**Demo on Confluent Kafka Producer and Consumers:**

Step 1:

Create table demo.product in MySQL database

Insert sample records into demo.product

create table demo.product

(

id INT,

`name` VARCHAR(50),

category VARCHAR(50),

price float,

last\_updated timestamp

);

INSERT INTO demo.product (id, name, category, price, last\_updated) VALUES

(31, 'Wireless Mouse', 'Electronics', 25.99,'2025-02-12'),

(32, 'Mechanical Keyboard', 'Electronics', 79.49, '2025-02-12'),

(33, 'Noise Cancelling Headphones', 'Electronics', 199.99, '2025-02-12'),

(34, 'Smart Watch', 'Wearables', 149.99, '2025-02-12'),

(35, '4K Monitor', 'Electronics', 349.99, '2025-02-12'),

(36, 'Bluetooth Speaker', 'Electronics', 59.99, '2025-02-12'),

(37, 'Laptop Stand', 'Accessories', 29.99, '2025-02-12'),

(38, 'External SSD', 'Storage', 129.99, '2025-02-12'),

(39, 'Gaming Chair', 'Furniture', 299.99, '2025-02-12'),

(40, 'Smartphone Gimbal', 'Accessories', 99.99, '2025-02-12'),

(41, 'Wireless Headphones', 'Electronics', 299.99, '2025-02-12'),

(42, 'Office desk', 'Furniture', 299.99, '2025-02-12');

Commit;

INSERT INTO demo.product (id, name, category, price, last\_updated) VALUES

(43, 'Mobile', 'Electronics', 299.99, '2025-02-13'),

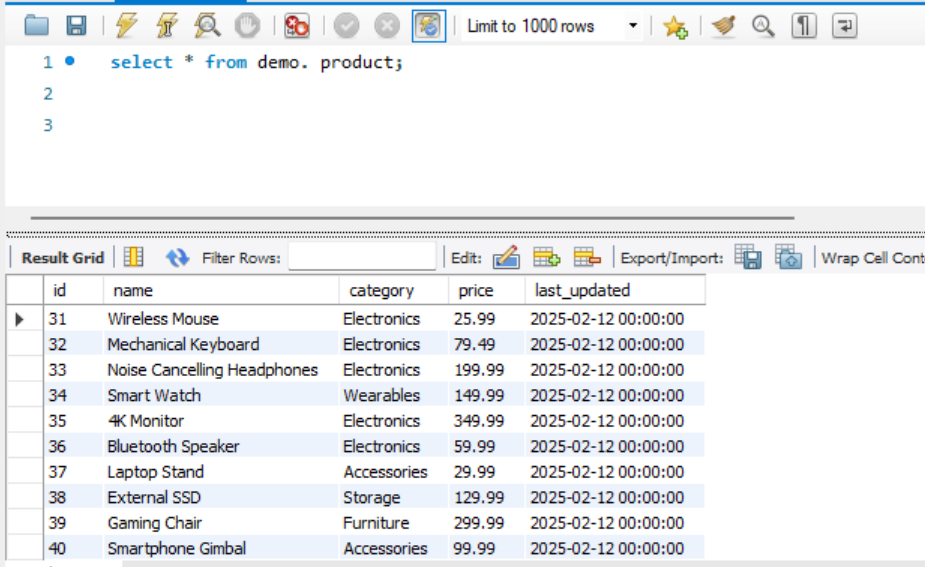
(44, 'Sofeset', 'Furniture', 399.99, '2025-02-13'),

(45, 'Smartphone Gimbal', 'Accessories', 199.99, '2025-02-13'),

(46, 'Television', 'Electronics', 399.99, '2025-02-14'),

(47, 'Bedframes', 'Furniture', 299.99, '2025-02-14');

Commit;



Step 2:

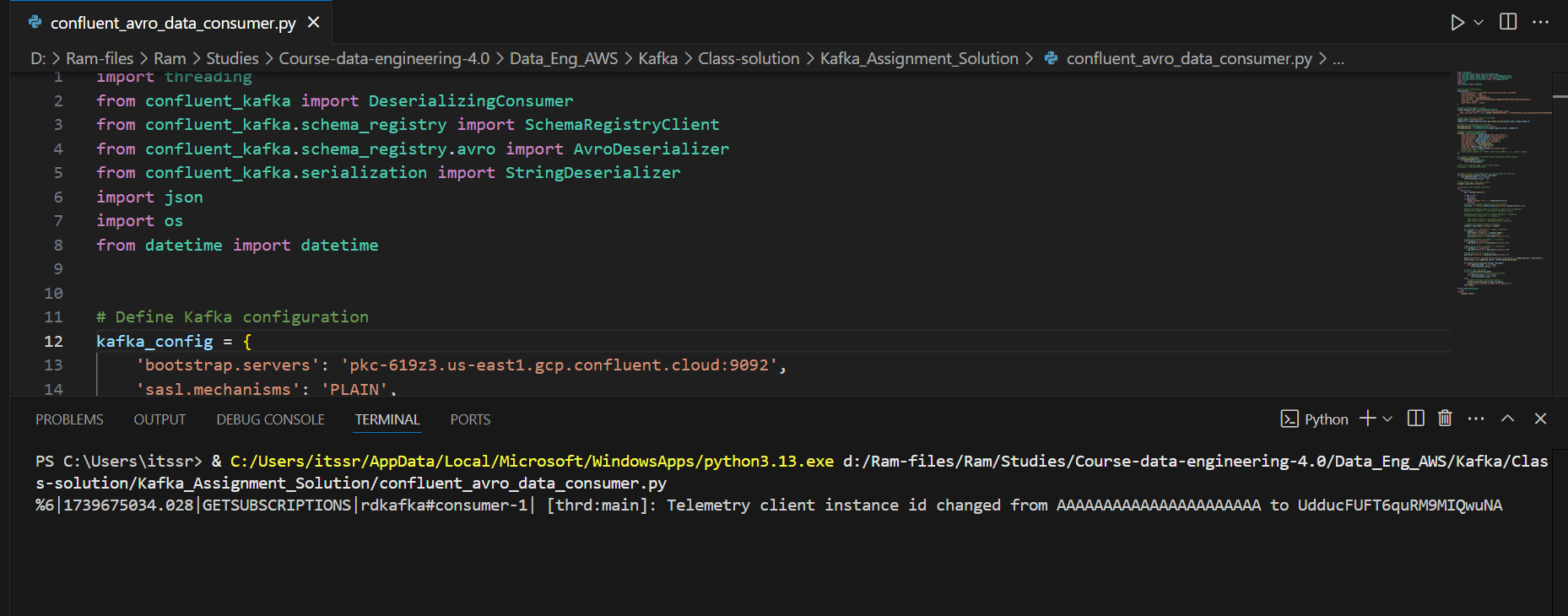
Kafka Producer and Consumer scripts:





