# HOW WAS YOUR ORDER?

## Querying transaction data from a generated database based on a Food Delivery App use case

CSCI 112-B Group 3

Cheng, Lee, Palo, Soto, Yu

# The use case

A food delivery app handles many transactions a day involving restaurants, riders, and customers. All three have registered information in the app.

Using a database containing the information of all transactions in years 2020-2021, a food delivery platform wishes to know the consumer behavior of its users, including the areas with more active users, the more popular types of cuisine among users, the frequency of their transactions, their general satisfaction level using the app, etc.

# WHY USE MONGODB?

01

FAMILIARITY WITH THE DATABASE TECHNOLOGY

02

PRIOR EXPERIENCE IN SETTING UP THE DEVELOPMENT ENVIRONMENT

03

TIME ALLOCATION TO STUDYING MORE COMPLEX OPERATORS AND COMMANDS

*Demonstration*

# SETTING UP THE DEVELOPMENT ENVIRONMENT

```
sudo nano /etc/yum.repos.d/mongodb-enterprise.repo

[mongodb-enterprise]
name=MongoDB Enterprise Repository baseurl=https://repo.mongodb.com/yum/amazon/2/mongodb-enterprise/5.0/$basearch/
gpgcheck=1
enabled=1
gpgkey=https://www.mongodb.org/static/pgp/server-5.0.asc

sudo yum install -y mongodb-enterprise

sudo systemctl start mongod.service

ss -ant

sudo nano /etc/mongod.conf

sudo systemctl restart mongod.service
```

```
wget -qO - https://www.mongodb.org/static/pgp/server-5.0.asc | sudo apt-key add -

echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/5.0 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-5.0.list

sudo apt update

sudo apt-get install -y mongodb-mongosh

mongo --host <ip address>

python3 -m venv <name venv>

source venv/bin/activate

pip3 install pymongo
```

# DETAILED EXPLANATION OF USE CASE

01

CUSTOMER SIGN-UP

02

RESTAURANT SIGN-UP

03

RIDER-SIGN UP

04

CUSTOMER COMPLETES TRANSACTION

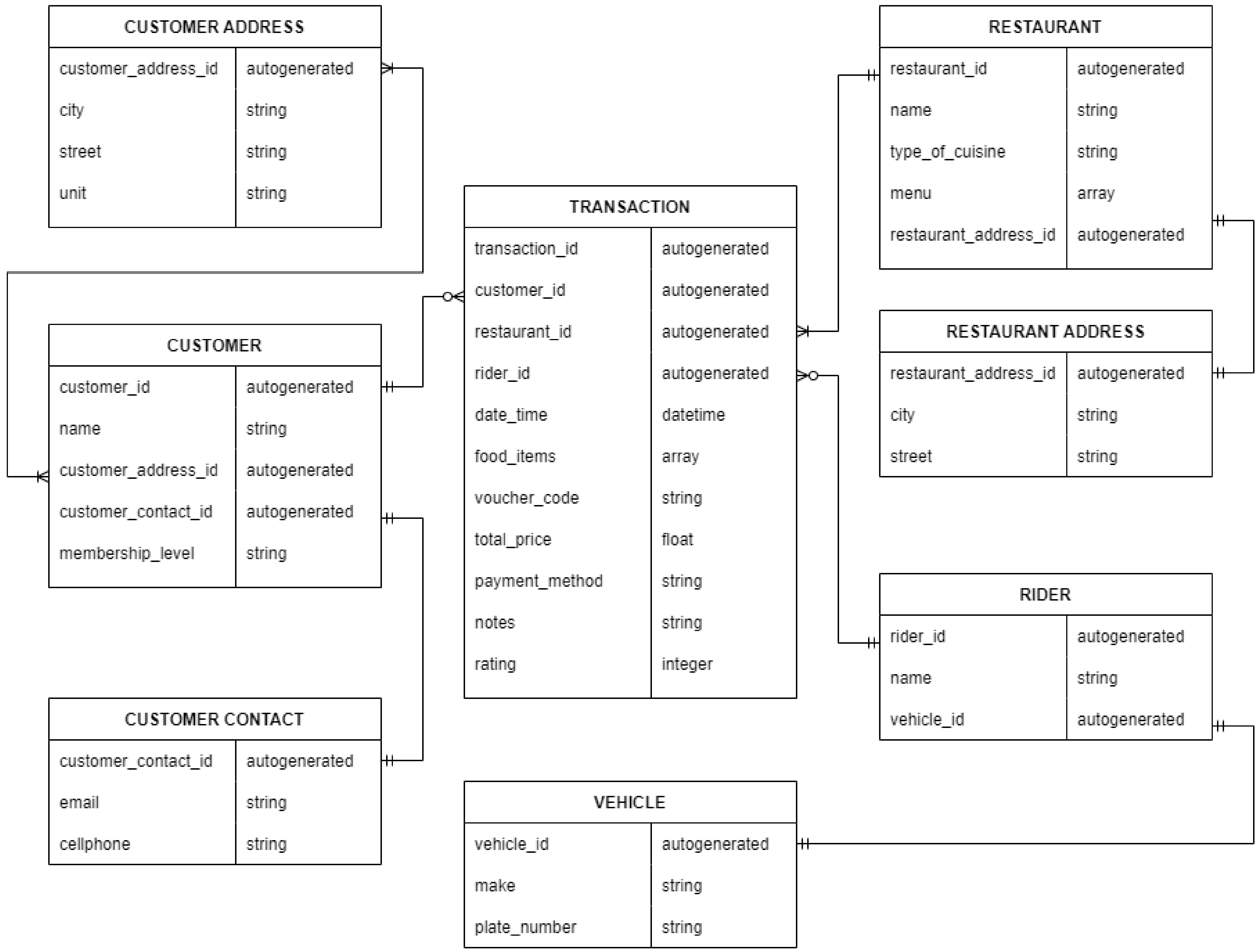
05

RESTAURANT HANDLES ORDER

06

RIDER DELIVERS ORDER

Selected Use Case: Meal Delivery App



| CUSTOMER ADDRESS | |
|---|---|
| customer_address_id | autogenerated |
| city | string |
| street | string |
| unit | string |

| CUSTOMER | |
|---|---|
| customer_id | autogenerated |
| name | string |
| customer_address_id | autogenerated |
| customer_contact_id | autogenerated |
| membership_level | string |

| CUSTOMER CONTACT | |
|---|---|
| customer_contact_id | autogenerated |
| email | string |
| cellphone | string |

| TRANSACTION | |
|---|---|
| transaction_id | autogenerated |
| customer_id | autogenerated |
| restaurant_id | autogenerated |
| rider_id | autogenerated |
| date_time | datetime |
| food_items | array |
| voucher_code | string |
| total_price | float |
| payment_method | string |
| notes | string |
| rating | integer |

| VEHICLE | |
|---|---|
| vehicle_id | autogenerated |
| make | string |
| plate_number | string |

| RESTAURANT | |
|---|---|
| restaurant_id | autogenerated |
| name | string |
| type_of_cuisine | string |
| menu | array |
| restaurant_address_id | autogenerated |

| RESTAURANT ADDRESS | |
|---|---|
| restaurant_address_id | autogenerated |
| city | string |
| street | string |

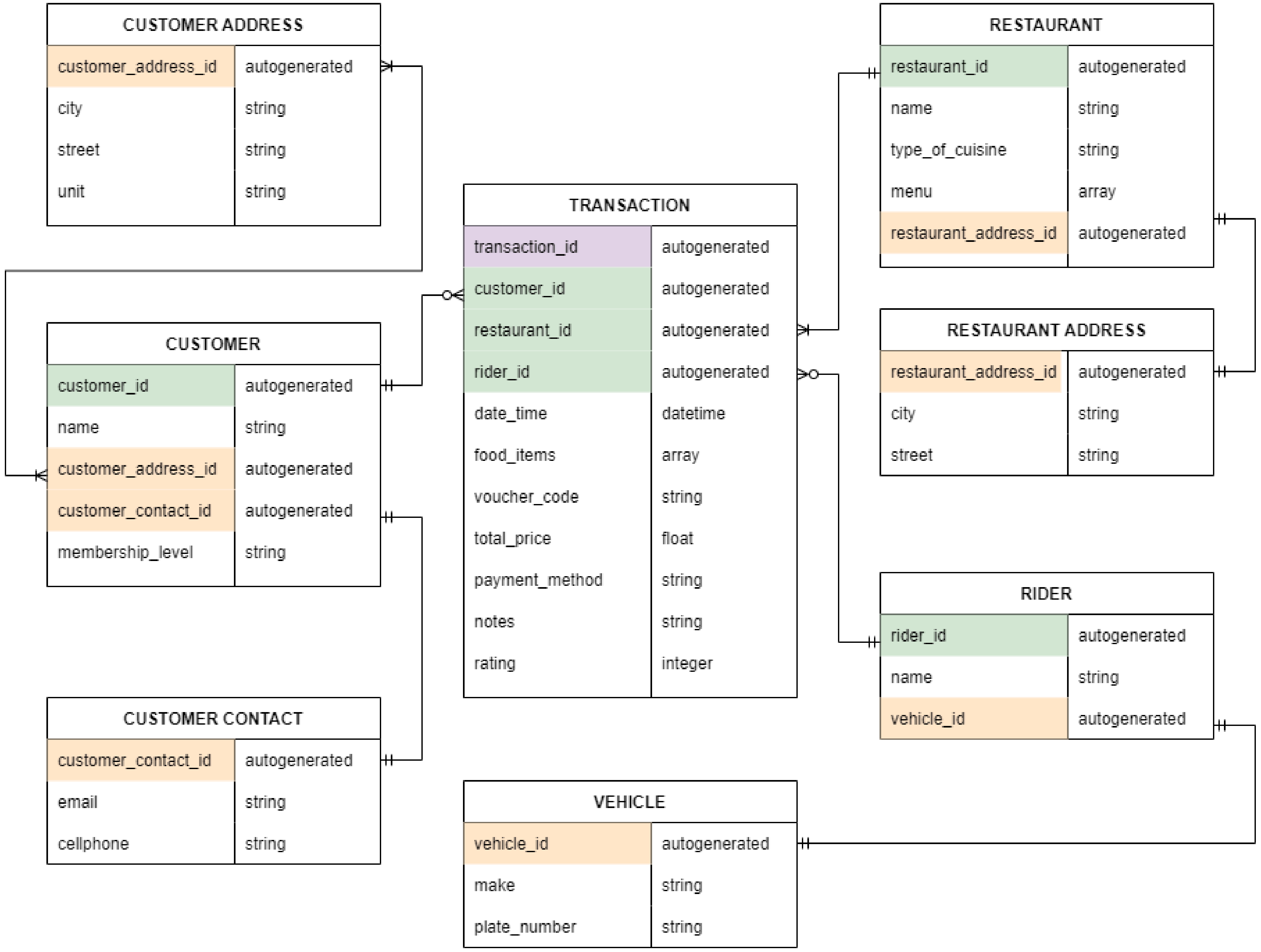
| RIDER | |
|---|---|
| rider_id | autogenerated |
| name | string |
| vehicle_id | autogenerated |

# ASSUMPTIONS MADE ABOUT THE DATA.

- Transactions were only made in 2020 or 2021
- The corporation employs 100 riders
- Riders are not limited by time and distance when delivering food
- Voucher codes are optional and are not limited to a certain time period
- Each of the five restaurants only have five items on their menu
- A food item can only be ordered up to four times in one transaction

# DATA GENERATION

**Selected Use Case: Meal Delivery App**



| CUSTOMER ADDRESS | |
|---|---|
| customer_address_id | autogenerated |
| city | string |
| street | string |
| unit | string |

| CUSTOMER | |
|---|---|
| customer_id | autogenerated |
| name | string |
| customer_address_id | autogenerated |
| customer_contact_id | autogenerated |
| membership_level | string |

| CUSTOMER CONTACT | |
|---|---|
| customer_contact_id | autogenerated |
| email | string |
| cellphone | string |

