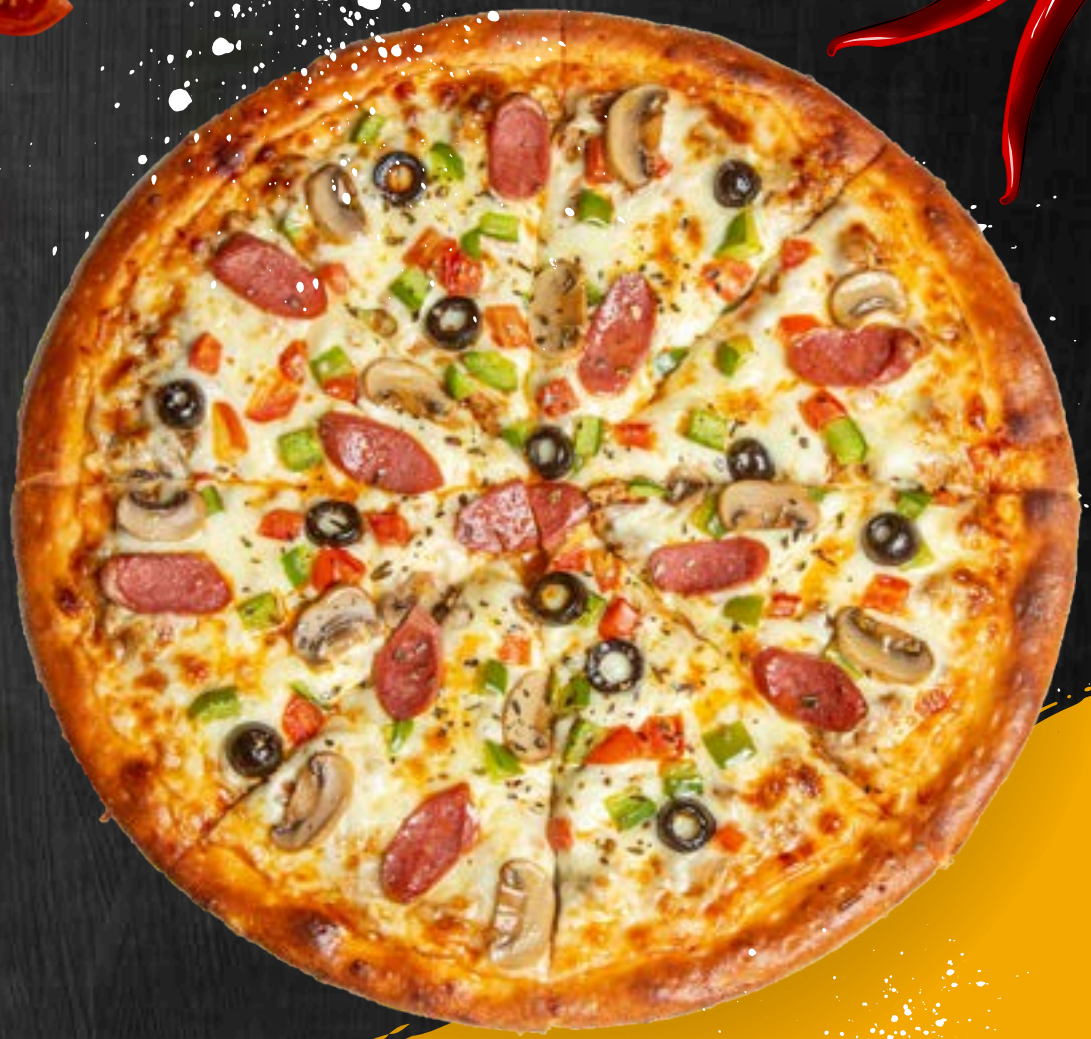
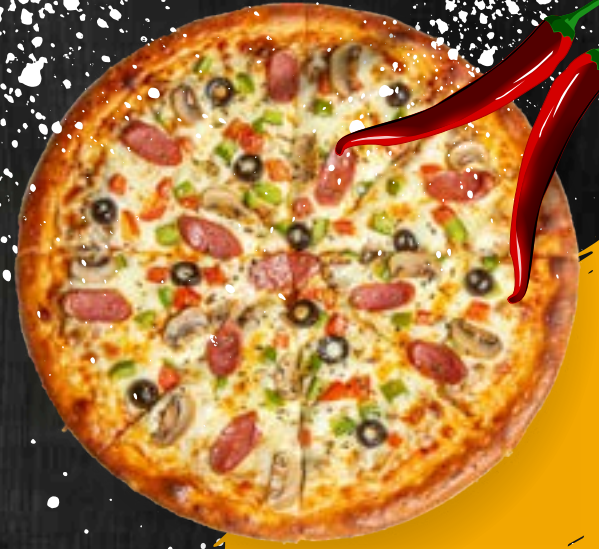
PIZZA SALES



