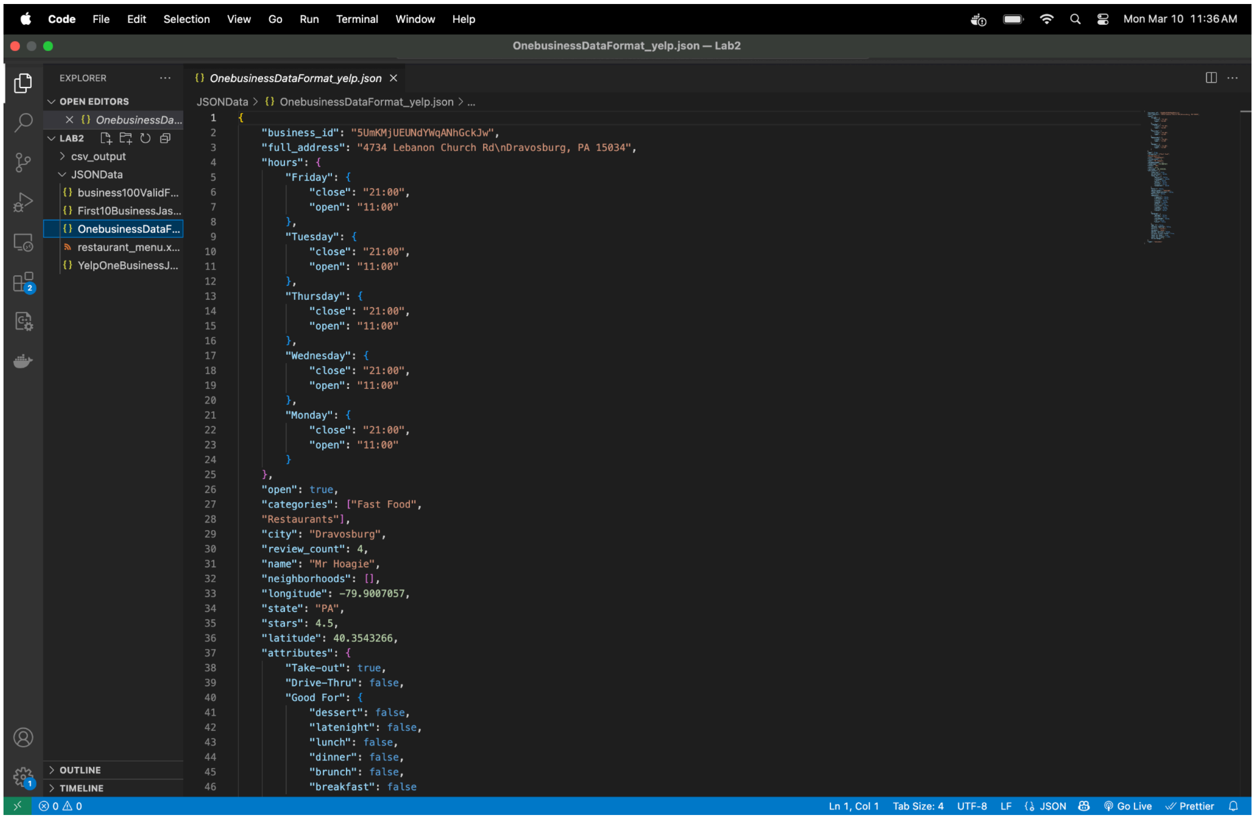
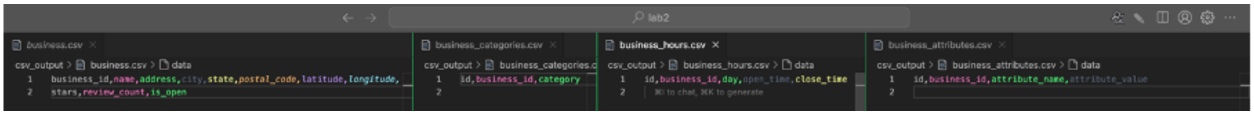
Santhosh Andavar
2818372
LAB2

JSON file



```json
{
    "business_id": "5UmKMjUEUNdYWqANhGckJw",
    "full_address": "4734 Lebanon Church Rd\nDravosburg, PA 15034",
    "hours": {
        "Friday": {
            "close": "21:00",
            "open": "11:00"
        },
        "Tuesday": {
            "close": "21:00",
            "open": "11:00"
        },
        "Thursday": {
            "close": "21:00",
            "open": "11:00"
        },
        "Wednesday": {
            "close": "21:00",
            "open": "11:00"
        },
        "Monday": {
            "close": "21:00",
            "open": "11:00"
        }
    },
    "open": true,
    "categories": ["Fast Food",
    "Restaurants"],
    "city": "Dravosburg",
    "review_count": 4,
    "name": "Mr Hoagie",
    "neighborhoods": [],
    "longitude": -79.9007057,
    "state": "PA",
    "stars": 4.5,
    "latitude": 40.3543266,
    "attributes": {
        "Take-out": true,
        "Drive-Thru": false,
        "Good For": {
            "dessert": false,
            "latenight": false,
            "lunch": false,
            "dinner": false,
            "brunch": false,
            "breakfast": false
```