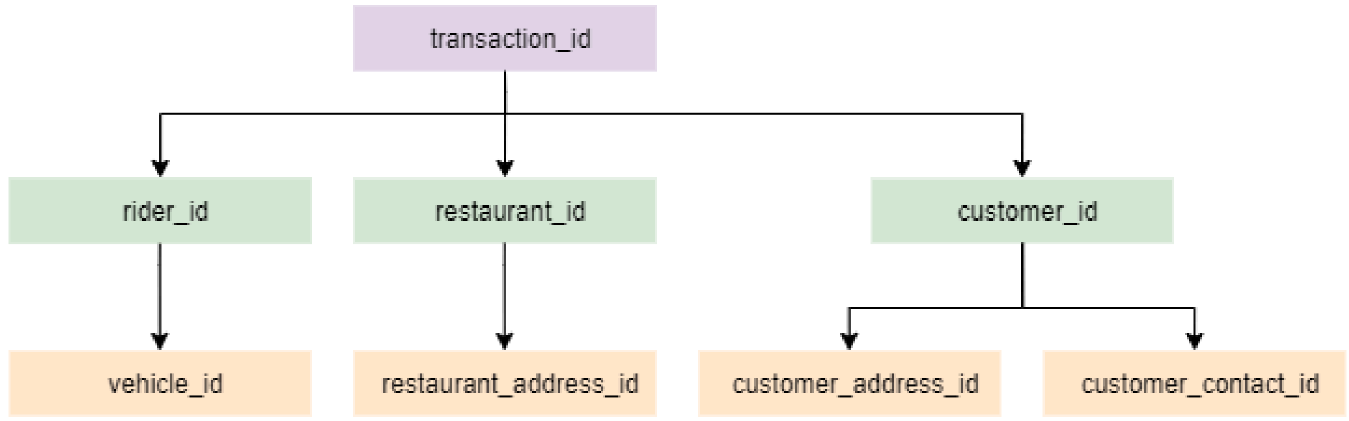
| TRANSACTION | |
|---|---|
| transaction_id | autogenerated |
| customer_id | autogenerated |
| restaurant_id | autogenerated |
| rider_id | autogenerated |
| date_time | datetime |
| food_items | array |
| voucher_code | string |
| total_price | float |
| payment_method | string |
| notes | string |
| rating | integer |

| VEHICLE | |
|---|---|
| vehicle_id | autogenerated |
| make | string |
| plate_number | string |

| RESTAURANT | |
|---|---|
| restaurant_id | autogenerated |
| name | string |
| type_of_cuisine | string |
| menu | array |
| restaurant_address_id | autogenerated |

| RESTAURANT ADDRESS | |
|---|---|
| restaurant_address_id | autogenerated |
| city | string |
| street | string |

| RIDER | |
|---|---|
| rider_id | autogenerated |
| name | string |
| vehicle_id | autogenerated |



transaction_id

rider_id

restaurant_id

customer_id

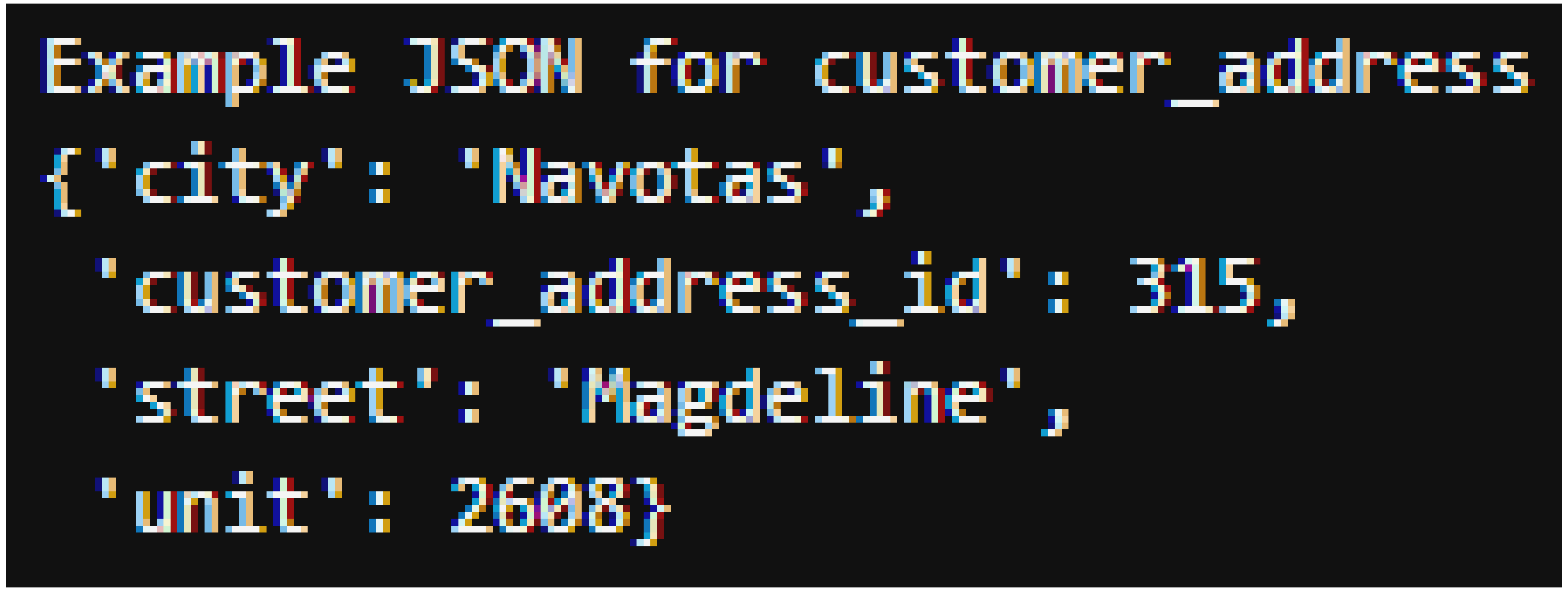
vehicle_id

restaurant_address_id

customer_address_id

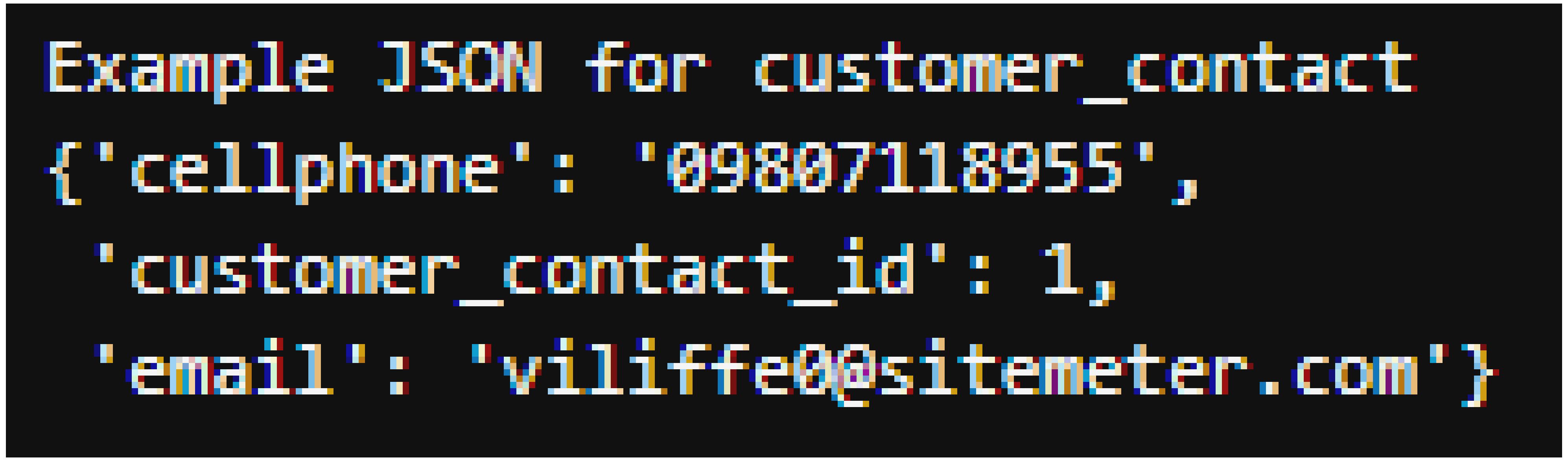
customer_contact_id

# CUSTOMER_ADDRESS ENTITY



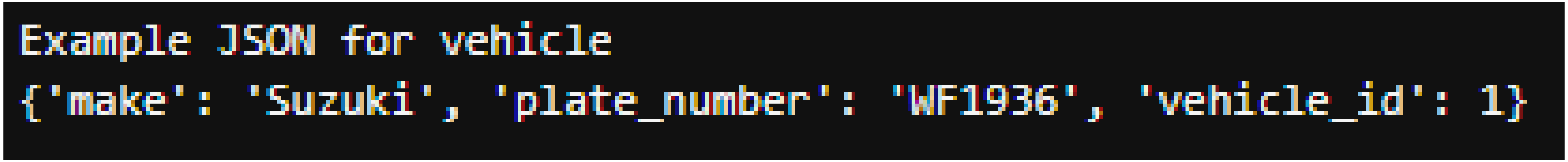
```
Example JSON for customer_address
{'city': 'Navotas',
 'customer_address_id': 315,
 'street': 'Magdeline',
 'unit': 2608}
```

# CUSTOMER_CONTACT ENTITY



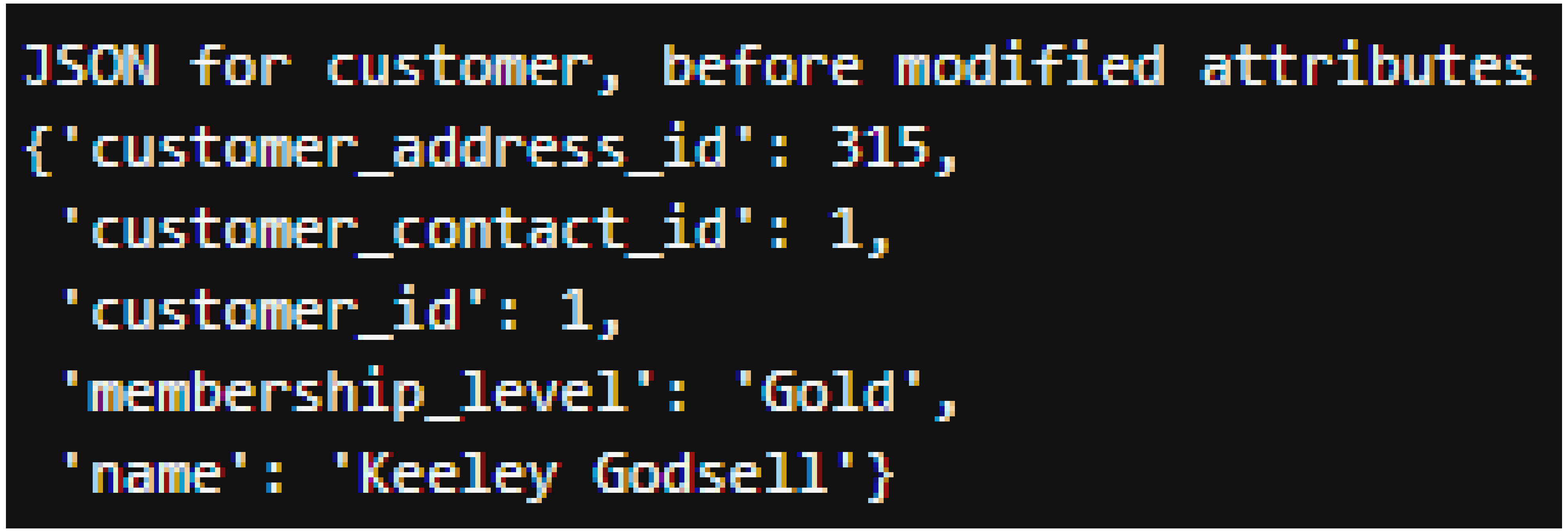
```
Example JSON for customer_contact
{'cellphone': '09807118955',
 'customer_contact_id': 1,
 'email': 'viliffe0@sitemeter.com'}
```

# VEHICLE ENTITY



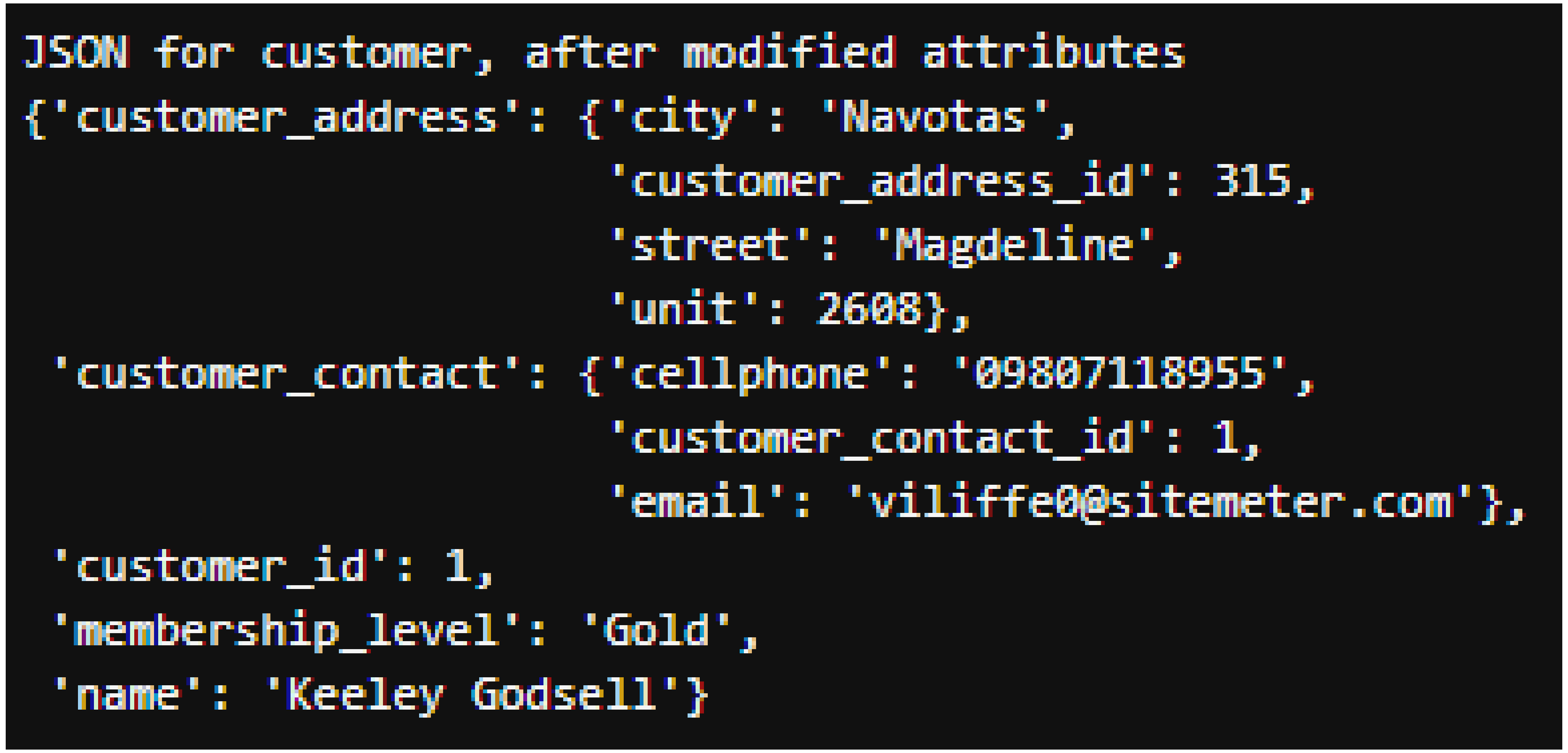
```
Example JSON for vehicle
{'make': 'Suzuki', 'plate_number': 'WF1936', 'vehicle_id': 1}
```

