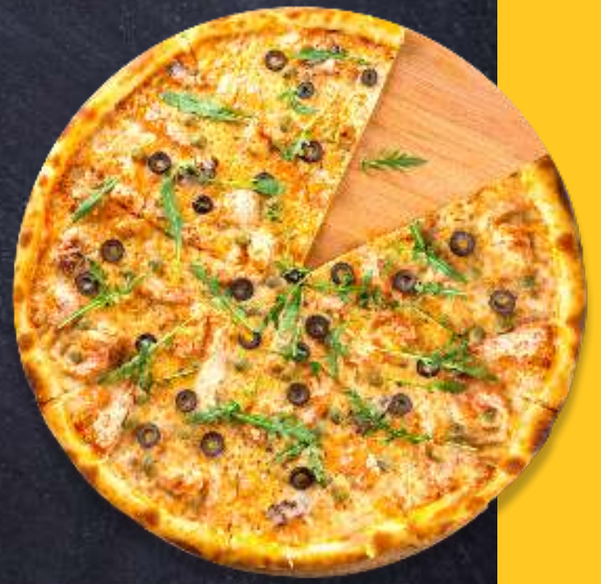
HELLO!

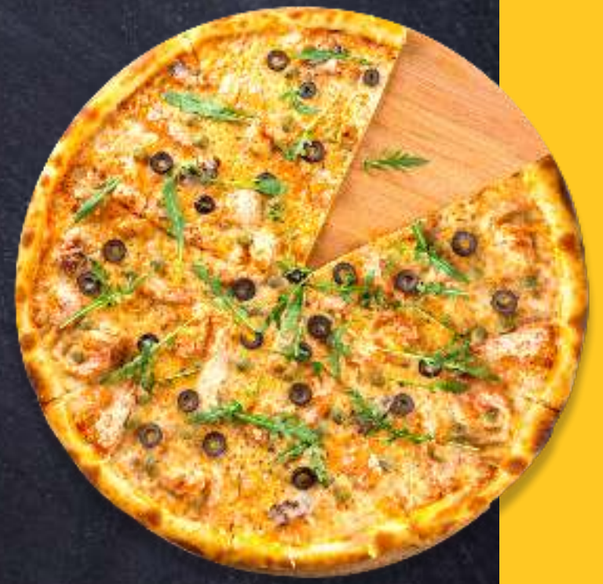
MY NAME IS NIKHI KUMAR, AND I AM EXCITED TO PRESENT MY WORK ON THE PIZZA SALES SQL PROJECT. I SPECIALIZE IN DATA ANALYSIS AND DATABASE MANAGEMENT, AND IN THIS PROJECT, I USED SQL TO ANALYZE AND REPORT ON PIZZA SALES DATA TO UNCOVER VALUABLE INSIGHTS THAT CAN HELP OPTIMIZE PIZZA SALES, CUSTOMER PREFERENCES, AND BUSINESS STRATEGY.



RETRIEVE THE TOTAL NO. OF ORDER PLACED.

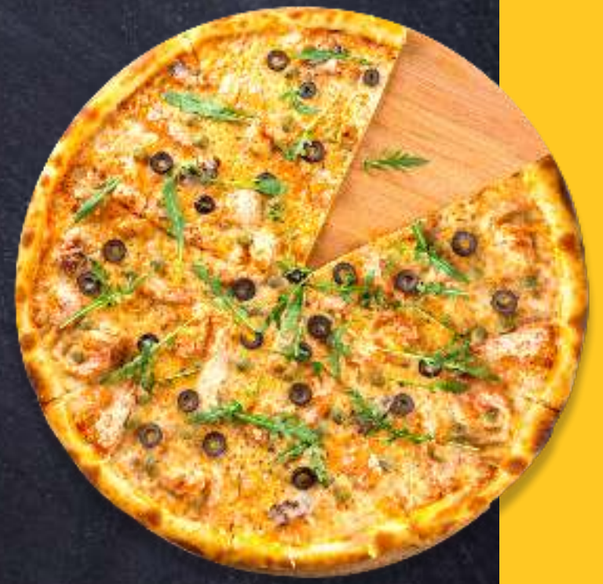
```
select count(*) as total_orders from orders;
```