Schema


lab2
business.csv
csv_output > business.csv > data
business_categories.csv
csv_output > business_categories.c
business_hours.csv
csv_output > business_hours.csv > data
business_attributes.csv
csv_output > business_attributes.csv > data

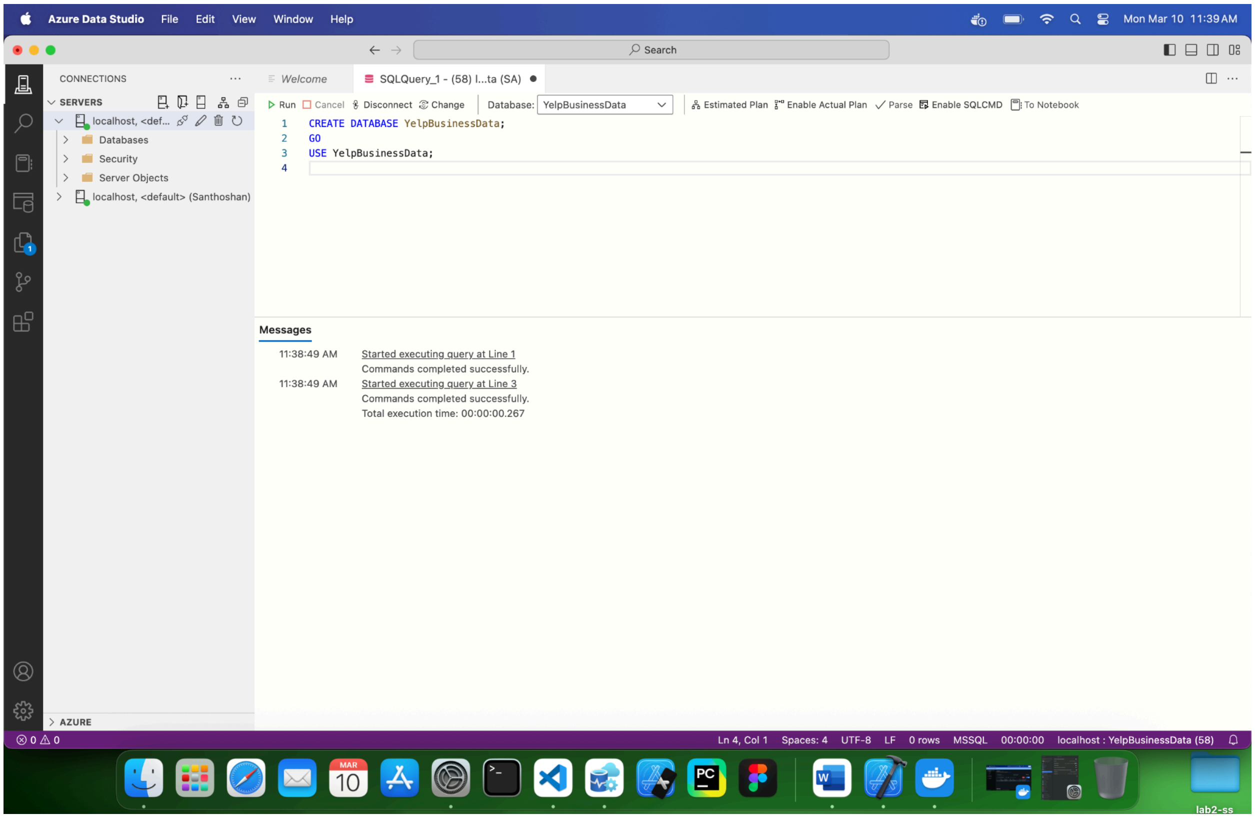
```
business_id,name,address,city,state,postal_code,latitude,longitude,
stars,review_count,is_open
```

```
id,business_id,category
```

```
id,business_id,day,open_time,close_time
⌘I to chat, ⌘K to generate
```

```
id,business_id,attribute_name,attribute_value
```

