Indian Institute of Information Technology Surat

Lab Report on
Advanced Database Management (CS 604)
Practical

Submitted by

[RAHUL KUMAR SINGH] (UI21CS44)

Course Faculty

Mr. Rishi Sharma

Department of Computer Science and Engineering
Indian Institute of Information Technology Surat
Gujarat-394190, India

Jan-2024

Lab No: 5 (Part 2)

Aim: How to analyze ecommerce Inventory

Description:

- Q1) What are the top 5 products with the highest inventory levels on the most recent inventory date?
- Q2) What is the total inventory level for each product category on the most recent inventory date?
- Q3) What is the average inventory level for each product category for the month of January 2022?
- Q4) Which products had a decrease in inventory level from the previous inventory date to the current inventory date?
- Q5) What is the overall trend in inventory levels for each product category over the month of January 2022?
- Tables Created:
  - The `Products` table stores information about each product including its ID, name, and category.
  - The `Inventory` table keeps track of the inventory levels of each product on different dates.

Source Code:

Create Tables:

```
CREATE TABLE Products (
   product_id INT AUTO_INCREMENT,
   product_name VARCHAR(255),
   category VARCHAR(255),
   PRIMARY KEY (product_id)
);
CREATE TABLE Inventory (
   inventory_id INT AUTO_INCREMENT,
   product_id INT,
   inventory_date DATETIME,
   inventory_level INT,
   PRIMARY KEY (inventory_id),
   FOREIGN KEY (product_id) REFERENCES Products(product_id)
);
```

Q1) Top 5 products with the highest inventory levels on the most recent inventory date:

```
SELECT p.product_name, i.inventory_level, i.inventory_date
FROM Inventory i
JOIN Products p ON p.product_id = i.product_id
WHERE DATE(i.inventory_date) = (SELECT MAX(DATE(inventory_date)) FROM Inventory)
ORDER BY i.inventory_level DESC
LIMIT 5;
```

Q2) Total inventory level for each product category on the most recent inventory date:

```
SELECT p.category, SUM(i.inventory_level) as total_inventory
FROM Products p
JOIN Inventory i ON p.product_id = i.product_id
WHERE DATE(i.inventory_date) = (SELECT MAX(DATE(inventory_date)) FROM Inventory ic JOIN Products pc ON pc.product_id =
ic.product_id WHERE pc.category = p.category)
GROUP BY p.category;
```

Q3) Average inventory level for each product category for the month of January 2022:

```
SELECT p.category, AVG(i.inventory_level) as avg_inventory
```

```
FROM Products p
JOIN Inventory i ON p.product_id = i.product_id
WHERE MONTH(i.inventory_date) = 1 AND YEAR(i.inventory_date) = 2022
GROUP BY p.category;
```

Q4) Products that had a decrease in inventory level from the previous inventory date to the current inventory date:

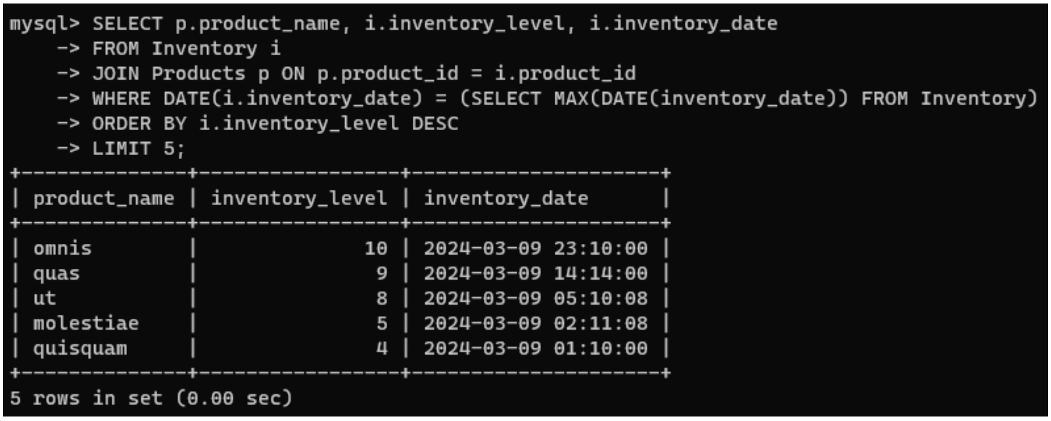
```
SELECT DISTINCT(p.product_name)
FROM Products p
JOIN Inventory i1 ON p.product_id = i1.product_id
JOIN Inventory i2 ON p.product_id = i2.product_id
WHERE i1.inventory_date < i2.inventory_date AND i1.inventory_level > i2.inventory_level;
```