Step 3:

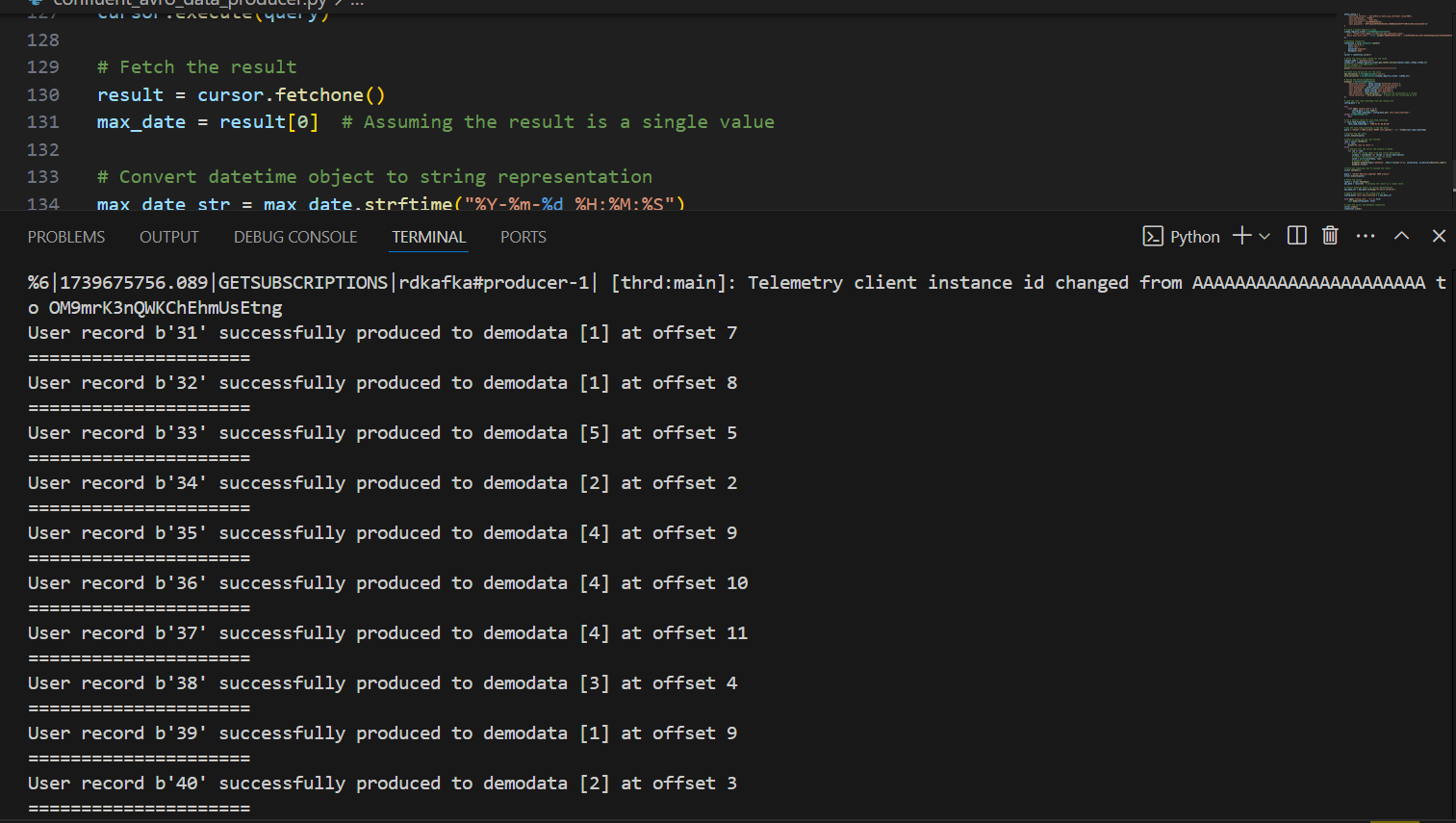
After inserting the above records in MySQL demo.product table. Run all consumer python code in 5 different sessions as below ( only sample one session is open here ).



Step 4:

Run the python producer code and see the 12 records distributed across the multiple consumers under a group id

confluent\_avro\_data\_producer.py – producer execution to load all 12 records

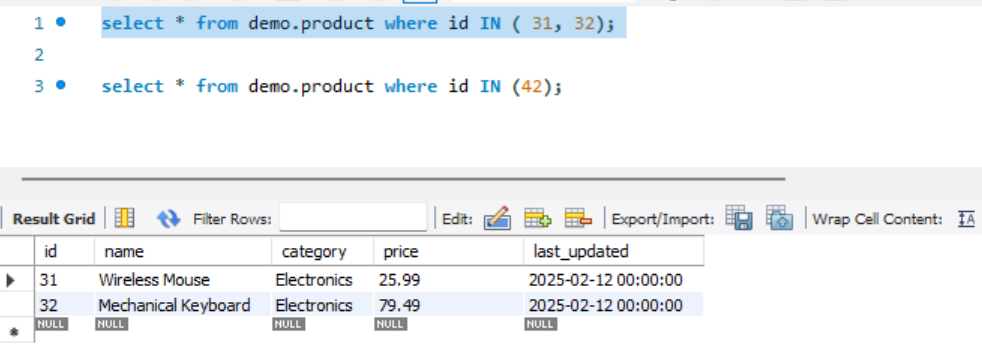


Consumers has started consuming the records in different sessions invoked.

we can see the business logic like

1. Increase price by 20% for electronics

For ex: if you look at id =31, the original price is 25.99 , after running the consumer the price has increased to 31.19



%6|1739675050.373|GETSUBSCRIPTIONS|rdkafka#consumer-1| [thrd:main]: Telemetry client instance id changed from AAAAAAAAAAAAAAAAAAAAAA to wT0p/a+BRKmFYqdS62iZrQ

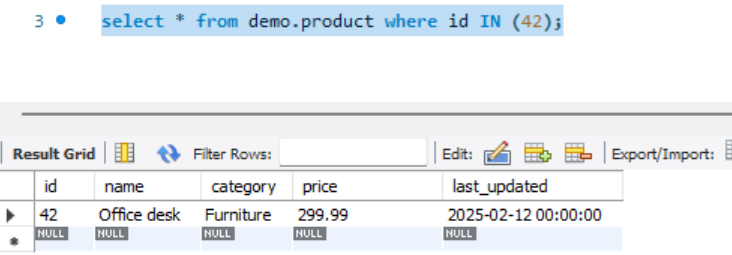
Successfully consumed record with key 31 and value {'id': 31, 'name': 'Wireless Mouse', 'category': 'ELECTRONICS', 'price': 31.19, 'last\_updated': datetime.datetime(2025, 2, 12, 0, 0, tzinfo=datetime.timezone.utc)}

Successfully consumed record with key 32 and value {'id': 32, 'name': 'Mechanical Keyboard', 'category': 'ELECTRONICS', 'price': 95.39, 'last\_updated': datetime.datetime(2025, 2, 12, 0, 0, tzinfo=datetime.timezone.utc)}

json\_string data is added to the JSON file.