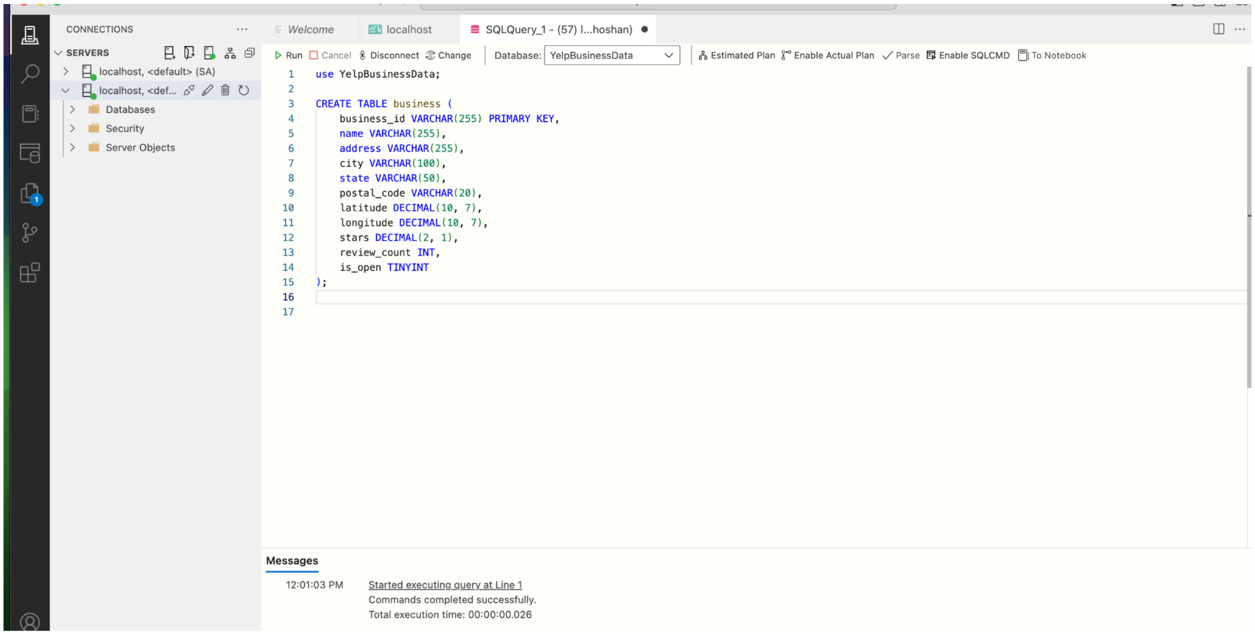
Created Database:



```sql
CREATE DATABASE YelpBusinessData;
GO
USE YelpBusinessData;
```

Messages

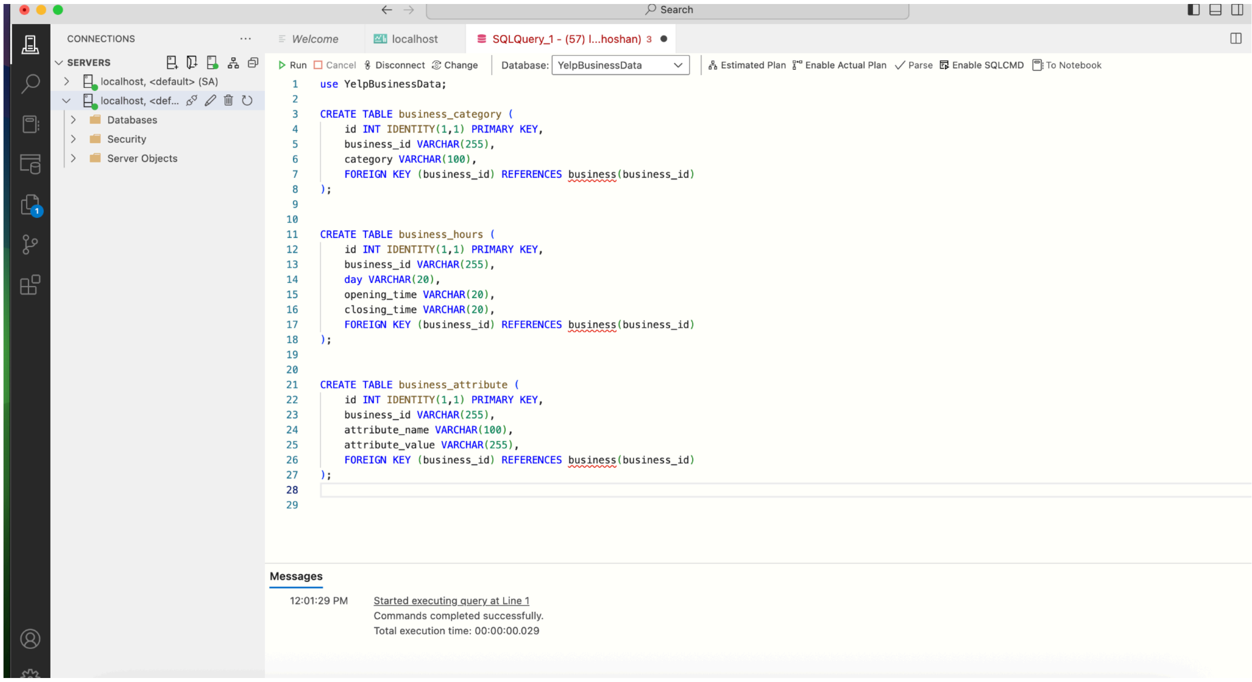
```
11:38:49 AM    Started executing query at Line 1
               Commands completed successfully.
11:38:49 AM    Started executing query at Line 3
               Commands completed successfully.
               Total execution time: 00:00:00.267
```



```sql
use YelpBusinessData;

CREATE TABLE business (
    business_id VARCHAR(255) PRIMARY KEY,
    name VARCHAR(255),
    address VARCHAR(255),
    city VARCHAR(100),
    state VARCHAR(50),
    postal_code VARCHAR(20),
    latitude DECIMAL(10, 7),
    longitude DECIMAL(10, 7),
    stars DECIMAL(2, 1),
    review_count INT,
    is_open TINYINT
);
```