	total_orders bigint
1	21350



CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

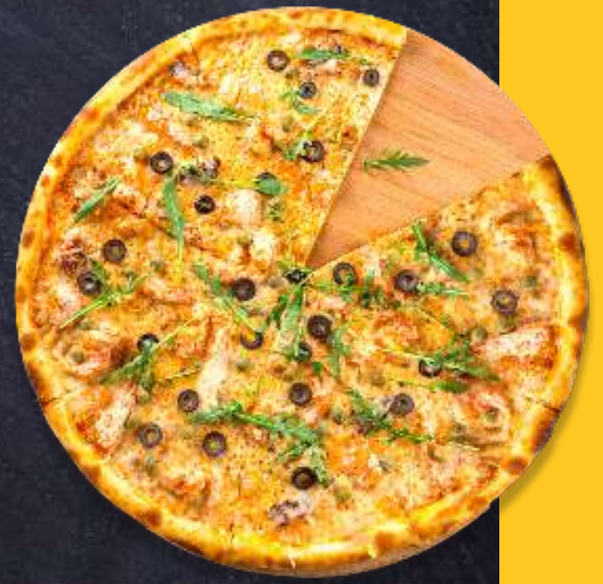
```sql
select round(sum(order_details.quantity * pizzas.price),2) as total_sales
from order_details join pizzas
on pizzas.pizza_id = order_details.pizza_id;
```



	total_sales numeric
1	817860.05

IDENTIFY THE HIGHEST PRICE PIZZAS.

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



	name character varying (100)	price numeric (10,2)
1	The Greek Pizza	35.95

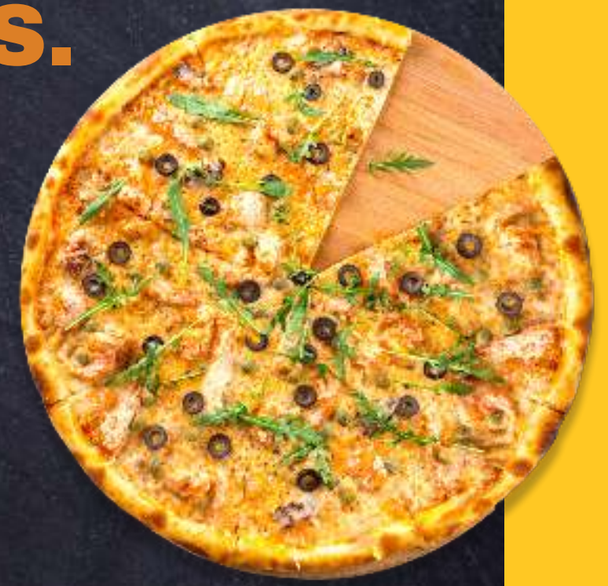
IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```sql
select pizzas.size, count(order_details.order_details_id) as order_count
from pizzas join order_details
on pizzas.pizza_id = order_details.pizza_id
group by size order by order_count desc;
```

	size character varying (10)	order_count bigint
	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28

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

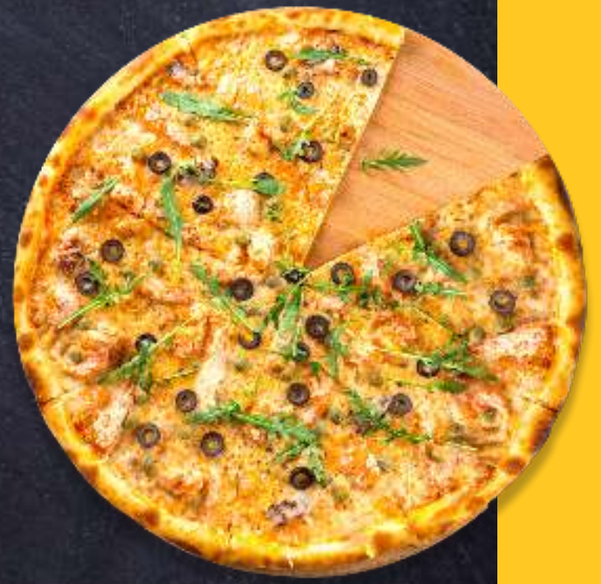
```sql
select pizza_type.name, sum(order_details.quantity) as quantity
from pizza_type join pizzas
on pizza_type.pizza_type_id = pizzas.pizza_type_id
join order_details
on order_details.pizza_id = pizzas.pizza_id
group by name order by quantity desc limit 5;
```



	name character varying (100)	quantity bigint
1	The Classic Deluxe Pizza	2453
2	The Barbecue Chicken Pizza	2432
3	The Hawaiian Pizza	2422
4	The Pepperoni Pizza	2418
5	The Thai Chicken Pizza	2371

RETRIEVE THE TOTAL NO. OF ORDER PLACED.

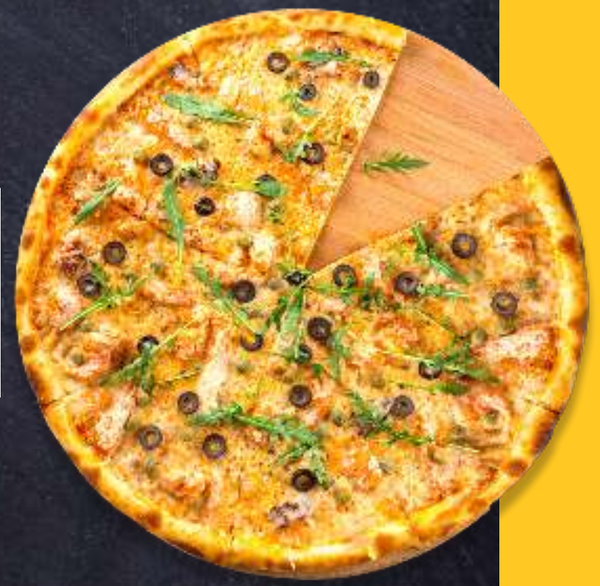
```
select count(*) as total_orders from orders;
```



	total_orders bigint
1	21350

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```
select round(sum(order_details.quantity * pizzas.price),2) as total_sales
from order_details join pizzas
on pizzas.pizza_id = order_details.pizza_id;
```