1. Apply 50% discount if category is 'furniture'

For ex: if look at id =42 , the Furniture ( office desk) price is 299.99, after running the consumer the price has decreased with 50% discount



Successfully consumed record with key 42 and value {'id': 42, 'name': 'Office desk', 'category': 'Furniture', 'price': 149.99, 'last\_updated': datetime.datetime(2025, 2, 12, 0, 0, tzinfo=datetime.timezone.utc)}

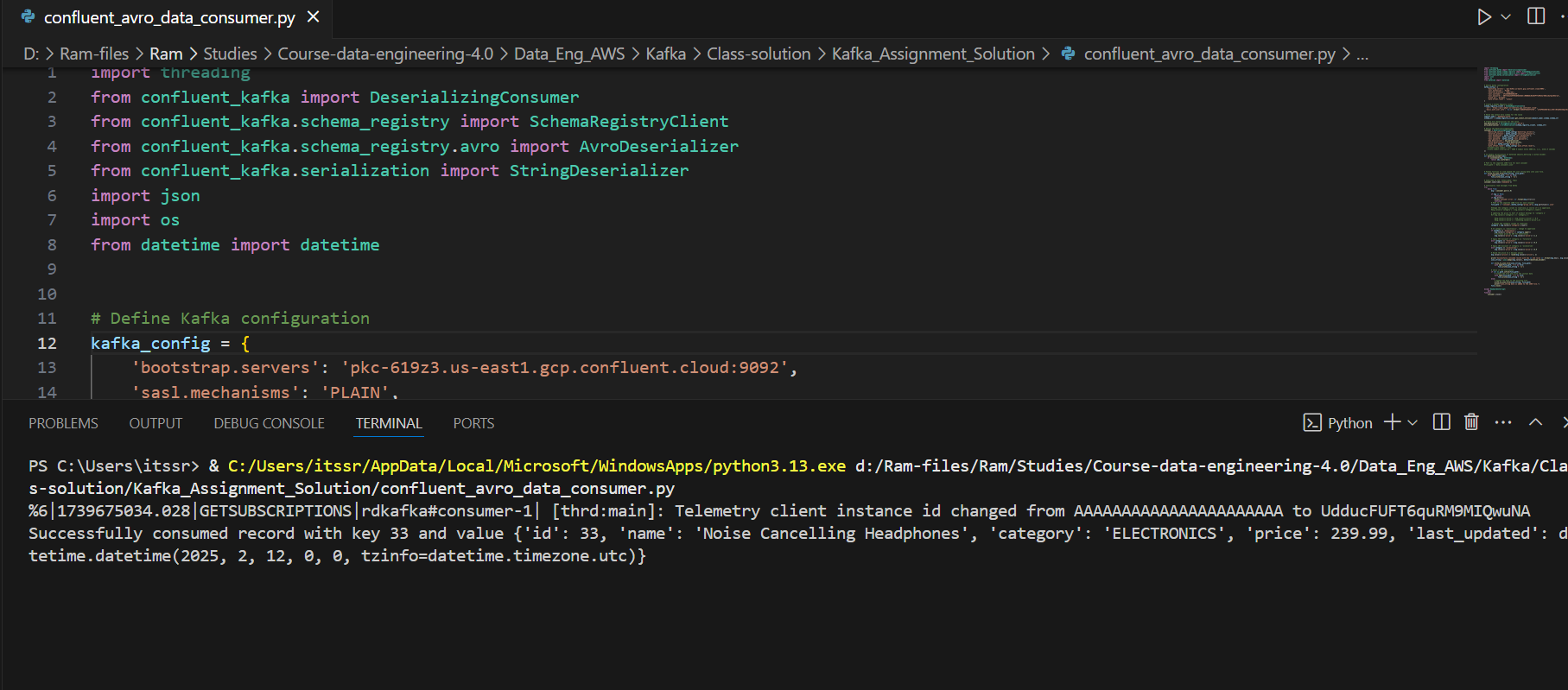
json\_string data is added to the JSON file.

1. Apply 10% discount if category is 'accessories'

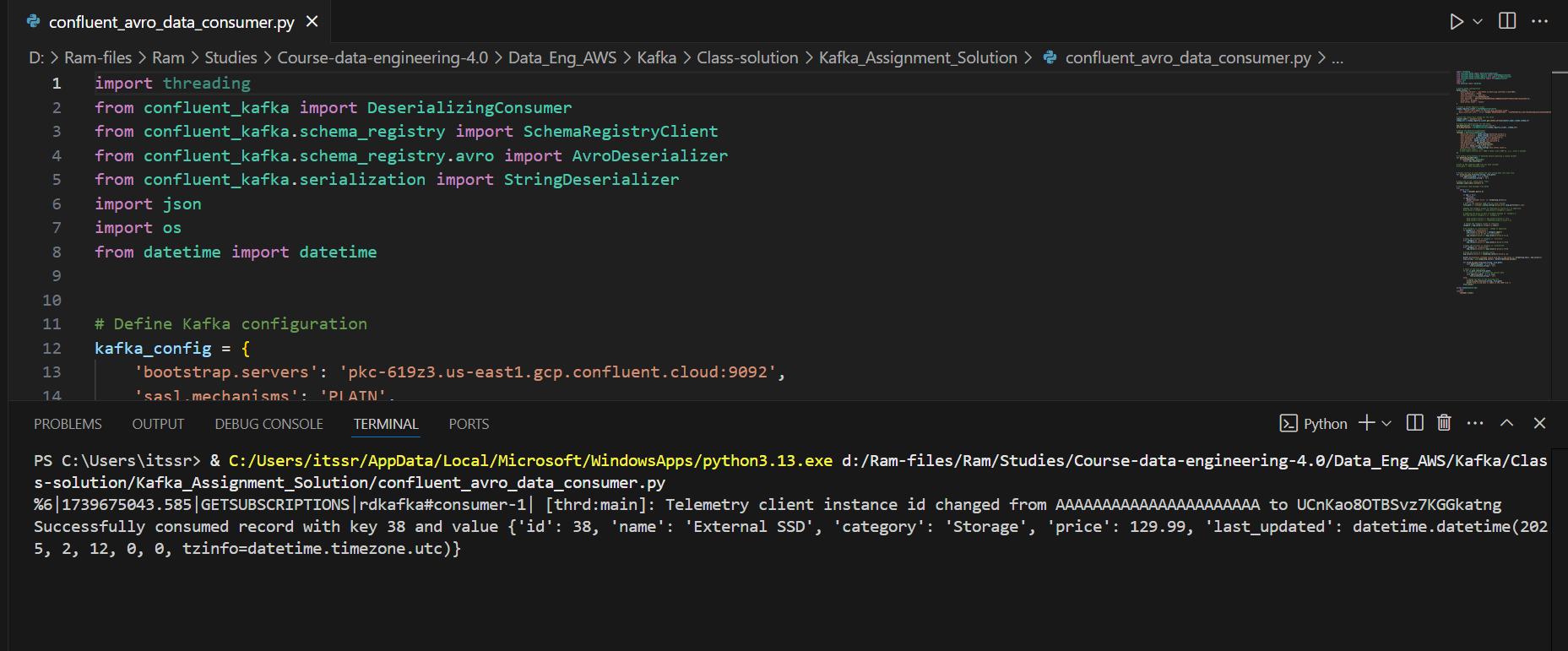
For ex: if look at id=40, the Accessories price is 99.99. after running the consumer, the price has decreased with 10% discount.