Messages

```
12:01:03 PM    Started executing query at Line 1
               Commands completed successfully.
               Total execution time: 00:00:00.026
```

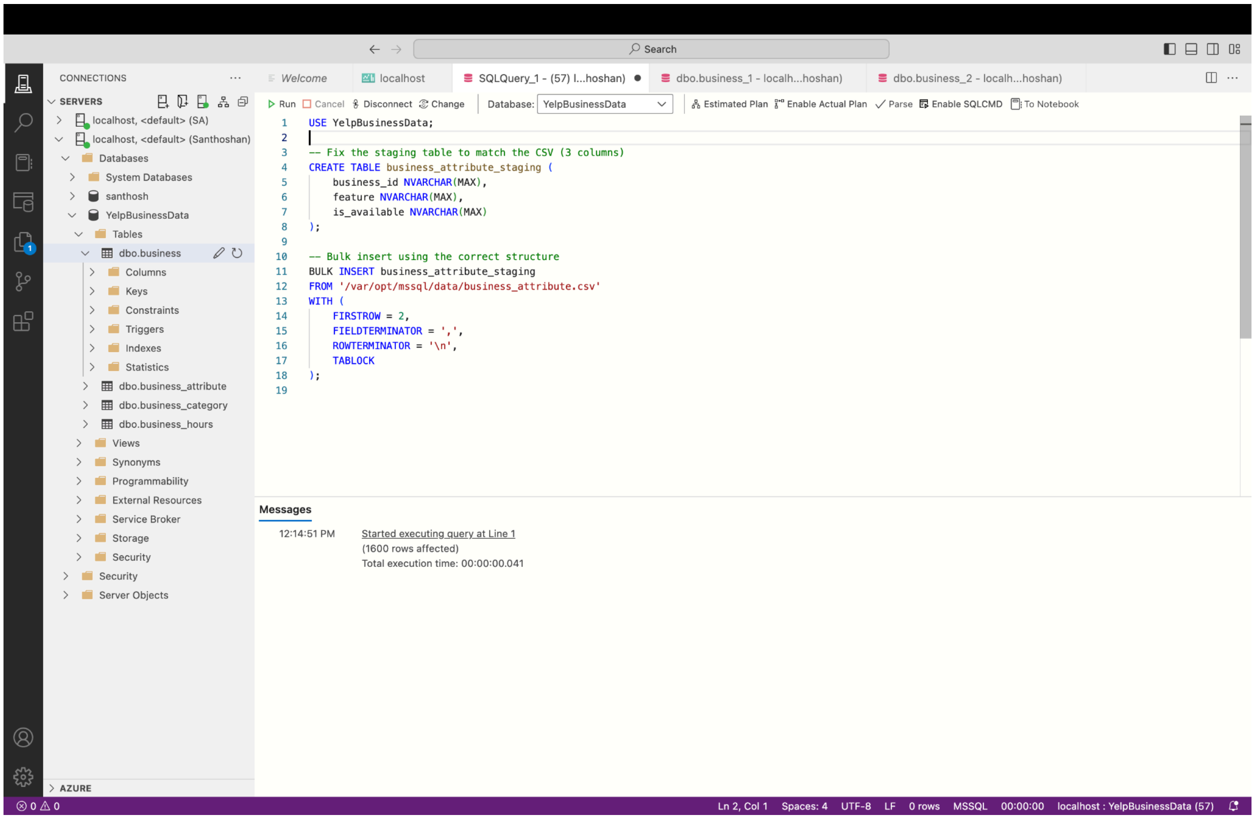


Database: YelpBusinessData

```sql
use YelpBusinessData;

CREATE TABLE business_category (
    id INT IDENTITY(1,1) PRIMARY KEY,
    business_id VARCHAR(255),
    category VARCHAR(100),
    FOREIGN KEY (business_id) REFERENCES business(business_id)
);


CREATE TABLE business_hours (
    id INT IDENTITY(1,1) PRIMARY KEY,
    business_id VARCHAR(255),
    day VARCHAR(20),
    opening_time VARCHAR(20),
    closing_time VARCHAR(20),
    FOREIGN KEY (business_id) REFERENCES business(business_id)
);


CREATE TABLE business_attribute (
    id INT IDENTITY(1,1) PRIMARY KEY,
    business_id VARCHAR(255),
    attribute_name VARCHAR(100),
    attribute_value VARCHAR(255),
    FOREIGN KEY (business_id) REFERENCES business(business_id)
);
```