	total_sales numeric
1	817860.05

IDENTIFY THE HIGHEST PRICE PIZZAS.

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

	name character varying (100)	price numeric (10,2)
1	The Greek Pizza	35.95

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```
select pizzas.size, count(order_details.order_details_id) as order_count
from pizzas join order_details
on pizzas.pizza_id = order_details.pizza_id
group by size order by order_count desc;
```

	size character varying (10)	order_count bigint
1	L	18526
2	M	15385
3	S	14137
4	XL	544
5	XXL	28

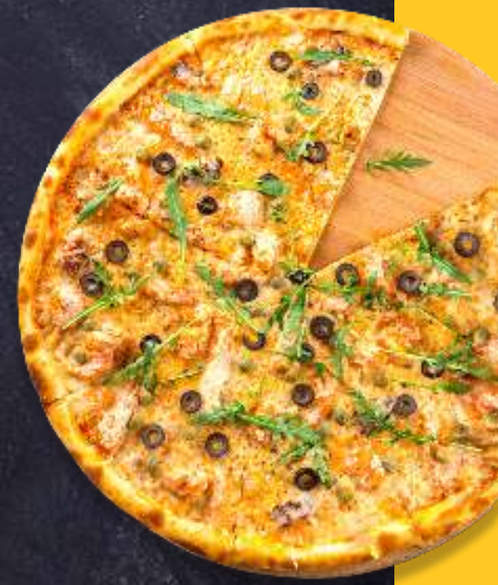
LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

