Pizza Sales: SQL Insights

Unlocking valuable business data through SQL queries

Basic Query 1: Total Orders

Retrieve the total number of orders placed.

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

This query provides a quick snapshot of overall business volume, essential for high-level performance tracking.

Basic Query 2: Total Revenue

Calculate the total revenue generated from pizza sales.

```
SELECT ROUND(SUM(o.quantity * p.price), 2) AS total_revenue FROM order_details o JOIN pizzas p ON
o.pizza_id = p.pizza_id;
```

Understanding total revenue is fundamental for financial analysis and strategic planning.

Basic Query 3 & 4: Top Pizza Insights

Identify the highest-priced pizza.

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

Identify the most common pizza size ordered.

```
SELECT p.size, COUNT(od.order_details_id)FROM
pizzas p JOIN order_details od ON p.pizza_id =
od.pizza_id GROUP BY p.size ORDER BY
COUNT(od.order_details_id) DESC LIMIT 1;
```

Basic Query 5: Top 5 Pizza Types

List the top 5 most ordered pizza types along with their quantities.

```
SELECT pt.name, SUM(od.quantity)FROM pizza_types pt JOIN pizzas p ON pt.pizza_type_id = p.pizza_type_id
JOIN order_details od ON p.pizza_id = od.pizza_id GROUP BY pt.name ORDER BY SUM(od.quantity) DESC LIMIT
5;
```

Actionable Insight: This query highlights popular items, informing inventory management and marketing strategies.

Intermediate Query 6 & 7: Category & Time Distribution

Total quantity of each pizza category ordered.

```
SELECT pt.category, SUM(od.quantity)FROM
pizza_types pt JOIN pizzas p ON
pt.pizza_type_id = p.pizza_type_id JOIN
order_details od ON p.pizza_id = od.pizza_id
GROUP BY pt.category;
```

Distribution of orders by hour of the day.

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

Intermediate Query 8 & 9: Pizza Categories & Daily Averages

Category-wise distribution of pizzas.

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

Average number of pizzas ordered per day.

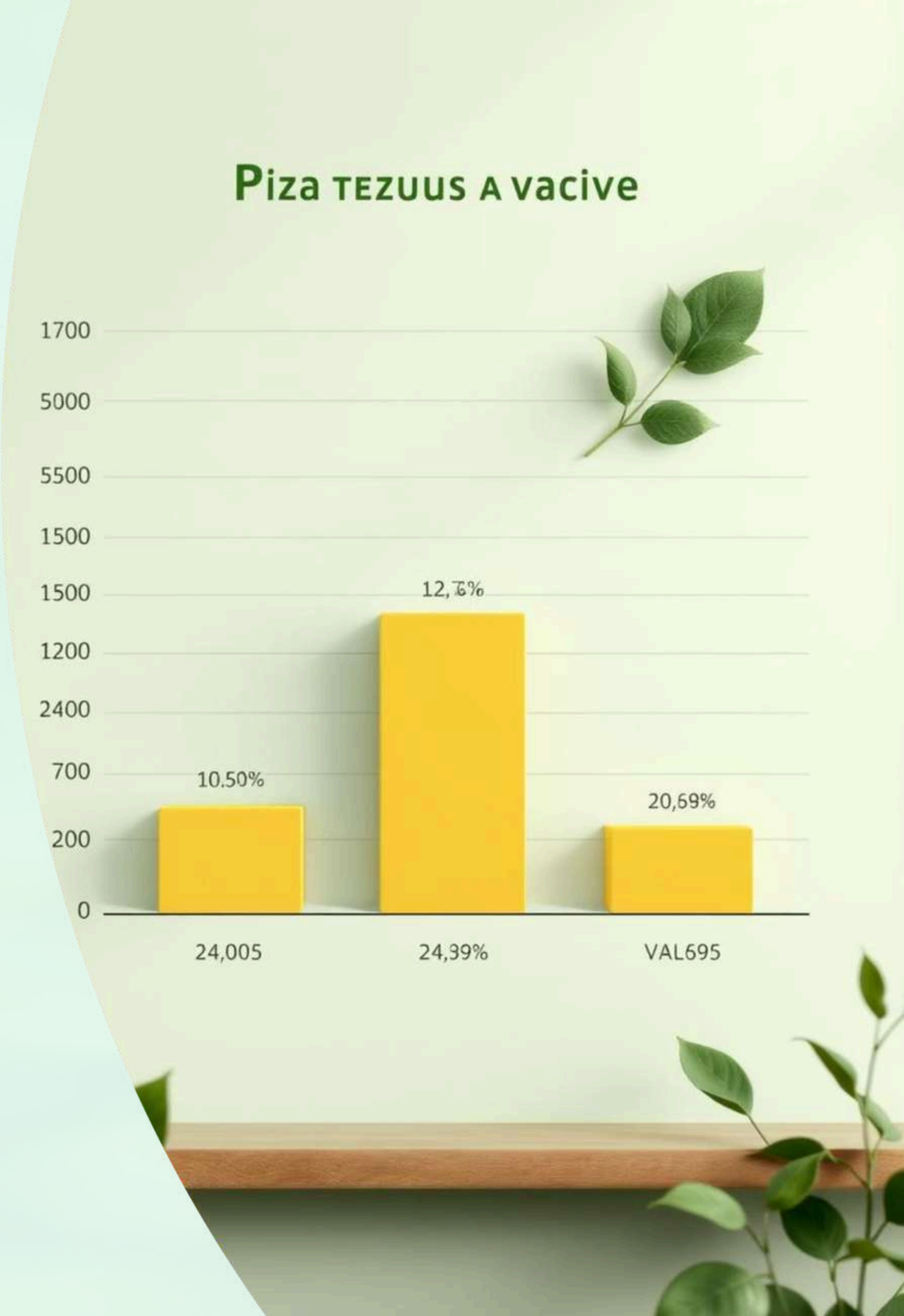
```
SELECT DAY(order_date), AVG(order_id)FROM
orders GROUP BY DAY(order_date);
```