Messages

```
12:01:29 PM    Started executing query at Line 1
               Commands completed successfully.
               Total execution time: 00:00:00.029
```

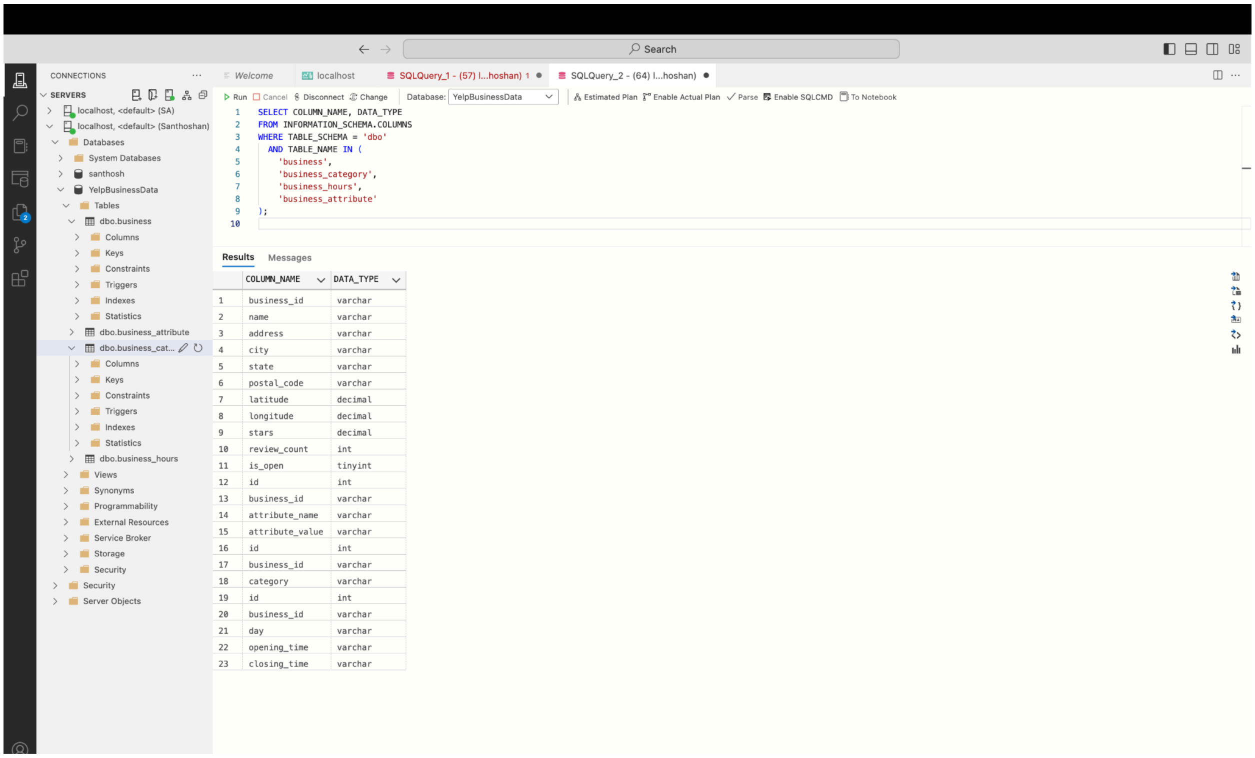


Database: YelpBusinessData

```sql
USE YelpBusinessData;

-- Fix the staging table to match the CSV (3 columns)
CREATE TABLE business_attribute_staging (
    business_id NVARCHAR(MAX),
    feature NVARCHAR(MAX),
    is_available NVARCHAR(MAX)
);

-- Bulk insert using the correct structure
BULK INSERT business_attribute_staging
FROM '/var/opt/mssql/data/business_attribute.csv'
WITH (
    FIRSTROW = 2,
    FIELDTERMINATOR = ',',
    ROWTERMINATOR = '\n',
    TABLOCK
);
```

