CS434 – Data Base Theory and Design

Project #4

Team Database Application (TDA): Part 4 – Loading Large Data Sets

Team

Lipika Baniya | 800794205 | lbaniya@siue.edu

The domain I would like to manage with the TDA is **Washington DC Crime Datasets 2024** by the District of Columbia Metropolitan Police Department (MPD).

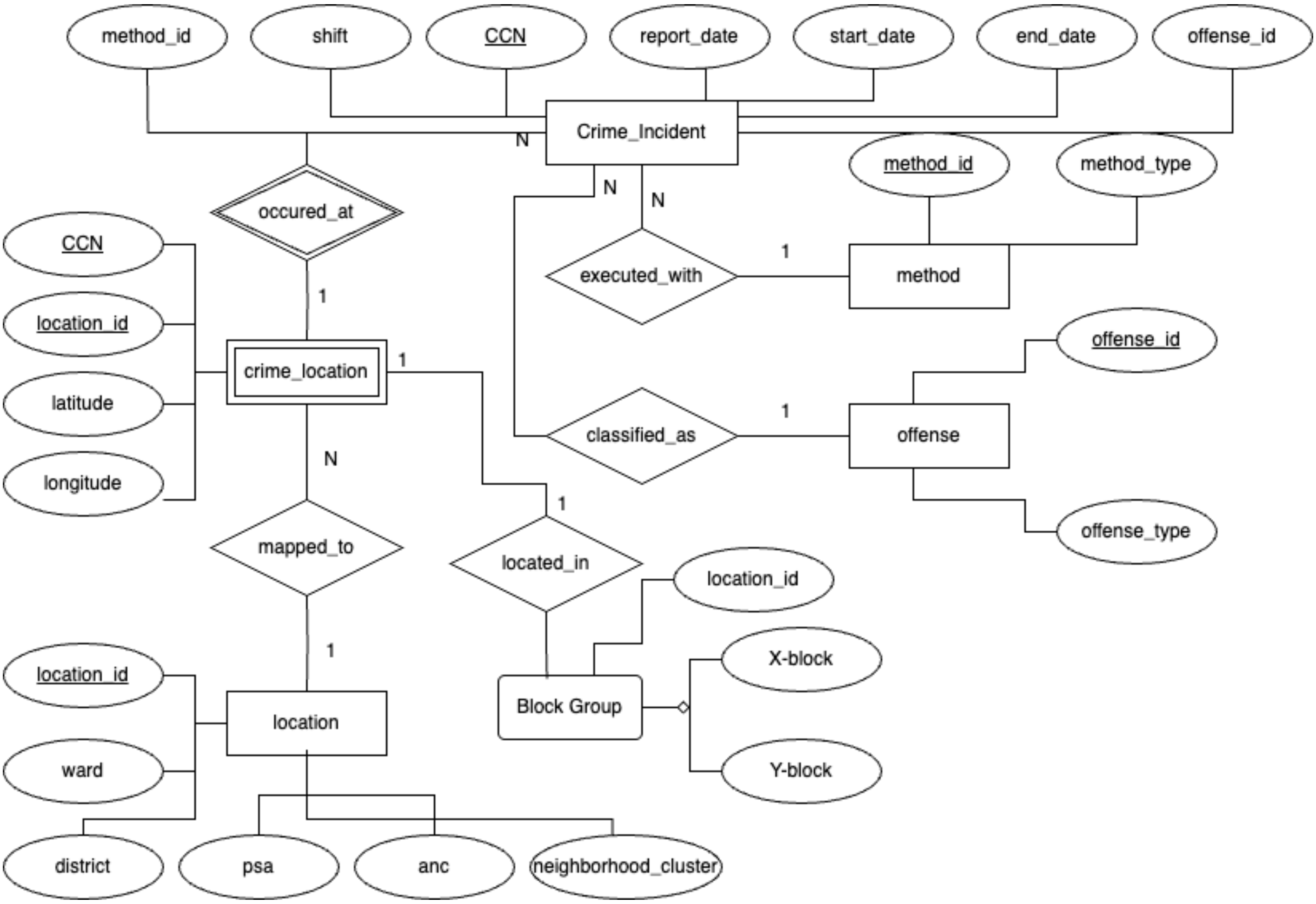
General Nature of application

The main goal of an Entity Relationship Diagram (ER Diagram) is to explain the relationship between entities; it is a structural design of the database. Through the help of specialized symbols, it helps to define the relationship between entities. It is based on three main principles entities, attributes and relationships, these help to design the database that would be required before implementing the database. It is a systematic process to design a database as it would require analyzing all requirements.

About Data

Washington, D.C. has been facing significant challenges in ensuring public safety due to the varying and growing crime rates in different neighborhoods and time periods. It is important for law enforcement agencies to understand when and where crimes occur so that it can respond efficiently and allocate limited resources wisely. Imagine a robust database system that is designed to handle this task effectively, because without a data-driven approach and structured database, policing efforts may remain reactive, which would result in delays or gaps in coverage in high-risk areas. This database includes various entities, each representing a key component of crime data management.

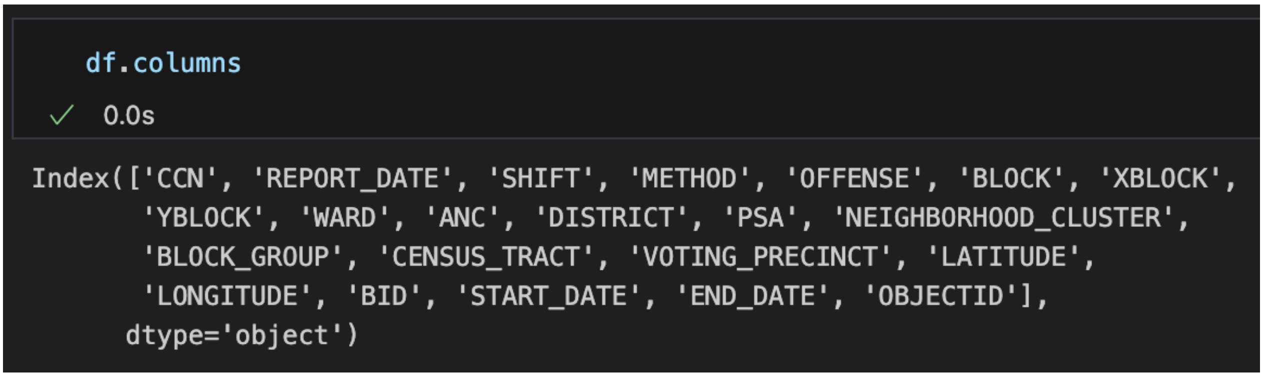
ER Diagram



1. Dataset

The dataset is available in one csv file Crime_Incidents_in_2024.csv

- Size: (7,306,043 bytes)
- Columns:



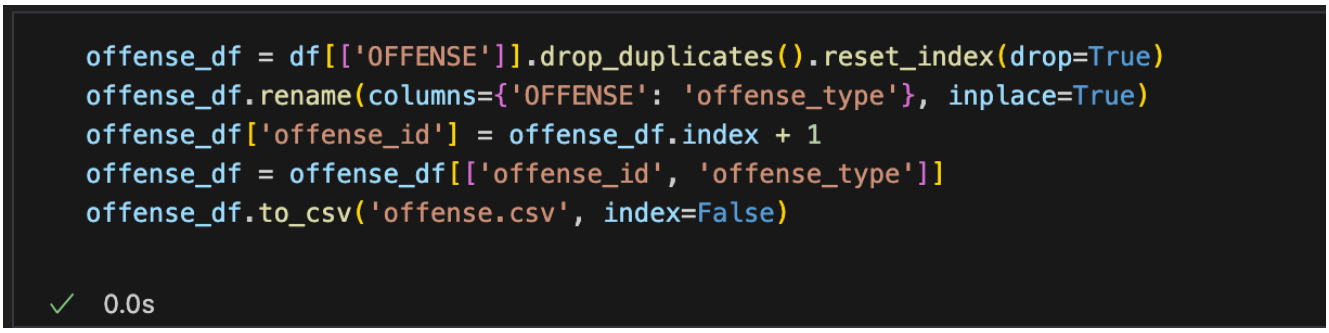
```
df.columns
✓ 0.0s

Index(['CCN', 'REPORT_DATE', 'SHIFT', 'METHOD', 'OFFENSE', 'BLOCK', 'XBLOCK',
       'YBLOCK', 'WARD', 'ANC', 'DISTRICT', 'PSA', 'NEIGHBORHOOD_CLUSTER',
       'BLOCK_GROUP', 'CENSUS_TRACT', 'VOTING_PRECINCT', 'LATITUDE',
       'LONGITUDE', 'BID', 'START_DATE', 'END_DATE', 'OBJECTID'],
      dtype='object')
```

2. Data Cleaning and Separating into Different csv files for Each Table

I used Python to separate the tables from the csv files. Python was used to:

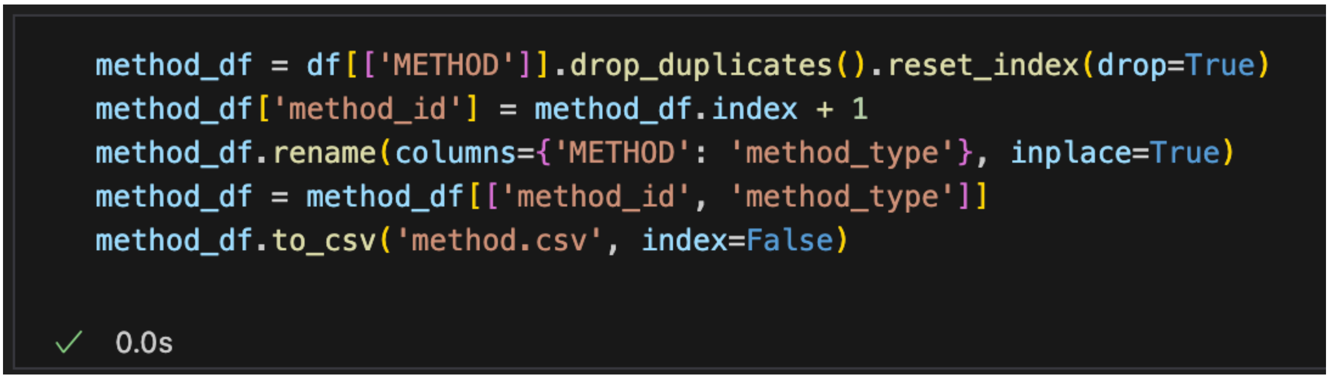
- Separate the csv based on table
- Assign primary and foreign keys
- Remove duplicate values that may occur in keys

