

Pizza Sales MySQL Analysis



Gaytri Mohapatra

Sales Analysis using MySQL queries



Retrieve the total number of orders placed.

```
mysql> use pizza_hut;
Database changed
mysql> select count(order_id) as total_orders from orders;
+-----+
| total_orders |
+-----+
|      671   |
+-----+
1 row in set (0.08 sec)
```

Calculate the total revenue generated from pizza sales.

```
mysql> 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 |
+-----+
|      507.00 |
+-----+
1 row in set (0.03 sec)
```

Identify the highest-priced pizza.

```
mysql> 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;
+-----+-----+
| name      | price |
+-----+-----+
| The Greek Pizza | 35.95 |
+-----+-----+
1 row in set (0.08 sec)
```

Identify the most common pizza size ordered.

```
mysql> 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;
+-----+
| size | count(order_details.order_details_id) |
+-----+
| L   |          12 |
| M   |          7  |
| S   |          11 |
+-----+
3 rows in set (0.11 sec)
```

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

```
mysql> 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;
+-----+-----+
| name           | quantity |
+-----+-----+
| The California Chicken Pizza |      4 |
| The Italian Supreme Pizza   |      4 |
| The Greek Pizza          |      3 |
| The Barbecue Chicken Pizza |      2 |
| The Spinach Supreme Pizza  |      2 |
+-----+-----+
5 rows in set (0.04 sec)
```

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

```
mysql> 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;
+-----+-----+
| category | quantity |
+-----+-----+
| Supreme   |      9 |
| Veggie    |      9 |
| Chicken   |      7 |
| Classic   |      6 |
+-----+-----+
4 rows in set (0.02 sec)
```

Determine the distribution of orders by hour of the day.

```
mysql> select hour(time), count(order_id)
-> FROM orders
-> GROUP BY hour(time);
+-----+-----+
| hour(time) | count(order_id) |
+-----+-----+
|      11    |        34    |
|      12    |        80    |
|      13    |        80    |
|      14    |        56    |
|      15    |        48    |
|      16    |        57    |
|      17    |        73    |
|      18    |        74    |
|      19    |        61    |
|      20    |        53    |
|      21    |        30    |
|      22    |        25    |
+-----+-----+
12 rows in set (0.03 sec)
```

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

```
mysql> 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;
+-----+-----+
| category | revenue |
+-----+-----+
| Supreme  | 30.18  |
| Veggie   | 29.24  |
| Chicken  | 24.70  |
| Classic  | 15.88  |
+-----+-----+
4 rows in set (0.05 sec)
```

Thank you!