Successfully consumed record with key 40 and value {'id': 40, 'name': 'Smartphone Gimbal', 'category': 'Accessories', 'price': 89.99, 'last\_updated': datetime.datetime(2025, 2, 12, 0, 0, tzinfo=datetime.timezone.utc)}

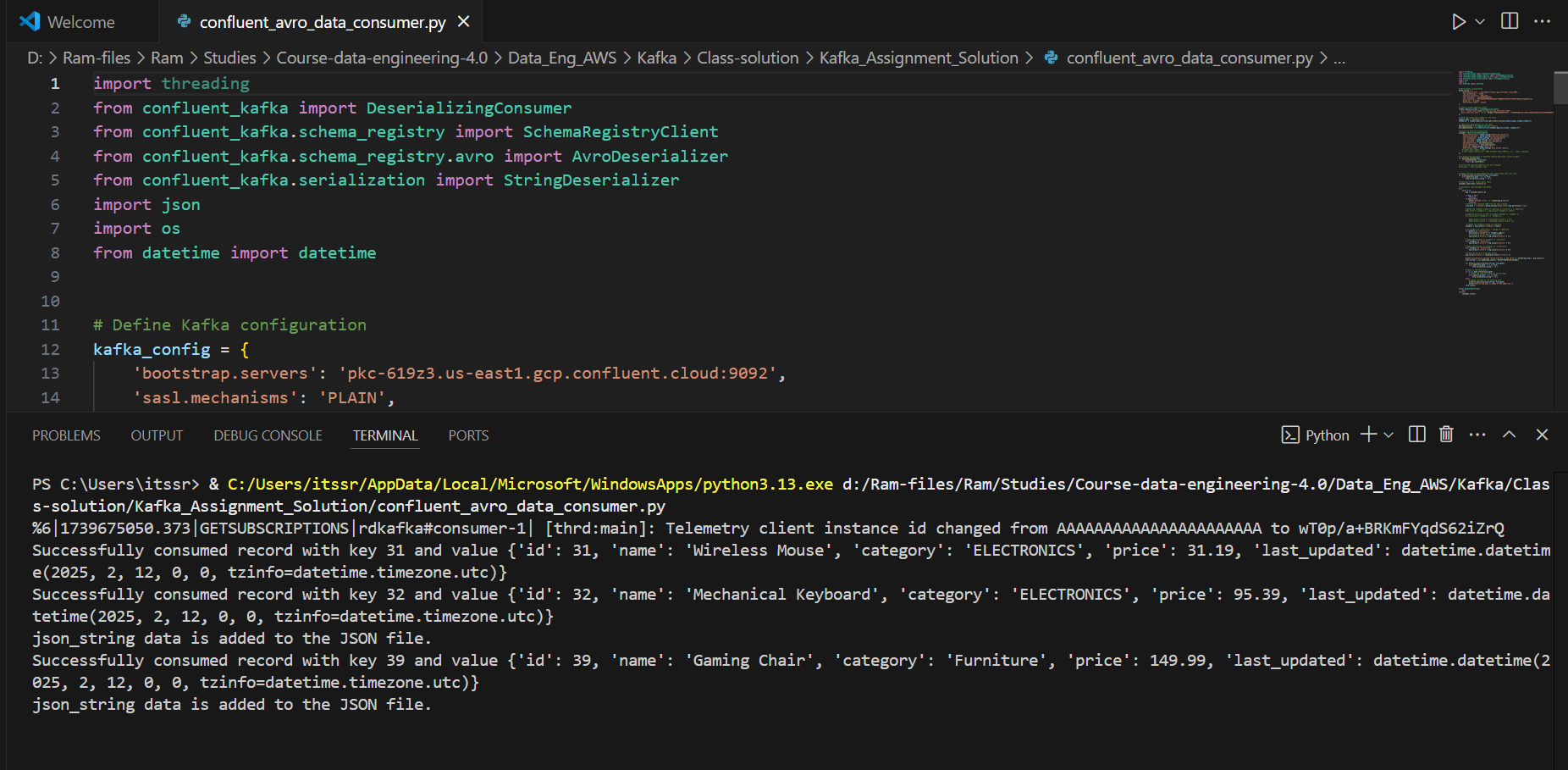
json\_string data is added to the JSON file.

confluent\_avro\_data\_consumer-1 : 

