The Building and Analysis of the Atlanta Police Department's Database Management System

XIN XIONG

Introduction

Atlanta traces its earliest years to the 1830s when it was originally called Terminus due to the fact that it was the southern end of a rail line connecting the North to the South. It would not be for another ten years that the city was officially renamed Atlanta by the state legislature and another five years until the police department was established. Crime was minimal and only a night policeman roamed the streets. Of course, the population was small, less than 10,000, and worries of crime were at a minimum.

Moving 150 years later, things have changed. The World Population Review states that more than 5.6 million people live in the metropolitan area today, making Atlanta the 9th largest city in the United States.[1]

Under the leadership of Atlanta Mayor Kasim Reed is Atlanta Police Chief, George Turner. A 35 year veteran of the department, Chief Turner has been the Police Chief since 2010 and is responsible for the $164 million department budget and its 2,200 personnel.[2]

Our group took on the project of exploring the Atlanta Police Department's (APD) open source databases with two primary intentions. First, we wanted to learn more about where crimes occur throughout Atlanta, who is committing the crimes, what times of the day, and many other questions along these lines. Secondly, and maybe more importantly, we wanted to build a database system on the open data provided by the APD and simulate solutions for victims' or officers' or analysts' inquiries on these crime cases.

Requirements Gathering and Analysis

The APD's data is open to the public, making sourcing of the data easily achievable. Located on the Department's website, weekly crime reports, annual reports, and an historical crime data file going back to the year 2009 are all available at www.atlantapd.org/crimedatadownloads.aspx. The APD maintains over 200 databases for its use, many of them duplicative in nature. In fact, it appears that relations have been created over time and not consolidated where appropriate.

Conceptual (ER Model) and Logical Design

Creating a logical model begins with an understanding of the business requirements that need to be modeled. To get a grasp of these requirements and how they would be linked together, we set out to identify the entities necessary to accomplish our task, and we created a relation around each entity.

We were able to create our ER model using MySQL. After importing our relations into MySQL, we were able to drag our relations from the left hand pane into the mapping diagram on the right.

[1] World Population Review, http://worldpopulationreview.com/us-cities/atlanta-population/, accessed November 18, 2016.
[2] Atlanta Police Department, http://www.atlantapd.org/chiefofpolice-georgeturner.aspx, accessed November 18, 2016.

Because our relations already had primary and foreign keys included within them, MySQL was easily able to create the logical design by connecting the relations together.

For our purposes, we chose eleven relations from the aggregate list. These relations include the following (in alphabetical order):

1. Address[3]
2. Crime Data
3. Crime Location
4. Crime Status
5. Offense
6. Offense Type
7. Police Officer
8. Police Station
9. Police Station Mapping
10. Shift
11. Victim

Although some of these are intuitive in nature, in the next section we will further define what each of these relations contains. It is worth noting here, however, that by shrinking the number of relations from all the potential relations in existence to these eleven primary relations does not eliminate or reduce our analytical efforts. In addition, the data provided by the APD includes more than 26,000 tuples of data. We filtered that down to approximately 200 and created some new data to prove the effectiveness of our effort.

Our ERD, Appendix A, shows the myriad relationships connecting the relations together. Most relations are tied, directly or indirectly, to the offense that was committed. The primary keys that are used in each table include the following: AddressID for the Address table, CriminalID for the Criminal table, CrimeLocID for the Crime Location table, CrimeStatusID for the Crime Status table, MappingID for PSO Mapping, OffenseID for the Office table, OffenseTypeID for the Offense Type table, PoliceOfficerID for the Police Officer table, PoliceStationID for the Police Station table, ShiftID for the Shift table, and VictimID for the Victim table. In the sake of space, we will not list all of the attributes in each relation here, but they can found in their entirety in Appendix A, as can the relationships that bind the various relations together.

After completing our ER model, we set out to develop a context diagram and a Level 0 data flow diagram to better depict the process by which the information would flow through the DBMS.