Following is a snippet of code use for data transformation:



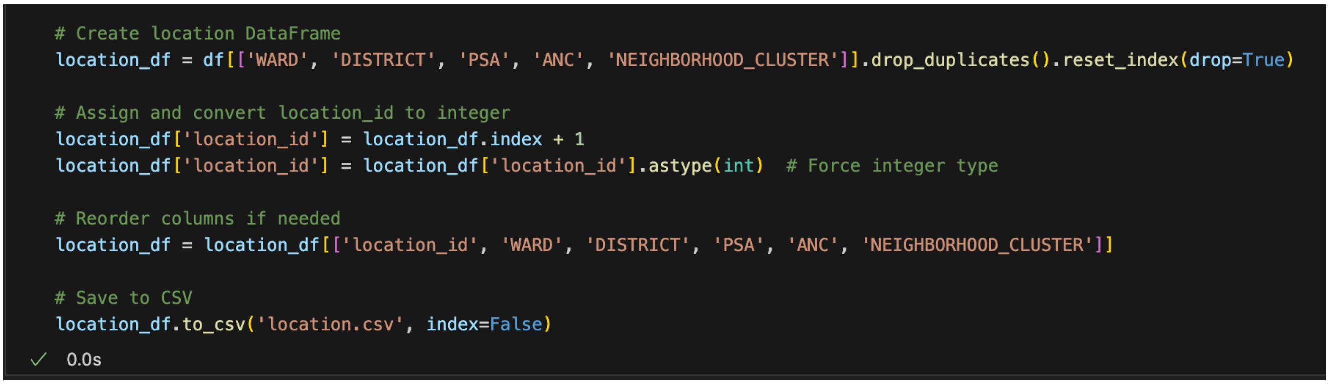
```python
offense_df = df[['OFFENSE']].drop_duplicates().reset_index(drop=True)
offense_df.rename(columns={'OFFENSE': 'offense_type'}, inplace=True)
offense_df['offense_id'] = offense_df.index + 1
offense_df = offense_df[['offense_id', 'offense_type']]
offense_df.to_csv('offense.csv', index=False)
```

✓ 0.0s



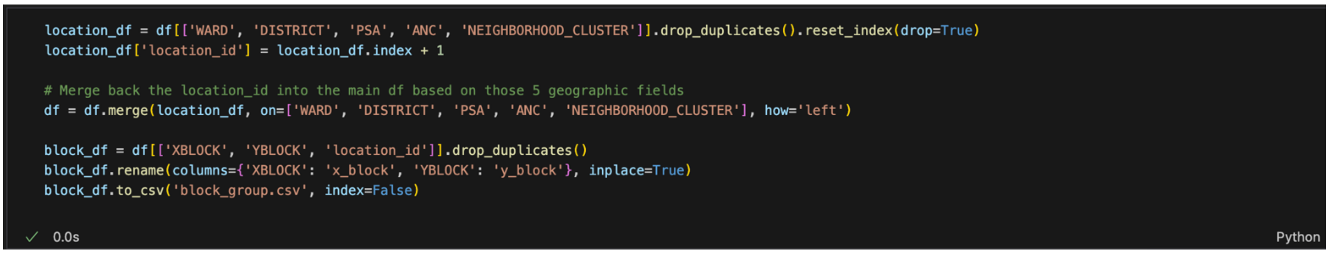
```python
method_df = df[['METHOD']].drop_duplicates().reset_index(drop=True)
method_df['method_id'] = method_df.index + 1
method_df.rename(columns={'METHOD': 'method_type'}, inplace=True)
method_df = method_df[['method_id', 'method_type']]
method_df.to_csv('method.csv', index=False)
```

✓ 0.0s



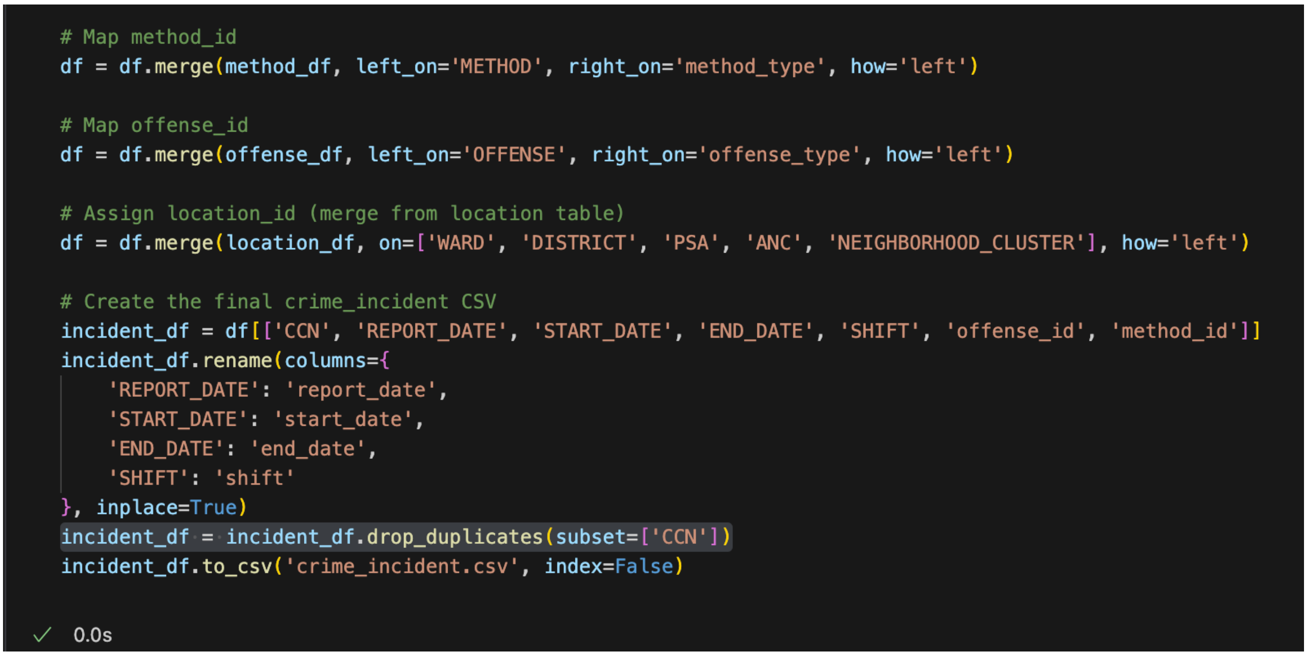
```python
# Create location DataFrame
location_df = df[['WARD', 'DISTRICT', 'PSA', 'ANC', 'NEIGHBORHOOD_CLUSTER']].drop_duplicates().reset_index(drop=True)

# Assign and convert location_id to integer
location_df['location_id'] = location_df.index + 1
location_df['location_id'] = location_df['location_id'].astype(int)  # Force integer type

# Reorder columns if needed
location_df = location_df[['location_id', 'WARD', 'DISTRICT', 'PSA', 'ANC', 'NEIGHBORHOOD_CLUSTER']]

# Save to CSV
location_df.to_csv('location.csv', index=False)
```

✓ 0.0s



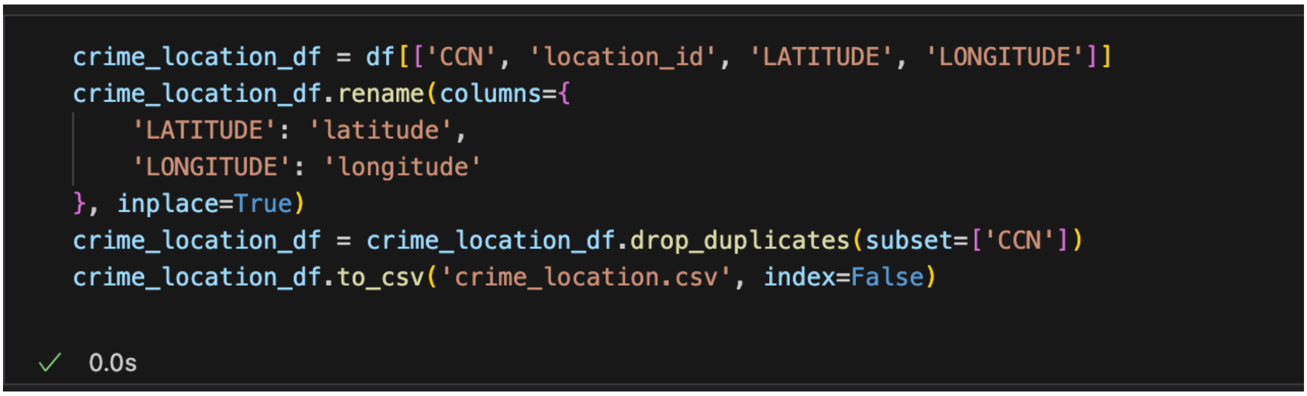
```python
location_df = df[['WARD', 'DISTRICT', 'PSA', 'ANC', 'NEIGHBORHOOD_CLUSTER']].drop_duplicates().reset_index(drop=True)
location_df['location_id'] = location_df.index + 1

# Merge back the location_id into the main df based on those 5 geographic fields
df = df.merge(location_df, on=['WARD', 'DISTRICT', 'PSA', 'ANC', 'NEIGHBORHOOD_CLUSTER'], how='left')

block_df = df[['XBLOCK', 'YBLOCK', 'location_id']].drop_duplicates()
block_df.rename(columns={'XBLOCK': 'x_block', 'YBLOCK': 'y_block'}, inplace=True)
block_df.to_csv('block_group.csv', index=False)
```