```sql
select pizza_type.name, sum(order_details.quantity) as quantity
from pizza_type join pizzas
on pizza_type.pizza_type_id = pizzas.pizza_type_id
join order_details
on order_details.pizza_id = pizzas.pizza_id
group by name order by quantity desc limit 5;
```

	name character varying (100)	quantity bigint
1	The Classic Deluxe Pizza	2453
2	The Barbecue Chicken Pizza	2432
3	The Hawaiian Pizza	2422
4	The Pepperoni Pizza	2418
5	The Thai Chicken Pizza	2371

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

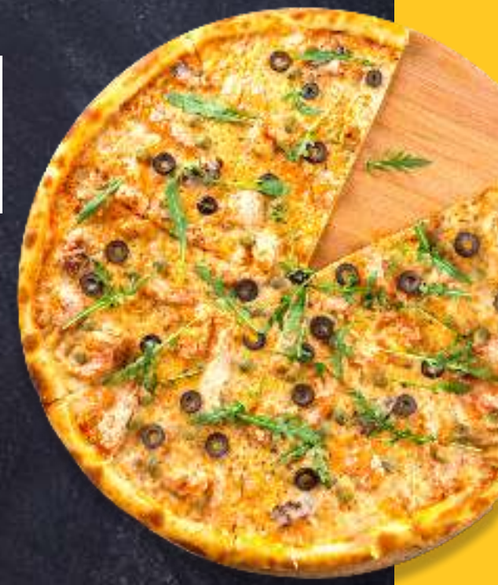
```sql
select pizza_type.category, sum(order_details.quantity) as quantity
from pizza_type join pizzas
on pizza_type.pizza_type_id = pizzas.pizza_type_id
join order_details
on pizzas.pizza_id = order_details.pizza_id
group by category
order by quantity desc;
```



	category character varying (50)	quantity bigint
1	Classic	14888
2	Supreme	11987
3	Veggie	11649
4	Chicken	11050

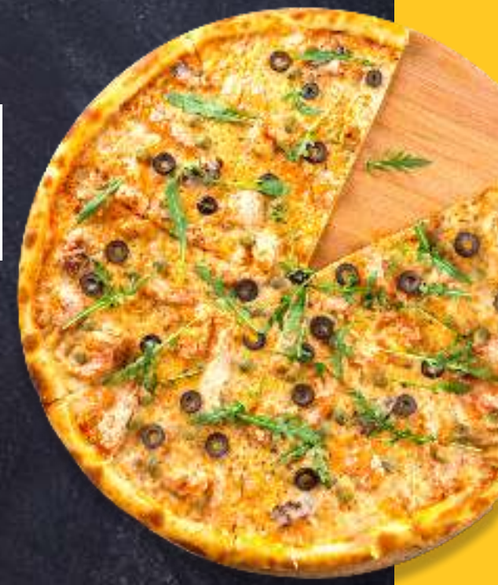
DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

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



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