The context diagram, found in Appendix B, is a high-level view of the system and how it interacts with the entities feeding into it. In other words, a context diagram is a data flow diagram that shows the scope of an organizational system. In our case, we call this system the "system process." Our system process has four external entities that interact with the system: Victim, Suspect, Case Department, and Officer. With regard to the victim, it communicates with the system through a

[3] Address includes the home address of the victim as well as the police officer who responded to the call and that officer's police station. This differs from the Crime Location table that includes data such as the X and Y coordinates and neighborhoods of the crimes.

victim inquiry. In return, the victim obtains feedback on the final result of the inquiry. The suspect communicates with the system process by receiving information from the system and by providing feedback into the system. Third, the case department interacts with the system process by updating a case that was given to the department to investigate. Finally, the officer interacts with the system process by receiving cases from the system and providing feedback into the system process. Taken together, this system process is the process by which victims and suspects of offenses are tracked and handled by the APD and its officials.

We present our Level 0 diagram in Appendix C. A Level 0 diagram shows all the processes that comprise the overall system. It shows how the information moves from and to each process. We kept three rules in mind when creating our Level 0 diagram. Namely, that processes can only be connected to other processes, that a process can be connected to an external entity, and that a process can be connected to a data store, which we had two. The entire process begins when a victim files a first information report, or FIR. Once the APD receives the FIR, it opens a case for the Case Department to investigate. When an investigation begins, the case is stored in an officer's information file and the case is submitted to the officer to work. The officer executes the case by requesting information from the suspect through an interrogation or some other means and then filing/storing the information gleaned in a database for future keeping. A resulting report is then created which is ultimately sent back to the victim for case closure.

Now that the tables have all been conceptualized and the data flow process has been created, it is time to turn our attention to how these relations can be normalized so that physical tables can be created, data entry can be completed, and queries can be conducted.

Normalization

With regard to normalization, we initially had the raw data and denormalized it into a single sheet. We then set out to find all of the repetitive data and removed it. We also removed all of the dependencies and were left with the table structure that contained a total of eleven tables as stated in previous sections of this report.

One significant step we took that is worth mentioning by way of an example was dealing with the ternary relationship between police station, police officer, and offense. Our way of handling this was to create a relation, "PSO Mapping," which contains the keys off all the other three tables that we used as foreign keys. Another significant example we implemented was how we dealt with both police officers and police stations. We felt it best to bring these two together in a single relation that further normalized it and removed the dependencies that were previously there. Finally, we initially had offense and offense location details in a single relation, so we normalized it by creating two tables, one containing the offense details and the other containing the offense location details.

Taken together, we normalized our tables into the third normal form, finding this to be more than sufficient for our purposes.

Creating Physical Tables

When we set out to create our physical tables, we first created a model in the MySQL Workbench where we created all of the logical tables that were required for this project. This, as stated earlier, generated the ER diagram that we have used throughout this entire project.

After fixing any minor issues that we saw, we then moved this logical schema to a physical schema. This was completed by loaded the data in Excel and then using a MySQL plug-in in Excel which imports the data directly from Excel to our database tables. Two examples physical tables that we created can be found in Appendix D.

Sample Data Entry

In order to ensure our physical tables worked properly, we took some sample data -- around 200 tuples -- for the corresponding fields from Atlanta crimes sites. Based on that data, we generated our own sample dummy data in order to run some queries and get the desired output for which we were looking. Two examples of this can be found in Appendix D.