✓ 0.0s Python



```python
# Map method_id
df = df.merge(method_df, left_on='METHOD', right_on='method_type', how='left')

# Map offense_id
df = df.merge(offense_df, left_on='OFFENSE', right_on='offense_type', how='left')

# Assign location_id (merge from location table)
df = df.merge(location_df, on=['WARD', 'DISTRICT', 'PSA', 'ANC', 'NEIGHBORHOOD_CLUSTER'], how='left')

# Create the final crime_incident CSV
incident_df = df[['CCN', 'REPORT_DATE', 'START_DATE', 'END_DATE', 'SHIFT', 'offense_id', 'method_id']]
incident_df.rename(columns={
    'REPORT_DATE': 'report_date',
    'START_DATE': 'start_date',
    'END_DATE': 'end_date',
    'SHIFT': 'shift'
}, inplace=True)
incident_df = incident_df.drop_duplicates(subset=['CCN'])
incident_df.to_csv('crime_incident.csv', index=False)
```

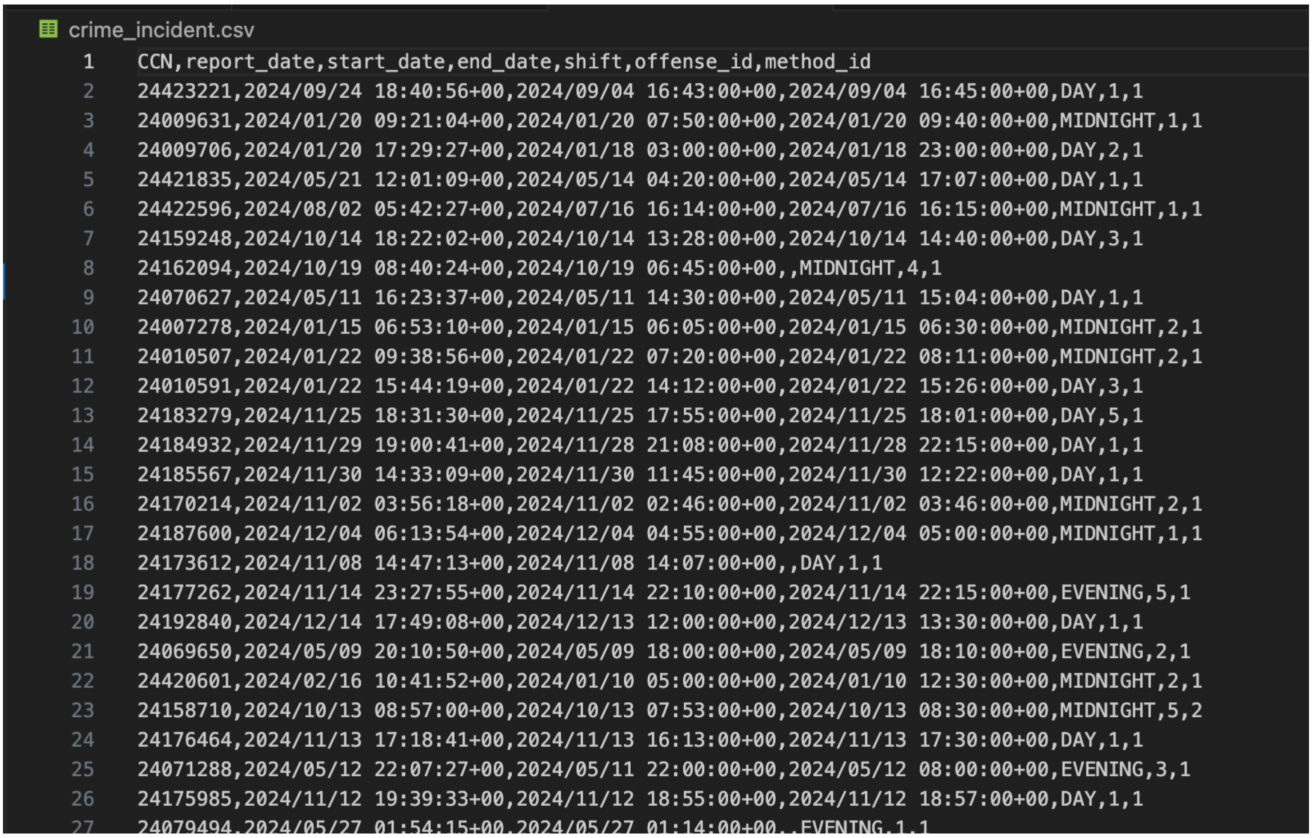
✓ 0.0s



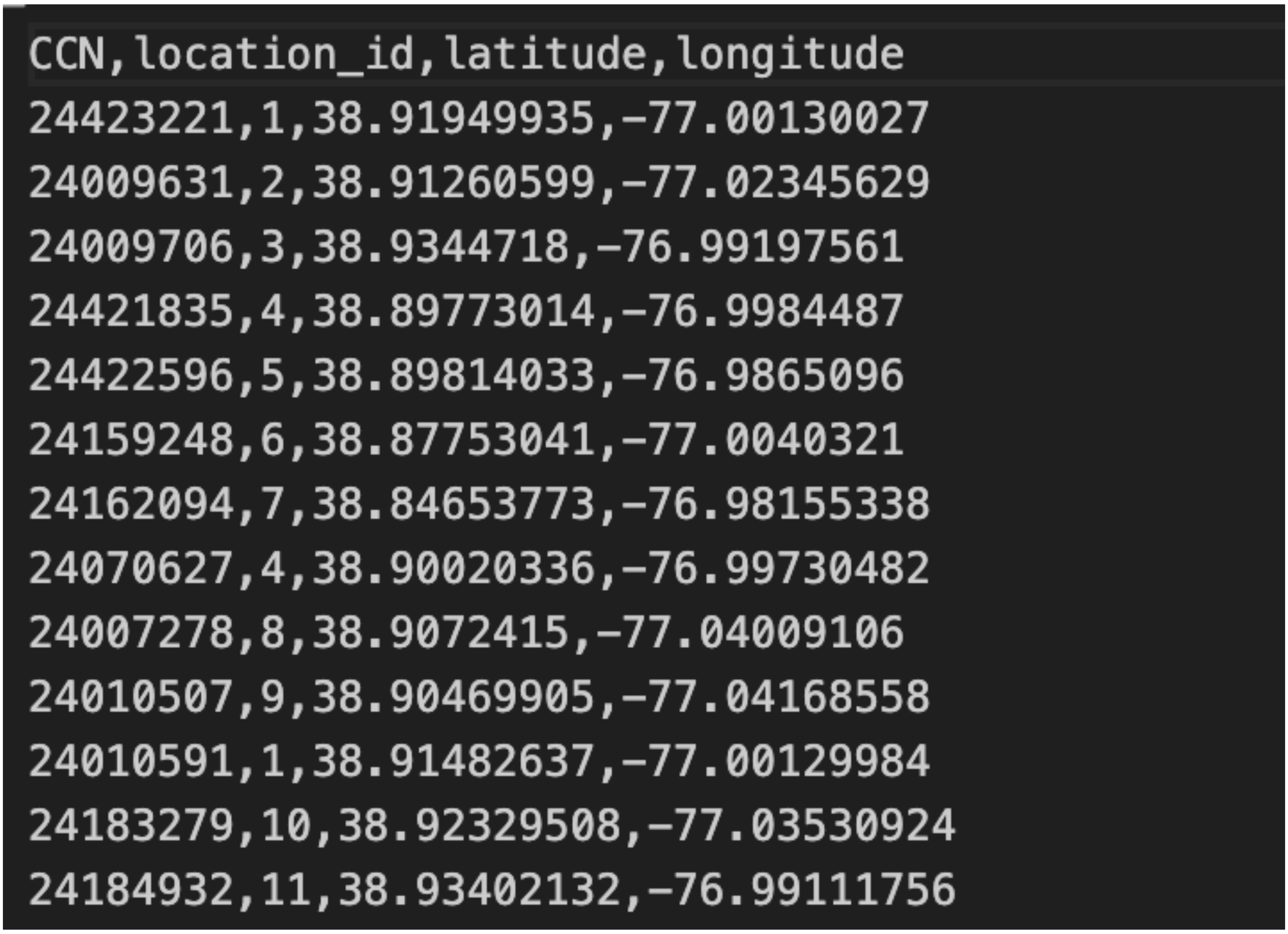
```python
crime_location_df = df[['CCN', 'location_id', 'LATITUDE', 'LONGITUDE']]
crime_location_df.rename(columns={
    'LATITUDE': 'latitude',
    'LONGITUDE': 'longitude'
}, inplace=True)
crime_location_df = crime_location_df.drop_duplicates(subset=['CCN'])
crime_location_df.to_csv('crime_location.csv', index=False)
```

✓ 0.0s

Modified csv file on tables looked as follows:



```
CCN,report_date,start_date,end_date,shift,offense_id,method_id
24423221,2024/09/24 18:40:56+00,2024/09/04 16:43:00+00,2024/09/04 16:45:00+00,DAY,1,1
24009631,2024/01/20 09:21:04+00,2024/01/20 07:50:00+00,2024/01/20 09:40:00+00,MIDNIGHT,1,1
24009706,2024/01/20 17:29:27+00,2024/01/18 03:00:00+00,2024/01/18 23:00:00+00,DAY,2,1
24421835,2024/05/21 12:01:09+00,2024/05/14 04:20:00+00,2024/05/14 17:07:00+00,DAY,1,1
24422596,2024/08/02 05:42:27+00,2024/07/16 16:14:00+00,2024/07/16 16:15:00+00,MIDNIGHT,1,1
24159248,2024/10/14 18:22:02+00,2024/10/14 13:28:00+00,2024/10/14 14:40:00+00,DAY,3,1
24162094,2024/10/19 08:40:24+00,2024/10/19 06:45:00+00,,MIDNIGHT,4,1
24070627,2024/05/11 16:23:37+00,2024/05/11 14:30:00+00,2024/05/11 15:04:00+00,DAY,1,1
24007278,2024/01/15 06:53:10+00,2024/01/15 06:05:00+00,2024/01/15 06:30:00+00,MIDNIGHT,2,1
24010507,2024/01/22 09:38:56+00,2024/01/22 07:20:00+00,2024/01/22 08:11:00+00,MIDNIGHT,2,1
24010591,2024/01/22 15:44:19+00,2024/01/22 14:12:00+00,2024/01/22 15:26:00+00,DAY,3,1
24183279,2024/11/25 18:31:30+00,2024/11/25 17:55:00+00,2024/11/25 18:01:00+00,DAY,5,1
24184932,2024/11/29 19:00:41+00,2024/11/28 21:08:00+00,2024/11/28 22:15:00+00,DAY,1,1
24185567,2024/11/30 14:33:09+00,2024/11/30 11:45:00+00,2024/11/30 12:22:00+00,DAY,1,1
24170214,2024/11/02 03:56:18+00,2024/11/02 02:46:00+00,2024/11/02 03:46:00+00,MIDNIGHT,2,1
24187600,2024/12/04 06:13:54+00,2024/12/04 04:55:00+00,2024/12/04 05:00:00+00,MIDNIGHT,1,1
24173612,2024/11/08 14:47:13+00,2024/11/08 14:07:00+00,,DAY,1,1
24177262,2024/11/14 23:27:55+00,2024/11/14 22:10:00+00,2024/11/14 22:15:00+00,EVENING,5,1
24192840,2024/12/14 17:49:08+00,2024/12/13 12:00:00+00,2024/12/13 13:30:00+00,DAY,1,1
24069650,2024/05/09 20:10:50+00,2024/05/09 18:00:00+00,2024/05/09 18:10:00+00,EVENING,2,1
24420601,2024/02/16 10:41:52+00,2024/01/10 05:00:00+00,2024/01/10 12:30:00+00,MIDNIGHT,2,1
24158710,2024/10/13 08:57:00+00,2024/10/13 07:53:00+00,2024/10/13 08:30:00+00,MIDNIGHT,5,2
24176464,2024/11/13 17:18:41+00,2024/11/13 16:13:00+00,2024/11/13 17:30:00+00,DAY,1,1
24071288,2024/05/12 22:07:27+00,2024/05/11 22:00:00+00,2024/05/12 08:00:00+00,EVENING,3,1
24175985,2024/11/12 19:39:33+00,2024/11/12 18:55:00+00,2024/11/12 18:57:00+00,DAY,1,1
24079494,2024/05/27 01:54:15+00,2024/05/27 01:14:00+00,,EVENING,1,1
```



