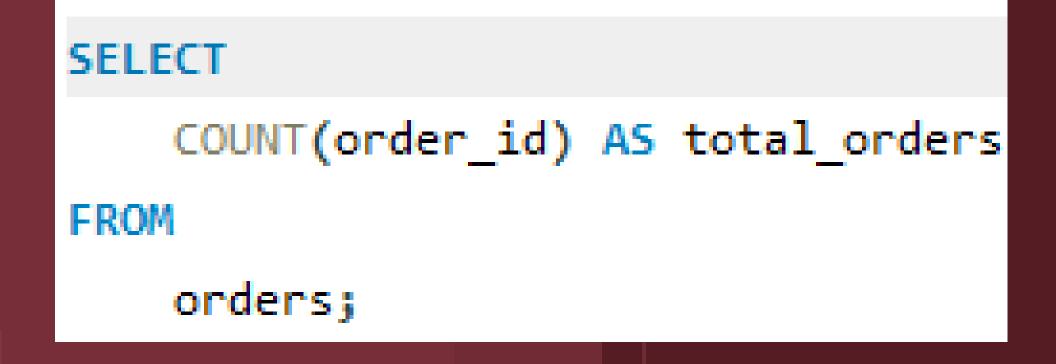
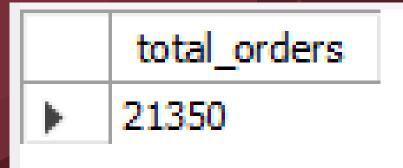
### PIZZA SALES ANALYSIS

Using SQL

This performs a detailed analysis of pizza sales data using SQL to uncover trends and insights about a year's worth of sales from a fictitious pizza place, including the date and time of each order and the pizzas served, with additional details on the type, size, quantity, price, and ingredients.

### RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED





#### 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

▶ 817860.05

#### IDENTIFY THE HIGHEST-PRICED PIZZA

	name	price
<b>&gt;</b>	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 pizzas.size
ORDER BY order_count DESC
LIMIT 1;
```

	size	order_count
•	L	18526

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

```
SELECT
   pizza_types.name, SUM(order_details.quantity) A5 quantity
FROM
   pizza_types
        JOTN
   pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOTN
   order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza types.name
ORDER BY quantity DESC
LIMIT 5;
```

	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

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

```
SELECT
    pizza_types.category,
    SUM(order_details.quantity) A5 total_quantity
FROM
    pizza types
        JOIN
    pizzas ON pizza types.pizza type id = pizzas.pizza type id
        JOTN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza types.category
ORDER BY total quantity DESC;
```

	category	total_quantity
•	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

### DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY

```
SELECT

HOUR(time) AS hour, COUNT(order_id) AS order_count

FROM

orders

GROUP BY HOUR(time);
```

	hour	order_count
<b>&gt;</b>	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
	10	8
	9	1

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

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

	category	COUNT(name)
•	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

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

```
ROUND(AVG(quantity), 0) AS avg_pizzas_ordered_per_day

FROM

(SELECT

orders.date, SUM(order_details.quantity) AS quantity

FROM

orders

JOIN order_details ON orders.order_id = order_details.order_id

GROUP BY orders.date) AS order_quantity;
```

```
avg_pizzas_ordered_per_day

138
```

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

```
SELECT
    pizza_types.name,
    SUM(order details.quantity * pizzas.price) AS revenue
FROM
    pizza types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza types.name
ORDER BY revenue DESC
LIMIT 3;
```

	name	revenue
•	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5

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

```
SELECT
   pizza_types.category,
   ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
                    ROUND(SUM(order_details.quantity * pizzas.price),
                                2) AS total sales
                FROM
                    order details
                        JOTN
                    pizzas ON pizzas.pizza_id = order_details.pizza_id) * 100,
            2) AS revenue FROM
   pizza types
        JOIN
   pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOTN
   order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza types.category
ORDER BY revenue DESC;
```

	category	revenue
•	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

#### ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME

```
select date, sum(revenue) over( order by date) as cum_revenue from (SELECT
    orders.date,
    SUM(order_details.quantity * pizzas.price) AS revenue
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id
        JOIN
    orders ON orders.order_id = order_details.order_id
GROUP BY orders.date) as sales;
```

	date	cum_revenue
•	2015-01-01 00:00:00	2713.8500000000004
	2015-01-02 00:00:00	5445.75
	2015-01-03 00:00:00	8108.15
	2015-01-04 00:00:00	9863.6
	2015-01-05 00:00:00	11929.55
	2015-01-06 00:00:00	14358.5
	1	

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

```
select name, revenue from (select category, name, revenue,
rank() over(partition by category order by revenue desc) as rn
from
(select pizza_types.category, pizza_types.name,
sum((order_details.quantity) * pizzas.price) as revenue
from pizza types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza type id
join order_details
on order details.pizza id = pizzas.pizza id
group by pizza_types.category, pizza_types.name) as a) as b
where rn<=3;
```

	name	revenue
<b>&gt;</b>	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5
	The Classic Deluxe Pizza	38180.5
	The Hawaiian Pizza	32273.25
	The Pepperoni Pizza	30161.75
	The Spicy Italian Pizza	34831.25
	The Italian Supreme Pizza	33476.75
	The Sicilian Pizza	30940.5
	The Four Cheese Pizza	32265.70000000065
	The Mexicana Pizza	26780.75