# CUSTOMER ENTITY



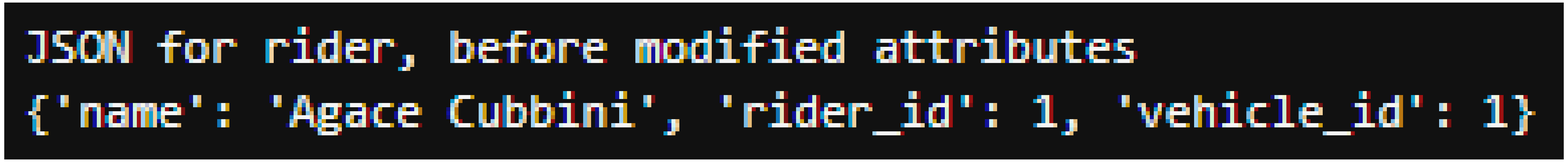
```
JSON for customer, before modified attributes
{'customer_address_id': 315,
 'customer_contact_id': 1,
 'customer_id': 1,
 'membership_level': 'Gold',
 'name': 'Keeley Godsell'}
```

# CUSTOMER ENTITY



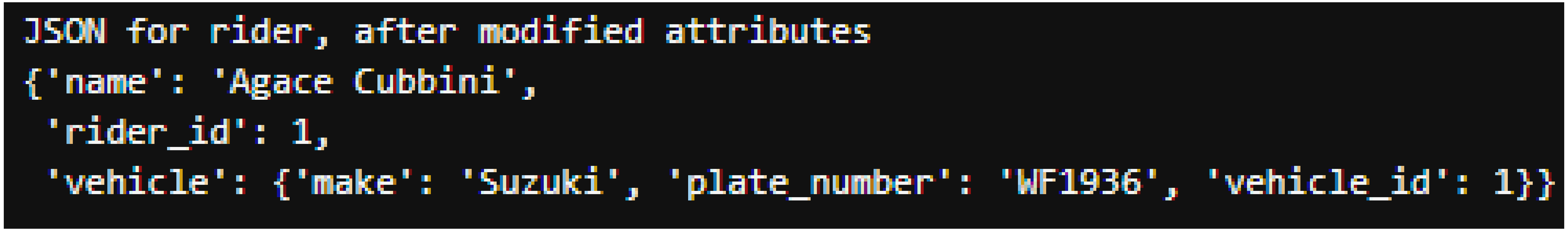
```
JSON for customer, after modified attributes
{'customer_address': {'city': 'Navotas',
                      'customer_address_id': 315,
                      'street': 'Magdeline',
                      'unit': 2608},
 'customer_contact': {'cellphone': '09807118955',
                      'customer_contact_id': 1,
                      'email': 'viliffe0@sitemeter.com'},
 'customer_id': 1,
 'membership_level': 'Gold',
 'name': 'Keeley Godsell'}
```

# RIDER ENTITY



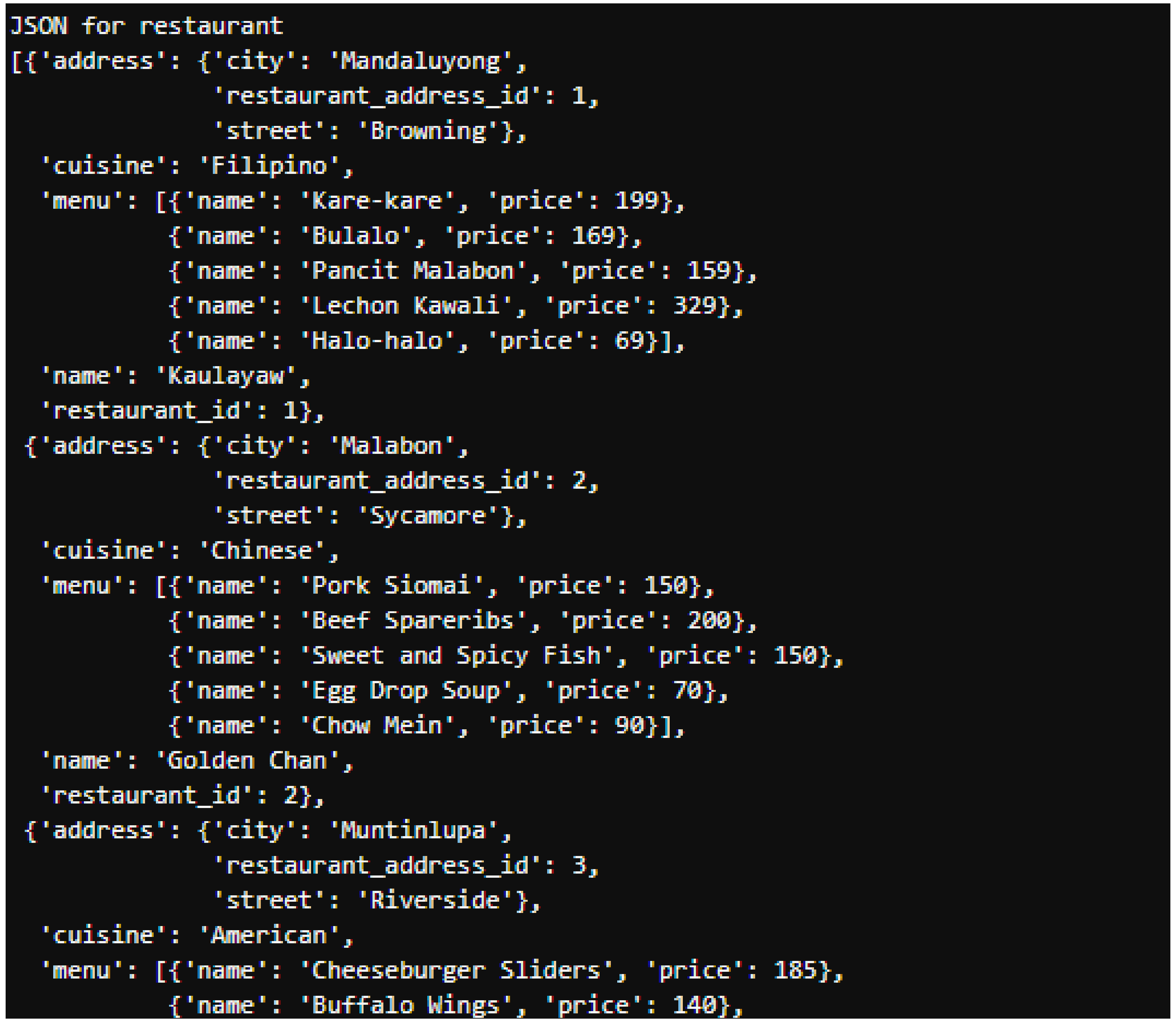
```
JSON for rider, before modified attributes
{'name': 'Agace Cubbini', 'rider_id': 1, 'vehicle_id': 1}
```

# RIDER ENTITY

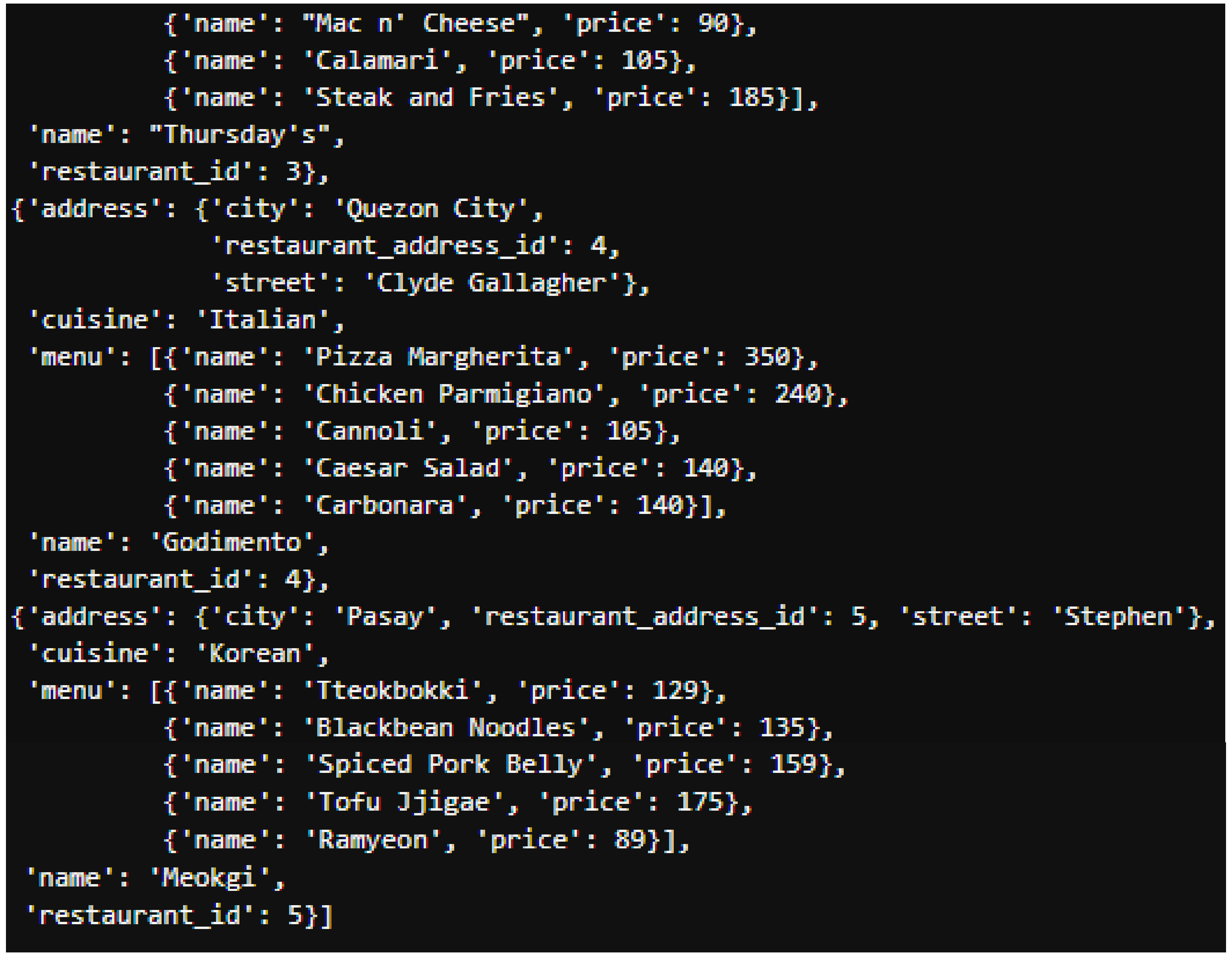


```
JSON for rider, after modified attributes
{'name': 'Agace Cubbini',
 'rider_id': 1,
 'vehicle': {'make': 'Suzuki', 'plate_number': 'WF1936', 'vehicle_id': 1}}
```