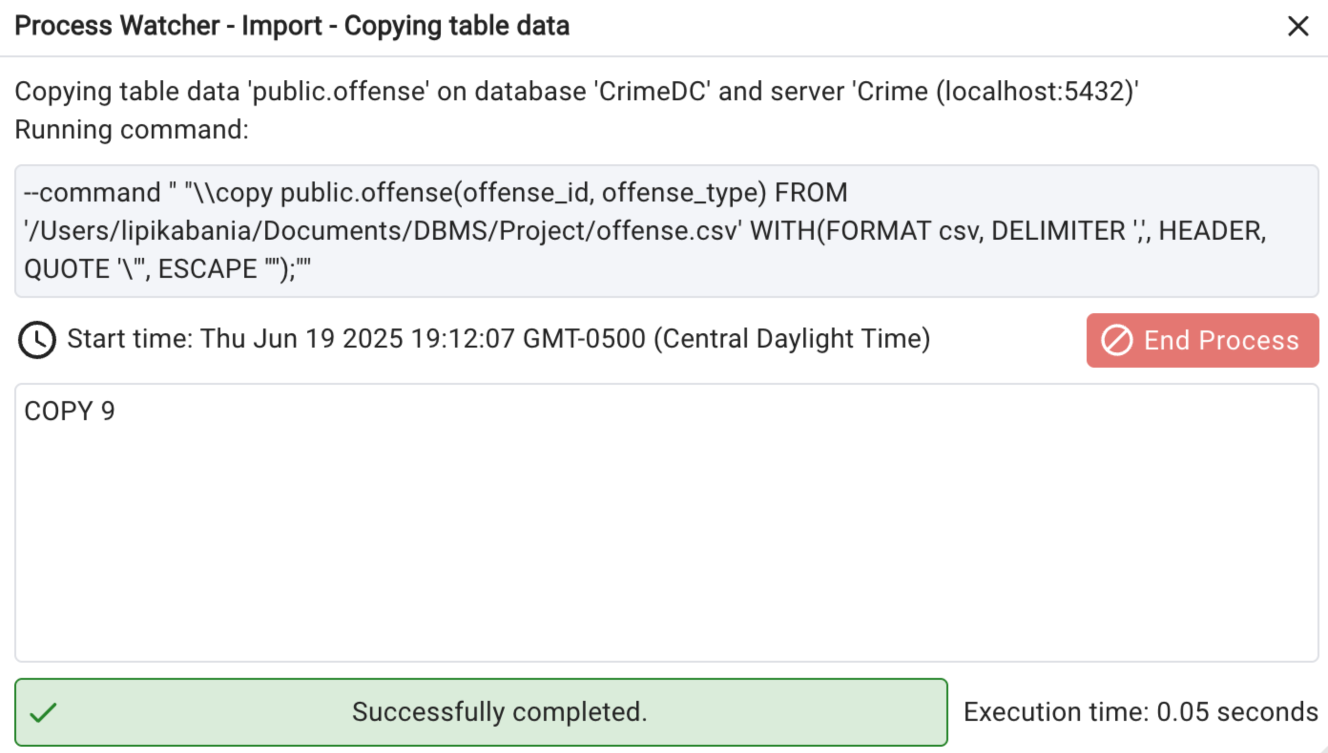
```
CCN,location_id,latitude,longitude
24423221,1,38.91949935,-77.00130027
24009631,2,38.91260599,-77.02345629
24009706,3,38.9344718,-76.99197561
24421835,4,38.89773014,-76.9984487
24422596,5,38.89814033,-76.9865096
24159248,6,38.87753041,-77.0040321
24162094,7,38.84653773,-76.98155338
24070627,4,38.90020336,-76.99730482
24007278,8,38.9072415,-77.04009106
24010507,9,38.90469905,-77.04168558
24010591,1,38.91482637,-77.00129984
24183279,10,38.92329508,-77.03530924
24184932,11,38.93402132,-76.99111756
```

3. Adding Data into Database

I imported data from the PostgreSQL GUI pgAdmin 4 to import csv files in bulk.

3.1. Table Offense

Number of tuples added: **9**



Process Watcher - Import - Copying table data

Copying table data 'public.offense' on database 'CrimeDC' and server 'Crime (localhost:5432)'
Running command:

```
--command " "\\copy public.offense(offense_id, offense_type) FROM '/Users/lipikabania/Documents/DBMS/Project/offense.csv' WITH(FORMAT csv, DELIMITER ',', HEADER, QUOTE '\"', ESCAPE '''');""
```

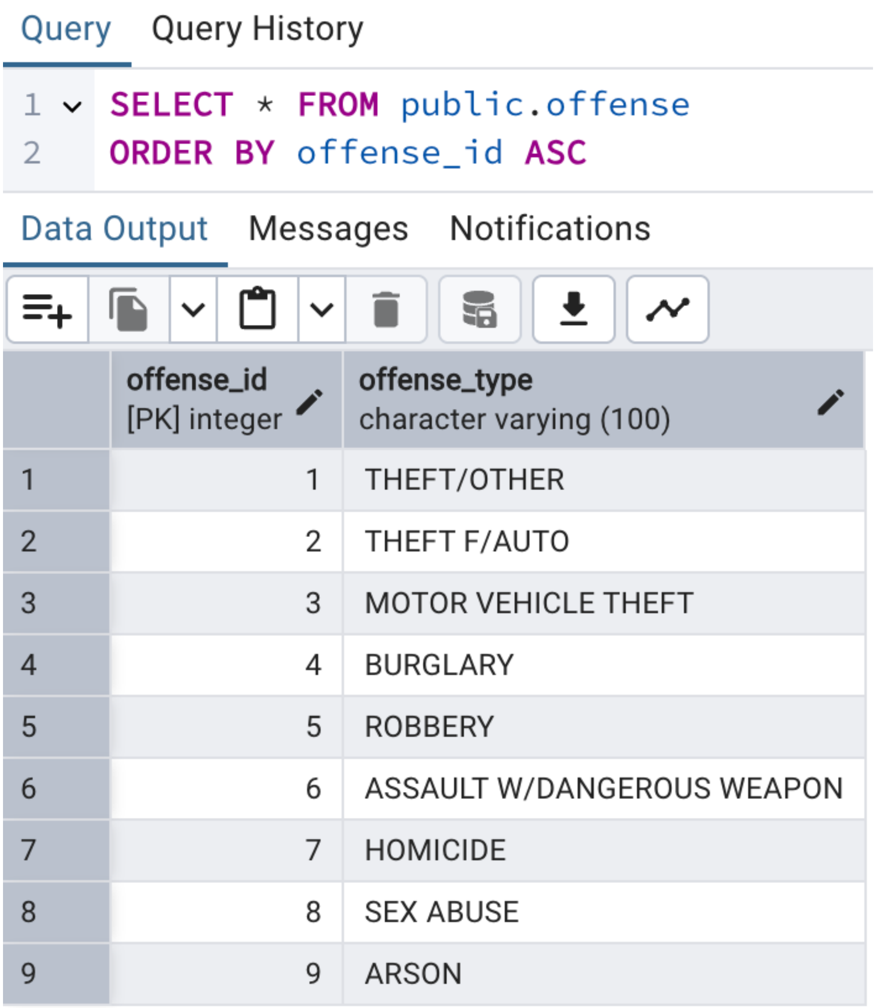
Start time: Thu Jun 19 2025 19:12:07 GMT-0500 (Central Daylight Time)

```
COPY 9
```

✓ Successfully completed.