Q5) Overall trend in inventory levels for each product category over the month of January 2022:

```
SELECT p.category, AVG(i.inventory_level) as avg_inventory, COUNT(i.inventory_level) as count_inventory, SUM(i.inventory_level) as total_inventory, MAX(i.inventory_level) as max_inventory, MIN(i.inventory_level) as min_inventory
FROM Products p
JOIN Inventory i ON p.product_id = i.product_id
WHERE MONTH(i.inventory_date) = 1 AND YEAR(i.inventory_date) = 2022
GROUP BY p.category
ORDER BY avg_inventory DESC;
```

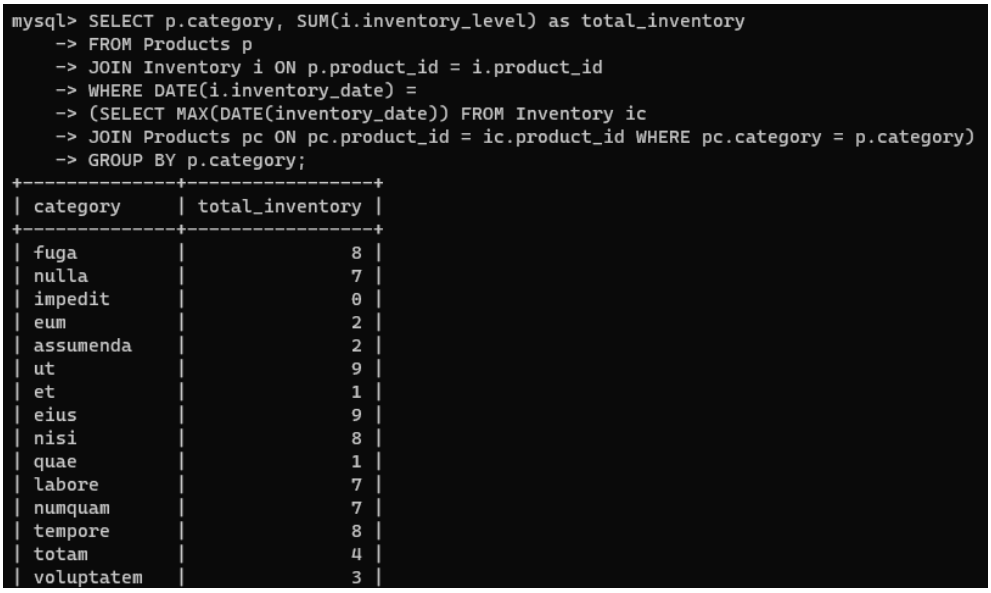
Output:

Q1) Top 5 products with the highest inventory levels on the most recent inventory date:



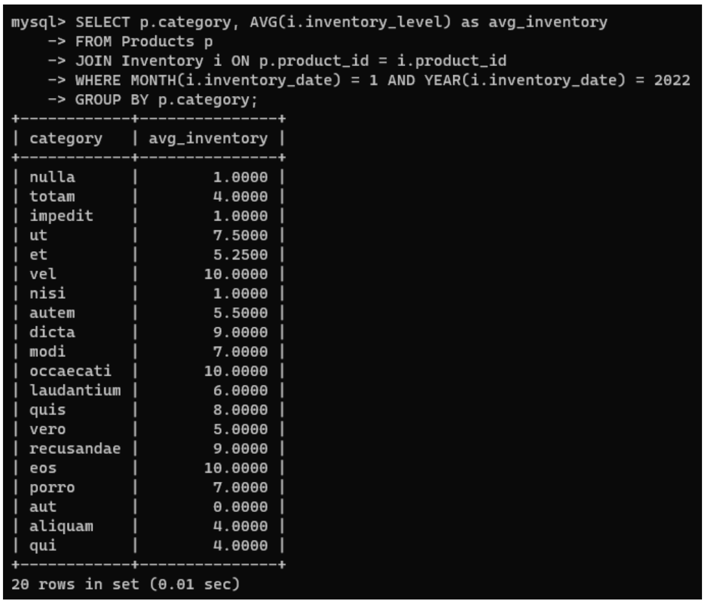
```
mysql> SELECT p.product_name, i.inventory_level, i.inventory_date
    -> FROM Inventory i
    -> JOIN Products p ON p.product_id = i.product_id
    -> WHERE DATE(i.inventory_date) = (SELECT MAX(DATE(inventory_date)) FROM Inventory)
    -> ORDER BY i.inventory_level DESC
    -> LIMIT 5;
+--------------+-----------------+---------------------+
| product_name | inventory_level | inventory_date      |
+--------------+-----------------+---------------------+
| omnis        |              10 | 2024-03-09 23:10:00 |
| quas         |               9 | 2024-03-09 14:14:00 |
| ut           |               8 | 2024-03-09 05:10:08 |
| molestiae    |               5 | 2024-03-09 02:11:08 |
| quisquam     |               4 | 2024-03-09 01:10:00 |
+--------------+-----------------+---------------------+
5 rows in set (0.00 sec)
```

