

KPI

```
SELECT SUM(total_price) FROM pizza_sales;
```

| | |
|---|------------------|
| ⋮ | SUM(total_price) |
| | 817860.05 |

```
SELECT SUM(total_price) / COUNT(DISTINCT order_id) as Avg_Order_Value from pizza_sales;
```

| | |
|---|-------------------|
| ⋮ | Avg_Order_Value |
| | 38.30726229508197 |

```
SELECT SUM(quantity) as Total_Pizza_Sold from pizza_sales;
```

| | |
|---|------------------|
| ⋮ | Total_Pizza_Sold |
| | 49574 |

```
SELECT COUNT(DISTINCT order_id) as Total_Orders from pizza_sales;
```

| | |
|---|--------------|
| ⋮ | Total_Orders |
| | 21350 |

```
SELECT ROUND(1.0 * SUM(quantity) / COUNT(DISTINCT order_id), 2) AS Vvg_Pizzas_Per_order  
FROM pizza_sales;
```

| | |
|---|---------------------|
| ⋮ | avg_items_per_order |
| | 2.32 |

– Daily Trend

```
SELECT
CASE strftime('%w', order_date)
WHEN '0' THEN 'Sun'
WHEN '1' THEN 'Mon'
WHEN '2' THEN 'Tue'
WHEN '3' THEN 'Wed'
WHEN '4' THEN 'Thu'
WHEN '5' THEN 'Fri'
WHEN '6' THEN 'Sat'
END AS order_day,
COUNT(DISTINCT order_id) AS total_orders
FROM pizza_sales
GROUP BY order_day
ORDER BY CAST(strftime('%w', order_date) AS INTEGER);
```

but actually, if i hadn't worked at sqlite, I could use

```
SELECT DATENAME(dw, order_date) as order_day, COUNT(DISTINCT order_id) AS
Total_orders
from pizza_sales;
GROUP BY DATENAME(dw, order_date);
```

| order_day | total_orders |
|-----------|--------------|
| Sun | 2624 |
| Mon | 2794 |
| Tue | 2973 |
| Wed | 3024 |
| Thu | 3239 |
| Fri | 3538 |
| Sat | 3158 |

– Hourly Trend

```
SELECT DATEPART(HOUR, order_time) AS Order_Hours, COUNT(DISTINCT order_id)
AS Total_orders
from pizza_sales;
GROUP BY DATEPART(HOUR, order_time);
```

OR

```
SELECT
CAST(strftime('%H', order_time) AS INTEGER) AS order_hour,
COUNT(DISTINCT order_id) AS total_orders
```

```
FROM pizza_sales  
GROUP BY order_hour  
ORDER BY order_hour;
```

| order_hour | total_orders |
|------------|--------------|
| 9 | 1 |
| 10 | 8 |
| 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 |

```
SELECT  
pizza_category,  
SUM(total_price) as Total_Sales,  
ROUND(SUM(total_price) * 100.0 / (SELECT SUM(total_price) FROM pizza_sales), 2) AS  
percent_share  
FROM pizza_sales  
GROUP BY pizza_category;
```

| pizza_category | Total_Sales | percent_share |
|----------------|-------------|---------------|
| Chicken | 195919.5 | 23.96 |
| Classic | 220053.1 | 26.91 |
| Supreme | 208197 | 25.46 |
| Veggie | 193690.45 | 23.68 |