Messages

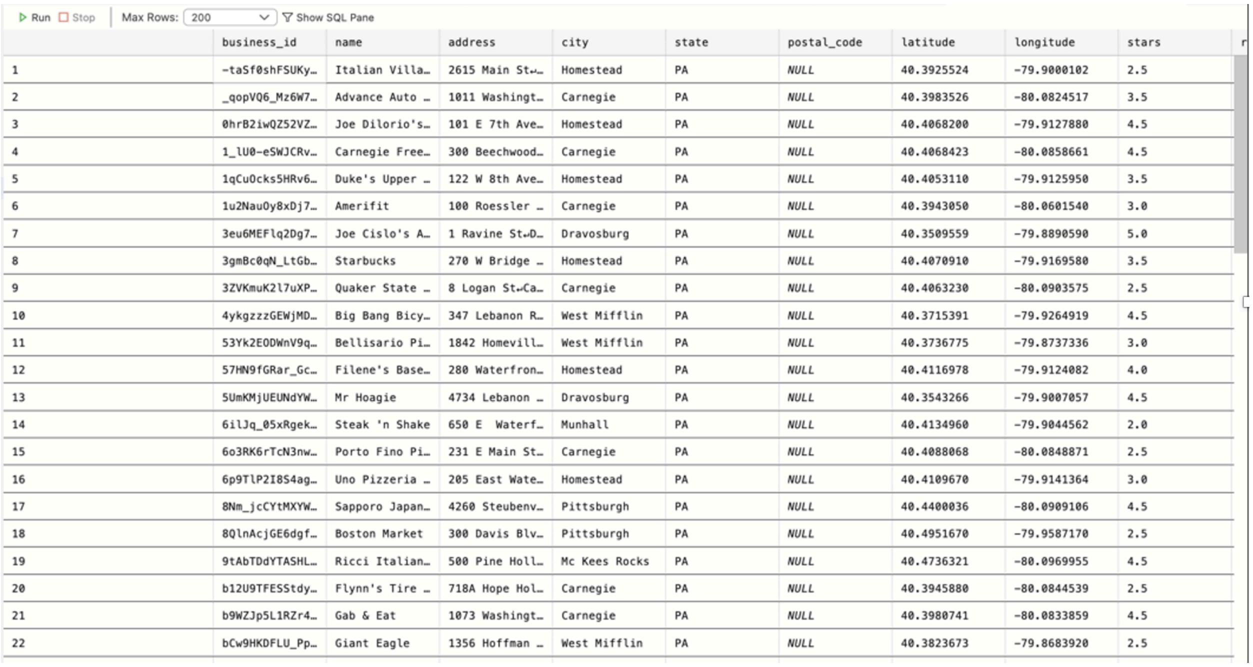
```
12:14:51 PM    Started executing query at Line 1
               (1600 rows affected)
               Total execution time: 00:00:00.041
```



Database: YelpBusinessData

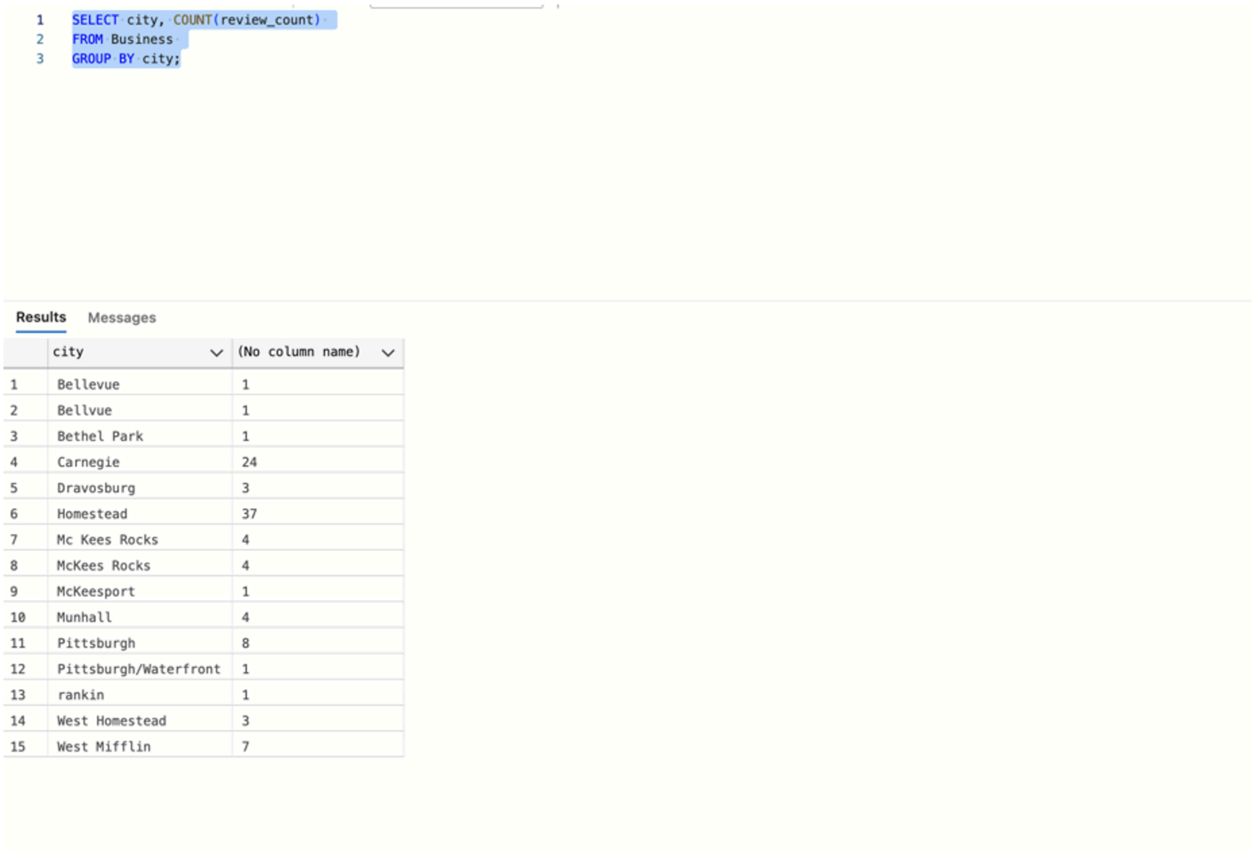
```sql
SELECT COLUMN_NAME, DATA_TYPE
FROM INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_SCHEMA = 'dbo'
  AND TABLE_NAME IN (
    'business',
    'business_category',
    'business_hours',
    'business_attribute'
);
```