Why it matters: These queries help identify peak hours for staffing and popular categories for menu optimization.

Intermediate Query 10: Top 3 Pizza Types by Revenue

Determine the top 3 most ordered pizza types based on revenue.

```
SELECT pt.name, SUM(od.quantity * p.price)FROM pizza_types
pt JOIN pizzas p ON pt.pizza_type_id = p.pizza_type_id
JOIN order_details od ON p.pizza_id = od.pizza_id GROUP BY
pt.name ORDER BY SUM(od.quantity * p.price) DESC LIMIT 3;
```



Advanced Query 11: Revenue Contribution by Pizza Type

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

```
SELECT pt.name, ROUND(SUM(od.quantity * p.price) / (SELECT ROUND(SUM(o.quantity * p.price), 2)FROM
order_details o JOIN pizzas p ON o.pizza_id = p.pizza_id) * 100, 2) AS revenue_percentage FROM
pizza_types pt JOIN pizzas p ON pt.pizza_type_id = p.pizza_type_id JOIN order_details od ON p.pizza_id
= od.pizza_id GROUP BY pt.name ORDER BY revenue_percentage DESC;
```

How does each pizza type contribute to the overall financial success of the business?

Advanced Query 12 & 13: Cumulative Revenue & Categorized Top Revenue

Analyze the cumulative revenue generated over time.

```
SELECT order_date, SUM(revenue) OVER(ORDER BY
order_date) AS cumulative_revenue FROM (SELECT
orders.order_date, SUM(order_details.quantity *
pizzas.price) AS revenue FROM orders JOIN
order_details ON orders.order_id =
order_details.order_id JOIN pizzas ON
order_details.pizza_id = pizzas.pizza_id GROUP BY
orders.order_date) AS sales;
```

Top 3 most ordered pizza types based on revenue for each category.

```
SELECT category, name, rn FROM (SELECT category,
name, 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 pizzas.pizza_id =
order_details.pizza_id GROUP BY pizza_types.category,
pizza_types.name) AS a) AS b WHERE rn <= 3;
```