SUMAN RAI

THE PIZZA CAFE

SALES REPORT

ORDER NOW

I AM SUMAN RAI, A DATA ANALYST. THIS IS MY SQL PROJECT DEVELOPED FOR THE PIZZA CAFE ANALYZING THE SALES AND REVENUE OF PIZZAS.

BASICS

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

```sql
-- Retrieve the total number of orders placed.

SELECT * FROM orders;
SELECT count(order_id) as total_orders from orders o;
```

Results 1

SELECT count(order_id) as total_orders f | Enter a SQL expression to filter results (use Ctrl+Space)

	123 total_orders
1	21,350

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```
--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
```

Results 1

SELECT ROUND(SUM(order_details.quar | Enter a SQL expression to filter results (use Ctrl+Space)

	123 total_sales
1	817,860.05

IDENTIFY THE HIGHEST-PRICED PIZZA.

```
--Identify the highest-priced pizza.

SELECT * FROM pizzas
ORDER BY price DESC LIMIT 1;
```

zas 1

LECT * FROM pizzas ORDER BY price [| Enter a SQL expression to filter results (use Ctrl+Space)

pizza_id	pizza_type_id	size	price
the_greek_xxl	the_greek	XXL	35.95

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```
--Identify the most common pizza size ordered.

SELECT pizzas."size", count(order_details.order_details_id)
FROM pizzas JOIN order_details
ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas."size"
ORDER BY "size" ASC LIMIT 1;
```

pizzas 1

SELECT pizzas."size", count(order_detail: | Enter a SQL expression to filter results (use Ctrl+Sp

	size	count(order_details.order_details_id)
1	L	18,526

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

```sql
SELECT pizza_types.name,
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.name ORDER BY quantity DESC LIMIT 5;
```

pizza_types 1

SELECT pizza_types.name, SUM(order_d | Enter a SQL expression to filter results (use Ctrl+Space)

	name	quantity
1	The Classic Deluxe Pizza	2,453
2	The Barbecue Chicken Pizza	2,432
3	The Hawaiian Pizza	2,422
4	The Pepperoni Pizza	2,418
5	The Thai Chicken Pizza	2,371

INTERMEDIATE

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;
```

a_types 1

ECT pizza_types.category, SUM(orde | Enter a SQL expression to filter results (use Ctrl+Space)

category	quantity
Classic	14,888
Supreme	11,987
Veggie	11,649
Chicken	11,050

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```sql
SELECT HOUR(orders.time), count(orders.order_id)
FROM orders
GROUP BY HOUR(orders.time);
```

Results 1

SELECT TIME(orders.time), count(order: | Enter a SQL expression to filter results (use Ctrl+Space)

	TIME(orders.time)	count(orders.order_id)
1	09:52:21	1
2	10:25:19	1
3	10:34:34	1
4	10:43:04	1
5	10:50:46	1
6	10:52:26	1
7	10:54:03	1
8	10:54:15	1