Q2) Total inventory level for each product category on the most recent inventory date:



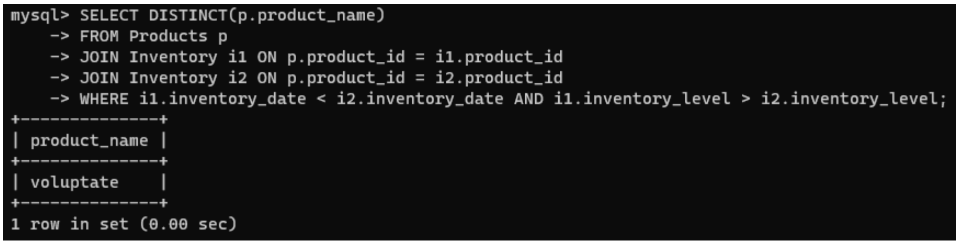
```
mysql> SELECT p.category, SUM(i.inventory_level) as total_inventory
    -> FROM Products p
    -> JOIN Inventory i ON p.product_id = i.product_id
    -> WHERE DATE(i.inventory_date) =
    -> (SELECT MAX(DATE(inventory_date)) FROM Inventory ic
    -> JOIN Products pc ON pc.product_id = ic.product_id WHERE pc.category = p.category)
    -> GROUP BY p.category;
+--------------+-----------------+
| category     | total_inventory |
+--------------+-----------------+
| fuga         |               8 |
| nulla        |               7 |
| impedit      |               0 |
| eum          |               2 |
| assumenda    |               2 |
| ut           |               9 |
| et           |               1 |
| eius         |               9 |
| nisi         |               8 |
| quae         |               1 |
| labore       |               7 |
| numquam      |               7 |
| tempore      |               8 |
| totam        |               4 |
| voluptatem   |               3 |
```

Q3) Average inventory level for each product category for the month of January 2022:



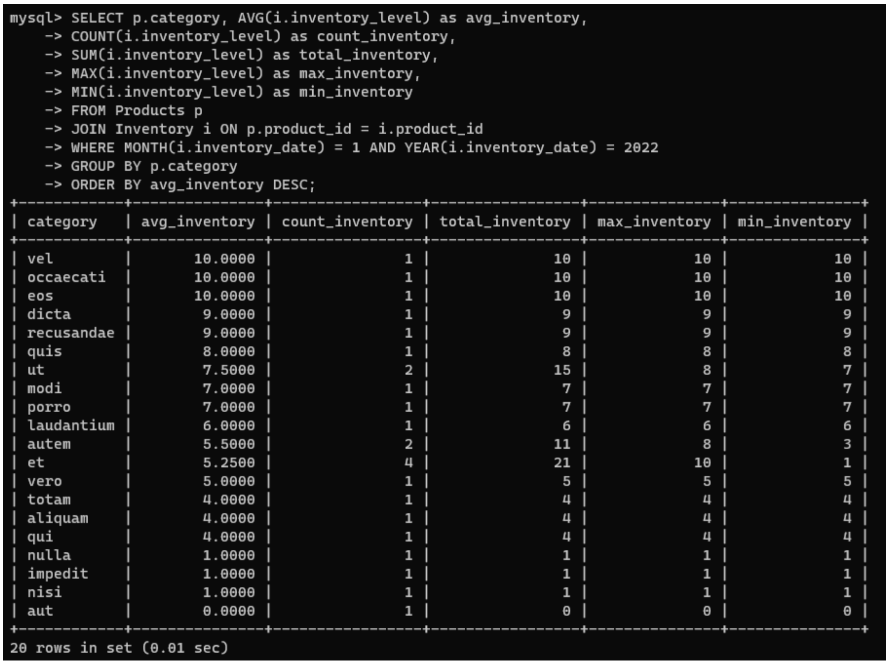
```
mysql> SELECT p.category, AVG(i.inventory_level) as avg_inventory
    -> FROM Products p
    -> JOIN Inventory i ON p.product_id = i.product_id
    -> WHERE MONTH(i.inventory_date) = 1 AND YEAR(i.inventory_date) = 2022
    -> GROUP BY p.category;
+------------+---------------+
| category   | avg_inventory |
+------------+---------------+
| nulla      |        1.0000 |
| totam      |        4.0000 |
| impedit    |        1.0000 |
| ut         |        7.5000 |
| et         |        5.2500 |
| vel        |       10.0000 |
| nisi       |        1.0000 |
| autem      |        5.5000 |
| dicta      |        9.0000 |
| modi       |        7.0000 |
| occaecati  |       10.0000 |
| laudantium |        6.0000 |
| quis       |        8.0000 |
| vero       |        5.0000 |
| recusandae |        9.0000 |
| eos        |       10.0000 |
| porro      |        7.0000 |
| aut        |        0.0000 |
| aliquam    |        4.0000 |
| qui        |        4.0000 |
+------------+---------------+
20 rows in set (0.01 sec)
```