# RESTAURANT AND RESTAURANT_ADDRESS ENTITIES

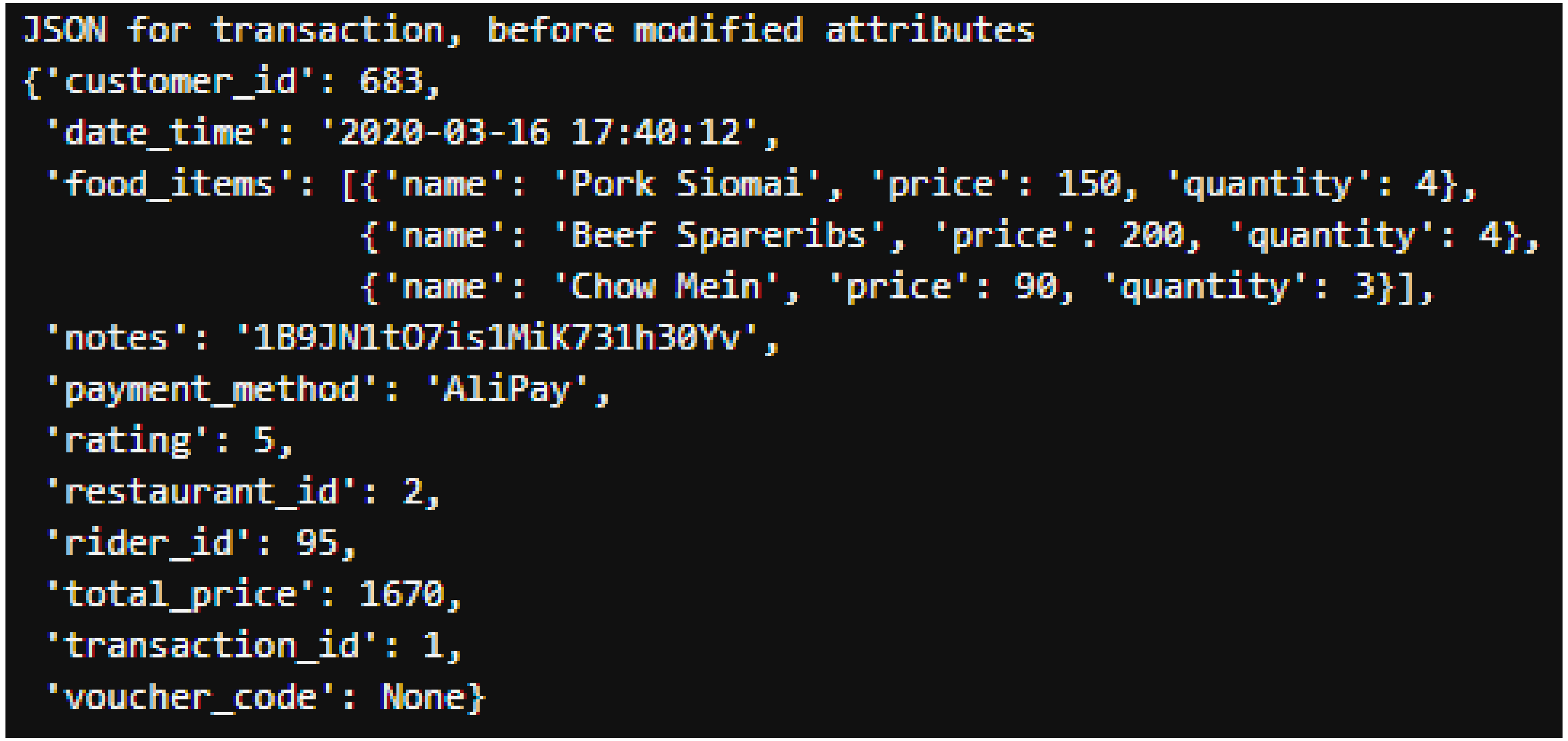


```
JSON for restaurant
[{'address': {'city': 'Mandaluyong',
              'restaurant_address_id': 1,
              'street': 'Browning'},
  'cuisine': 'Filipino',
  'menu': [{'name': 'Kare-kare', 'price': 199},
           {'name': 'Bulalo', 'price': 169},
           {'name': 'Pancit Malabon', 'price': 159},
           {'name': 'Lechon Kawali', 'price': 329},
           {'name': 'Halo-halo', 'price': 69}],
  'name': 'Kaulayaw',
  'restaurant_id': 1},
 {'address': {'city': 'Malabon',
              'restaurant_address_id': 2,
              'street': 'Sycamore'},
  'cuisine': 'Chinese',
  'menu': [{'name': 'Pork Siomai', 'price': 150},
           {'name': 'Beef Spareribs', 'price': 200},
           {'name': 'Sweet and Spicy Fish', 'price': 150},
           {'name': 'Egg Drop Soup', 'price': 70},
           {'name': 'Chow Mein', 'price': 90}],
  'name': 'Golden Chan',
  'restaurant_id': 2},
 {'address': {'city': 'Muntinlupa',
              'restaurant_address_id': 3,
              'street': 'Riverside'},
  'cuisine': 'American',
  'menu': [{'name': 'Cheeseburger Sliders', 'price': 185},
           {'name': 'Buffalo Wings', 'price': 140},
```



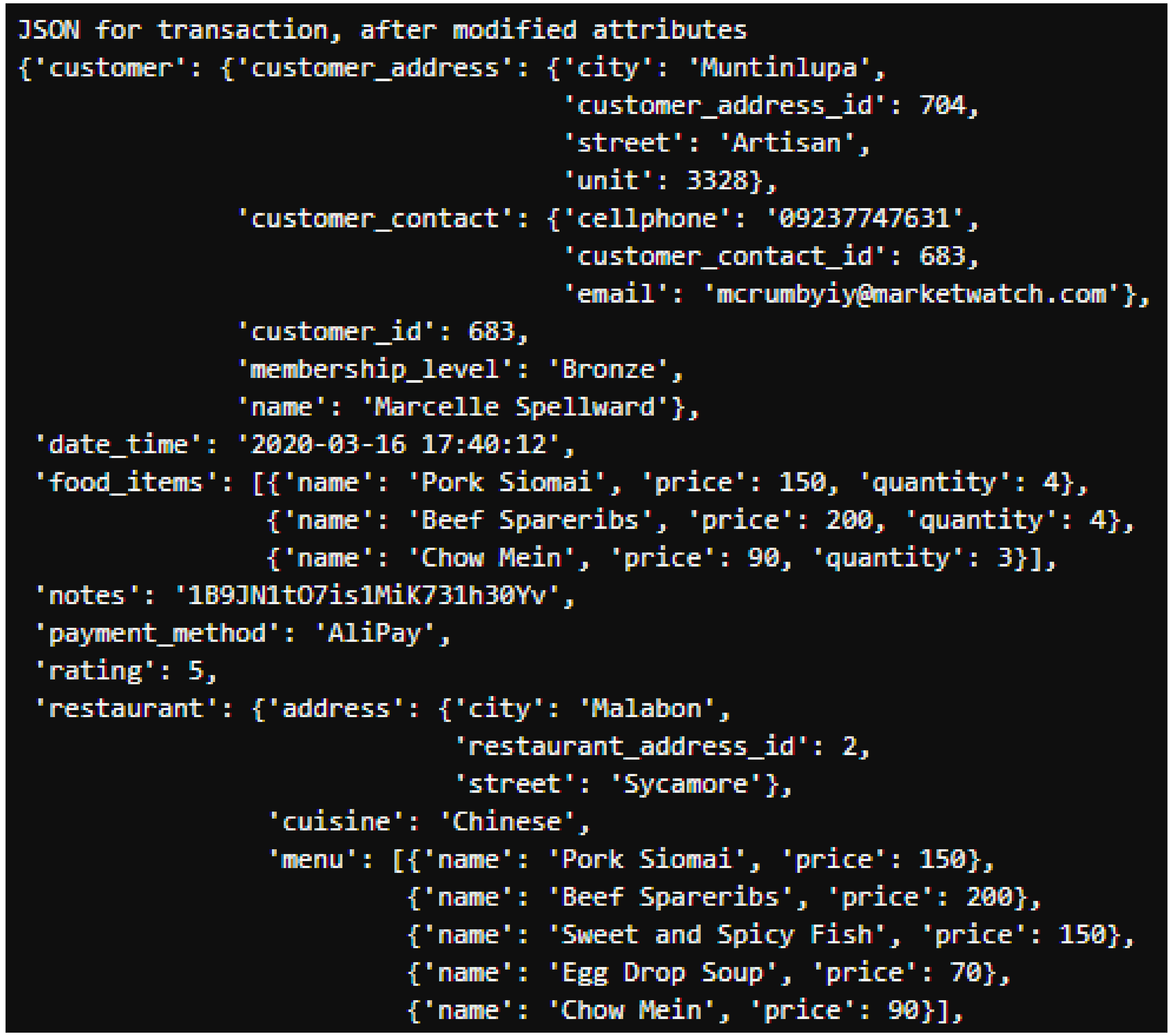
```
          {'name': "Mac n' Cheese", 'price': 90},
          {'name': 'Calamari', 'price': 105},
          {'name': 'Steak and Fries', 'price': 185}],
 'name': "Thursday's",
 'restaurant_id': 3},
{'address': {'city': 'Quezon City',
             'restaurant_address_id': 4,
             'street': 'Clyde Gallagher'},
 'cuisine': 'Italian',
 'menu': [{'name': 'Pizza Margherita', 'price': 350},
          {'name': 'Chicken Parmigiano', 'price': 240},
          {'name': 'Cannoli', 'price': 105},
          {'name': 'Caesar Salad', 'price': 140},
          {'name': 'Carbonara', 'price': 140}],
 'name': 'Godimento',
 'restaurant_id': 4},
{'address': {'city': 'Pasay', 'restaurant_address_id': 5, 'street': 'Stephen'},
 'cuisine': 'Korean',
 'menu': [{'name': 'Tteokbokki', 'price': 129},
          {'name': 'Blackbean Noodles', 'price': 135},
          {'name': 'Spiced Pork Belly', 'price': 159},
          {'name': 'Tofu Jjigae', 'price': 175},
          {'name': 'Ramyeon', 'price': 89}],
 'name': 'Meokgi',
 'restaurant_id': 5}]
```

# TRANSACTION ENTITY

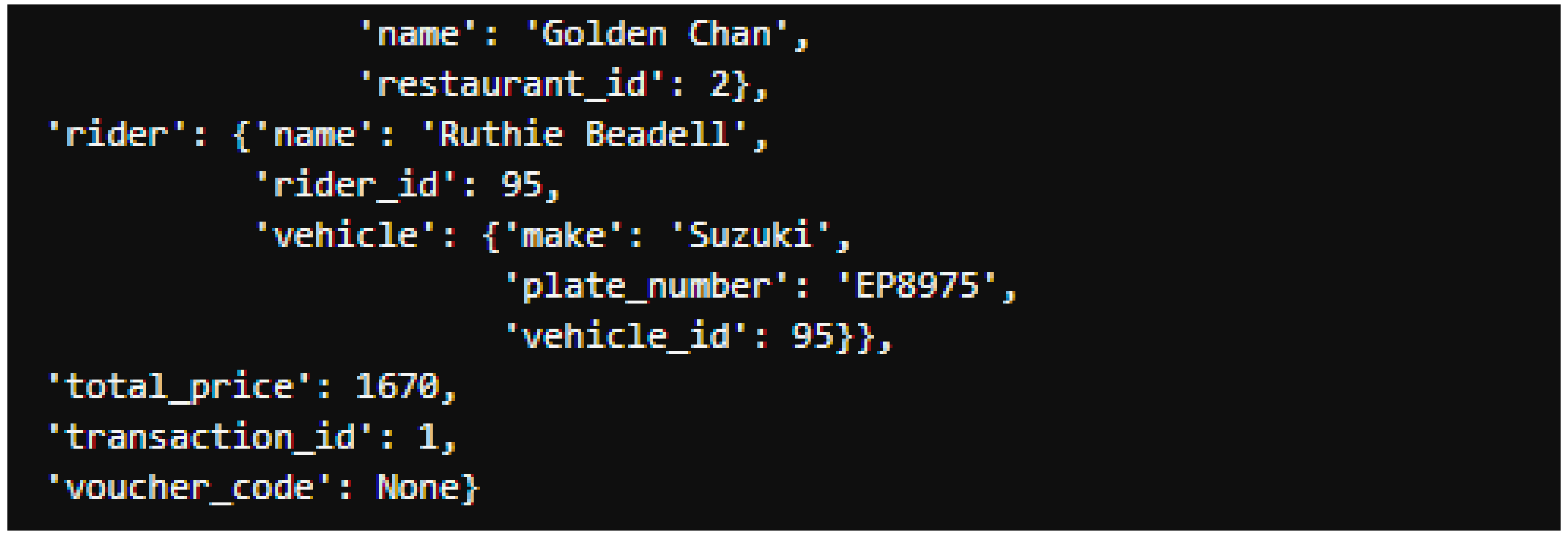


```
JSON for transaction, before modified attributes
{'customer_id': 683,
 'date_time': '2020-03-16 17:40:12',
 'food_items': [{'name': 'Pork Siomai', 'price': 150, 'quantity': 4},
                {'name': 'Beef Spareribs', 'price': 200, 'quantity': 4},
                {'name': 'Chow Mein', 'price': 90, 'quantity': 3}],
 'notes': '1B9JN1tO7is1MiK731h30Yv',
 'payment_method': 'AliPay',
 'rating': 5,
 'restaurant_id': 2,
 'rider_id': 95,
 'total_price': 1670,
 'transaction_id': 1,
 'voucher_code': None}
```

# TRANSACTION ENTITY

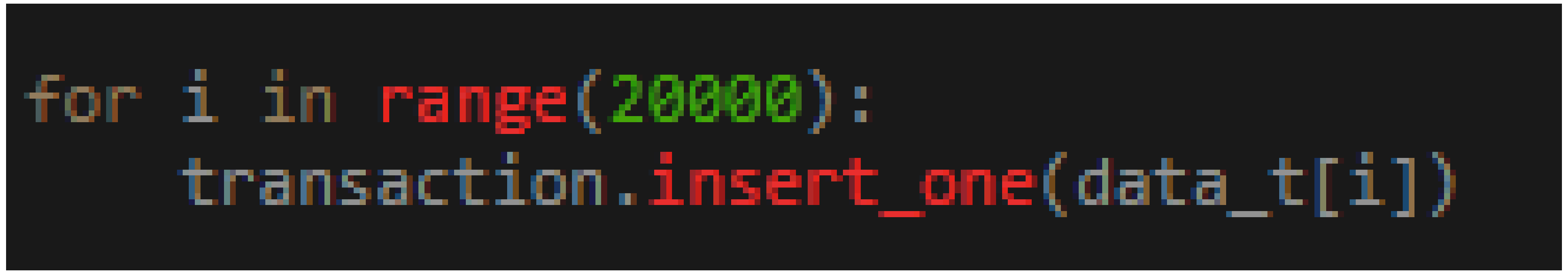


```
JSON for transaction, after modified attributes
{'customer': {'customer_address': {'city': 'Muntinlupa',
                                   'customer_address_id': 704,
                                   'street': 'Artisan',
                                   'unit': 3328},
              'customer_contact': {'cellphone': '09237747631',
                                   'customer_contact_id': 683,
                                   'email': 'mcrumbyiy@marketwatch.com'},
              'customer_id': 683,
              'membership_level': 'Bronze',
              'name': 'Marcelle Spellward'},
 'date_time': '2020-03-16 17:40:12',
 'food_items': [{'name': 'Pork Siomai', 'price': 150, 'quantity': 4},
                {'name': 'Beef Spareribs', 'price': 200, 'quantity': 4},
                {'name': 'Chow Mein', 'price': 90, 'quantity': 3}],
 'notes': '1B9JN1tO7is1MiK731h30Yv',
 'payment_method': 'AliPay',
 'rating': 5,
 'restaurant': {'address': {'city': 'Malabon',
                            'restaurant_address_id': 2,
                            'street': 'Sycamore'},
                'cuisine': 'Chinese',
                'menu': [{'name': 'Pork Siomai', 'price': 150},
                         {'name': 'Beef Spareribs', 'price': 200},
                         {'name': 'Sweet and Spicy Fish', 'price': 150},
                         {'name': 'Egg Drop Soup', 'price': 70},
                         {'name': 'Chow Mein', 'price': 90}],
```



```
                'name': 'Golden Chan',
                'restaurant_id': 2},
 'rider': {'name': 'Ruthie Beadell',
           'rider_id': 95,
           'vehicle': {'make': 'Suzuki',
                       'plate_number': 'EP8975',
                       'vehicle_id': 95}},
 'total_price': 1670,
 'transaction_id': 1,
 'voucher_code': None}
```