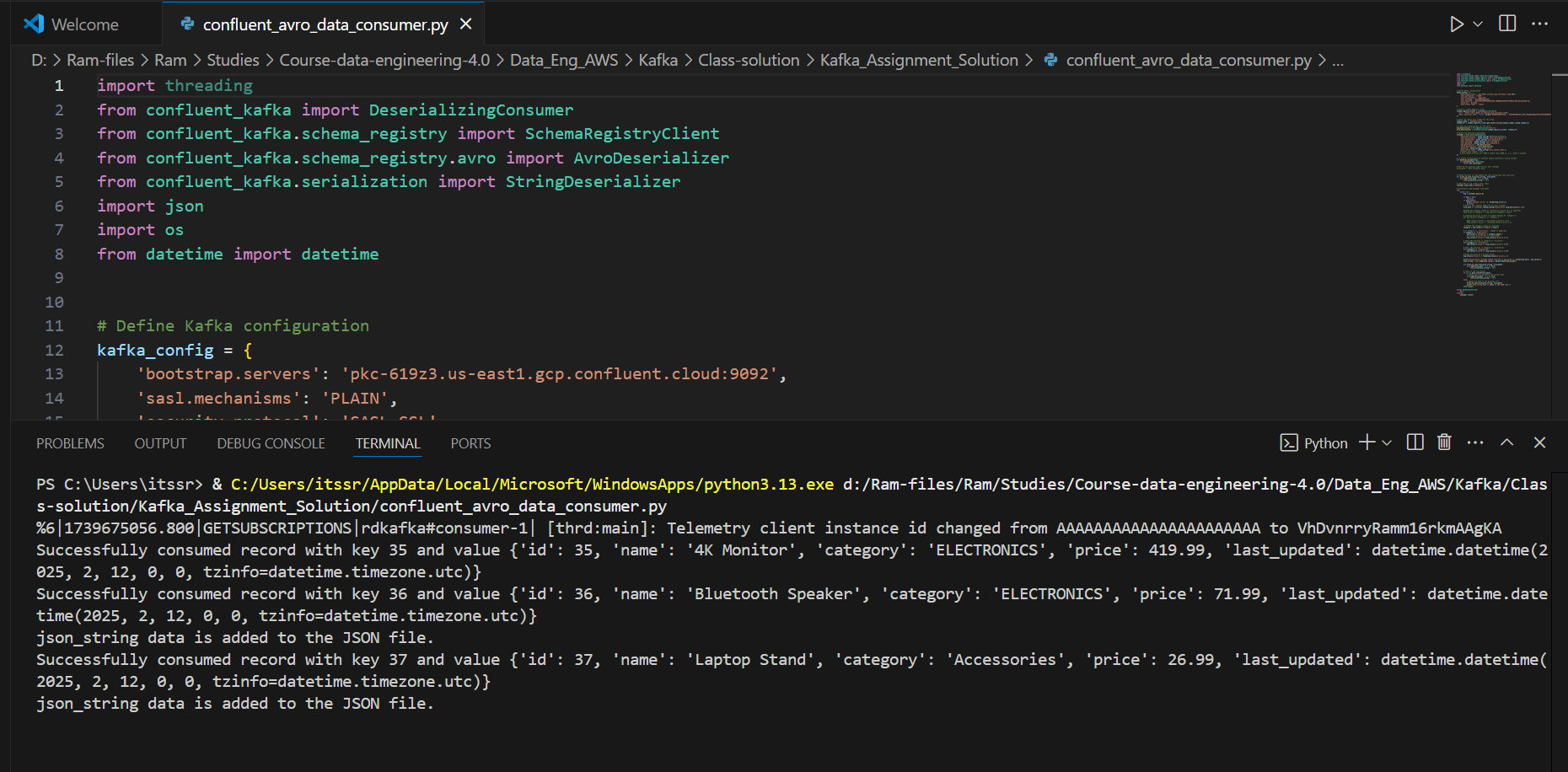
confluent\_avro\_data\_consumer-2



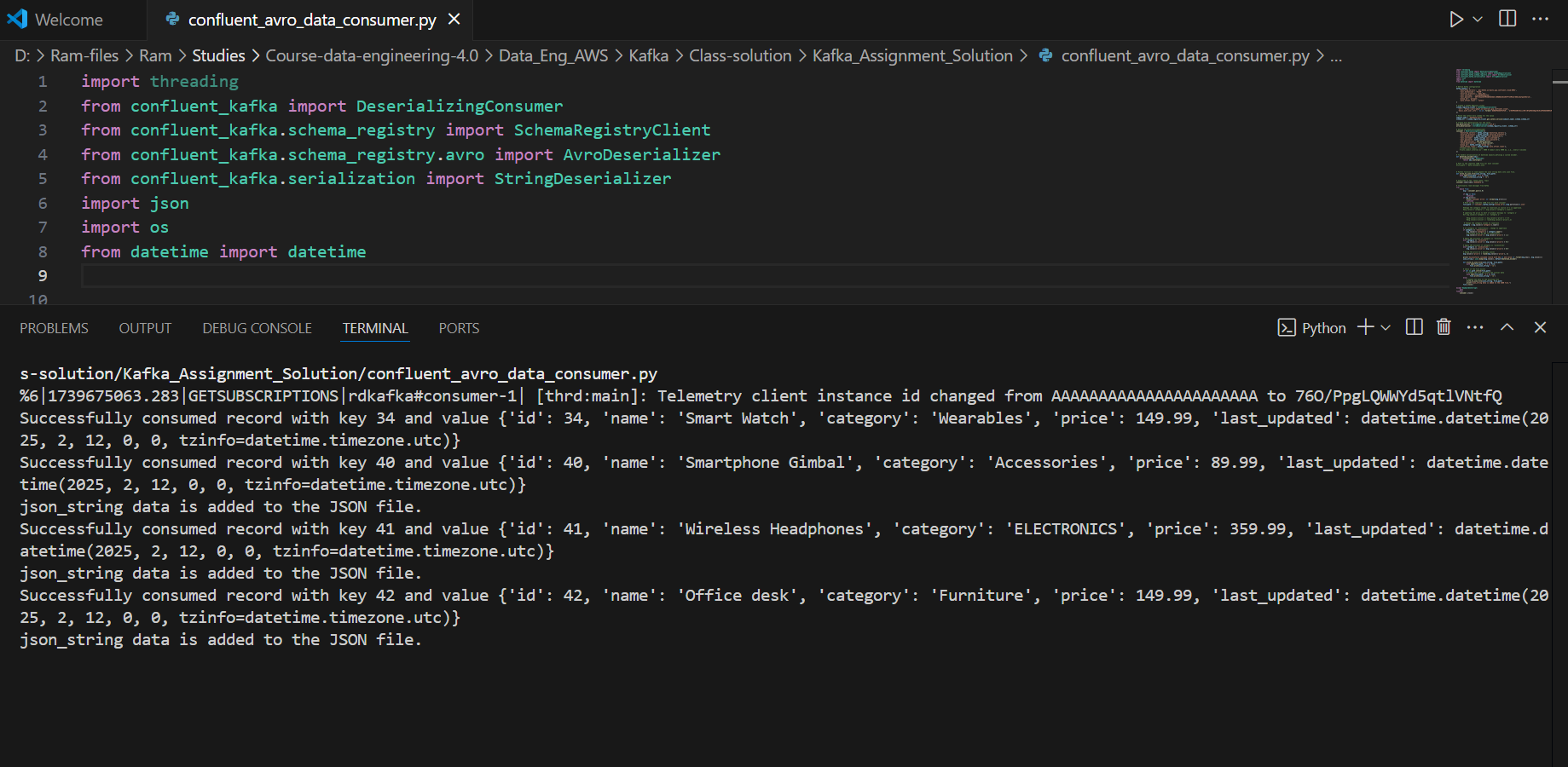
confluent\_avro\_data\_consumer-3



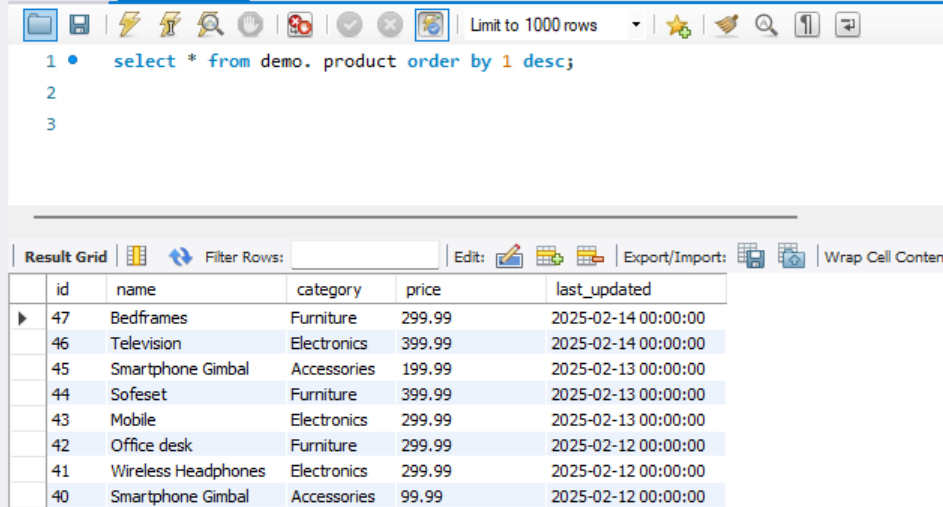