	COLUMN_NAME	DATA_TYPE
1	business_id	varchar
2	name	varchar
3	address	varchar
4	city	varchar
5	state	varchar
6	postal_code	varchar
7	latitude	decimal
8	longitude	decimal
9	stars	decimal
10	review_count	int
11	is_open	tinyint
12	id	int
13	business_id	varchar
14	attribute_name	varchar
15	attribute_value	varchar
16	id	int
17	business_id	varchar
18	category	varchar
19	id	int
20	business_id	varchar
21	day	varchar
22	opening_time	varchar
23	closing_time	varchar



Max Rows: 200

	business_id	name	address	city	state	postal_code	latitude	longitude	stars
1	-taSf0shFSUKy…	Italian Villa…	2615 Main St↵…	Homestead	PA	NULL	40.3925524	-79.9000102	2.5
2	_qopVQ6_Mz6W7…	Advance Auto …	1011 Washingt…	Carnegie	PA	NULL	40.3983526	-80.0824517	3.5
3	0hrB2iwQZ52VZ…	Joe Dilorio's…	101 E 7th Ave…	Homestead	PA	NULL	40.4068200	-79.9127880	4.5
4	1_lU0-eSWJCRv…	Carnegie Free…	300 Beechwood…	Carnegie	PA	NULL	40.4068423	-80.0858661	4.5
5	1qCuOcks5HRv6…	Duke's Upper …	122 W 8th Ave…	Homestead	PA	NULL	40.4053110	-79.9125950	3.5
6	1u2NauOy8xDj7…	Amerifit	100 Roessler …	Carnegie	PA	NULL	40.3943050	-80.0601540	3.0
7	3eu6MEFlq2Dg7…	Joe Cislo's A…	1 Ravine St↵D…	Dravosburg	PA	NULL	40.3509559	-79.8890590	5.0
8	3gmBc0qN_LtGb…	Starbucks	270 W Bridge …	Homestead	PA	NULL	40.4070910	-79.9169580	3.5
9	3ZVKmuK2l7uXP…	Quaker State …	8 Logan St↵Ca…	Carnegie	PA	NULL	40.4063230	-80.0903575	2.5
10	4ykgzzzGEWjMD…	Big Bang Bicy…	347 Lebanon R…	West Mifflin	PA	NULL	40.3715391	-79.9264919	4.5
11	53Yk2EODWnV9q…	Bellisario Pi…	1842 Homevill…	West Mifflin	PA	NULL	40.3736775	-79.8737336	3.0
12	57HN9fGRar_Gc…	Filene's Base…	280 Waterfron…	Homestead	PA	NULL	40.4116978	-79.9124082	4.0
13	5UmKMjUEUNdYW…	Mr Hoagie	4734 Lebanon …	Dravosburg	PA	NULL	40.3543266	-79.9007057	4.5
14	6ilJq_05xRgek…	Steak 'n Shake	650 E Waterf…	Munhall	PA	NULL	40.4134960	-79.9044562	2.0
15	6o3RK6rTcN3nw…	Porto Fino Pi…	231 E Main St…	Carnegie	PA	NULL	40.4088068	-80.0848871	2.5
16	6p9TlP2I8S4ag…	Uno Pizzeria …	205 East Wate…	Homestead	PA	NULL	40.4109670	-79.9141364	3.0
17	8Nm_jcCYtMXYW…	Sapporo Japan…	4260 Steubenv…	Pittsburgh	PA	NULL	40.4400036	-80.0909106	4.5
18	8QlnAcjGE6dgf…	Boston Market	300 Davis Blv…	Pittsburgh	PA	NULL	40.4951670	-79.9587170	2.5
19	9tAbTDdYTASHL…	Ricci Italian…	500 Pine Holl…	Mc Kees Rocks	PA	NULL	40.4736321	-80.0969955	4.5
20	b12U9TFESStdy…	Flynn's Tire …	718A Hope Hol…	Carnegie	PA	NULL	40.3945880	-80.0844539	2.5
21	b9WZJp5L1RZr4…	Gab & Eat	1073 Washingt…	Carnegie	PA	NULL	40.3980741	-80.0833859	4.5
22	bCw9HKDFLU_Pp…	Giant Eagle	1356 Hoffman …	West Mifflin	PA	NULL	40.3823673	-79.8683920	2.5



```
SELECT city, COUNT(review_count)
FROM Business
GROUP BY city;
```