Execution time: 0.05 seconds

Screenshot of Table Offense

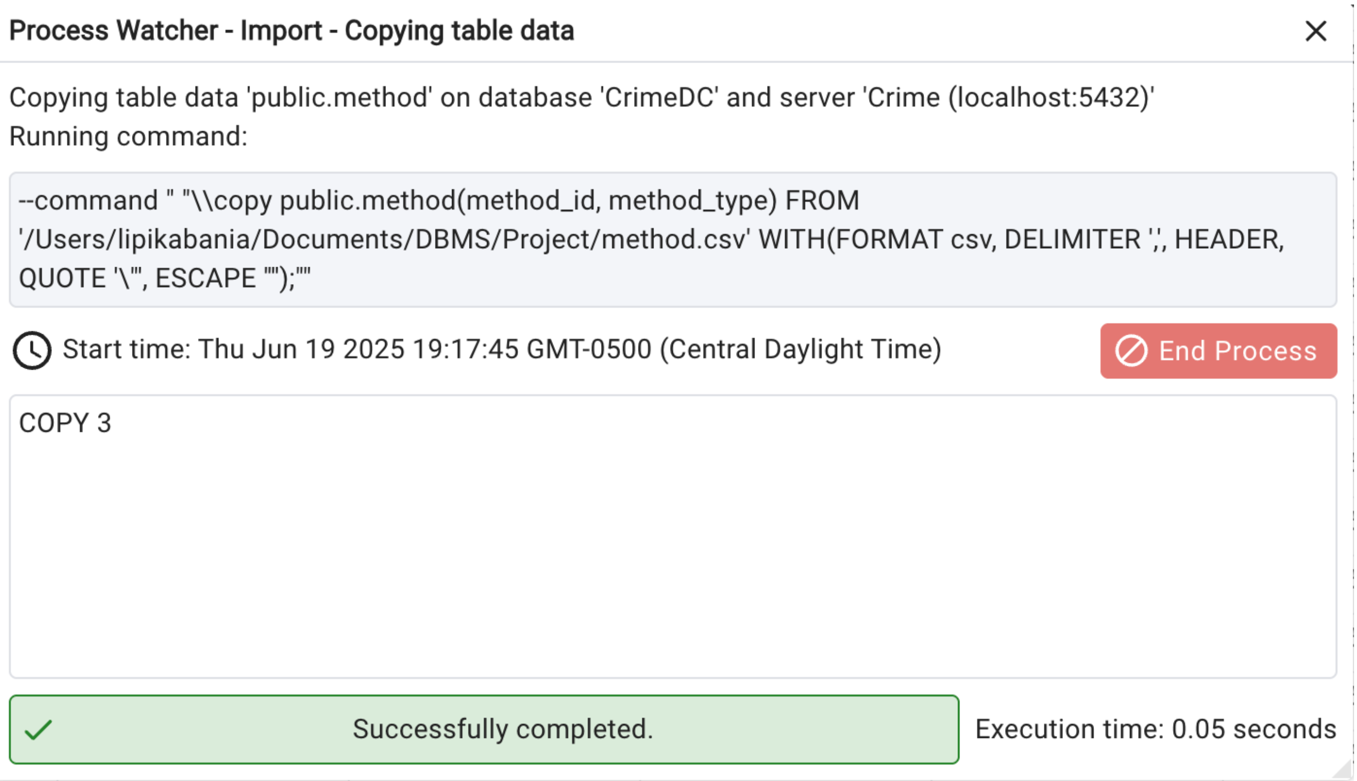


```
SELECT * FROM public.offense
ORDER BY offense_id ASC
```

	offense_id [PK] integer	offense_type character varying (100)
1	1	THEFT/OTHER
2	2	THEFT F/AUTO
3	3	MOTOR VEHICLE THEFT
4	4	BURGLARY
5	5	ROBBERY
6	6	ASSAULT W/DANGEROUS WEAPON
7	7	HOMICIDE
8	8	SEX ABUSE
9	9	ARSON

3.2.Table Method

Number of tuples: **3**



Copying table data 'public.method' on database 'CrimeDC' and server 'Crime (localhost:5432)'
Running command:

```
--command " "\\copy public.method(method_id, method_type) FROM
'/Users/lipikabania/Documents/DBMS/Project/method.csv' WITH(FORMAT csv, DELIMITER ',', HEADER,
QUOTE '\"', ESCAPE '''');""
```

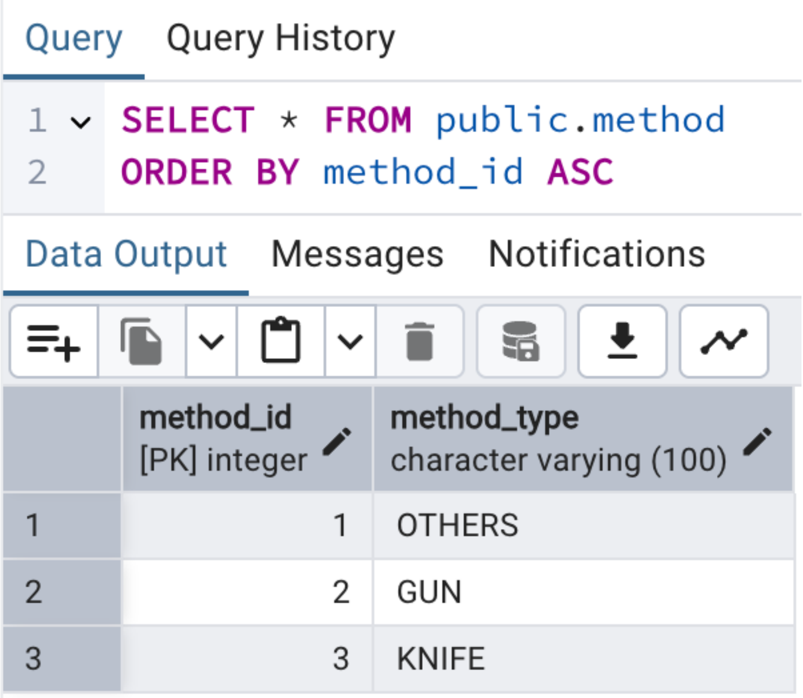
Start time: Thu Jun 19 2025 19:17:45 GMT-0500 (Central Daylight Time)

```
COPY 3
```

✓ Successfully completed.

Execution time: 0.05 seconds

Screenshot of Table Method

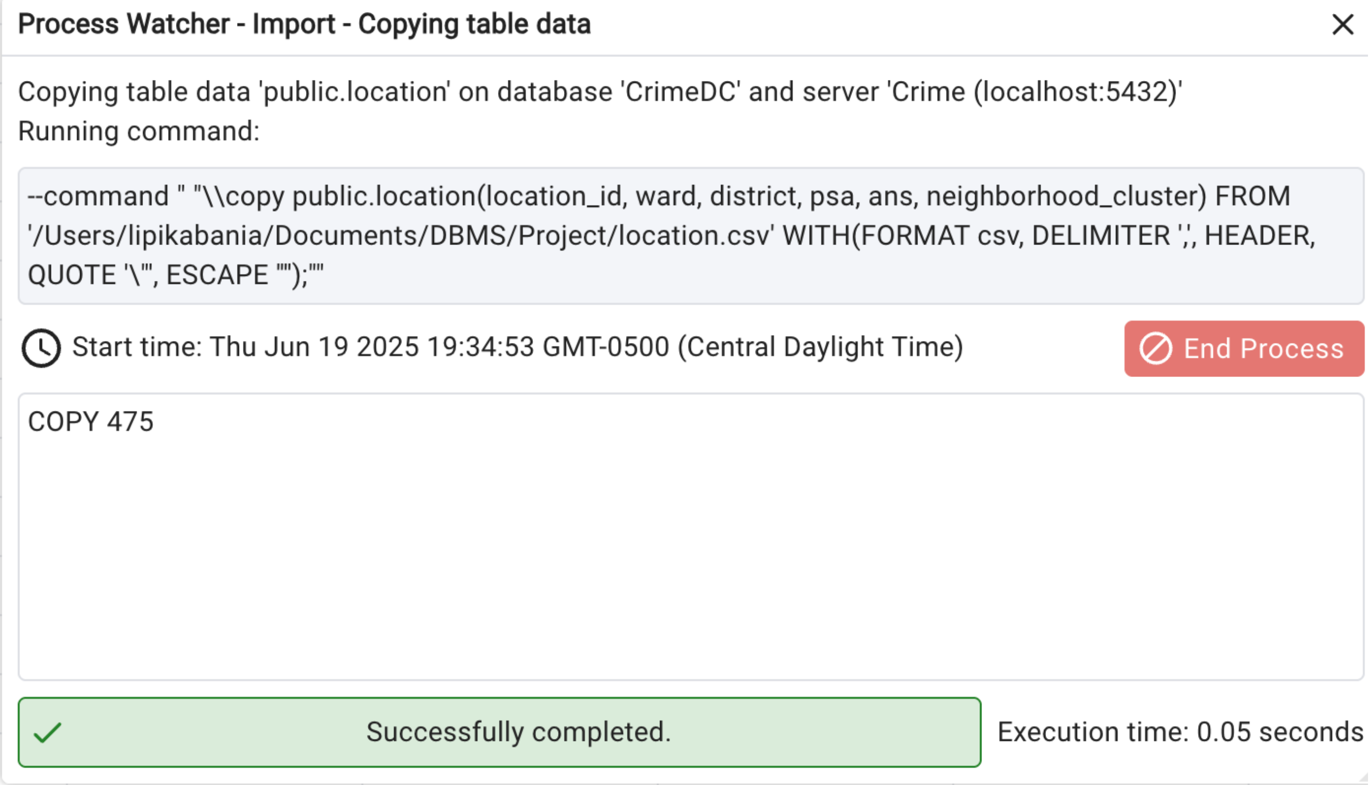


```
SELECT * FROM public.method
ORDER BY method_id ASC
```

	method_id [PK] integer	method_type character varying (100)
1	1	OTHERS
2	2	GUN
3	3	KNIFE

3.3. Table Location

Number of tuples: **475**



Copying table data 'public.location' on database 'CrimeDC' and server 'Crime (localhost:5432)'
Running command:

```
--command " "\\copy public.location(location_id, ward, district, psa, ans, neighborhood_cluster) FROM
'/Users/lipikabania/Documents/DBMS/Project/location.csv' WITH(FORMAT csv, DELIMITER ',', HEADER,
QUOTE '\"', ESCAPE '"');""
```