# INSERTION OF JSON FILE TO MONGODB



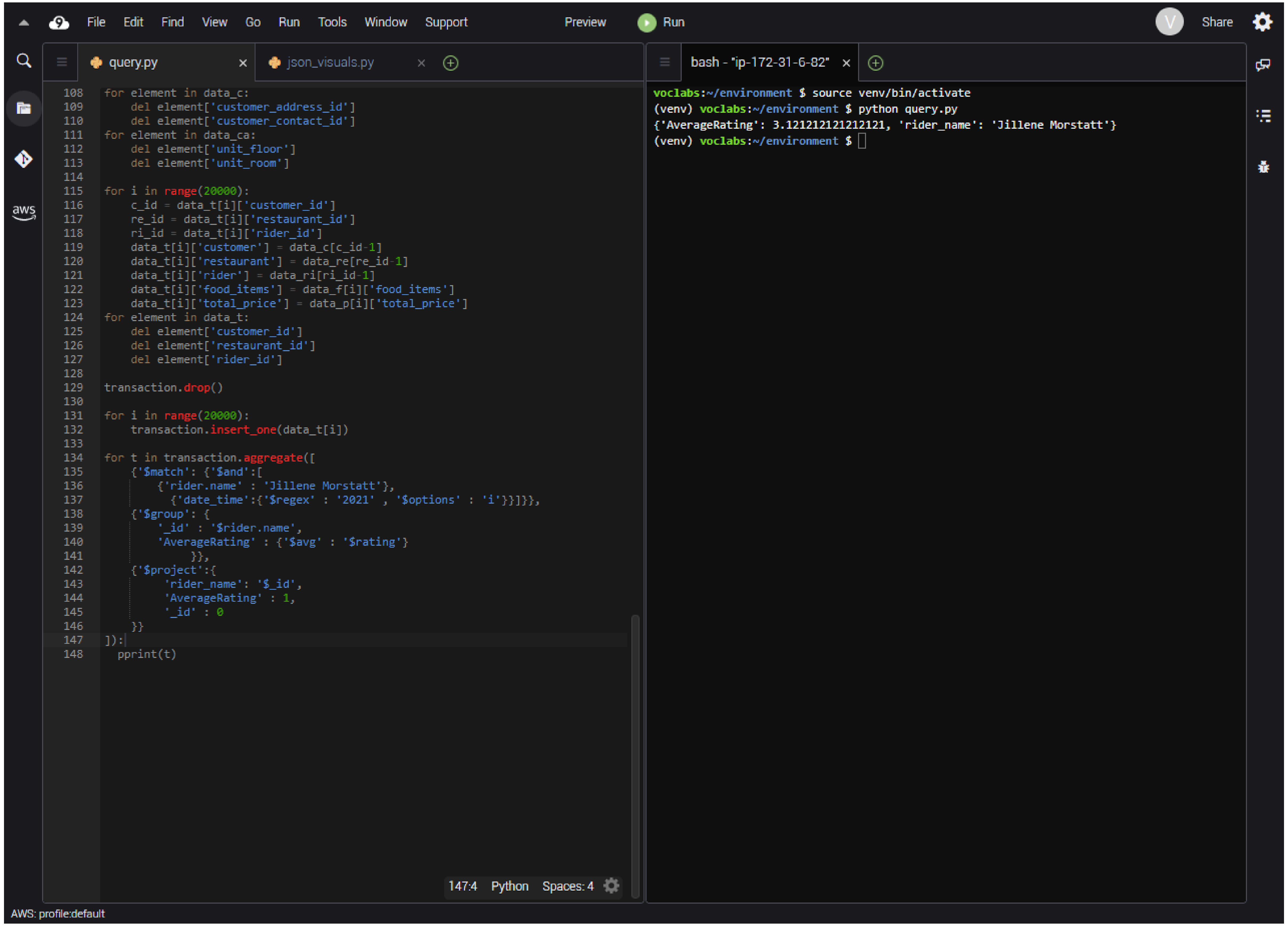
```
for i in range(20000):
    transaction.insert_one(data_t[i])
```

- The command shown above inserts one document into the database called "transaction"
- The process is repeated 20,000 times.

# SAMPLE QUERIES USING THE DATABASE

# THE RIDER JILLENE MORSTATT'S AVERAGE TRANSACTION RATING FOR 2021

To inform Morstatt and the company about their performance in 2021.

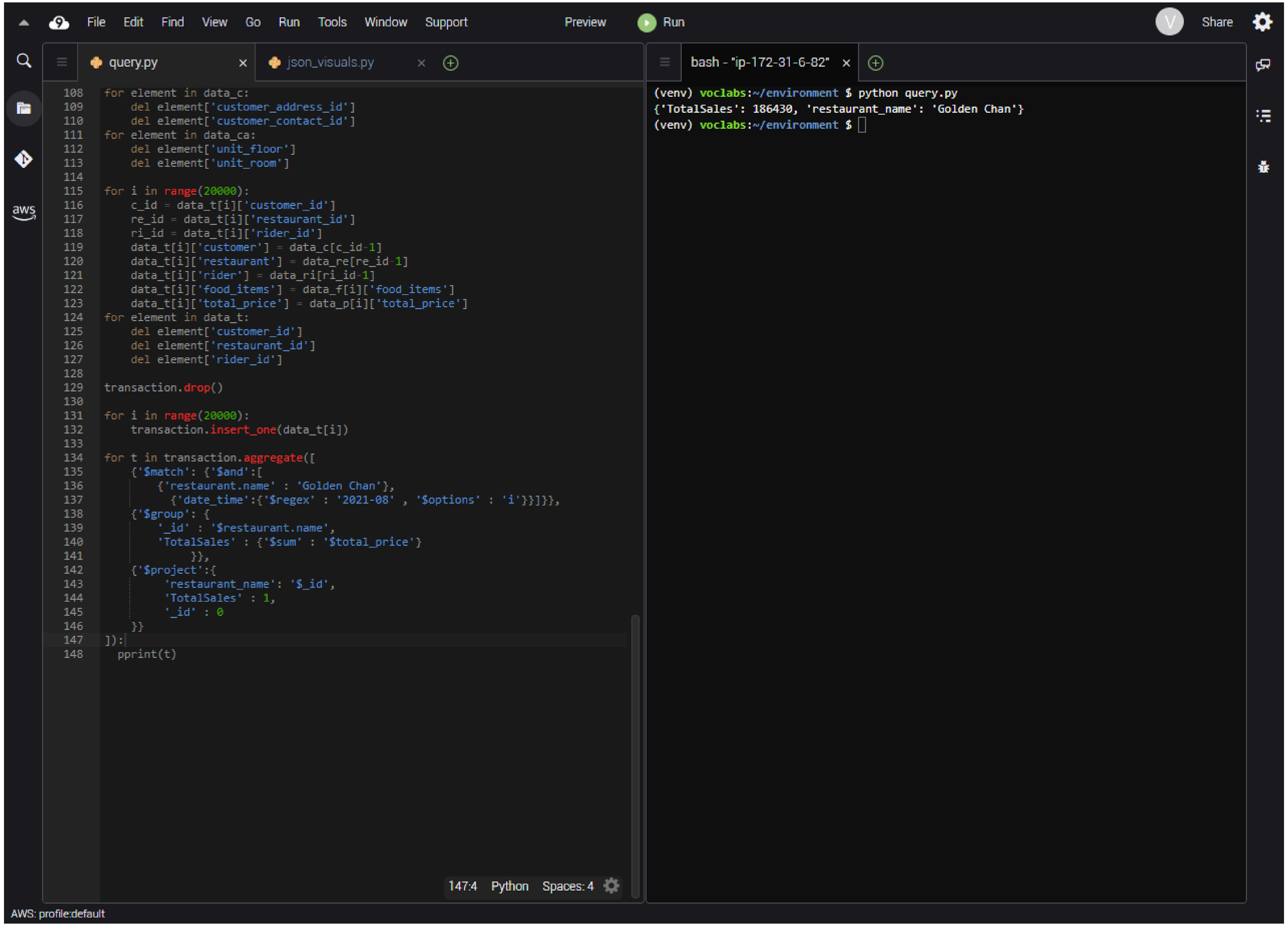


```python
for element in data_c:
    del element['customer_address_id']
    del element['customer_contact_id']
for element in data_ca:
    del element['unit_floor']
    del element['unit_room']

for i in range(20000):
    c_id = data_t[i]['customer_id']
    re_id = data_t[i]['restaurant_id']
    ri_id = data_t[i]['rider_id']
    data_t[i]['customer'] = data_c[c_id-1]
    data_t[i]['restaurant'] = data_re[re_id-1]
    data_t[i]['rider'] = data_ri[ri_id-1]
    data_t[i]['food_items'] = data_f[i]['food_items']
    data_t[i]['total_price'] = data_p[i]['total_price']
for element in data_t:
    del element['customer_id']
    del element['restaurant_id']
    del element['rider_id']

transaction.drop()

for i in range(20000):
    transaction.insert_one(data_t[i])

for t in transaction.aggregate([
    {'$match': {'$and':[
        {'rider.name' : 'Jillene Morstatt'},
          {'date_time':{'$regex' : '2021' , '$options' : 'i'}}]}},
    {'$group': {
        '_id' : '$rider.name',
        'AverageRating' : {'$avg' : '$rating'}
             }},
    {'$project':{
         'rider_name': '$_id',
         'AverageRating' : 1,
         '_id' : 0
    }}
]):
  pprint(t)
```

```
voclabs:~/environment $ source venv/bin/activate
(venv) voclabs:~/environment $ python query.py
{'AverageRating': 3.121212121212121, 'rider_name': 'Jillene Morstatt'}
(venv) voclabs:~/environment $ 
```

# TOTAL SALES OF GOLDEN CHAN IN AUGUST 2021

To attain the total revenue and examine if the change in season affects the sales of Golden Chan.

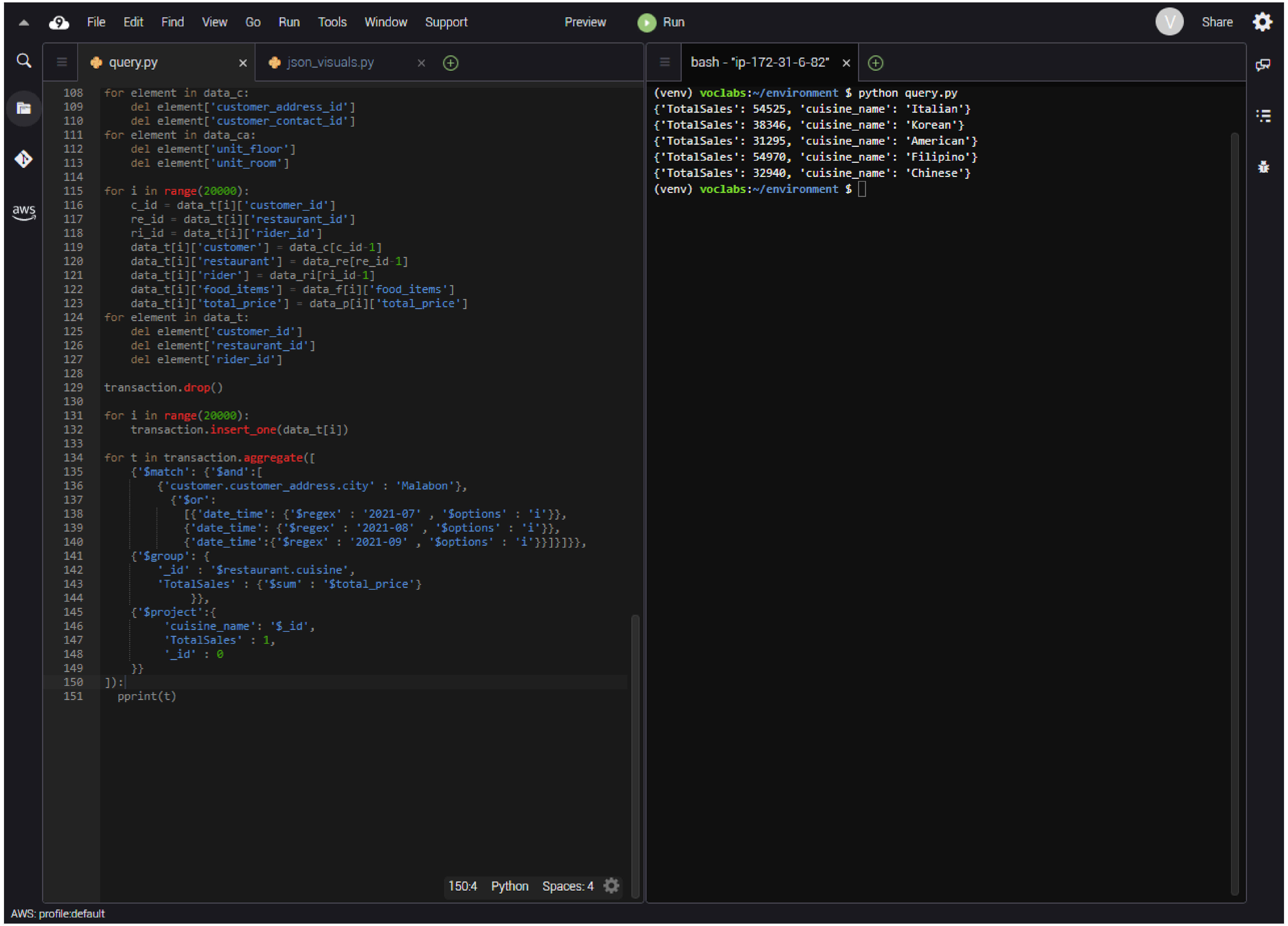


```python
for element in data_c:
    del element['customer_address_id']
    del element['customer_contact_id']
for element in data_ca:
    del element['unit_floor']
    del element['unit_room']

for i in range(20000):
    c_id = data_t[i]['customer_id']
    re_id = data_t[i]['restaurant_id']
    ri_id = data_t[i]['rider_id']
    data_t[i]['customer'] = data_c[c_id-1]
    data_t[i]['restaurant'] = data_re[re_id-1]
    data_t[i]['rider'] = data_ri[ri_id-1]
    data_t[i]['food_items'] = data_f[i]['food_items']
    data_t[i]['total_price'] = data_p[i]['total_price']
for element in data_t:
    del element['customer_id']
    del element['restaurant_id']
    del element['rider_id']

transaction.drop()

for i in range(20000):
    transaction.insert_one(data_t[i])

for t in transaction.aggregate([
    {'$match': {'$and':[
        {'restaurant.name' : 'Golden Chan'},
          {'date_time':{'$regex' : '2021-08' , '$options' : 'i'}}]}},
    {'$group': {
        '_id' : '$restaurant.name',
        'TotalSales' : {'$sum' : '$total_price'}
             }},
    {'$project':{
         'restaurant_name': '$_id',
         'TotalSales' : 1,
         '_id' : 0
    }}
]):
  pprint(t)
```

```
(venv) voclabs:~/environment $ python query.py
{'TotalSales': 186430, 'restaurant_name': 'Golden Chan'}
(venv) voclabs:~/environment $ 
```

