Project Part 2 Submission

Eesun Moon (em3907), Chaeeun Ryu (cr3413)

PostgreSQL account name: em3907

Three "interesting" SQL queries over our database

(1) Peak Times for Crime

SQL Query:

```
SELECT EXTRACT(HOUR FROM
CMPLNT_FR_TM) AS crime_hour,
COUNT(CRIME_SCENE_PID) AS
num_crimes FROM Crime_Scene
GROUP BY crime_hour ORDER BY
num_crimes DESC;
```

Output:

```
 crime_hour | num_crimes
------------+------------
         19 |        866
         16 |        726
         12 |        425
         17 |        236
         13 |        177
          4 |         84
          1 |         66
(7 rows)
```

About: `CMPLNT_FR_TM` indicates exact time of occurrence for the reported event. So, the code calculates the peak times for crime, and from the code output above it was interesting to figure out that at 7:00 pm, the number of crimes was the highest.

(2) Crimes by Incident Type with Average Dispatch Time and Crime Locations

SQL Query:

```
SELECT i.TYP_DESC,
    COUNT(i.INCIDENT_PID) AS total_incidents,
    AVG(EXTRACT(EPOCH FROM (dd.ARRIVD_TS - dd.DISP_TS))) / 60 AS
```

```
avg_dispatch_time_minutes,
    COUNT(DISTINCT cs.LOC_OF_OCCUR_DESC) AS distinct_locations
FROM Incident i
JOIN Crime_Scene cs ON i.INCIDENT_PID = cs.CRIME_SCENE_PID
JOIN Dispatch_Duration dd ON cs.CRIME_SCENE_PID =
dd.DISPATCH_DURATION_PID
WHERE cs.CMPLNT_FR_DT BETWEEN '2023-01-01' AND '2023-12-31' --
Incidents in 2023
GROUP BY i.TYP_DESC
ORDER BY total_incidents DESC;
```

Output:

typ_desc	total_incidents	avg_dispatch_time_minutes	distinct_locations
TRAIN ORDER MAINTENANCE SWEEP (TOMS)	238	0.0000000000000000000	1
STATION INSPECTION BY TRANSIT BUREAU PERSONNEL	191	0.0000000000000000000	1
TRAIN RUN/MOBILE ORDER MAINTENANCE SWEEP	123	0.0000000000000000000	1
TRANSIT PATROL/INSPECTION BY NON-TRANSIT BUREAU PERSONNEL	108	0.09830246913580246833	1
DISORDERLY: PERSON/INSIDE	105	34.2936507936507937	2
SEE COMPLAINANT: OTHER/INSIDE	68	4.5534313725490196	2
VERIFY AMB NEEDED: TRANSIT	64	32.6890625000000000	1
VISIBILITY PATROL: DIRECTED	55	2.2687878787878788	2
INVESTIGATE/POSSIBLE CRIME: SERIOUS OTHER/TRANSIT	26	11.8865384615384615	1
OTHER CRIMES (IN PROGRESS): HARASSMENT/INSIDE	24	12.1375000000000000	2
COMMUNITY TIME	23	0.03188405797101449333	2
INVESTIGATE/POSSIBLE CRIME: SUSP PERSON/OUTSIDE (PROWLER)	22	0.55984848484848484833	2
AMBULANCE CASE: SERIOUS/TRANSIT	21	8.5206349206349206	1
INVESTIGATE/POSSIBLE CRIME: CALLS FOR HELP/INSIDE	17	79.9372549019607843	2
ALARMS: COMMERCIAL/BURGLARY	15	38.8611111111111111	2
LARCENY (PAST): OTHER/INSIDE	14	45.9273809523809524	1
OTHER CRIMES (IN PROGRESS): OTHER/TRANSIT	14	17.6226190476190476	1
DISORDERLY: GROUP/INSIDE	14	27.4500000000000000	1
AMBULANCE CASE: EDP/INSIDE	14	7.4476190476190476	2
DISORDERLY: PERSON/TRANSIT	13	16.7692307692307692	1
INVESTIGATE/POSSIBLE CRIME: SUSP VEHICLE/OUTSIDE	13	5.4884615384615385	2
LARCENY (IN PROGRESS): OTHER/INSIDE	12	7.5888888888888889	1
AMBULANCE CASE: EDP/TRANSIT	12	8.3541666666666667	1
INVESTIGATE/POSSIBLE CRIME: SERIOUS/OTHER	11	43.7833333333333333	2
AMBULANCE CASE: UNCONSCIOUS/TRANSIT	10	21.1650000000000000	1
INVESTIGATE/POSSIBLE CRIME: CALLS FOR HELP/OUTSIDE	10	77.1416666666666667	2
OTHER CRIMES (IN PROGRESS): HARASSMENT/TRANSIT	10	13.1683333333333333	1
DISPUTE: INSIDE	9	25.6037037037037037	2
INVESTIGATE/POSSIBLE CRIME: SUSP PERSON/TRANSIT	8	5.6645833333333333	1
OTHER CRIMES (IN PROGRESS): HARASSMENT/OUTSIDE	7	12.7428571428571429	2
SEE COMPLAINANT: OTHER/OUTSIDE	7	0.0000000000000000000	2
ASSAULT (IN PROGRESS): OTHER/TRANSIT	6	4.9250000000000000	1
DISORDERLY: GROUP/TRANSIT	5	22.6533333333333333	1
AMBULANCE CASE: INJURY/TRANSIT	5	15.9866666666666667	1
DISPUTE: OUTSIDE	5	30.1966666666666667	2
ASSAULT (IN PROGRESS): OTHER/INSIDE	4	9.8166666666666667	2

About: The above code returns the crimes by incident type with its average dispatch time and crime location in 2023, ordered by the total number of incidents in descending order. From the output of code attached above, it was interesting to figure out that it takes very long dispatch time on average to deal with calls for help compared to the other cases (Calls for help from inside takes on average 79 minutes and calls for help from outside on average 77 minutes).

(3) Incident Types with the Longest Closing Times

SQL Query:

```
SELECT i.TYP_DESC, AVG(EXTRACT(EPOCH FROM (cs.CMPLNT_TO_TM -
cs.CMPLNT_FR_TM))) / 3600 AS avg_closing_time_hours
FROM Incident i
JOIN Crime_Scene cs ON i.INCIDENT_PID = cs.CRIME_SCENE_PID
GROUP BY i.TYP_DESC
ORDER BY avg_closing_time_hours DESC
LIMIT 10;
```

Output:

```
                typ_desc                 | avg_closing_time_hours
-----------------------------------------+------------------------
 FIRE: EXPLOSION                         |     3.0000000000000000
 AMBULANCE CASE: INJURY/INSIDE           |     2.4166666666666667
 FIRE: FIRE MULTI DWELLING               |     2.4166666666666667
 ROBBERY (IN PROGRESS): OTHER/INSIDE     |     2.4166666666666667
 BURGLARY (PAST): COMMERCIAL             |     2.0333333333333333
 VEHICLE ACCIDENT: PERSON PINNED         |     2.0333333333333333
 INVESTIGATE/POSSIBLE CRIME: KNIFE/OUTSIDE |     2.0333333333333333
 BURGLARY (IN PROGRESS): COMMERCIAL      |     2.0333333333333333
 BURGLARY (PAST): OTHER                  |     2.0333333333333333
 AMBULANCE CASE: SERIOUS/INSIDE          |     1.9666666666666667
(10 rows)
```

About: The above code outputs incident types with the longest time spent on closing the case, with the output ordered in descending order with respect to the time spent for closing. It was interesting to figure out that it takes the longest time to close cases regarding fire.