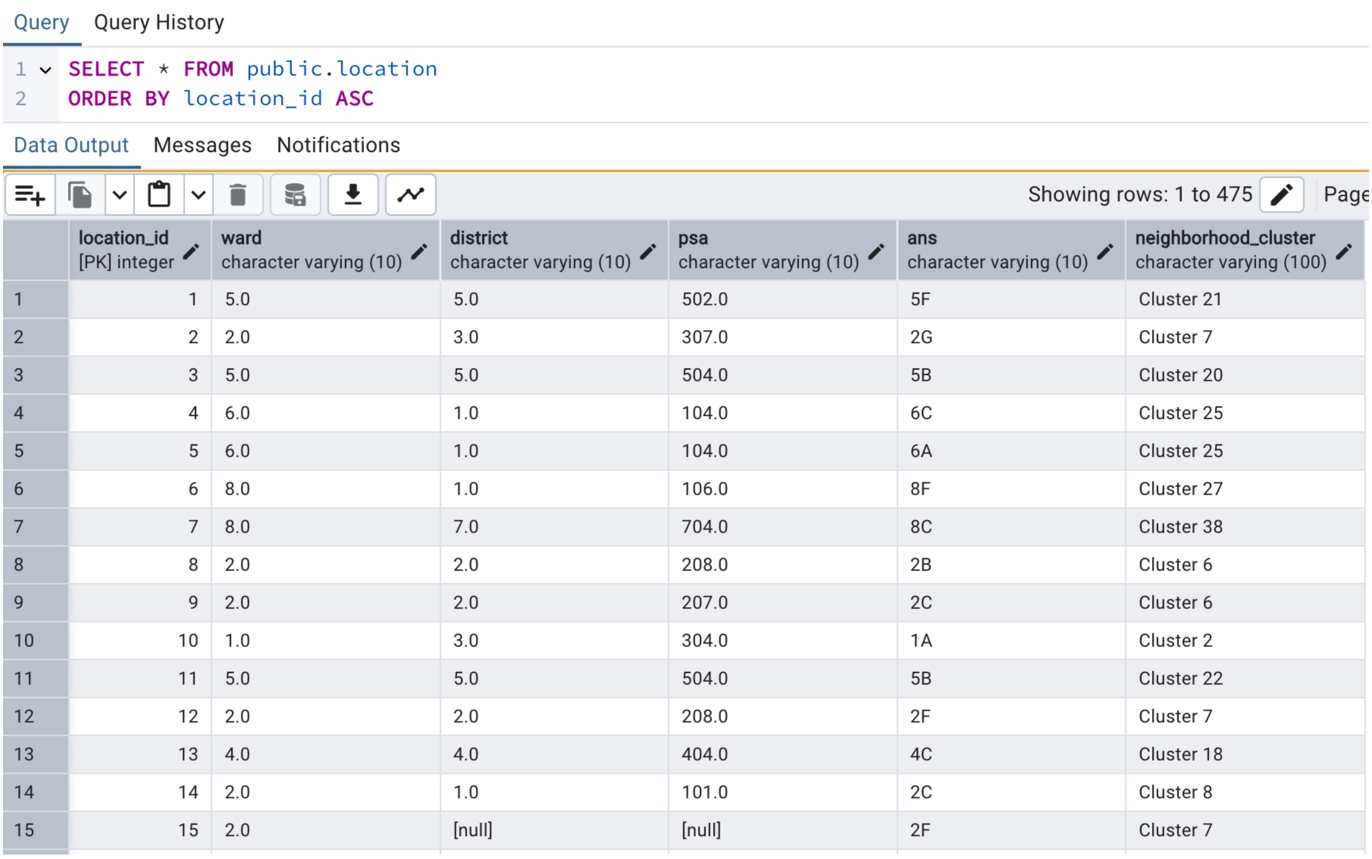
Start time: Thu Jun 19 2025 19:34:53 GMT-0500 (Central Daylight Time)

```
COPY 475
```

✓ Successfully completed.

Execution time: 0.05 seconds

Screenshot of Table Location



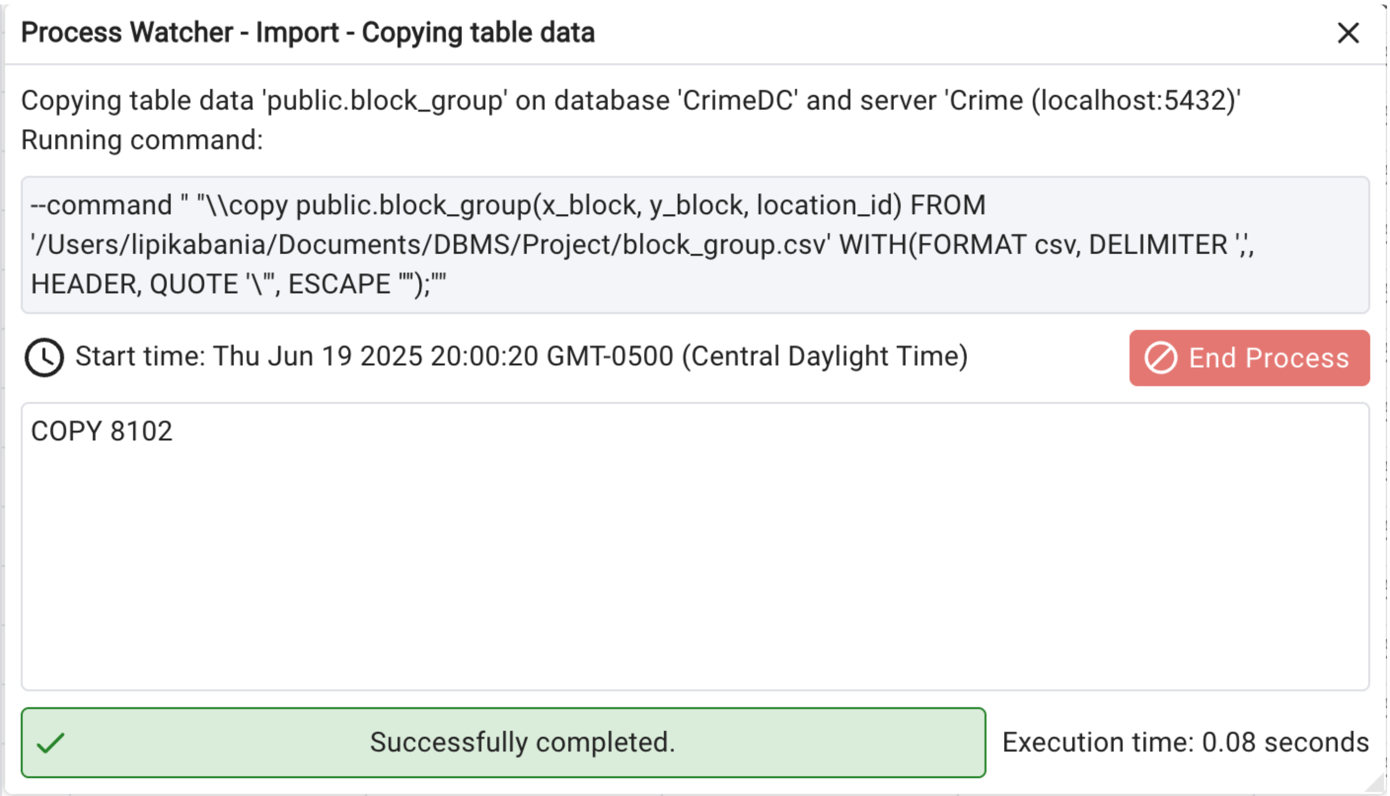
```sql
SELECT * FROM public.location
ORDER BY location_id ASC
```

Showing rows: 1 to 475

	location_id [PK] integer	ward character varying (10)	district character varying (10)	psa character varying (10)	ans character varying (10)	neighborhood_cluster character varying (100)
1	1	5.0	5.0	502.0	5F	Cluster 21
2	2	2.0	3.0	307.0	2G	Cluster 7
3	3	5.0	5.0	504.0	5B	Cluster 20
4	4	6.0	1.0	104.0	6C	Cluster 25
5	5	6.0	1.0	104.0	6A	Cluster 25
6	6	8.0	1.0	106.0	8F	Cluster 27
7	7	8.0	7.0	704.0	8C	Cluster 38
8	8	2.0	2.0	208.0	2B	Cluster 6
9	9	2.0	2.0	207.0	2C	Cluster 6
10	10	1.0	3.0	304.0	1A	Cluster 2
11	11	5.0	5.0	504.0	5B	Cluster 22
12	12	2.0	2.0	208.0	2F	Cluster 7
13	13	4.0	4.0	404.0	4C	Cluster 18
14	14	2.0	1.0	101.0	2C	Cluster 8
15	15	2.0	[null]	[null]	2F	Cluster 7

3.4.Table Block_Group

Number of tuples: **8102**



Copying table data 'public.block_group' on database 'CrimeDC' and server 'Crime (localhost:5432)'
Running command:

```
--command " "\\copy public.block_group(x_block, y_block, location_id) FROM
'/Users/lipikabania/Documents/DBMS/Project/block_group.csv' WITH(FORMAT csv, DELIMITER ',',
HEADER, QUOTE '\"', ESCAPE '''');""
```

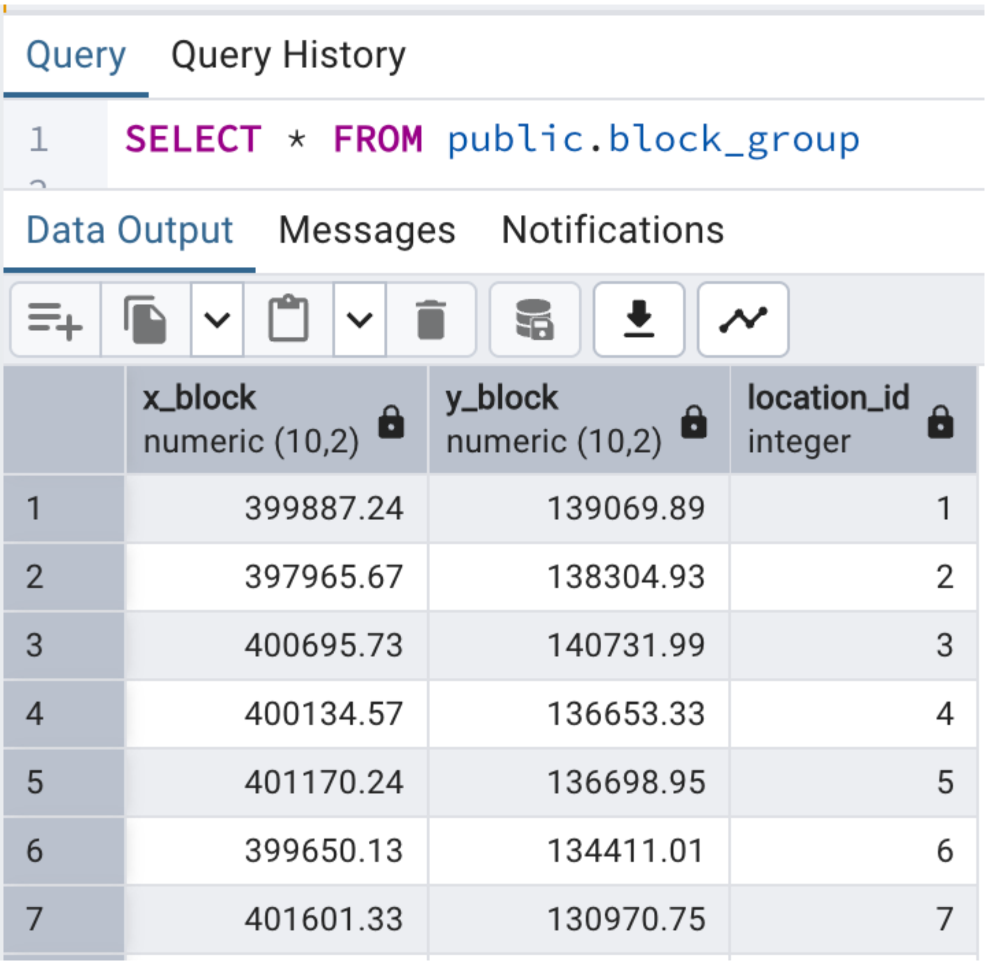
Start time: Thu Jun 19 2025 20:00:20 GMT-0500 (Central Daylight Time)

```
COPY 8102
```

✓ Successfully completed.