DATA ANALYST

SQL - PIZZA-SALES PROJECT

PIZZA SALES

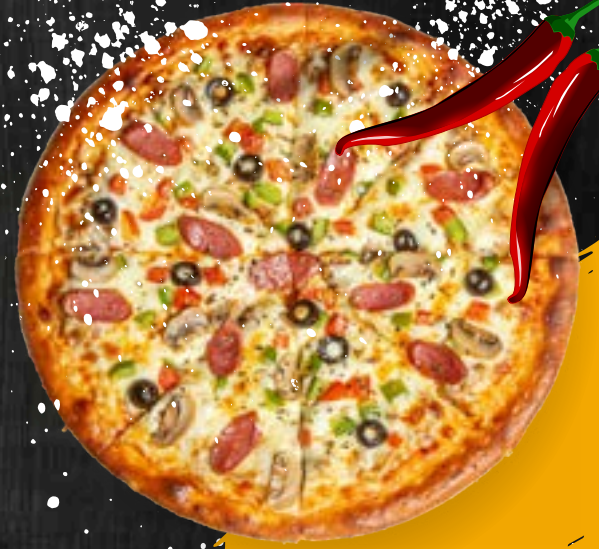


HOW TO CREATE A TABLE IN SQL

I HAVE CREATED THIS TABLE FOR ORDERS

```
create table orders(
order_id int not null,
order_date date not null,
order_time time not null,
primary key(order_id));
```

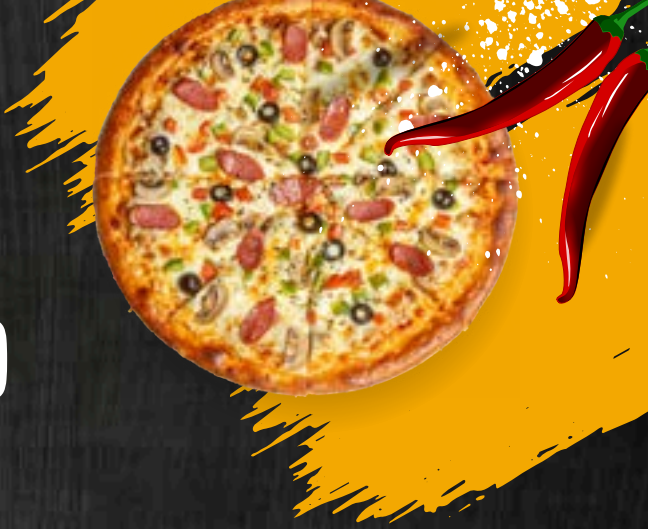
PIZZA SALES



THIS TABLE I HAVE CREATED FOR ORDERS_DETAILS TABLE

```
create table orders_details(
order_details_id int not null,
order_id int not null,
pizza_id text not null,
quantity int not null,
primary key(order_details_id));
```

PIZZA SALES

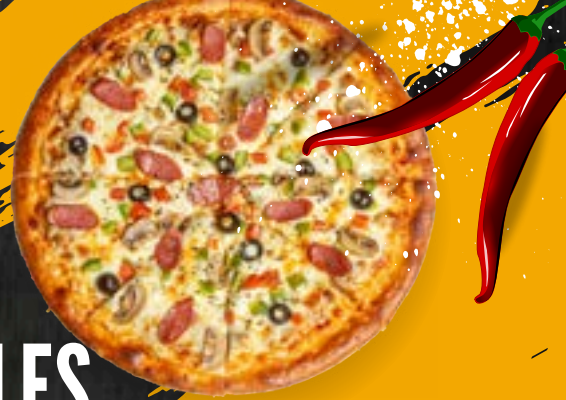


RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED

```
SELECT
    COUNT(order_id) AS Total_Order
FROM
    orders;
```