confluent\_avro\_data\_consumer-4



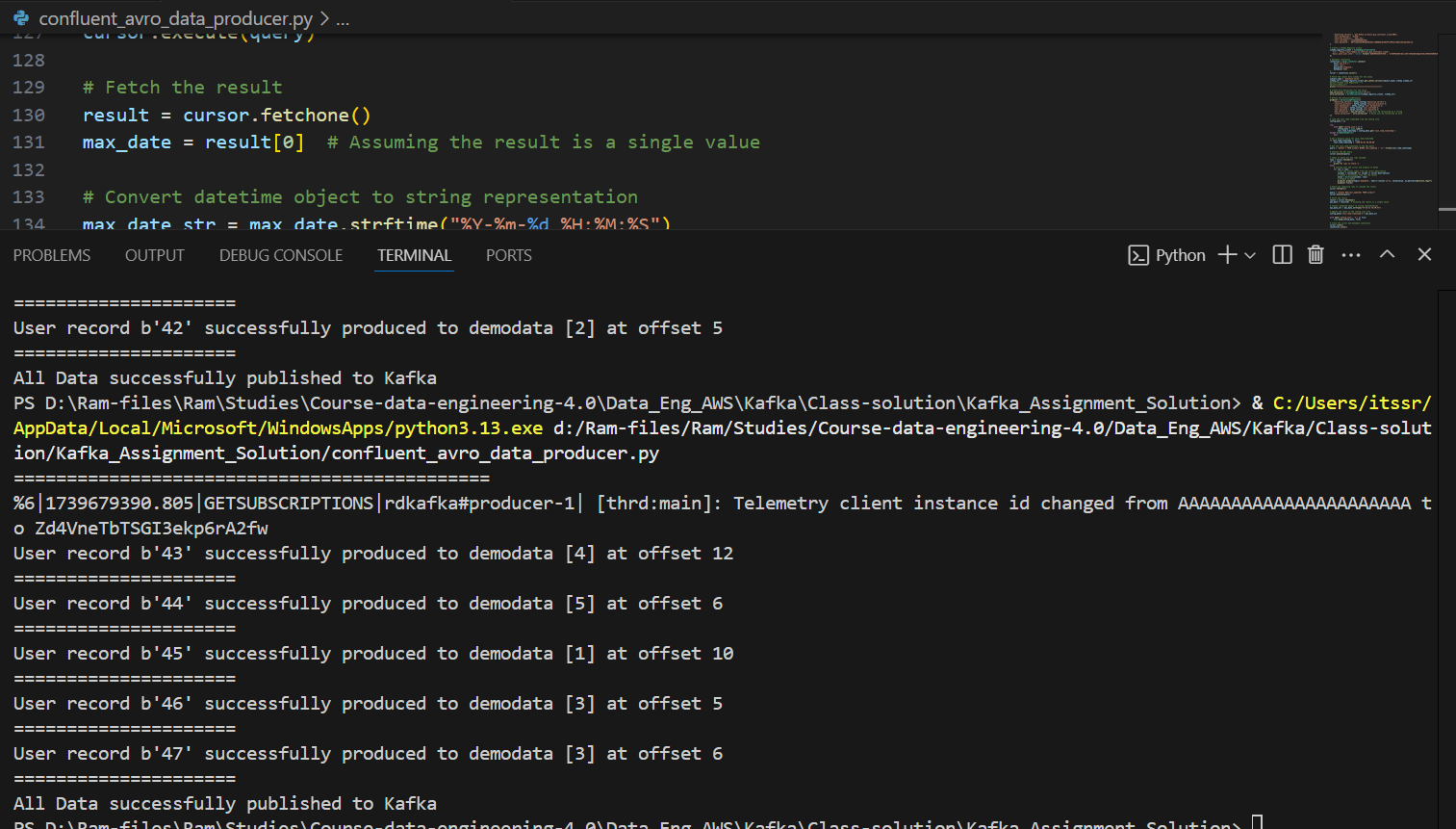
confluent\_avro\_data\_consumer-5



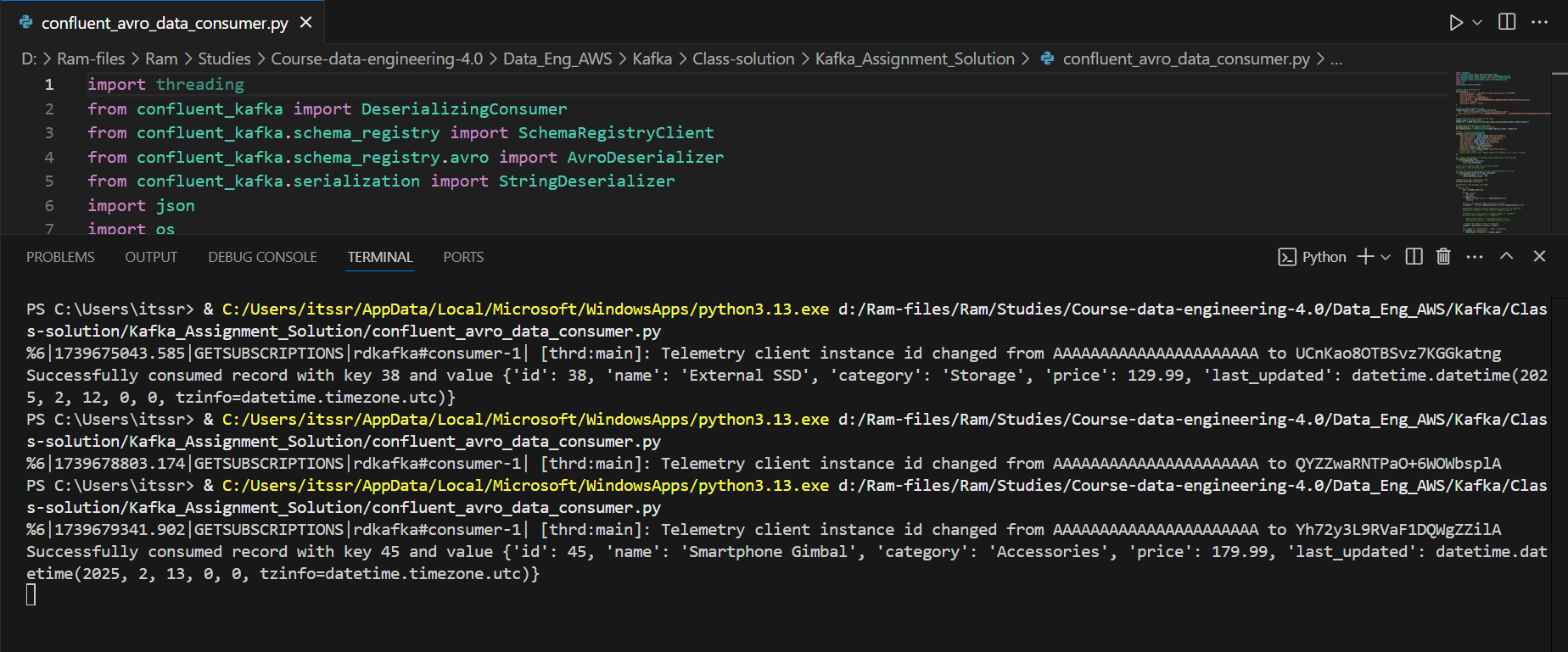
