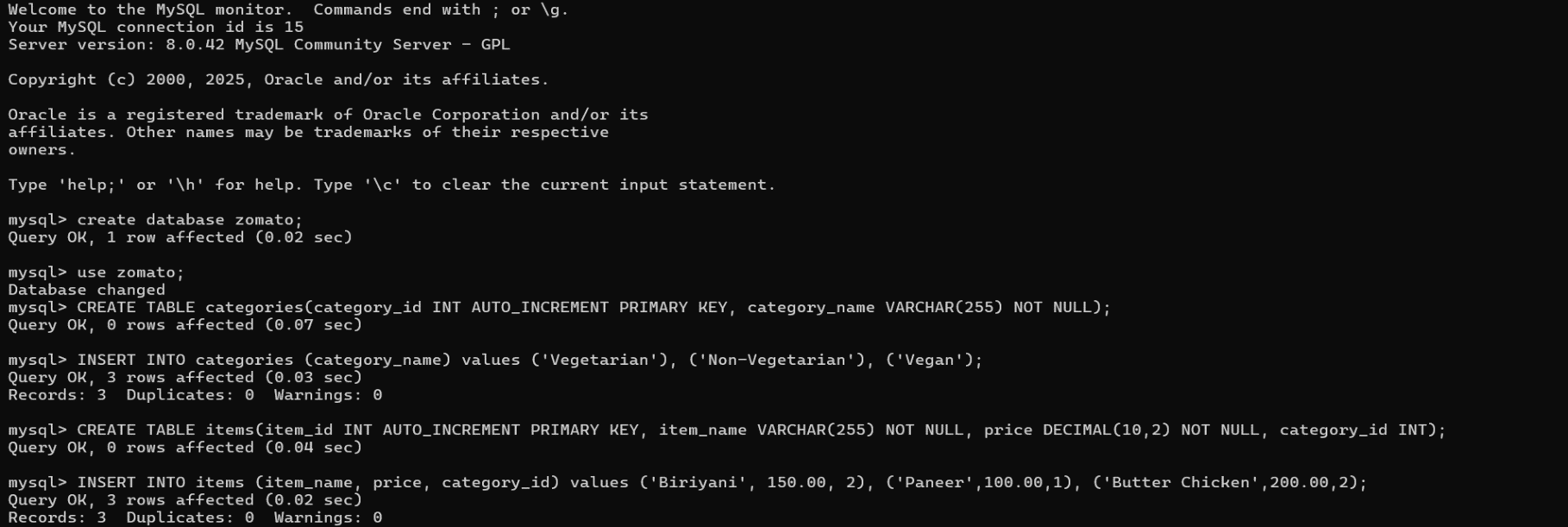
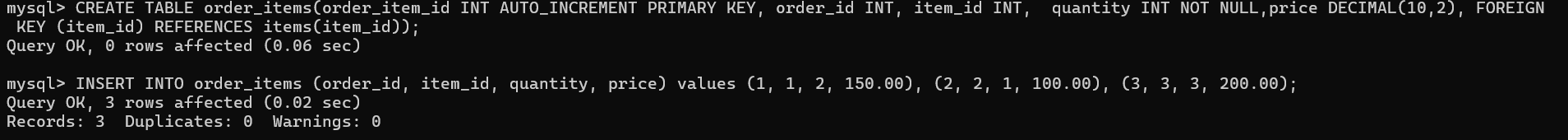
**SQL PROJECT**