Execution time: 0.08 seconds

Screenshot of Block_Group

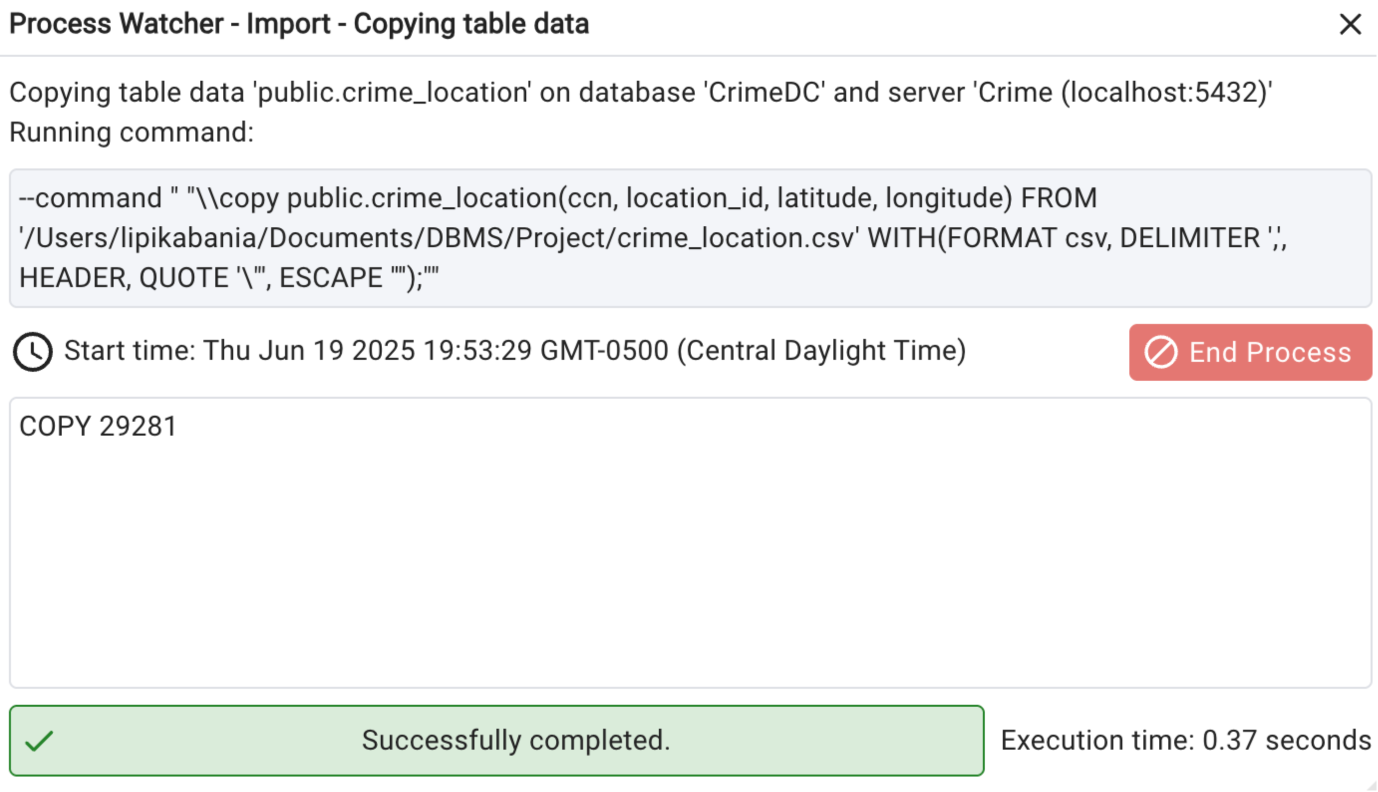


```
SELECT * FROM public.block_group
```

	x_block numeric (10,2)	y_block numeric (10,2)	location_id integer
1	399887.24	139069.89	1
2	397965.67	138304.93	2
3	400695.73	140731.99	3
4	400134.57	136653.33	4
5	401170.24	136698.95	5
6	399650.13	134411.01	6
7	401601.33	130970.75	7

3.5.Table Crime_Location

Number of tuples: **29281**



Process Watcher - Import - Copying table data

Copying table data 'public.crime_location' on database 'CrimeDC' and server 'Crime (localhost:5432)'
Running command:

```
--command " "\\copy public.crime_location(ccn, location_id, latitude, longitude) FROM
'/Users/lipikabania/Documents/DBMS/Project/crime_location.csv' WITH(FORMAT csv, DELIMITER ',',
HEADER, QUOTE '\"', ESCAPE '''');""
```

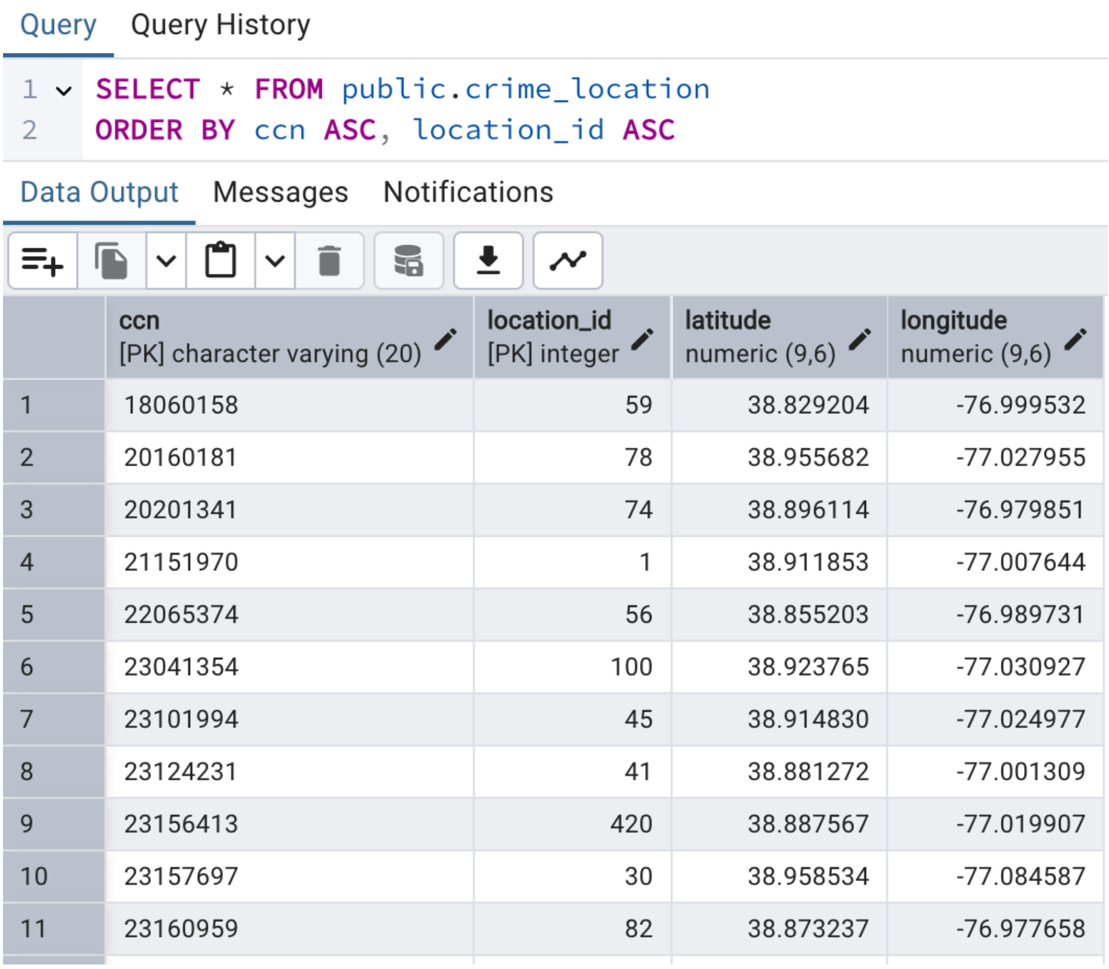
Start time: Thu Jun 19 2025 19:53:29 GMT-0500 (Central Daylight Time)

```
COPY 29281
```

Successfully completed.