# CUISINE WITH HIGHEST SALES IN MALABON IN Q3 OF 2021

To prepare for an influx of orders coming from Malabon.

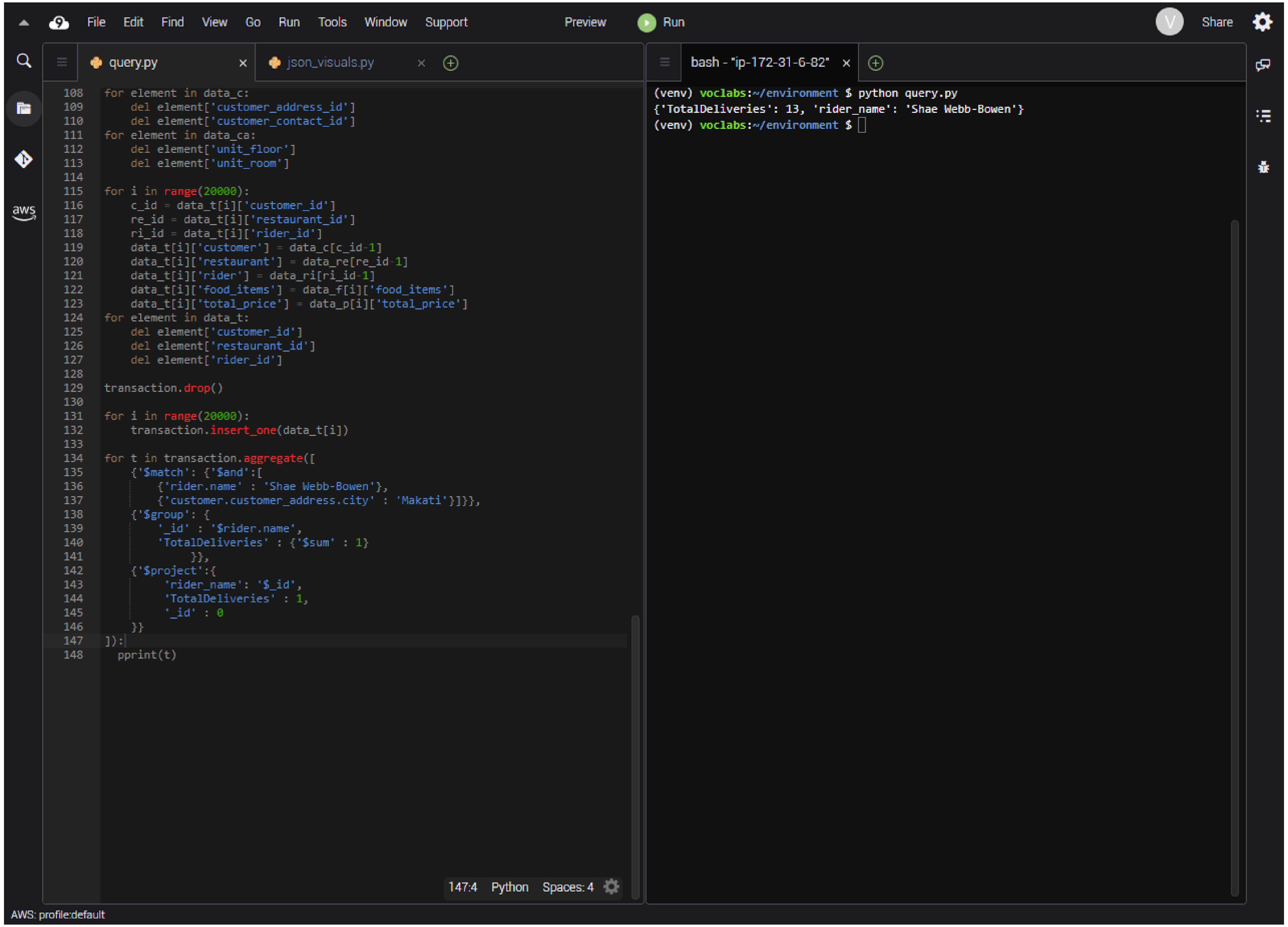


```python
for element in data_c:
    del element['customer_address_id']
    del element['customer_contact_id']
for element in data_ca:
    del element['unit_floor']
    del element['unit_room']

for i in range(20000):
    c_id = data_t[i]['customer_id']
    re_id = data_t[i]['restaurant_id']
    ri_id = data_t[i]['rider_id']
    data_t[i]['customer'] = data_c[c_id-1]
    data_t[i]['restaurant'] = data_re[re_id-1]
    data_t[i]['rider'] = data_ri[ri_id-1]
    data_t[i]['food_items'] = data_f[i]['food_items']
    data_t[i]['total_price'] = data_p[i]['total_price']
for element in data_t:
    del element['customer_id']
    del element['restaurant_id']
    del element['rider_id']

transaction.drop()

for i in range(20000):
    transaction.insert_one(data_t[i])

for t in transaction.aggregate([
    {'$match': {'$and':[
        {'customer.customer_address.city' : 'Malabon'},
          {'$or':
            [{'date_time': {'$regex' : '2021-07' , '$options' : 'i'}},
            {'date_time': {'$regex' : '2021-08' , '$options' : 'i'}},
            {'date_time':{'$regex' : '2021-09' , '$options' : 'i'}}]}]}},
    {'$group': {
        '_id' : '$restaurant.cuisine',
        'TotalSales' : {'$sum' : '$total_price'}
             }},
    {'$project':{
         'cuisine_name': '$_id',
         'TotalSales' : 1,
         '_id' : 0
    }}
]):
  pprint(t)
```

```
(venv) voclabs:~/environment $ python query.py
{'TotalSales': 54525, 'cuisine_name': 'Italian'}
{'TotalSales': 38346, 'cuisine_name': 'Korean'}
{'TotalSales': 31295, 'cuisine_name': 'American'}
{'TotalSales': 54970, 'cuisine_name': 'Filipino'}
{'TotalSales': 32940, 'cuisine_name': 'Chinese'}
(venv) voclabs:~/environment $
```

# NUMBER OF DELIVERIES DONE BY RIDER SHAE WEBB-BOWEN IN MAKATI

The company is considering giving incentives to riders who complete a certain number of deliveries.

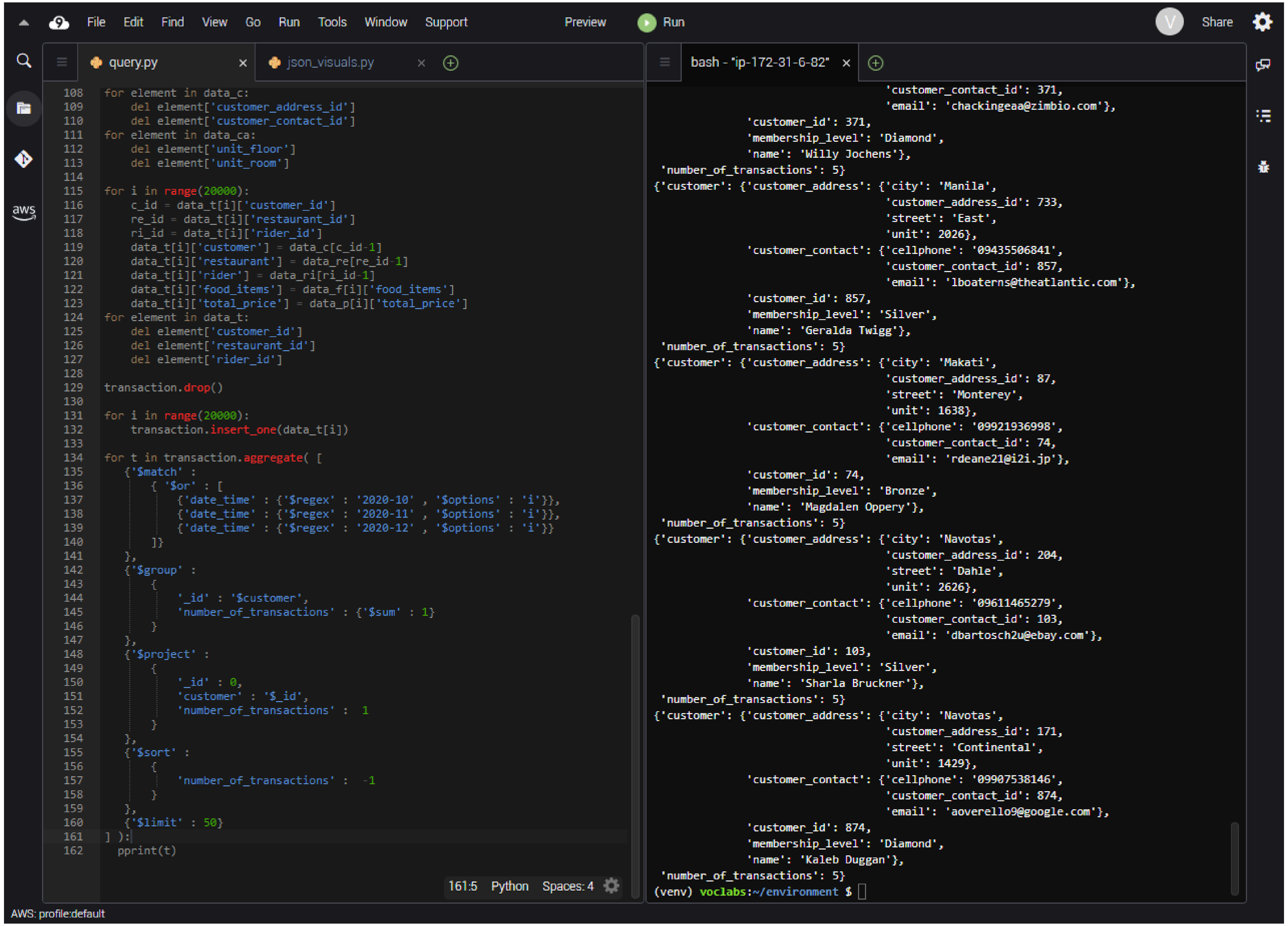


```python
for element in data_c:
    del element['customer_address_id']
    del element['customer_contact_id']
for element in data_ca:
    del element['unit_floor']
    del element['unit_room']

for i in range(20000):
    c_id = data_t[i]['customer_id']
    re_id = data_t[i]['restaurant_id']
    ri_id = data_t[i]['rider_id']
    data_t[i]['customer'] = data_c[c_id-1]
    data_t[i]['restaurant'] = data_re[re_id-1]
    data_t[i]['rider'] = data_ri[ri_id-1]
    data_t[i]['food_items'] = data_f[i]['food_items']
    data_t[i]['total_price'] = data_p[i]['total_price']
for element in data_t:
    del element['customer_id']
    del element['restaurant_id']
    del element['rider_id']

transaction.drop()

for i in range(20000):
    transaction.insert_one(data_t[i])

for t in transaction.aggregate([
    {'$match': {'$and':[
        {'rider.name' : 'Shae Webb-Bowen'},
        {'customer.customer_address.city' : 'Makati'}]}},
    {'$group': {
        '_id' : '$rider.name',
        'TotalDeliveries' : {'$sum' : 1}
             }},
    {'$project':{
         'rider_name': '$_id',
         'TotalDeliveries' : 1,
         '_id' : 0
    }}
]):
  pprint(t)
```

```
(venv) voclabs:~/environment $ python query.py
{'TotalDeliveries': 13, 'rider_name': 'Shae Webb-Bowen'}
(venv) voclabs:~/environment $ 
```

# TOP 50 CUSTOMERS BY NUMBER OF TRANSACTIONS MADE IN Q4 OF 2020

The top 50 customers will be rewarded with special discount vouchers.