Q4) Products that had a decrease in inventory level from the previous inventory date to the current inventory date:



```
mysql> SELECT DISTINCT(p.product_name)
    -> FROM Products p
    -> JOIN Inventory i1 ON p.product_id = i1.product_id
    -> JOIN Inventory i2 ON p.product_id = i2.product_id
    -> WHERE i1.inventory_date < i2.inventory_date AND i1.inventory_level > i2.inventory_level;
+--------------+
| product_name |
+--------------+
| voluptate    |
+--------------+
1 row in set (0.00 sec)
```

Q5) Overall trend in inventory levels for each product category over the month of January 2022:



```
mysql> SELECT p.category, AVG(i.inventory_level) as avg_inventory,
    -> COUNT(i.inventory_level) as count_inventory,
    -> SUM(i.inventory_level) as total_inventory,
    -> MAX(i.inventory_level) as max_inventory,
    -> MIN(i.inventory_level) as min_inventory
    -> FROM Products p
    -> JOIN Inventory i ON p.product_id = i.product_id
    -> WHERE MONTH(i.inventory_date) = 1 AND YEAR(i.inventory_date) = 2022
    -> GROUP BY p.category
    -> ORDER BY avg_inventory DESC;
+------------+---------------+-----------------+-----------------+---------------+---------------+
| category   | avg_inventory | count_inventory | total_inventory | max_inventory | min_inventory |
+------------+---------------+-----------------+-----------------+---------------+---------------+
| vel        |       10.0000 |               1 |              10 |            10 |            10 |
| occaecati  |       10.0000 |               1 |              10 |            10 |            10 |
| eos        |       10.0000 |               1 |              10 |            10 |            10 |
| dicta      |        9.0000 |               1 |               9 |             9 |             9 |
| recusandae |        9.0000 |               1 |               9 |             9 |             9 |
| quis       |        8.0000 |               1 |               8 |             8 |             8 |
| ut         |        7.5000 |               2 |              15 |             8 |             7 |
| modi       |        7.0000 |               1 |               7 |             7 |             7 |
| porro      |        7.0000 |               1 |               7 |             7 |             7 |
| laudantium |        6.0000 |               1 |               6 |             6 |             6 |
| autem      |        5.5000 |               2 |              11 |             8 |             3 |
| et         |        5.2500 |               4 |              21 |            10 |             1 |
| vero       |        5.0000 |               1 |               5 |             5 |             5 |
| totam      |        4.0000 |               1 |               4 |             4 |             4 |
| aliquam    |        4.0000 |               1 |               4 |             4 |             4 |
| qui        |        4.0000 |               1 |               4 |             4 |             4 |
| nulla      |        1.0000 |               1 |               1 |             1 |             1 |
| impedit    |        1.0000 |               1 |               1 |             1 |             1 |
| nisi       |        1.0000 |               1 |               1 |             1 |             1 |
| aut        |        0.0000 |               1 |               0 |             0 |             0 |
+------------+---------------+-----------------+-----------------+---------------+---------------+
20 rows in set (0.01 sec)
```

Conclusion:

- Help us to identify which products are currently overstocked.
- Gives a view of our current inventory situation broken down by category.
- Help us understand how our inventory levels fluctuate over time and identify any seasonal trends.
- Help us spot products that are selling well or have issues with restocking.
- Help us make informed decisions about inventory management in the future.