza_category  
order by pct desc
```

-- 4. percentage of sales by pizza size - this will help us understand customer preferences for pizza sizes and  
-- their impact on sales

```
select
    pizza_size,
    round(sum(total_price) / (select sum(total_price) from pizza_sales) * 100, 2) as pct
from pizza_sales
group by pizza_size
order by pct desc
```

-- 5. total pizzas sold by pizza category - this will allow us to compare the sales performance of different  
-- pizza categories

```
select
    pizza_category,
    sum(total_price) as total_revenue
from pizza_sales
group by pizza_category
order by total_revenue desc
```

-- 6. top 5 pizzas sold by pizza name - this will allow us to compare the sales performance of the most  
-- popular pizzas

```
select
    pizza_name,
    sum(total_price) as total_revenue
from pizza_sales
group by pizza_name
order by total_revenue desc
limit 5
```

-- 7. bottom 5 pizzas sold by pizza name - this will allow us to compare the sales performance of the  
-- underperforming pizzas

```
select
    pizza_name,
    sum(total_price) as total_revenue
from pizza_sales
group by pizza_name
order by total_revenue
limit 5
```

-- 8. top 5 pizzas by quantity - this will allow us to compare the performance of the most popular pizzas  
-- by quantity

```
select
    pizza_name,
    sum(quantity) as total_quantity
from pizza_sales
group by pizza_name
order by total_quantity desc
limit 5
```

-- 9. bottom 5 pizzas by quantity - this will allow us to compare the performance of the least popular pizzas

-- by quantity

```
select
    pizza_name,
    sum(quantity) as total_quantity
from pizza_sales
group by pizza_name
order by total_quantity
limit 5
```

-- 10. top 5 pizzas by order id - this will allow us to compare the performance of the most popular pizzas

-- by order id

```
select
    pizza_name,
    count(distinct order_id) as total_orders
from pizza_sales
group by pizza_name
order by total_orders desc
limit 5
```

-- 11. bottom 5 pizzas by order - this will allow us to compare the performance of the least popular pizzas

-- by order id

```
select
    pizza_name,
    count(distinct order_id) as total_orders
from pizza_sales
group by pizza_name
order by total_orders
limit 5
```