Queries

To complete our efforts, we were excited to answer some important questions about crime in Atlanta. With everything prepared, we set out to answer ten primary questions that we found perplexing at the outset of our efforts. Our MySQL script and output can be found in Appendix D, but we place here our questions and their respective answers:

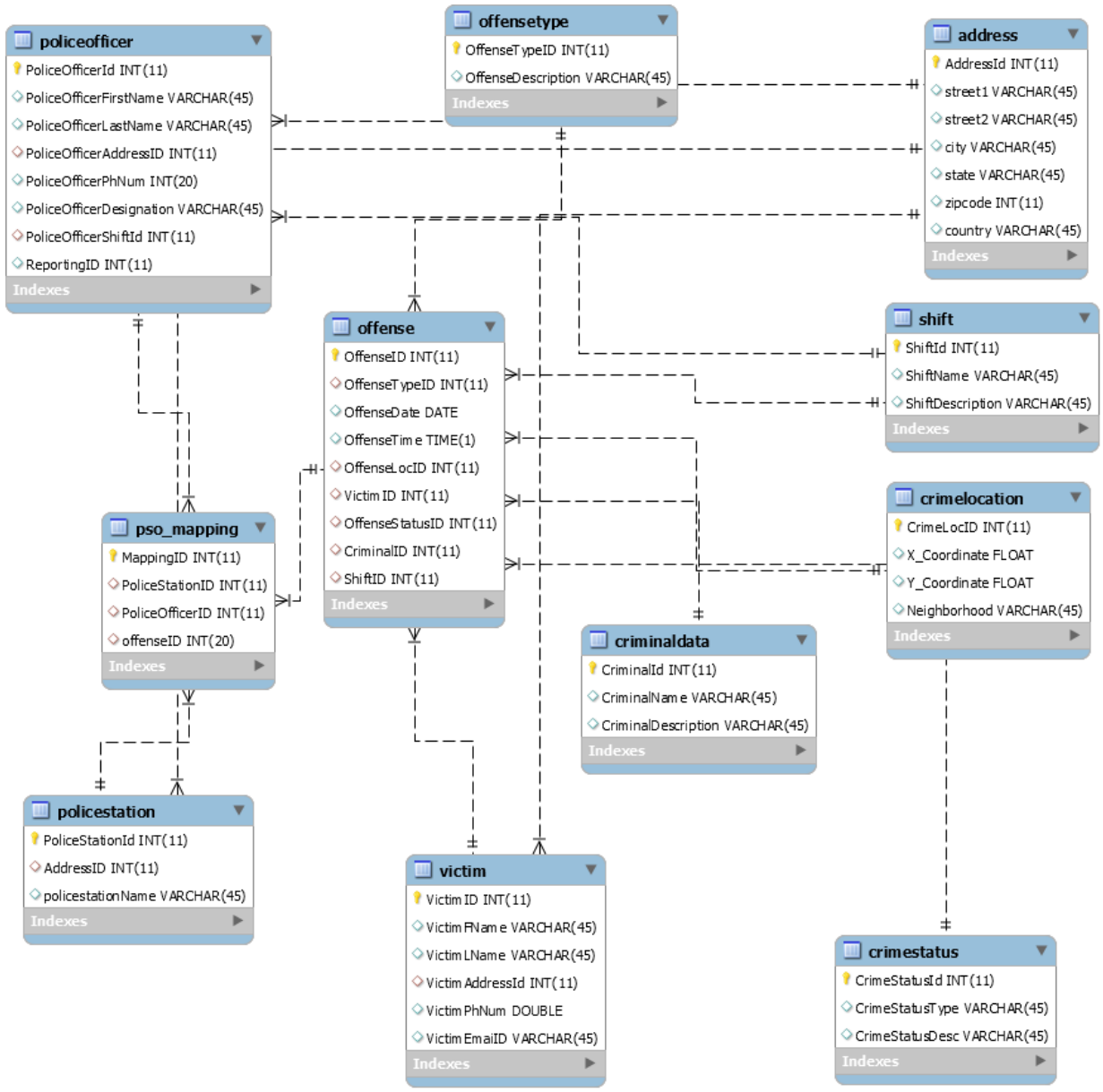
1) What are the total number of active crimes in Atlanta? Ans. 175;
2) What are the addresses of the victims whose crime status is not complete but whose file is closed? Ans. Eight addresses were produced, including their street number and name as well as the zip code inside Atlanta;
3) What are the addresses of the victims whose crime status is active? Ans. Our query produces 175 results in the city's limits;
4) What are the addresses of the victims whose crime status is both completed and whose file is closed? Ans. We found 13 results meeting this requirement;
5) What are the top three neighborhoods in Atlanta for crime? Ans. Three neighborhoods are produces: Downtown with 20 such instances, Midtown with 13, and along English Avenue with seven;
6) What are the number of crimes reported per police station in Atlanta? Ans. There are five police departments in Atlanta, and APD1 reported 48 crimes, APD2 reported 47 crimes, APD3 reported 42 crimes, APD4 reported 35 crimes, and APD5 reported 24 crimes.
7) Who are the top three police officers dealing with the highest number of offenses? Ans. Officer Jessica Johnson has recently handled five crime offenses. The same can be said of Officers William Smith and Thomas Johnson;
8) We were able to search for all offenses conducted after the date April 15, 2016;
9) What is the number of offenses that occur during the daytime shift (as opposed to the evening or night shifts)? Ans. 123; and finally,