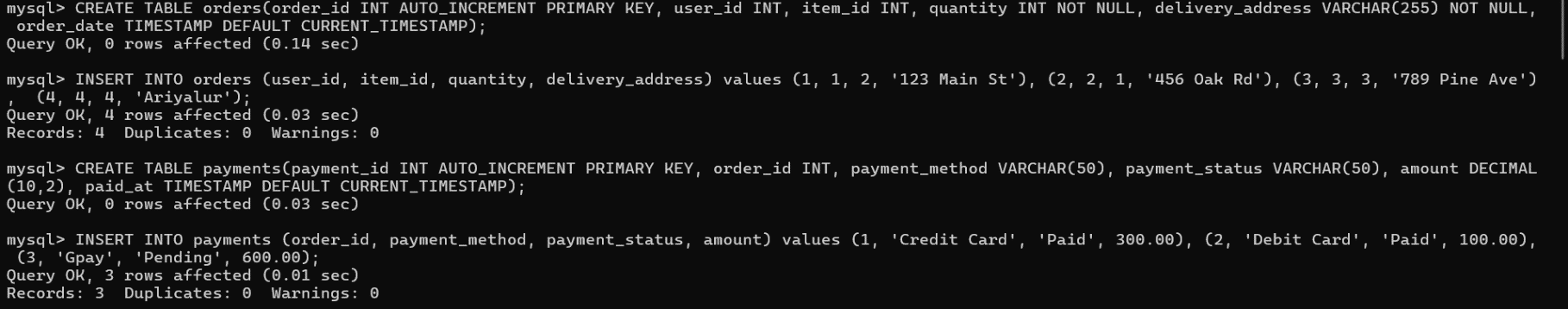
# **RESTAURANT’S ONLINE BOOKING APP (LIKE-ZOMATO PLATFORM)**

CREATING TABLES AND INSERTING DATA TO THOSE TABLES

Step 1: Creating a database named “zomato” and creating tables named “categories” and “items”. In this the categories tables has the categories of food like Veg, Non-veg, and Vegan and the table items has the list of items, its price, and its category id as shown in the below figure:

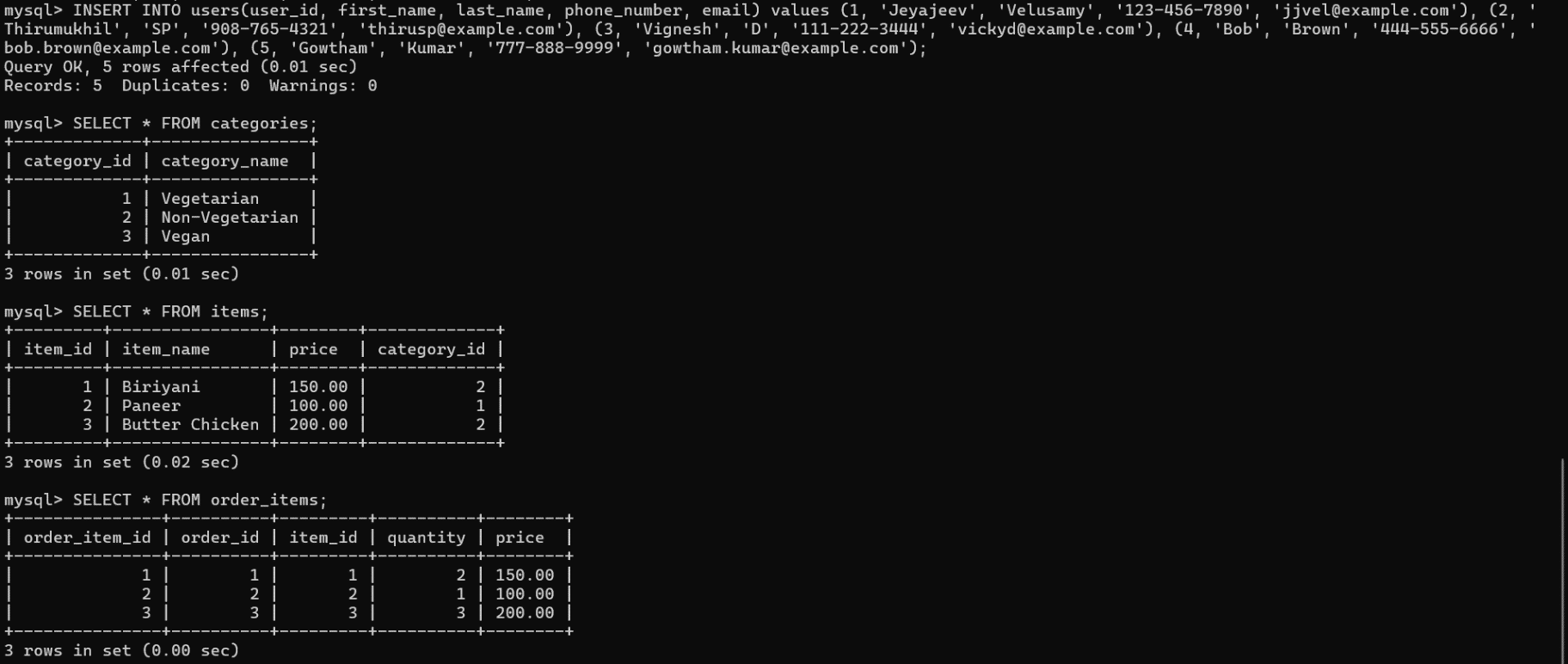
Step 2: Creating table named “order\_items” with details order item id, order id, item id referenced from the table items, quantity and the price. The values corresponding to this are inserted to the table as show in the code below:

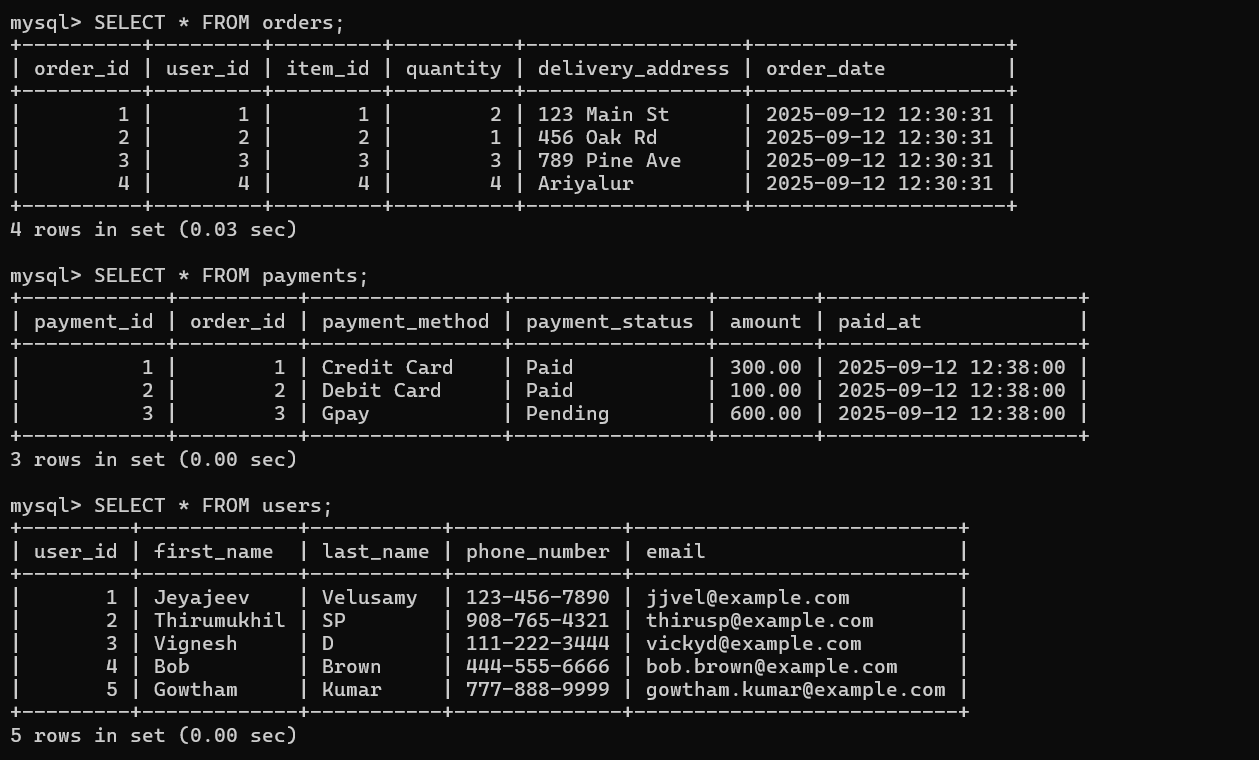
Step 3: Creating table named “orders” and “payments”. The table orders with details order id, user id, item id, quantity, delivery address and the date of order and the table payments has the details like payment id, order id, payment method (Debit card/Credit card/ Gpay), payment status (Paid/Pending/Failed), amount to be paid and the date of payment. The values corresponding to these tables are inserted to the table as show in the code below:



Step 4: Creating table named “users” with details of users who make orders like user id, first name, last name, phone number, email. The values corresponding for this table are also inserted to the table and the full structure of tables which values are as show in the code below:

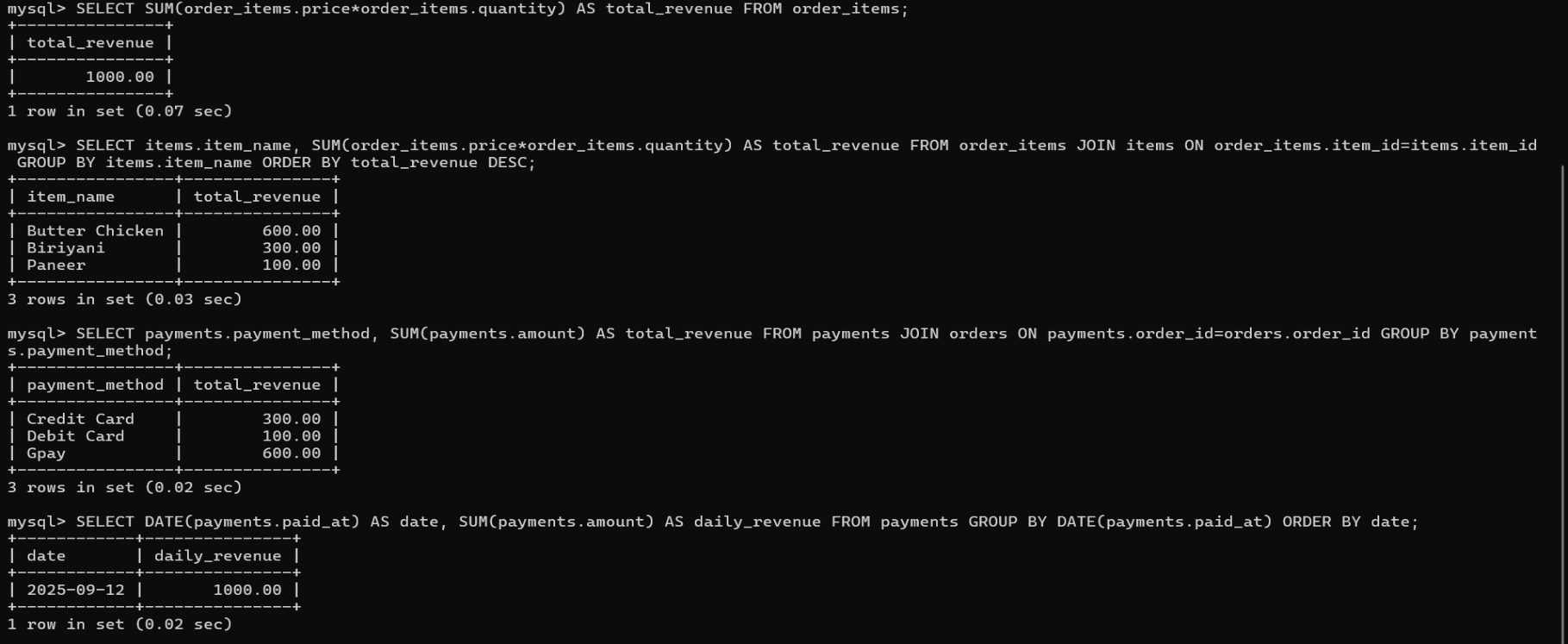






ETL PROCESS USING THESE TABLES AND DATA

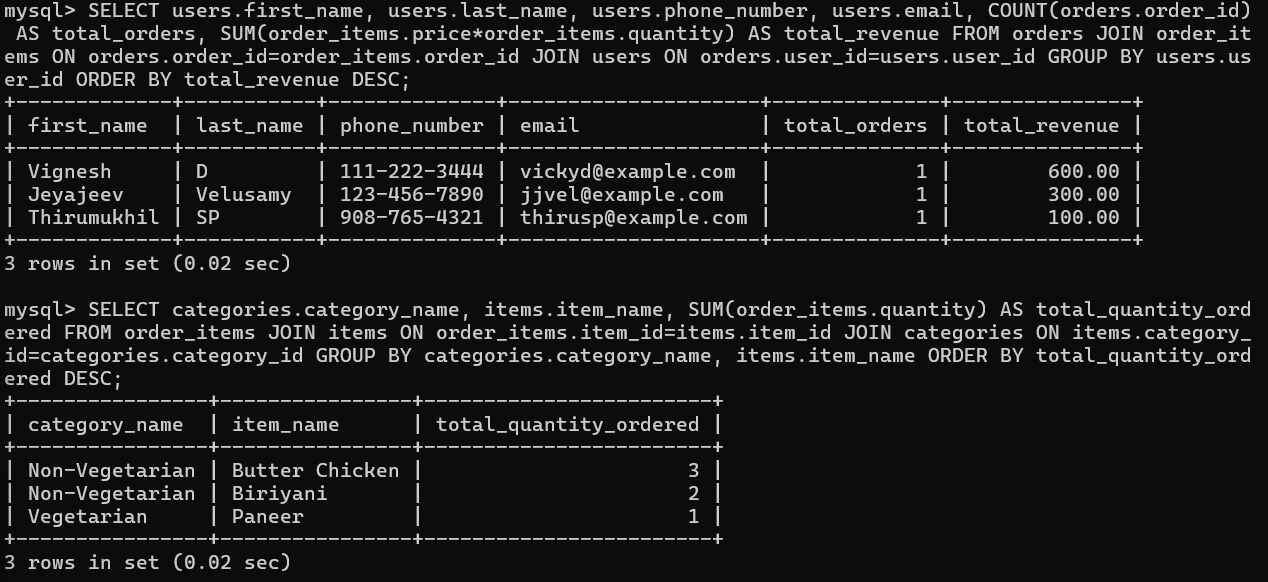
Here, various ETL processes have been performed using the above tables and data. The processes are as follows:

1.Total Revenue from All orders, 2. Revenue by Item, 3. Revenue by Payment Method, 4. Total Revenue by Date

In the above image, the first sql command calculates the complete revenue earned from all order items by summing each item’s price multiplied by its quantity, showing total revenue overall, the second displays each item’s name with total revenue earned by summing price times quantity across orders, grouped by item, then ordered descending by revenue, the third shows total revenue collected for each payment method by summing all amounts paid per method, grouped by method, giving revenue comparison between different methods, and the fourth command lists each payment date with the total revenue earned on that date by summing payment amounts per date, providing daily revenue tracking trends.

The next processes are:

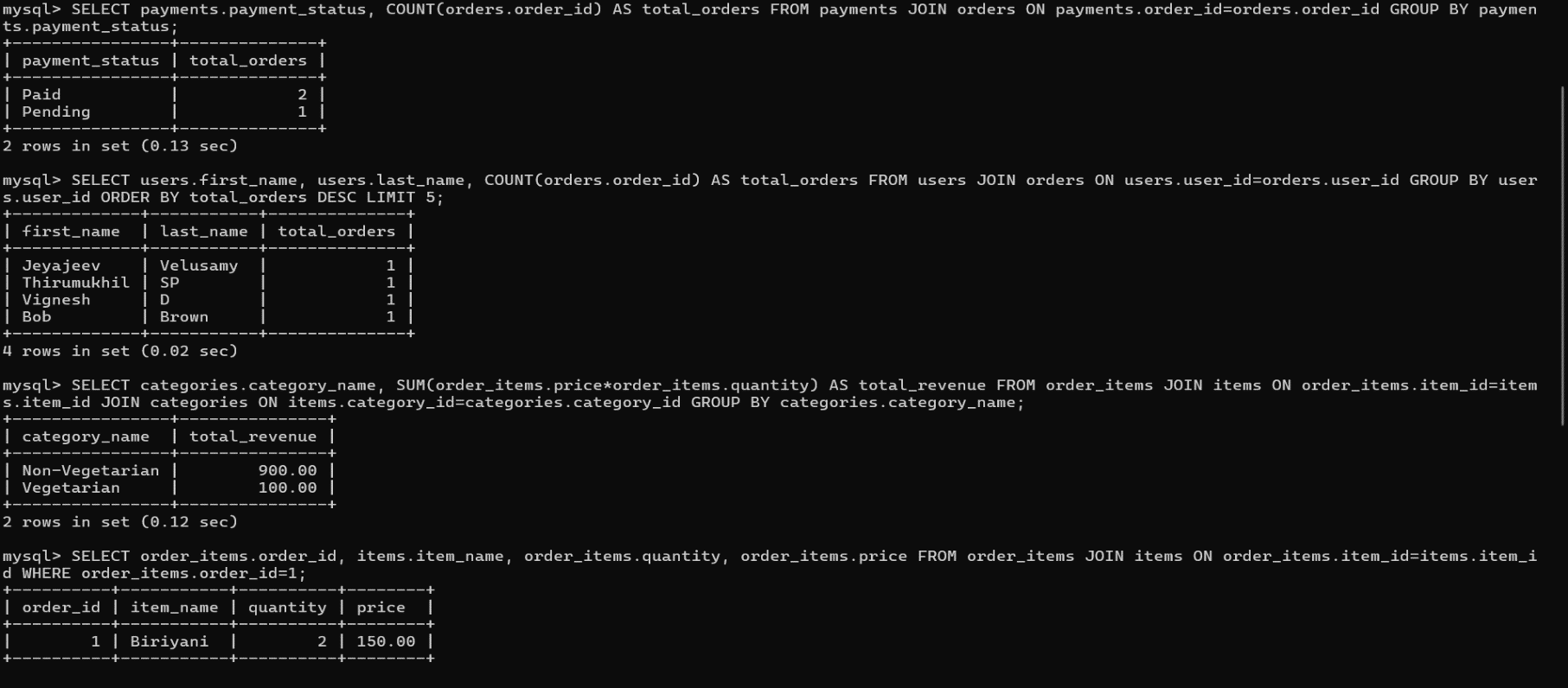
5. Total Orders and Revenue by User, 6. Items Ordered by Category



