



LARANA PIZZA

# PIZZA HUT



# RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

```
1   -- Retrieve the total number of order placed.  
2  
3 • select count(order_id) as total_orders from orders;
```

	total_orders
▶	1410





# CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```
-- Calculate the total revenue generated from pizza sales.

SELECT
    SUM(order_details.quantity * pizzas.price) AS Total_Sales
FROM
    pizzas
    JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id;
```

Total_Sales
10173.999999999998

# IDENTIFY THE HIGHEST-PRICED PIZZA.

-- Identify the highest-priced pizza.

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

Result Grid		
	name	price
▶	The Greek Pizza	35.95

# IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED

```
1 -- Identify the most common pizza size ordered.  
2  
3 • SELECT  
4     pizzas.size,  
5     COUNT(order_details.order_details_id) AS order_count  
6 FROM  
7     pizzas  
8     JOIN  
9         order_details ON pizzas.pizza_id = order_details.pizza_id  
10 GROUP BY pizzas.size  
11 ORDER BY order_count DESC  
12 LIMIT 1;
```

	size	order_count
▶	L	256



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

```
-- Join the necessary tables to find the total quantity of each pizza category ordered.

SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS quantity
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
ORDER BY quantity DESC
LIMIT 5;
```

	category	count(name)
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9



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

```
-- Determine the distribution of orders by hour of the day.  
  
SELECT  
    HOUR(order_time), COUNT(order_id)  
FROM  
    orders  
GROUP BY HOUR(order_time);
```

hour(order_time)	count(order_id)
11	75
12	171
13	151
14	127
15	108
16	124
17	160
18	150
19	137
20	101
21	64
22	41
23	1



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

```
-- Join relevant tables to find the category-wise distribution of pizzas.
```

```
SELECT
```

```
    category, COUNT(name)
```

```
FROM
```

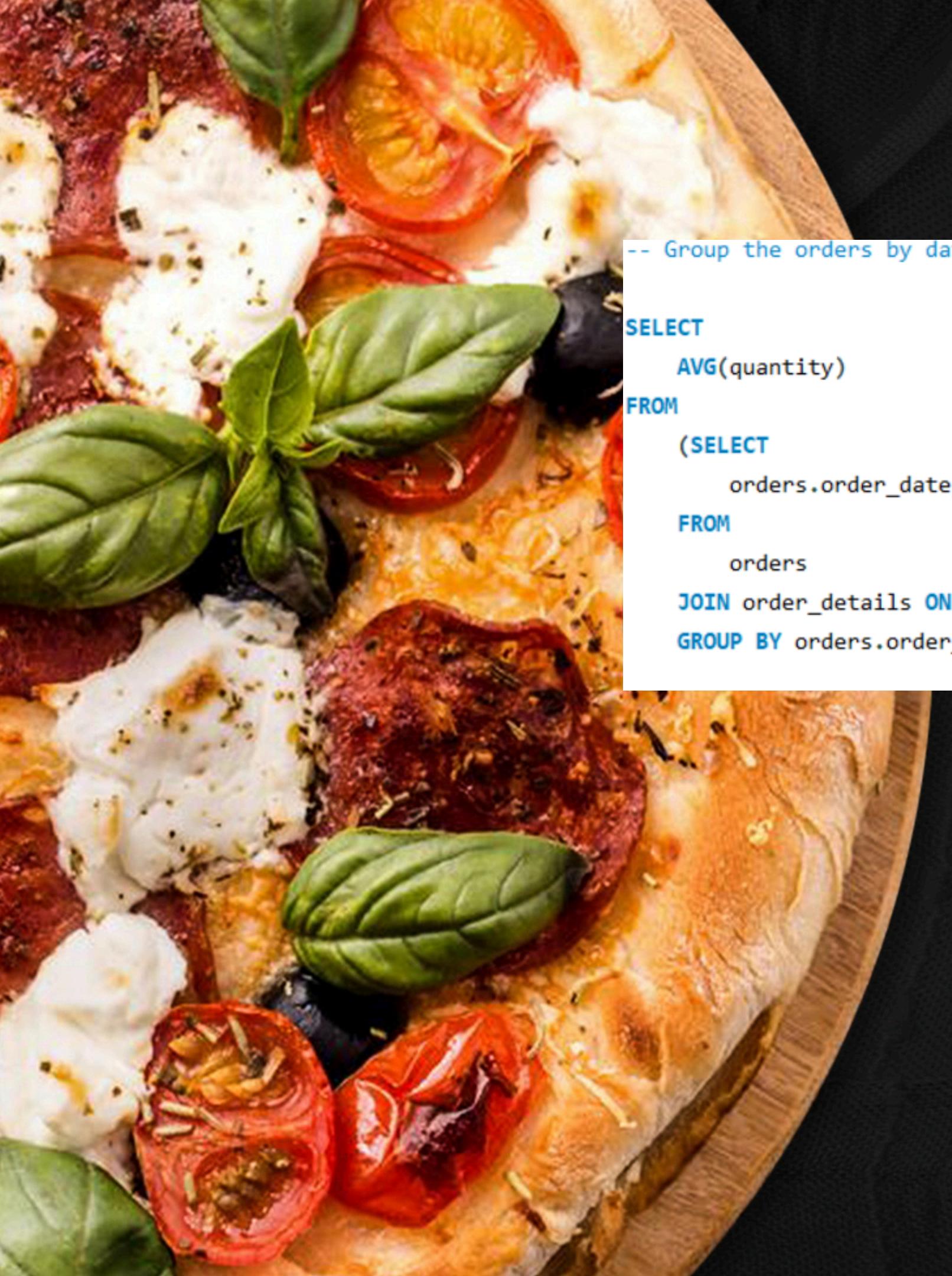
```
    pizza_types
```

```
GROUP BY category;
```

	category	count(name)
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9



LARANA PIZZA



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

```
-- Group the orders by date and calculate the average number of pizzas ordered per day.

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;
```

	<b>AVG(quantity)</b>
▶	<b>121.6000</b>

# ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

-- Analyze the cumulative revenue generated over time.

```
SELECT
    order_date,
    SUM(revenue) OVER(ORDER BY order_date) AS cum_revenue
FROM (
    SELECT
        orders.order_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.order_date
) AS sales;
```

	order_date	cum_revenue
▶	2015-01-01	2713.8500000000004
	2015-01-02	5445.75
	2015-01-03	8108.15
	2015-01-04	9863.6
	2015-01-05	10174