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

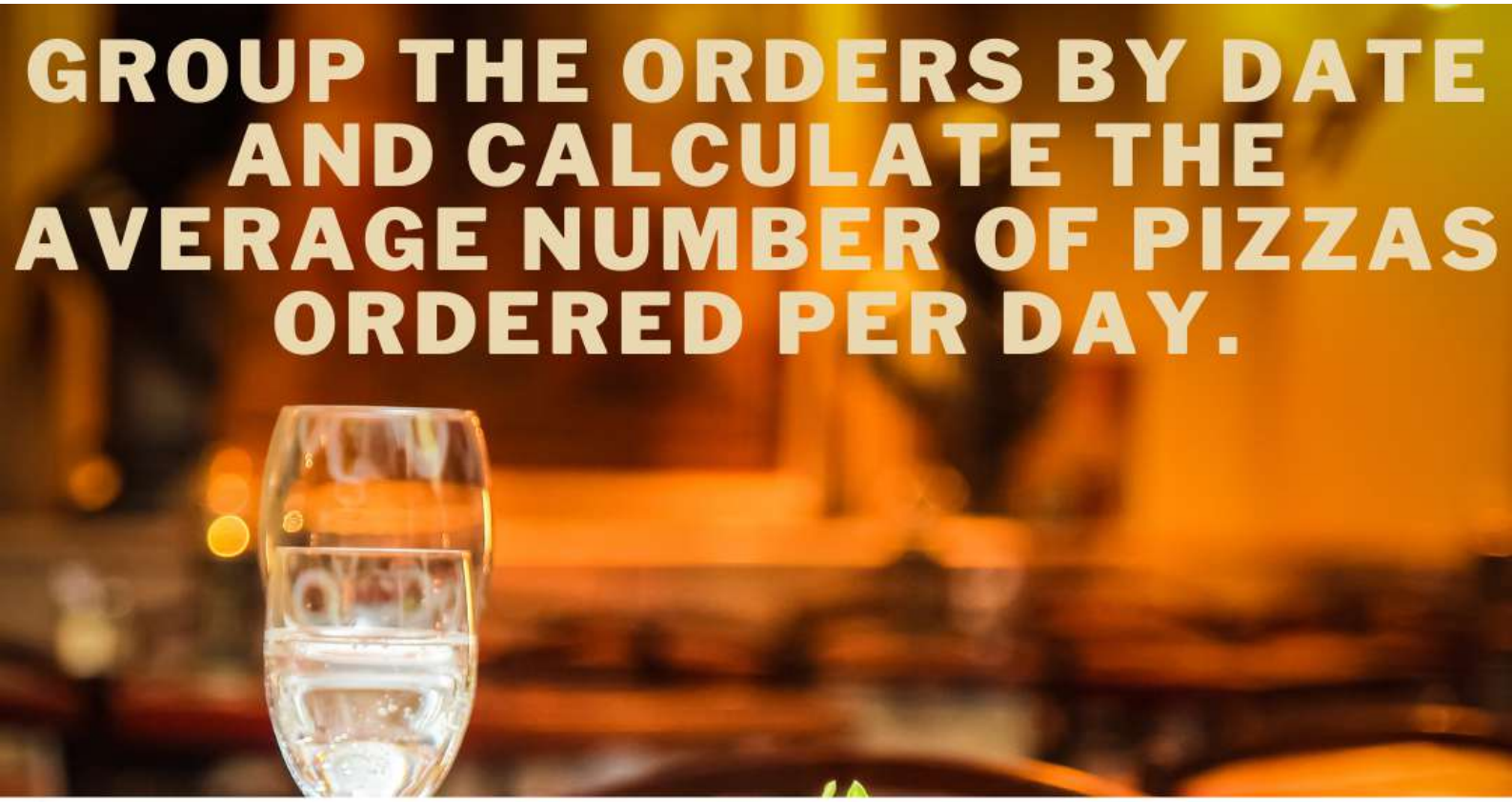
```
--Join relevant tables to find the category-wise distribution of pizzas.

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

za_types 1

LECT category, COUNT(name) FROM | Enter a SQL expression to filter results (use Ctrl+Space)

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.



```
SELECT AVG(quantity)
FROM(SELECT orders.date,orders.order_id ,SUM(order_details.quantity) as 'quantity'
FROM orders JOIN order_details
ON orders.order_id = order_details.order_id
GROUP BY orders.date);
```

rs 1

CT orders.date,orders.order_id ,SUM | Enter a SQL expression to filter results (use Ctrl+Space)

date	order_id	quantity
2015-01-01	1	49,574

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