Step 5: Now we are set to run few more inserts into product table



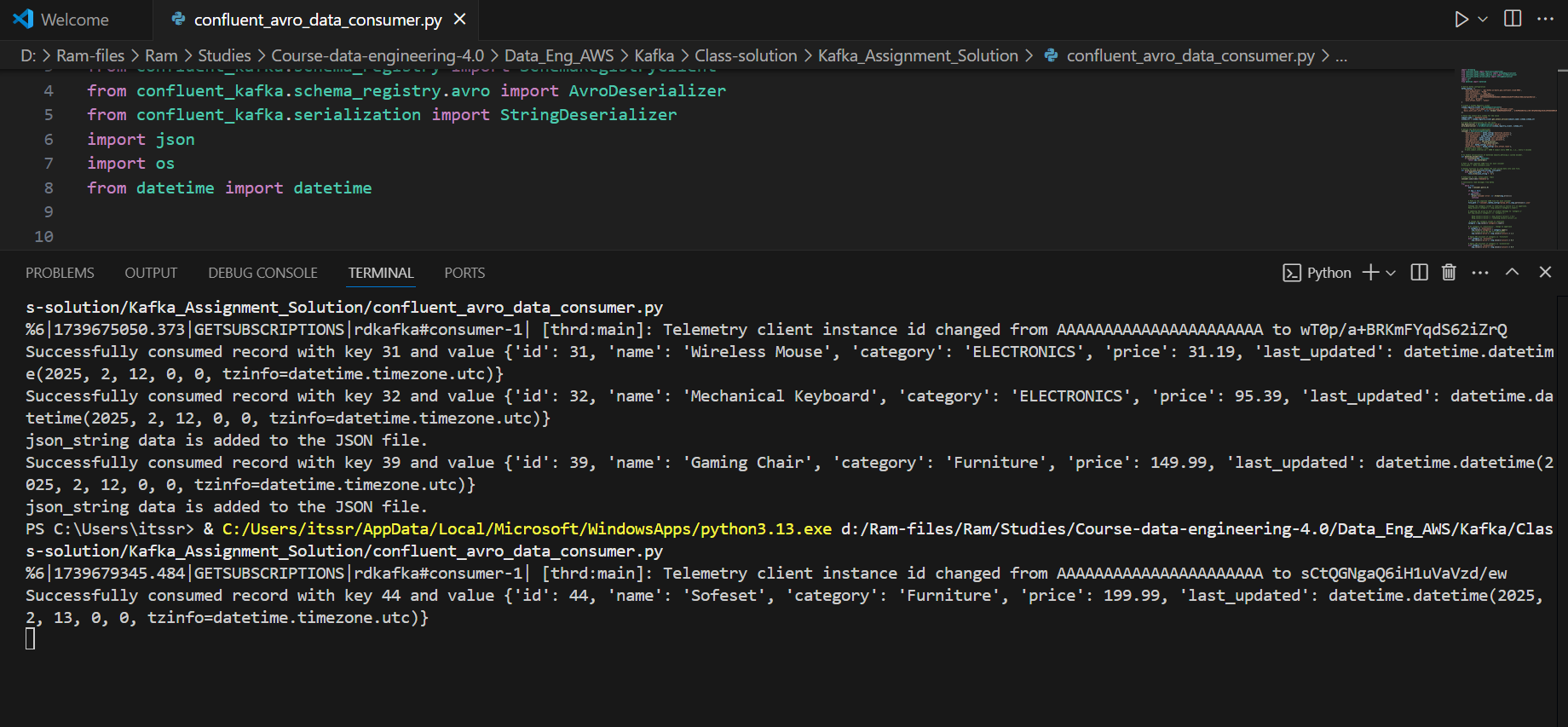
Step 6: let us run producer now to get the newly added records into consumer