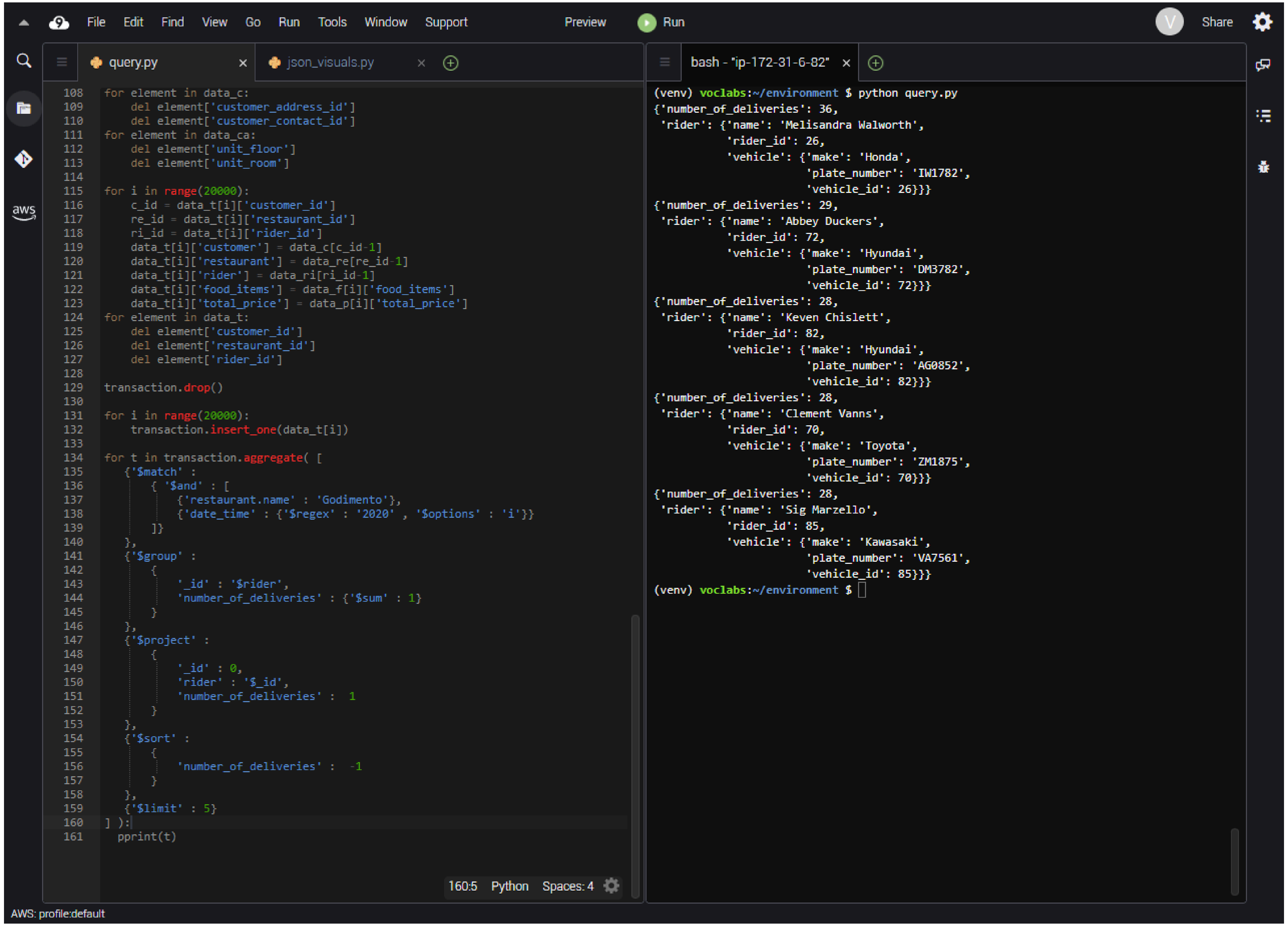


```python
for element in data_c:
    del element['customer_address_id']
    del element['customer_contact_id']
for element in data_ca:
    del element['unit_floor']
    del element['unit_room']

for i in range(20000):
    c_id = data_t[i]['customer_id']
    re_id = data_t[i]['restaurant_id']
    ri_id = data_t[i]['rider_id']
    data_t[i]['customer'] = data_c[c_id-1]
    data_t[i]['restaurant'] = data_re[re_id-1]
    data_t[i]['rider'] = data_ri[ri_id-1]
    data_t[i]['food_items'] = data_f[i]['food_items']
    data_t[i]['total_price'] = data_p[i]['total_price']
for element in data_t:
    del element['customer_id']
    del element['restaurant_id']
    del element['rider_id']

transaction.drop()

for i in range(20000):
    transaction.insert_one(data_t[i])

for t in transaction.aggregate( [
   {'$match' :
       { '$or' : [
           {'date_time' : {'$regex' : '2020-10' , '$options' : 'i'}},
           {'date_time' : {'$regex' : '2020-11' , '$options' : 'i'}},
           {'date_time' : {'$regex' : '2020-12' , '$options' : 'i'}}
       ]}
   },
   {'$group' :
       {
           '_id' : '$customer',
           'number_of_transactions' : {'$sum' : 1}
       }
   },
   {'$project' :
       {
           '_id' : 0,
           'customer' : '$_id',
           'number_of_transactions' :  1
       }
   },
   {'$sort' :
       {
           'number_of_transactions' :  -1
       }
   },
   {'$limit' : 50}
] ):
  pprint(t)
```

```
                                   'customer_contact_id': 371,
                                   'email': 'chackingeaa@zimbio.com'},
              'customer_id': 371,
              'membership_level': 'Diamond',
              'name': 'Willy Jochens'},
 'number_of_transactions': 5}
{'customer': {'customer_address': {'city': 'Manila',
                                   'customer_address_id': 733,
                                   'street': 'East',
                                   'unit': 2026},
              'customer_contact': {'cellphone': '09435506841',
                                   'customer_contact_id': 857,
                                   'email': 'lboaterns@theatlantic.com'},
              'customer_id': 857,
              'membership_level': 'Silver',
              'name': 'Geralda Twigg'},
 'number_of_transactions': 5}
{'customer': {'customer_address': {'city': 'Makati',
                                   'customer_address_id': 87,
                                   'street': 'Monterey',
                                   'unit': 1638},
              'customer_contact': {'cellphone': '09921936998',
                                   'customer_contact_id': 74,
                                   'email': 'rdeane21@i2i.jp'},
              'customer_id': 74,
              'membership_level': 'Bronze',
              'name': 'Magdalen Oppery'},
 'number_of_transactions': 5}
{'customer': {'customer_address': {'city': 'Navotas',
                                   'customer_address_id': 204,
                                   'street': 'Dahle',
                                   'unit': 2626},
              'customer_contact': {'cellphone': '09611465279',
                                   'customer_contact_id': 103,
                                   'email': 'dbartosch2u@ebay.com'},
              'customer_id': 103,
              'membership_level': 'Silver',
              'name': 'Sharla Bruckner'},
 'number_of_transactions': 5}
{'customer': {'customer_address': {'city': 'Navotas',
                                   'customer_address_id': 171,
                                   'street': 'Continental',
                                   'unit': 1429},
              'customer_contact': {'cellphone': '09907538146',
                                   'customer_contact_id': 874,
                                   'email': 'aoverello9@google.com'},
              'customer_id': 874,
              'membership_level': 'Diamond',
              'name': 'Kaleb Duggan'},
 'number_of_transactions': 5}
(venv) voclabs:~/environment $
```

# TOP 5 RIDERS BY NUMBER OF DELIVERIES FOR GODIMENTO IN 2020

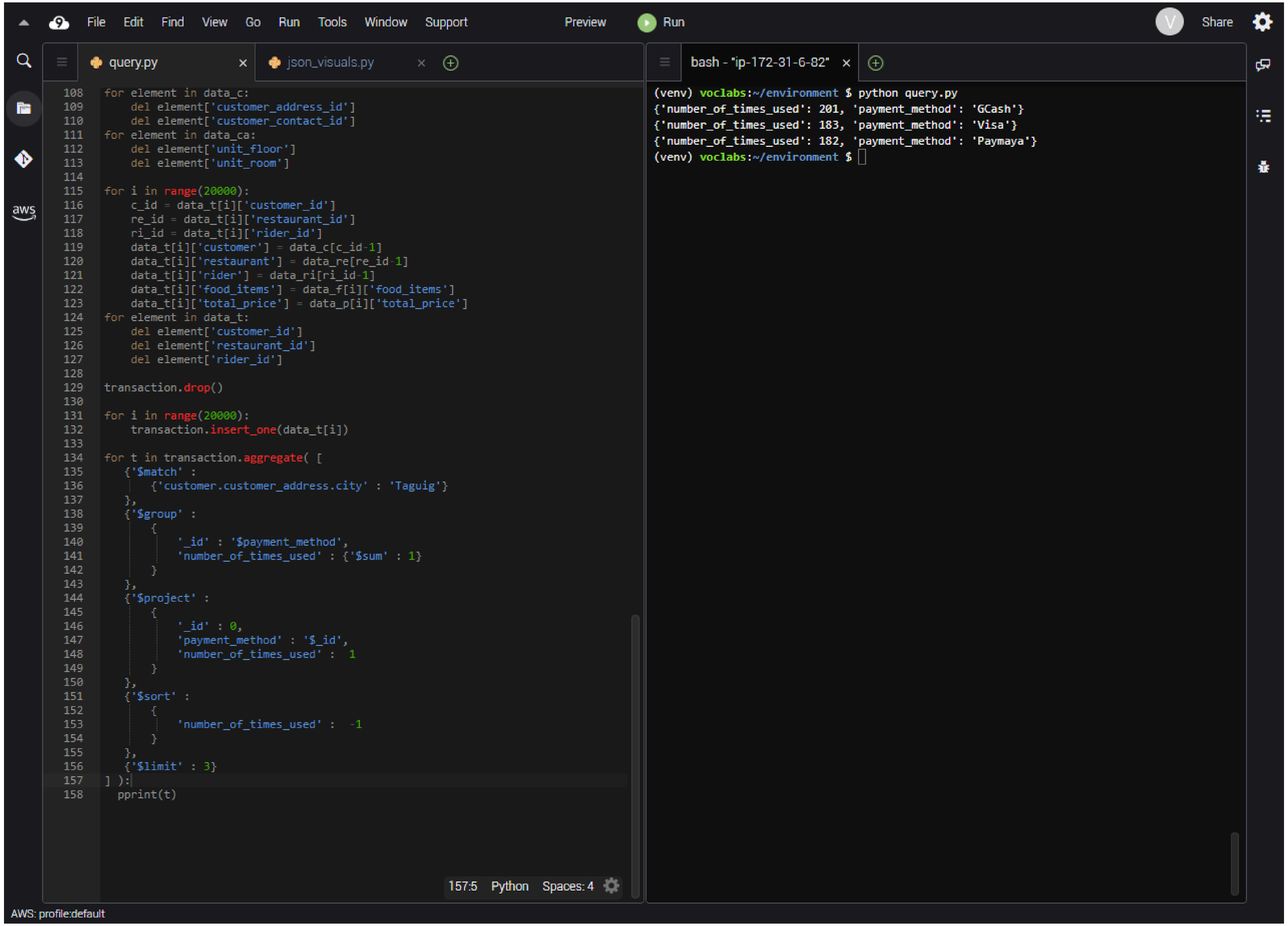
The top 5 riders will be rewarded with vouchers from Godimento.



```python
for element in data_c:
    del element['customer_address_id']
    del element['customer_contact_id']
for element in data_ca:
    del element['unit_floor']
    del element['unit_room']

for i in range(20000):
    c_id = data_t[i]['customer_id']
    re_id = data_t[i]['restaurant_id']
    ri_id = data_t[i]['rider_id']
    data_t[i]['customer'] = data_c[c_id-1]
    data_t[i]['restaurant'] = data_re[re_id-1]
    data_t[i]['rider'] = data_ri[ri_id-1]
    data_t[i]['food_items'] = data_f[i]['food_items']
    data_t[i]['total_price'] = data_p[i]['total_price']
for element in data_t:
    del element['customer_id']
    del element['restaurant_id']
    del element['rider_id']

transaction.drop()

for i in range(20000):
    transaction.insert_one(data_t[i])

for t in transaction.aggregate( [
   {'$match' :
       { '$and' : [
           {'restaurant.name' : 'Godimento'},
           {'date_time' : {'$regex' : '2020' , '$options' : 'i'}}
       ]}
   },
   {'$group' :
       {
           '_id' : '$rider',
           'number_of_deliveries' : {'$sum' : 1}
       }
   },
   {'$project' :
       {
           '_id' : 0,
           'rider' : '$_id',
           'number_of_deliveries' :  1
       }
   },
   {'$sort' :
       {
           'number_of_deliveries' :  -1
       }
   },
   {'$limit' : 5}
] ):
  pprint(t)
```

```
(venv) voclabs:~/environment $ python query.py
{'number_of_deliveries': 36,
 'rider': {'name': 'Melisandra Walworth',
           'rider_id': 26,
           'vehicle': {'make': 'Honda',
                       'plate_number': 'IW1782',
                       'vehicle_id': 26}}}
{'number_of_deliveries': 29,
 'rider': {'name': 'Abbey Duckers',
           'rider_id': 72,
           'vehicle': {'make': 'Hyundai',
                       'plate_number': 'DM3782',
                       'vehicle_id': 72}}}
{'number_of_deliveries': 28,
 'rider': {'name': 'Keven Chislett',
           'rider_id': 82,
           'vehicle': {'make': 'Hyundai',
                       'plate_number': 'AG0852',
                       'vehicle_id': 82}}}
{'number_of_deliveries': 28,
 'rider': {'name': 'Clement Vanns',
           'rider_id': 70,
           'vehicle': {'make': 'Toyota',
                       'plate_number': 'ZM1875',
                       'vehicle_id': 70}}}
{'number_of_deliveries': 28,
 'rider': {'name': 'Sig Marzello',
           'rider_id': 85,
           'vehicle': {'make': 'Kawasaki',
                       'plate_number': 'VA7561',
                       'vehicle_id': 85}}}
(venv) voclabs:~/environment $ 
```