```
--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;
```

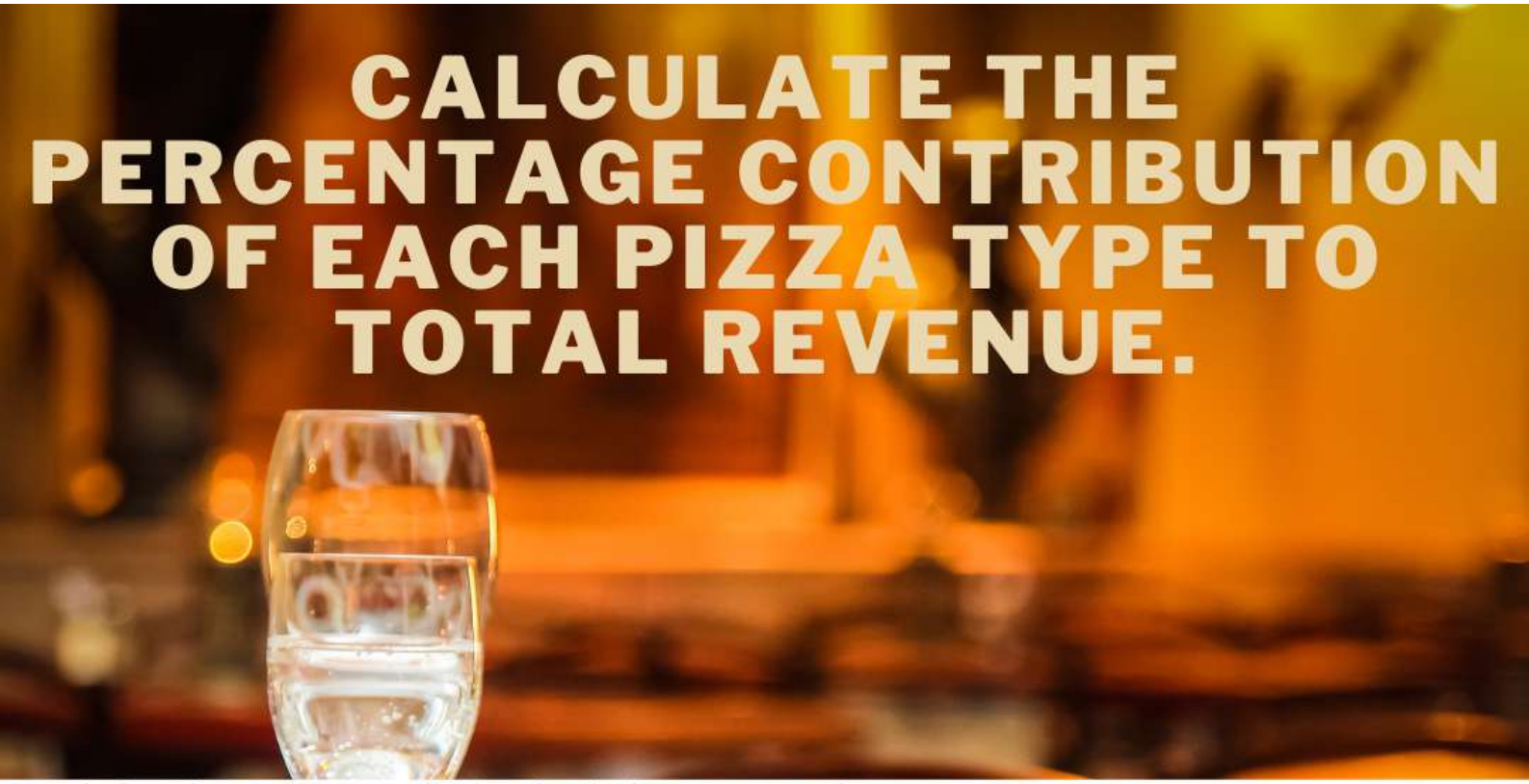
pizza_types 1

SELECT pizza_types.name, SUM(order_d | Enter a SQL expression to filter results (use Ctrl+Sp

Grid | Text

	name	revenue
1	The Thai Chicken Pizza	43,434.25
2	The Barbecue Chicken Pizza	42,768
3	The California Chicken Pizza	41,409.5

ADVANCED




```
--Calculate the percentage contribution of each pizza type to total revenue.
```


```
--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
JOIN pizzas
ON pizzas.pizza_id = order_details.pizza_id) *100,2) 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 ORDER BY revenue DESC;
```

pizza_types 1

SELECT pizza_types.category, ROUND(SI | Enter a SQL expression to filter results (use Ctrl+Space)

	category	revenue
1	Classic	26.91
2	Supreme	25.46
3	Chicken	23.96
4	Veggie	23.68

Value

Classic

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```sql
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;
```

ders 1

LECT date, SUM(revenue) OVER(ORDE | Enter a SQL expression to filter results (

date	cum_revenue
2015-01-01	2,713.85
2015-01-02	5,445.75
2015-01-03	8,108.15
2015-01-04	9,863.6
2015-01-05	11,929.55
2015-01-06	14,358.5
2015-01-07	16,560.7

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

```
--Determine the top 3 most ordered pizza types based on revenue for

SELECT name, revenue FROM
(SELECT category, name, revenue,
RANK() OVER(PARTITION BY category ORDER BY revenue DESC) AS r 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 r <=3;
```

pizza_types 1

ECT name, revenue FROM (SELECT c | Enter a SQL expression to filter results (use Ctrl+Space

name	revenue
The Thai Chicken Pizza	43,434.25
The Barbecue Chicken Pizza	42,768
The California Chicken Pizza	41,409.5
The Classic Deluxe Pizza	38,180.5
The Hawaiian Pizza	32,273.25
The Pepperoni Pizza	30,161.75
The Spicy Italian Pizza	34,831.25
The Italian Supreme Pizza	33,476.75