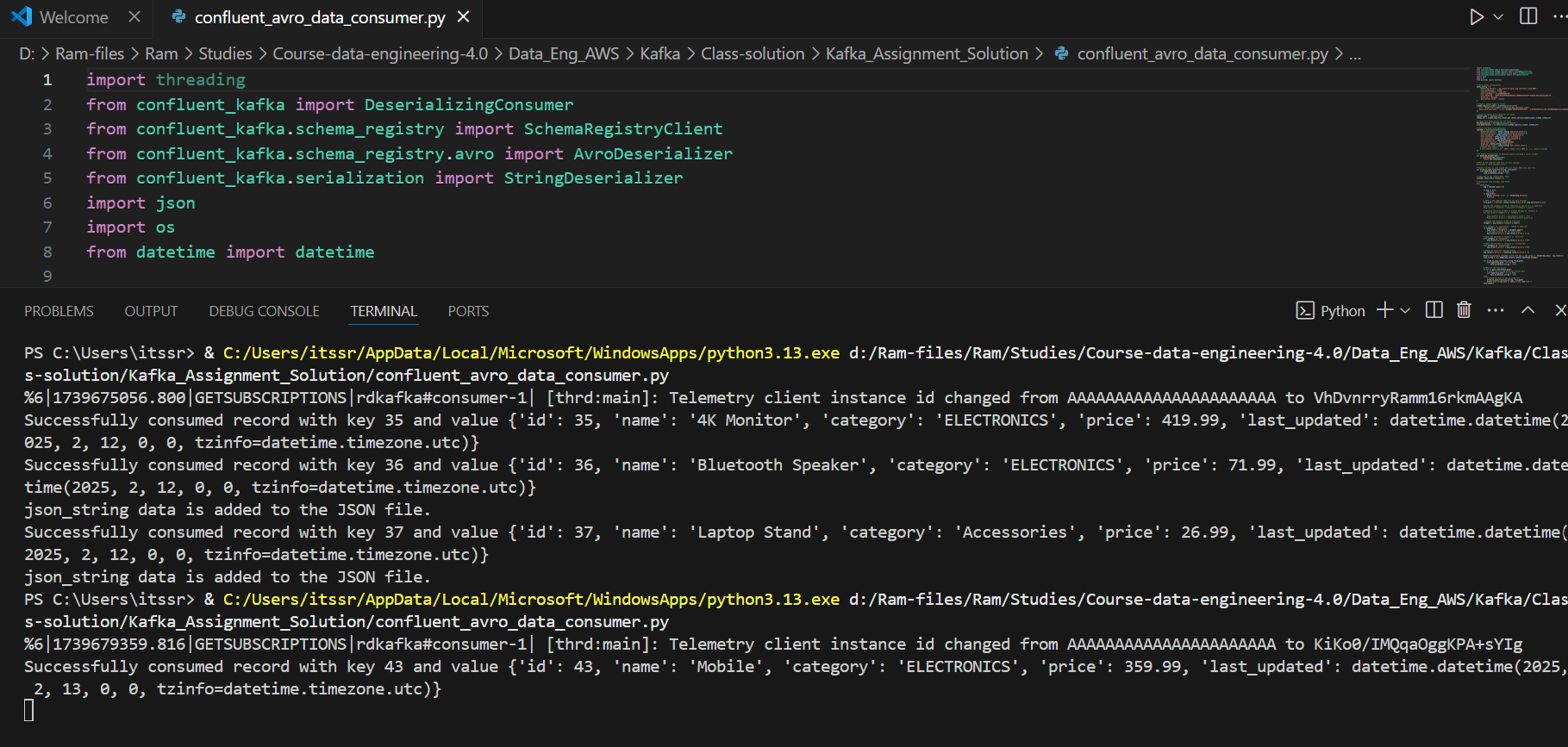
Step 7: Observe the newly added records are consumed by the consumer and below are the details

confluent\_avro\_data\_consumer-1 :

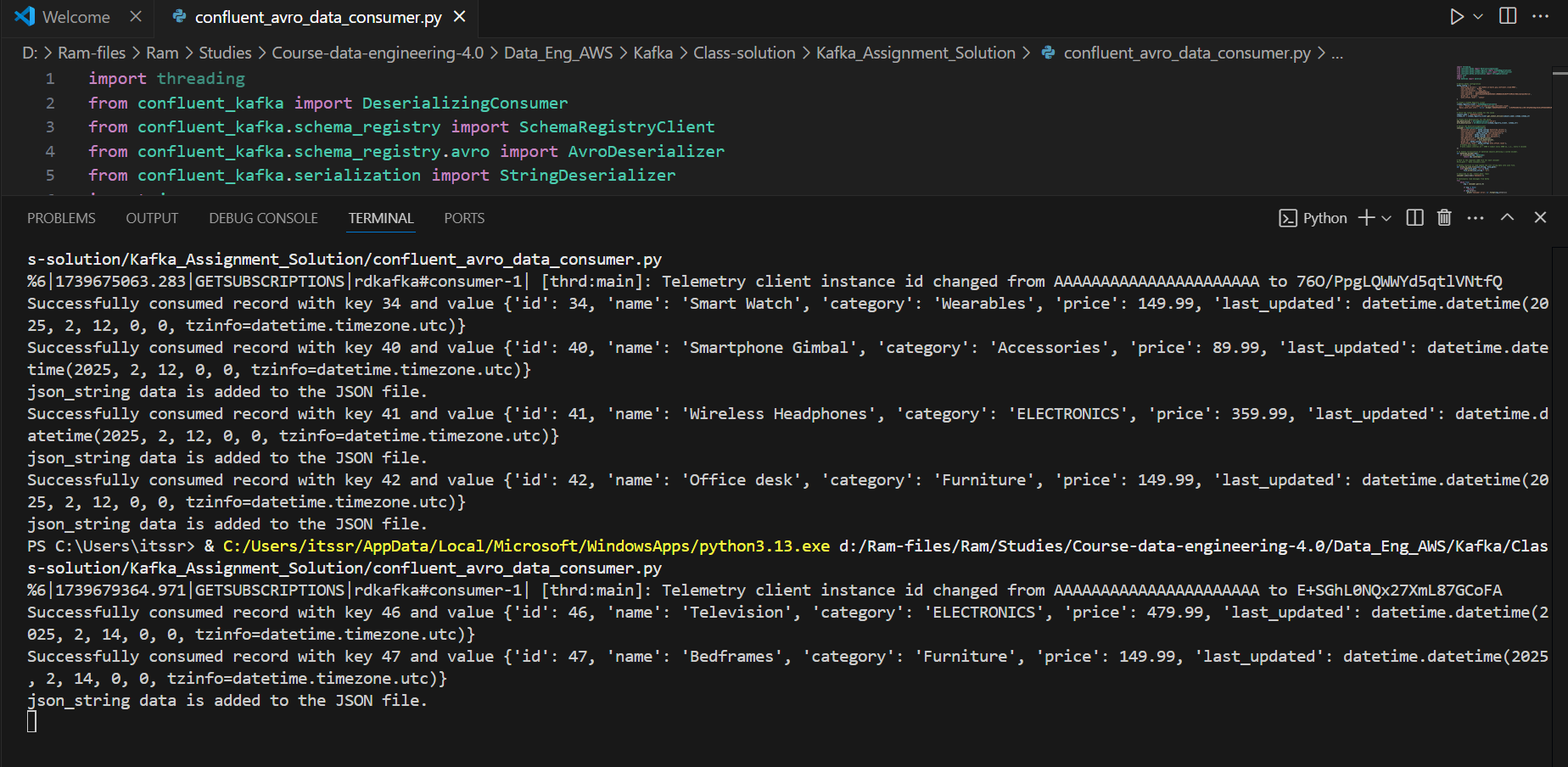
confluent\_avro\_data\_consumer-2 :



confluent\_avro\_data\_consumer-3 :



confluent\_avro\_data\_consumer-4 :



Attached the .json files created during the consumer execution:





Final review in the Topic of the confluent kafka and attached the .csv file of the messages received.