	city	(No column name)
1	Bellevue	1
2	Bellvue	1
3	Bethel Park	1
4	Carnegie	24
5	Dravosburg	3
6	Homestead	37
7	Mc Kees Rocks	4
8	McKees Rocks	4
9	McKeesport	1
10	Munhall	4
11	Pittsburgh	8
12	Pittsburgh/Waterfront	1
13	rankin	1
14	West Homestead	3
15	West Mifflin	7

JSON to CSV

```
#Main.py


import json
import csv
import pandas as pd
import os
import re

# Print current working directory
print(f"Current working directory: {os.getcwd()}")

# Function to read and fix JSON
def read_json_file(file_path):
    print(f"Opening file: {file_path}")
    try:
        with open(file_path, 'r', encoding='utf-8') as file:
            content = file.read()

            # Basic fixes for malformed JSON
            content = content.replace('"Price Range": $$', '"Price Range": 2')
            content = content.replace('"price_range": $$', '"price_range": 2')
            content = re.sub(r'}\s*{', '},{', content)

            # Try parsing JSON
            try:
                data = json.loads(content)
                return data.get("Business", data)
            except json.JSONDecodeError:
                print("Standard JSON parse failed, trying line-by-line repair...")

        # Try line-by-line parsing
        businesses = []
        with open(file_path, 'r', encoding='utf-8') as file:
            lines = file.readlines()

        obj_lines = []
        for line in lines:
            line = line.strip()
            if line.startswith('{') and line.endswith('}'):
                try:
                    business = json.loads(line)
                    businesses.append(business)
                except json.JSONDecodeError:
                    continue

        return businesses

    except FileNotFoundError:
        print(f"File not found: {file_path}")
        exit(1)
    except Exception as e:
        print(f"Unexpected error: {type(e).__name__}: {e}")
        exit(1)

# File path
```

```
json_file = 'business100ValidForm.json'

# Show files for context
print("Files in current directory:")
for file in os.listdir('.'):
    print(f"  - {file}")

# Read and process the JSON
businesses = read_json_file(json_file)

if not businesses:
    print("No business records found.")
    exit(1)

# Prepare data containers
business_data = []
category_data = []
hours_data = []
attribute_data = []

print(f"Processing {len(businesses)} business records...")

for biz in businesses:
    address = biz.get('address', biz.get('full_address', ''))
    business_data.append({
        'business_id': biz.get('business_id', ''),
        'name': biz.get('name', ''),
        'address': address,
        'city': biz.get('city', ''),
        'state': biz.get('state', ''),
        'postal_code': biz.get('postal_code', ''),
        'latitude': biz.get('latitude', 0),
        'longitude': biz.get('longitude', 0),
        'stars': biz.get('stars', 0),
        'review_count': biz.get('review_count', 0),
        'is_open': 1 if biz.get('open', False) else 0
    })

    # Categories
    categories = biz.get('categories', [])
```

```
if isinstance(categories, str):
    categories = categories.split(', ')
for category in categories:
    category_data.append({
        'business_id': biz.get('business_id'),
        'category': category
    })

# Hours
for day, hours in biz.get('hours', {}).items():
    if isinstance(hours, dict):
        hours_data.append({
            'business_id': biz.get('business_id'),
            'day': day,
            'open': hours.get('open', ''),
            'close': hours.get('close', '')
        })
    elif isinstance(hours, str):
        times = hours.split('-')
        hours_data.append({
            'business_id': biz.get('business_id'),
            'day': day,
            'open': times[0],
            'close': times[1] if len(times) > 1 else ''
        })

# Attributes
for attr, value in biz.get('attributes', {}).items():
    if isinstance(value, dict):
        for sub_attr, sub_val in value.items():
            attribute_data.append({
                'business_id': biz.get('business_id'),
                'attribute_name': f"{attr}.{sub_attr}",
                'attribute_value': str(sub_val)
            })
    else:
        attribute_data.append({
            'business_id': biz.get('business_id'),
            'attribute_name': attr,
            'attribute_value': str(value)
        })

# Save to CSV
pd.DataFrame(business_data).to_csv('business.csv', index=False)
pd.DataFrame(category_data).to_csv('business_category.csv', index=False)
pd.DataFrame(hours_data).to_csv('business_hours.csv', index=False)
pd.DataFrame(attribute_data).to_csv('business_attribute.csv', index=False)

print("CSV files created:")
print("  - business.csv")
print("  - business_category.csv")
print("  - business_hours.csv")
print("  - business_attribute.csv")
```