In the above image, the first sql command shows each user’s name, contact info, how many orders they made, and how much money they spent, sorted from highest to lowest spender and the second command shows categories and item names with total ordered quantity, grouped by category and item, summing quantities to highlight popular items under each category.

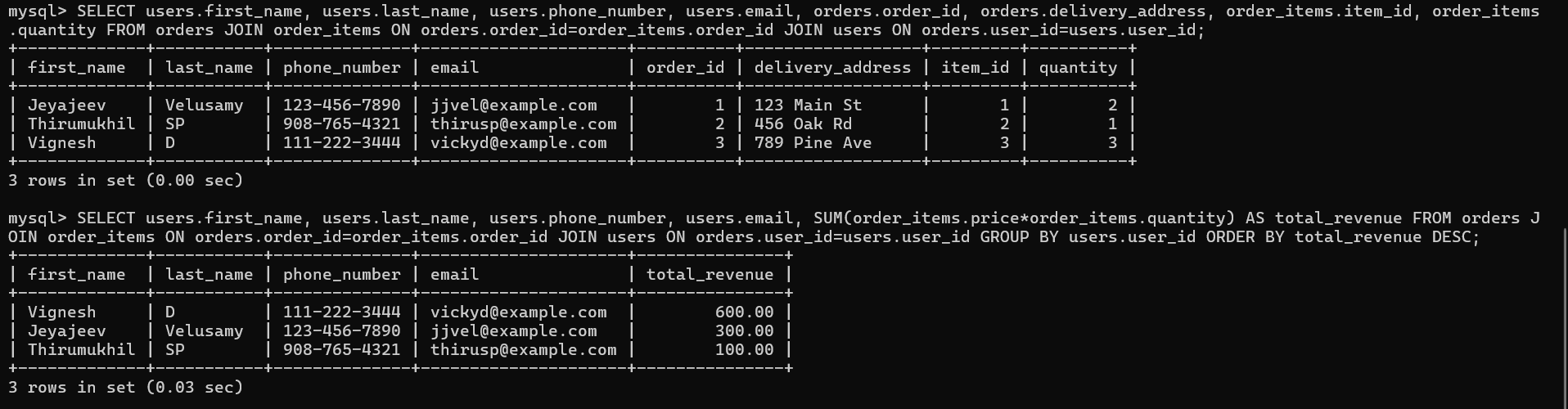
The next processes are:

7. Orders by Payment Status, 8. Users with Most Orders, 9. Revenue by Category, 10. Items Purchased in Specific Order



In the above image, the first sql command counts orders for each payment status by grouping orders with their payment status, providing total orders per status to analyze payment completion performance. The second command, lists top five users with first and last names, counting their total orders placed, grouped by user, ordered descending to show frequent buyers. The third command, displays each category with summed revenue across all items in it, grouping by category to show which categories generate the most revenue overall and the fourth command lists items, quantities, and prices for a particular order ID, showing detailed breakdown of products bought within that single order by customer.

The next processes are:

11. Customer Details with Orders, 12. Revenue by Customer.

In the image above, the first command shows each customer’s contact details, order ID, delivery address, item ID and quantity, joining orders and customers to display full order information together. The second command lists each customer’s contact details with their total revenue contribution by summing price times quantity for all items ordered, grouped by customer user ID.