••

```
SHOW DATABASES;

USE food_db;

SELECT * FROM food_sales LIMIT 1000;

-- Total Revenue done by Store

SELECT ROUND(SUM(total_price),2) as Revenue
FROM food_sales;

-- Total Pizzas sold

SELECT SUM(quantity) as total_pizzas_sold
FROM food_sales;

-- Total Orders placed

SELECT COUNT(DISTINCT order_id) as total_orders
FROM food_sales;

-- Average order value

SELECT ROUND(SUM(total_price) / COUNT(DISTINCT order_id),2) as
average_order_value
FROM food_sales;

-- average pizza per order
SELECT ROUND(SUM(quantity) / COUNT(DISTINCT order_id),2) as
average_pizza_per_order
FROM food_sales;

-- Daily trend for sales order
SELECT order_date, ROUND(SUM(total_price),2) as Daily_sales
FROM food_sales
```

```
GROUP BY order_date
ORDER BY order_date ASC;

-- Daily trend for sales and orders
DESC food_sales; -- order date is in text, convert str to date

SELECT DAYNAME(str_to_date(order_date,"%d-%m-%Y")) as day_name,
COUNT(DISTINCT order_id) as total_orders,
ROUND(SUM(total_price),2) as Daily_sales
FROM food_sales
GROUP BY day_name
ORDER BY total_orders DESC, Daily_sales DESC;

-- monthly sales and orders trend
SELECT MONTHNAME(str_to_date(order_date,"%d-%m-%Y")) as month_name,
-- YEAR(str_to_date(order_date,"%d-%m-%Y")) as year_date,
COUNT(DISTINCT order_id) as total_orders,
ROUND(SUM(total_price),2) as Daily_sales
FROM food_sales
GROUP BY month_name
ORDER BY str_to_date(order_date,"%d-%m-%Y") ASC;

-- Percentage of sales by pizza category
SELECT pizza_category,
ROUND((SUM(total_price) / (SELECT SUM(total_price) from food_sales))*100,2)
as percent_sales
FROM food_sales
GROUP BY pizza_category
ORDER BY pizza_category ASC;

-- Percentage of sales by pizza size
SELECT pizza_size,
ROUND((SUM(total_price) / (SELECT SUM(total_price) from food_sales))*100,2)
as percent_sales
FROM food_sales
GROUP BY pizza_size
ORDER BY pizza_size ASC;
```

```sql
-- top 5 best seller by revenue
SELECT pizza_name, ROUND(SUM(total_price),2) as revenue
FROM food_sales
GROUP BY pizza_name
ORDER BY revenue DESC
LIMIT 5;

-- bottom 5 best seller by revenue
SELECT pizza_name, ROUND(SUM(total_price),2) as revenue
FROM food_sales
GROUP BY pizza_name
ORDER BY revenue ASC
LIMIT 5;

-- top 5 by qty
SELECT pizza_name, SUM(quantity) as Qty
FROM food_sales
GROUP BY pizza_name
ORDER BY Qty DESC
LIMIT 5;

-- bottom 5 by qty
SELECT pizza_name, SUM(quantity) as Qty
FROM food_sales
GROUP BY pizza_name
ORDER BY Qty ASC
LIMIT 5;

-- most ordered pizzas
SELECT pizza_name, COUNT(distinct order_id) as orders
FROM food_sales
GROUP BY pizza_name
ORDER BY orders DESC
LIMIT 5;

-- least ordered pizzas
SELECT pizza_name, COUNT(distinct order_id) as orders
FROM food_sales
```

```
GROUP BY pizza_name
ORDER BY orders ASC
LIMIT 5;

-- total sales during weekdays and week ends
SELECT
CASE DAYNAME(str_to_date(order_date,"%d-%m-%Y"))
WHEN "Saturday" THEN "Weekend"
   WHEN "Sunday" THEN "Weekend"
   ELSE "Weekday"
END as Day_type,
COUNT(DISTINCT order_id) as total_orders,
ROUND(SUM(total_price),2) as Daily_sales
FROM food_sales
GROUP BY Day_type
ORDER BY total_orders DESC, Daily_sales DESC;

-- ------------------------------------------------------------------------
----RW

-- select all limit 1000
-- SELECT * from food_sales LIMIT 1000;

-- SELECT DISTINCT(YEAR(str_to_date(order_date,"%d-%m-%Y")))
-- FROM food_sales;

-- SELECT day(order_date) as day_name
-- FROm food_sales;
-- SELECT FORMAT(order_date,"dd-mm-yyyy") as date_
-- from food_sales
-- GROUP BY order_date;


-- DROP VIEW abc;
-- CREATE VIEW abc AS
-- (SELECT order_id, AVG(total_price) as AVGE
-- FROM food_sales
-- group by order_id) ;
```

```
-- SELECT AVG(AVGE) FROM abc;
```