# 3 MOST USED PAYMENT METHODS IN TAGUIG



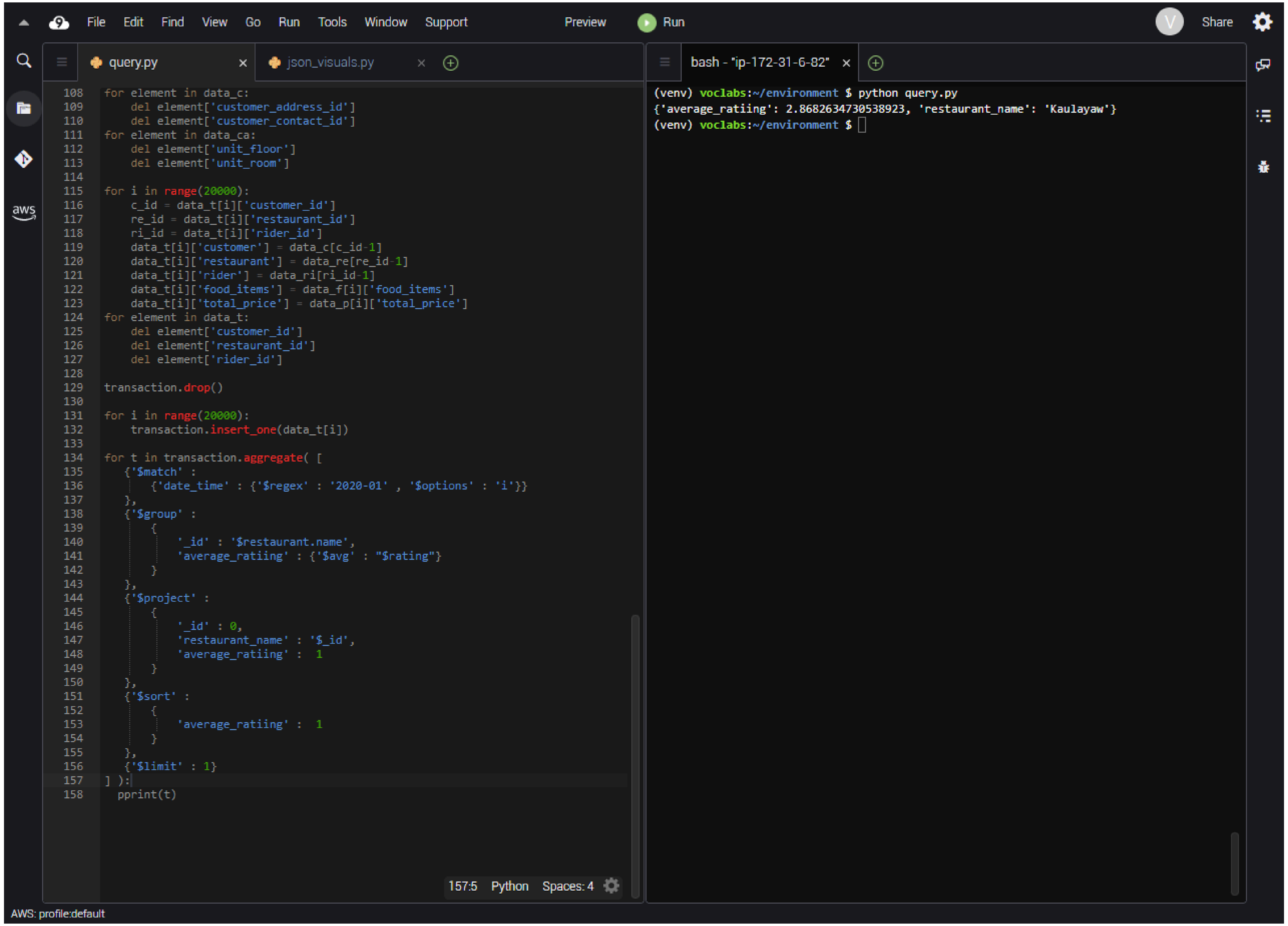
```python
for element in data_c:
    del element['customer_address_id']
    del element['customer_contact_id']
for element in data_ca:
    del element['unit_floor']
    del element['unit_room']

for i in range(20000):
    c_id = data_t[i]['customer_id']
    re_id = data_t[i]['restaurant_id']
    ri_id = data_t[i]['rider_id']
    data_t[i]['customer'] = data_c[c_id-1]
    data_t[i]['restaurant'] = data_re[re_id-1]
    data_t[i]['rider'] = data_ri[ri_id-1]
    data_t[i]['food_items'] = data_f[i]['food_items']
    data_t[i]['total_price'] = data_p[i]['total_price']
for element in data_t:
    del element['customer_id']
    del element['restaurant_id']
    del element['rider_id']

transaction.drop()

for i in range(20000):
    transaction.insert_one(data_t[i])

for t in transaction.aggregate( [
   {'$match' :
       {'customer.customer_address.city' : 'Taguig'}
   },
   {'$group' :
       {
           '_id' : '$payment_method',
           'number_of_times_used' : {'$sum' : 1}
       }
   },
   {'$project' :
       {
           '_id' : 0,
           'payment_method' : '$_id',
           'number_of_times_used' :  1
       }
   },
   {'$sort' :
       {
           'number_of_times_used' :  -1
       }
   },
   {'$limit' : 3}
] ):
  pprint(t)
```

```
(venv) voclabs:~/environment $ python query.py
{'number_of_times_used': 201, 'payment_method': 'GCash'}
{'number_of_times_used': 183, 'payment_method': 'Visa'}
{'number_of_times_used': 182, 'payment_method': 'Paymaya'}
(venv) voclabs:~/environment $
```

To improve the efficiency of payments in Taguig.

# RESTAURANT WITH LOWEST-RATED TRANSACTIONS IN JANUARY 2020

To identify which restaurant could benefit from suggestions of improvement.

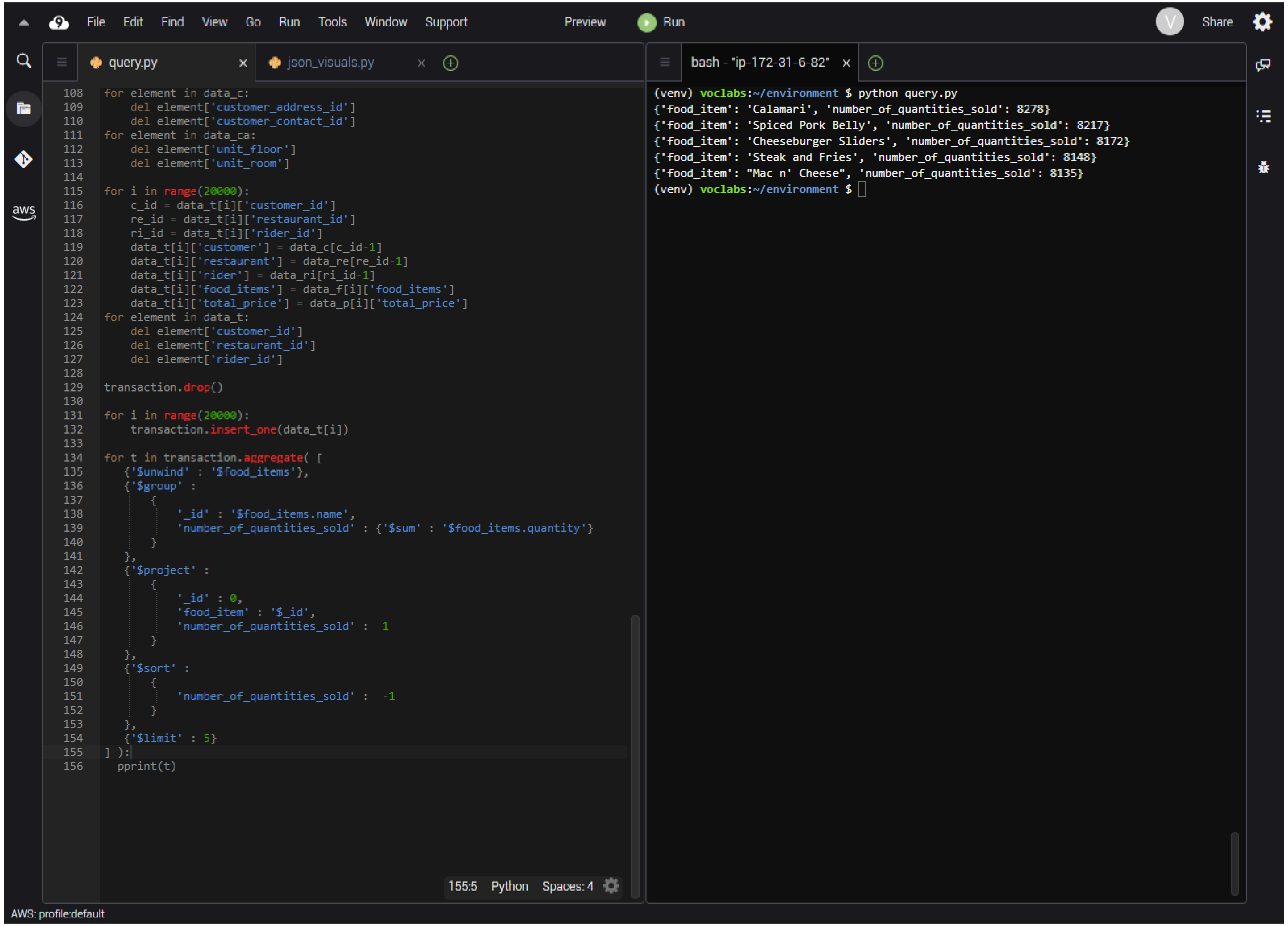


```python
for element in data_c:
    del element['customer_address_id']
    del element['customer_contact_id']
for element in data_ca:
    del element['unit_floor']
    del element['unit_room']

for i in range(20000):
    c_id = data_t[i]['customer_id']
    re_id = data_t[i]['restaurant_id']
    ri_id = data_t[i]['rider_id']
    data_t[i]['customer'] = data_c[c_id-1]
    data_t[i]['restaurant'] = data_re[re_id-1]
    data_t[i]['rider'] = data_ri[ri_id-1]
    data_t[i]['food_items'] = data_f[i]['food_items']
    data_t[i]['total_price'] = data_p[i]['total_price']
for element in data_t:
    del element['customer_id']
    del element['restaurant_id']
    del element['rider_id']

transaction.drop()

for i in range(20000):
    transaction.insert_one(data_t[i])

for t in transaction.aggregate( [
  {'$match' :
    {'date_time' : {'$regex' : '2020-01' , '$options' : 'i'}}
  },
  {'$group' :
    {
      '_id' : '$restaurant.name',
      'average_ratiing' : {'$avg' : "$rating"}
    }
  },
  {'$project' :
    {
      '_id' : 0,
      'restaurant_name' : '$_id',
      'average_ratiing' :  1
    }
  },
  {'$sort' :
    {
      'average_ratiing' :  1
    }
  },
  {'$limit' : 1}
] ):
  pprint(t)
```

```
(venv) voclabs:~/environment $ python query.py
{'average_ratiing': 2.8682634730538923, 'restaurant_name': 'Kaulayaw'}
(venv) voclabs:~/environment $ 
```

# 5 MOST POPULAR FOOD ITEMS AMONG ALL RESTAURANTS

To monitor demand of customerbase and inform decisions regarding use of resources.



```python
for element in data_c:
    del element['customer_address_id']
    del element['customer_contact_id']
for element in data_ca:
    del element['unit_floor']
    del element['unit_room']

for i in range(20000):
    c_id = data_t[i]['customer_id']
    re_id = data_t[i]['restaurant_id']
    ri_id = data_t[i]['rider_id']
    data_t[i]['customer'] = data_c[c_id-1]
    data_t[i]['restaurant'] = data_re[re_id-1]
    data_t[i]['rider'] = data_ri[ri_id-1]
    data_t[i]['food_items'] = data_f[i]['food_items']
    data_t[i]['total_price'] = data_p[i]['total_price']
for element in data_t:
    del element['customer_id']
    del element['restaurant_id']
    del element['rider_id']

transaction.drop()

for i in range(20000):
    transaction.insert_one(data_t[i])

for t in transaction.aggregate( [
   {'$unwind' : '$food_items'},
   {'$group' :
       {
           '_id' : '$food_items.name',
           'number_of_quantities_sold' : {'$sum' : '$food_items.quantity'}
       }
   },
   {'$project' :
       {
           '_id' : 0,
           'food_item' : '$_id',
           'number_of_quantities_sold' :  1
       }
   },
   {'$sort' :
       {
           'number_of_quantities_sold' :  -1
       }
   },
   {'$limit' : 5}
] ):
  pprint(t)
```

```
(venv) voclabs:~/environment $ python query.py
{'food_item': 'Calamari', 'number_of_quantities_sold': 8278}
{'food_item': 'Spiced Pork Belly', 'number_of_quantities_sold': 8217}
{'food_item': 'Cheeseburger Sliders', 'number_of_quantities_sold': 8172}
{'food_item': 'Steak and Fries', 'number_of_quantities_sold': 8148}
{'food_item': "Mac n' Cheese", 'number_of_quantities_sold': 8135}
(venv) voclabs:~/environment $
```