Total_Order
21350

PIZZA SALES

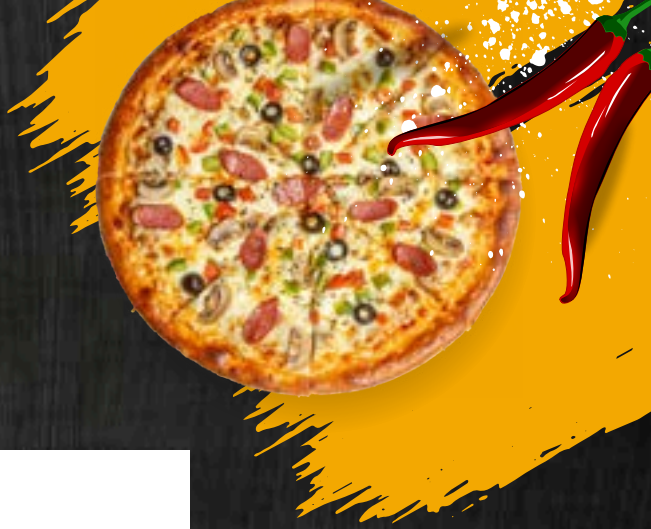


CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```sql
SELECT
    ROUND(SUM(orders_details.Quantity * Pizzas.Price),
            2) AS Total_Ravanue
FROM
    Pizzas
        JOIN
    orders_details ON orders_details.pizza_id = Pizzas.pizza_id;
```

	Total_Ravanue
▶	817860.05

PIZZA SALES

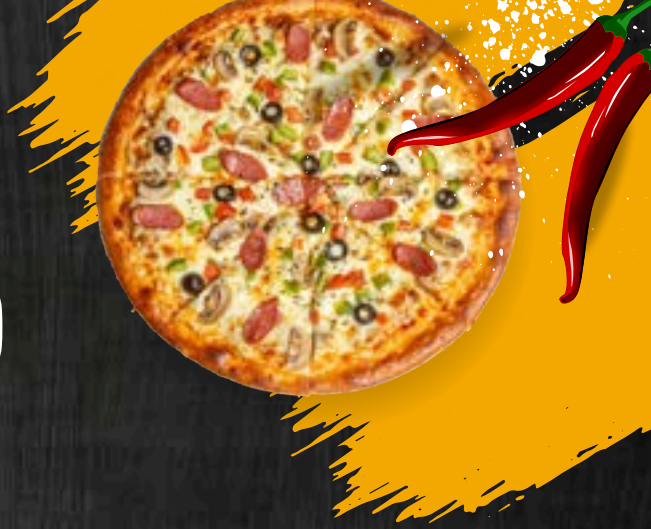


IDENTIFY THE HIGHEST-PRICED PIZZA.

```
SELECT
    pizza_types.name, pizzas.price
FROM
    pizzas
        JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

	name	price
▶	The Greek Pizza	35.95

PIZZA SALES

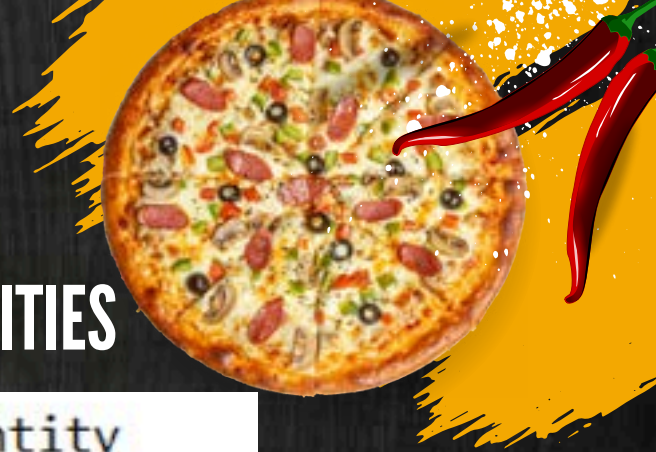


IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED

```sql
SELECT
    pizzas.size,
    COUNT(orders_details.order_details_id) AS order_count
FROM
    pizzas
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC;
```

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

PIZZA SALES

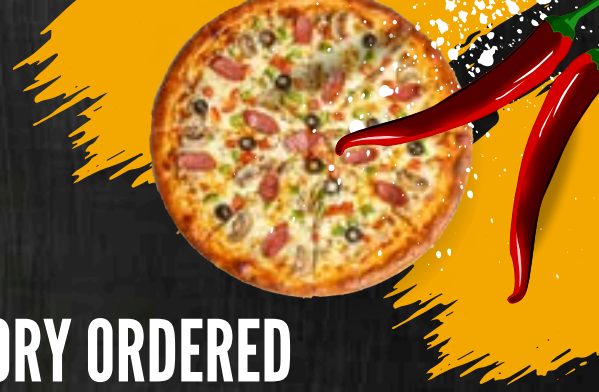


LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES

```
    pizza_types.name, SUM(orders_details.quantity) AS Quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
```