Execution time: 0.37 seconds

Screenshot of Table Crime_Location

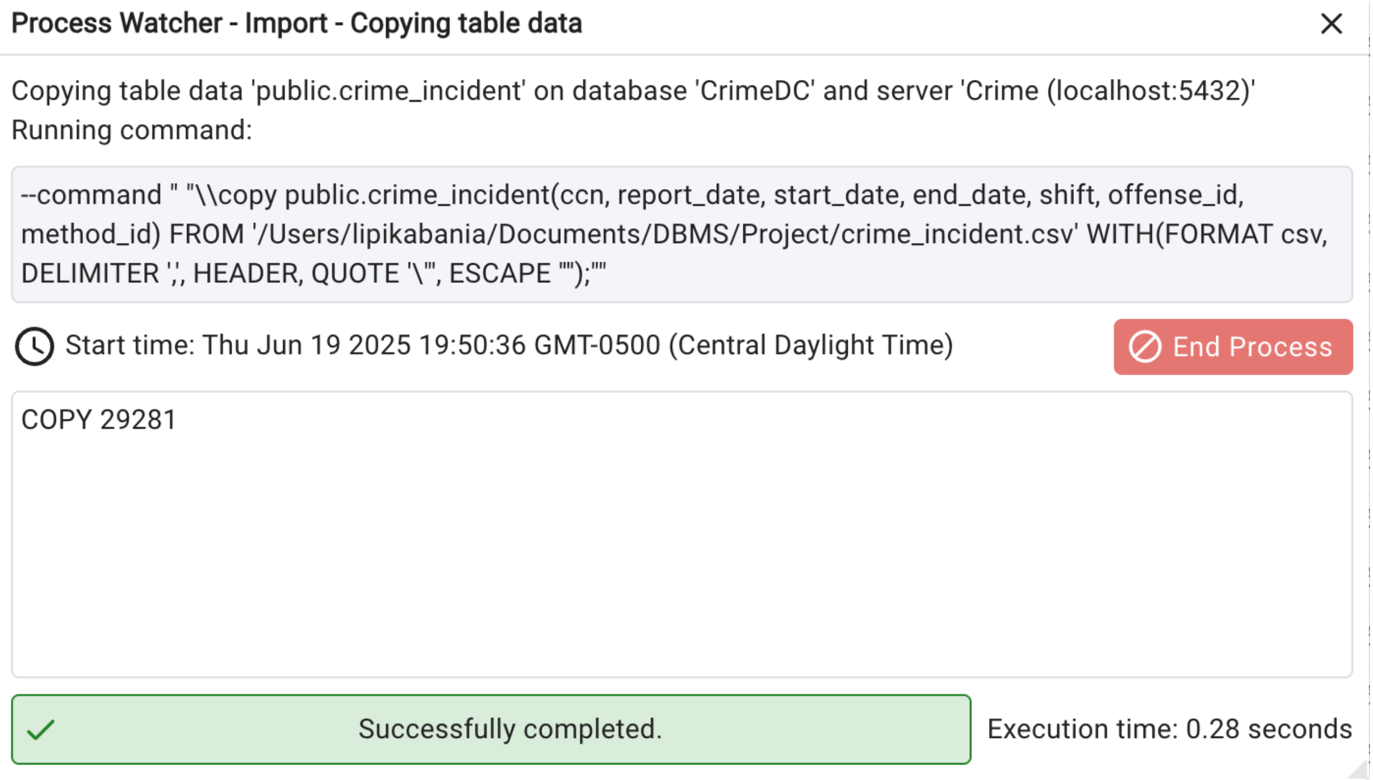


```sql
SELECT * FROM public.crime_location
ORDER BY ccn ASC, location_id ASC
```

	ccn [PK] character varying (20)	location_id [PK] integer	latitude numeric (9,6)	longitude numeric (9,6)
1	18060158	59	38.829204	-76.999532
2	20160181	78	38.955682	-77.027955
3	20201341	74	38.896114	-76.979851
4	21151970	1	38.911853	-77.007644
5	22065374	56	38.855203	-76.989731
6	23041354	100	38.923765	-77.030927
7	23101994	45	38.914830	-77.024977
8	23124231	41	38.881272	-77.001309
9	23156413	420	38.887567	-77.019907
10	23157697	30	38.958534	-77.084587
11	23160959	82	38.873237	-76.977658

3.6.Table Crime_Incident

Number of tuples: **29281**



Process Watcher - Import - Copying table data

Copying table data 'public.crime_incident' on database 'CrimeDC' and server 'Crime (localhost:5432)'
Running command:

```
--command " "\\copy public.crime_incident(ccn, report_date, start_date, end_date, shift, offense_id, method_id) FROM '/Users/lipikabania/Documents/DBMS/Project/crime_incident.csv' WITH(FORMAT csv, DELIMITER ',', HEADER, QUOTE '\"', ESCAPE '''');""
```

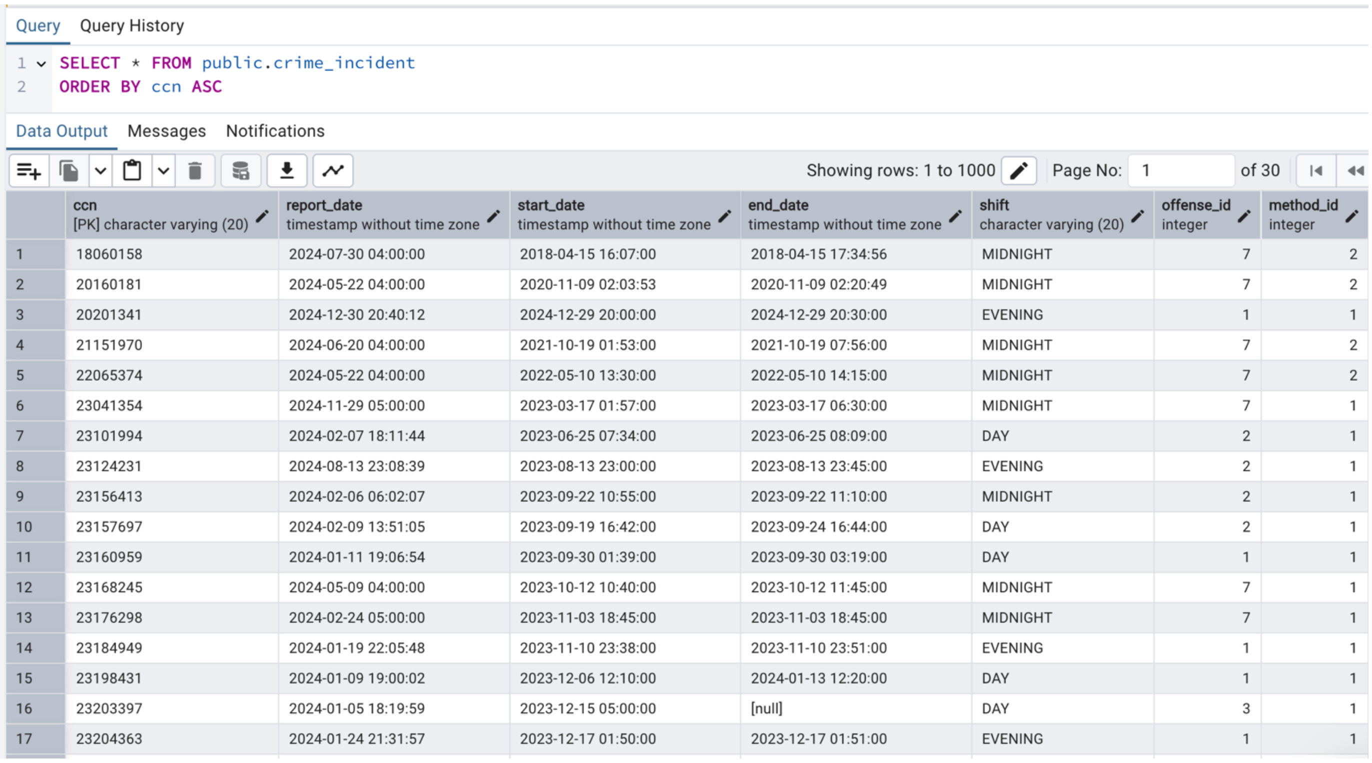
Start time: Thu Jun 19 2025 19:50:36 GMT-0500 (Central Daylight Time)

```
COPY 29281
```

Successfully completed.

Execution time: 0.28 seconds

Screenshot of Crime_Incident



```
SELECT * FROM public.crime_incident
ORDER BY ccn ASC
```

Showing rows: 1 to 1000 Page No: 1 of 30

	ccn [PK] character varying (20)	report_date timestamp without time zone	start_date timestamp without time zone	end_date timestamp without time zone	shift character varying (20)	offense_id integer	method_id integer
1	18060158	2024-07-30 04:00:00	2018-04-15 16:07:00	2018-04-15 17:34:56	MIDNIGHT	7	2
2	20160181	2024-05-22 04:00:00	2020-11-09 02:03:53	2020-11-09 02:20:49	MIDNIGHT	7	2
3	20201341	2024-12-30 20:40:12	2024-12-29 20:00:00	2024-12-29 20:30:00	EVENING	1	1
4	21151970	2024-06-20 04:00:00	2021-10-19 01:53:00	2021-10-19 07:56:00	MIDNIGHT	7	2
5	22065374	2024-05-22 04:00:00	2022-05-10 13:30:00	2022-05-10 14:15:00	MIDNIGHT	7	2
6	23041354	2024-11-29 05:00:00	2023-03-17 01:57:00	2023-03-17 06:30:00	MIDNIGHT	7	1
7	23101994	2024-02-07 18:11:44	2023-06-25 07:34:00	2023-06-25 08:09:00	DAY	2	1
8	23124231	2024-08-13 23:08:39	2023-08-13 23:00:00	2023-08-13 23:45:00	EVENING	2	1
9	23156413	2024-02-06 06:02:07	2023-09-22 10:55:00	2023-09-22 11:10:00	MIDNIGHT	2	1
10	23157697	2024-02-09 13:51:05	2023-09-19 16:42:00	2023-09-24 16:44:00	DAY	2	1
11	23160959	2024-01-11 19:06:54	2023-09-30 01:39:00	2023-09-30 03:19:00	DAY	1	1
12	23168245	2024-05-09 04:00:00	2023-10-12 10:40:00	2023-10-12 11:45:00	MIDNIGHT	7	1
13	23176298	2024-02-24 05:00:00	2023-11-03 18:45:00	2023-11-03 18:45:00	MIDNIGHT	7	1
14	23184949	2024-01-19 22:05:48	2023-11-10 23:38:00	2023-11-10 23:51:00	EVENING	1	1
15	23198431	2024-01-09 19:00:02	2023-12-06 12:10:00	2024-01-13 12:20:00	DAY	1	1
16	23203397	2024-01-05 18:19:59	2023-12-15 05:00:00	[null]	DAY	3	1
17	23204363	2024-01-24 21:31:57	2023-12-17 01:50:00	2023-12-17 01:51:00	EVENING	1	1