```
select category, count(name) as total_pizza_category from pizza_type
group by category;
```

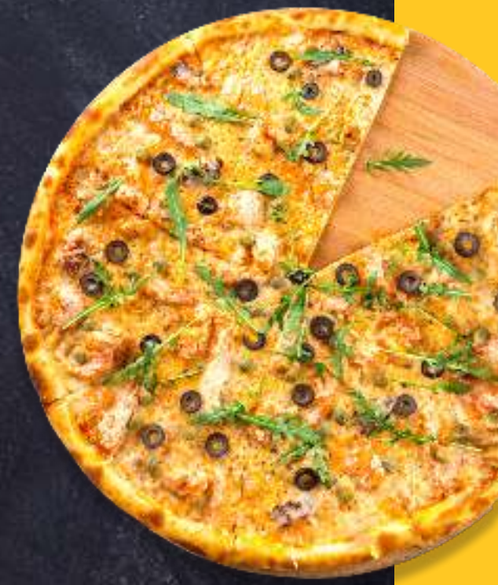


	category character varying (50)	total_pizza_category bigint
1	Supreme	9
2	Classic	8
3	Veggie	9
4	Chicken	6

GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.

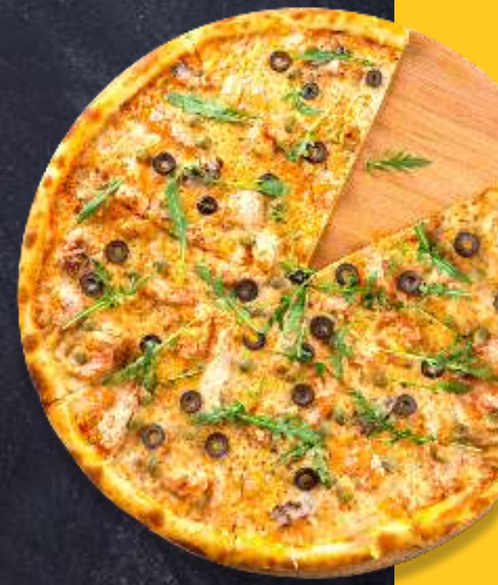
```sql
select avg(quantity) from
(select orders.order_date,sum(order_details.quantity) as quantity
from orders join order_details
on orders.order_id = order_details.order_id
group by orders.order_date) as order_quantity;
```

	avg numeric
1	138.4748603351955307



DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

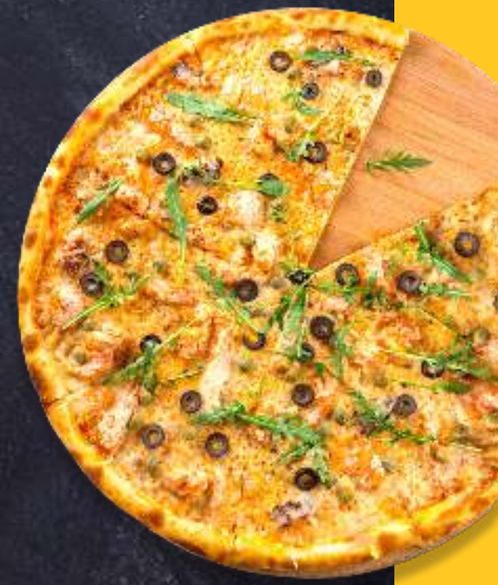
```sql
select pizza_type.name, sum(order_details.quantity * pizzas.price) as revenue
from pizza_type join pizzas
on pizza_type.pizza_type_id = pizzas.pizza_type_id
join order_details
on order_details.pizza_id = pizzas.pizza_id
group by name
order by revenue desc limit 3;
```



	name character varying (100)	revenue numeric
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768.00
3	The California Chicken Pizza	41409.50

CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

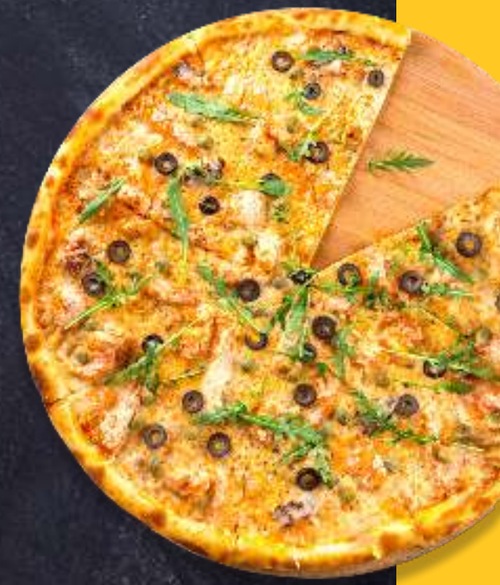
```sql
select pizza_type.category,
round(sum(order_details.quantity * pizzas.price)/(select round(sum(order_details.quantity * pizzas.price),2)
                                                                                    as total_sales
from order_details
join pizzas
on pizzas.pizza_id = order_details.pizza_id) * 100,2) as revenue

from pizza_type join pizzas
on pizza_type.pizza_type_id = pizzas.pizza_type_id
join order_details
on order_details.pizza_id = pizzas.pizza_id
group by category
order by revenue desc;
```



	category character varying (50)	revenue numeric
1	Classic	26.91
2	Supreme	25.46
3	Chicken	23.96
4	Veggie	23.68

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

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



DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

```sql
select name, revenue, category
from

(select category, name, revenue,
rank() over(partition by category order by revenue desc) as rn

from
(select pizza_type.category, pizza_type.name,
sum(order_details.quantity * pizzas.price) as revenue
from order_details join pizzas
on order_details.pizza_id = pizzas.pizza_id
join pizza_type
on pizza_type.pizza_type_id = pizzas.pizza_type_id
group by pizza_type.category, pizza_type.name) as a) as b

where rn<=3;
```