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

PIZZA SALES



JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED

```sql
SELECT
    pizza_types.category,
    SUM(orders_details.Quantity) AS Total_quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizza_types.category
ORDER BY Total_quantity DESC;
```

category	Total_quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

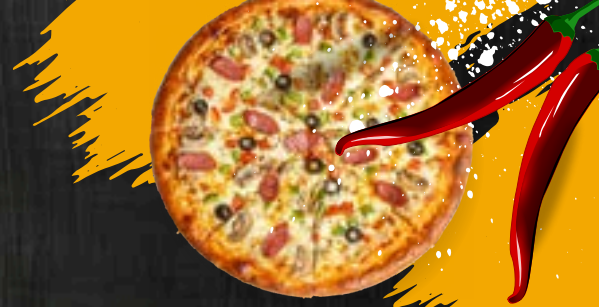
PIZZA SALES

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY

```
SELECT
    HOUR(order_time) AS Hour, COUNT(order_id) AS count_order
FROM
    orders
GROUP BY HOUR(order_time);
```

Hour	count_order
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642

PIZZA SALES

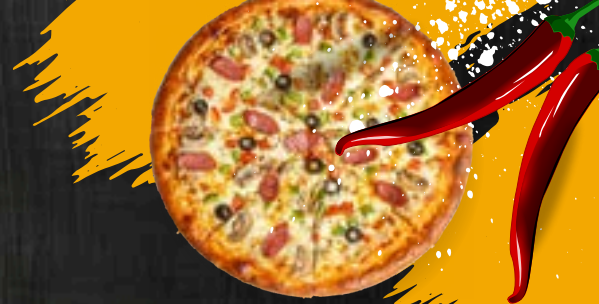


JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS

```sql
SELECT
    category, COUNT(name) count_pizza
FROM
    pizza_types
GROUP BY category;
```

category	count_pizza
Chicken	6
Classic	8
Supreme	9
Veggie	9

PIZZA SALES



GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIORDERED PER DAY

```sql
SELECT
    ROUND(AVG(Total_order), 0) Avg_Pizza_Orderd_Per_Day
FROM
    (SELECT
        orders.order_date,
            SUM(orders_details.Quantity) AS Total_order
    FROM
        orders
    JOIN orders_details ON orders.order_id = orders_details.order_id
    GROUP BY order_date) AS Pizza_order_data;
```

Avg_Pizza_Orderd_Per_Day
138

PIZZA SALES

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE

```sql
SELECT
    pizza_types.name,
    SUM(orders_details.quantity * pizzas.price) AS Total_Ravanue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizza_types.name
ORDER BY Total_Ravanue DESC
LIMIT 3;
```

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

PIZZA SALES

CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE

```sql
SELECT
    pizza_types.category,
    Round((SUM(orders_details.quantity * pizzas.price) / (SELECT
    ROUND(SUM(orders_details.Quantity * Pizzas.Price),
            2) AS Total_Ravanue
FROM
    Pizzas
        JOIN
    orders_details ON orders_details.pizza_id = Pizzas.pizza_id))*100,2) as Total_Ravanue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizza_types.category
ORDER BY Total_Ravanue DESC;
```

category	Total_Ravanue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

PIZZA SALES

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```sql
select order_date,
sum(Revenue) over (order by order_date) as Cum_Rvevenu
from
(select orders.order_date,
sum(orders_details.Quantity*pizzas.price) as Revenue
from orders_details
join pizzas
on orders_details.pizza_id = pizzas.pizza_id
join orders
on orders.order_id  = orders_details.order_id
group by orders.order_date) as Sales;
```

order_date	Cum_Rvevenu
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
2015-01-10	23990.350000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.300000000003
2015-01-14	32358.700000000004
2015-01-15	34343.50000000001
2015-01-16	36937.65000000001
2015-01-17	39001.75000000001
2015-01-18	40978.600000000006
2015-01-19	43365.75000000001
2015-01-20	45763.65000000001
2015-01-21	47804.20000000001
2015-01-22	50300.90000000001
2015-01-23	52724.600000000006
2015-01-24	55013.850000000006
2015-01-25	56631.40000000001
2015-01-26	58515.80000000001
2015-01-27	61043.85000000001
2015-01-28	63059.85000000001
2015-01-29	65105.150000000016
2015-01-30	67375.45000000001
2015-01-31	69793.30000000002