10) Can we determine the neighborhoods where at least two crimes took place, ordered by LocationID? Ans. Yes. Our output included 38 total such cases.

Conclusion

This effort was immensely valuable in better helping us understand crime in Atlanta, a relatively young city whose population continues to balloon in size. Using these databases, we believe that crime can better be understood and perhaps alleviated to some degree. By understanding where crimes occur and in which police department precinct or zone, resources can better be allocated to assist with fighting crime. Additionally, by knowing which files are closed but whose crime status is not yet completed, the APD can double back on its efforts to ensure no case remains open longer than is completely necessary.

Appendix A - ER Diagram

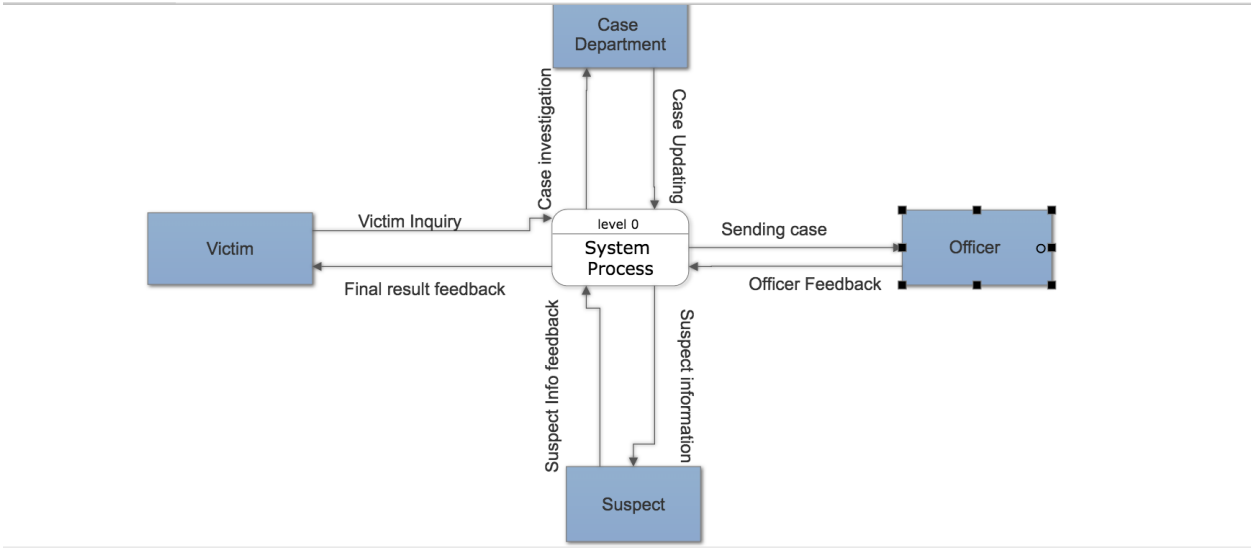


Assumptions:

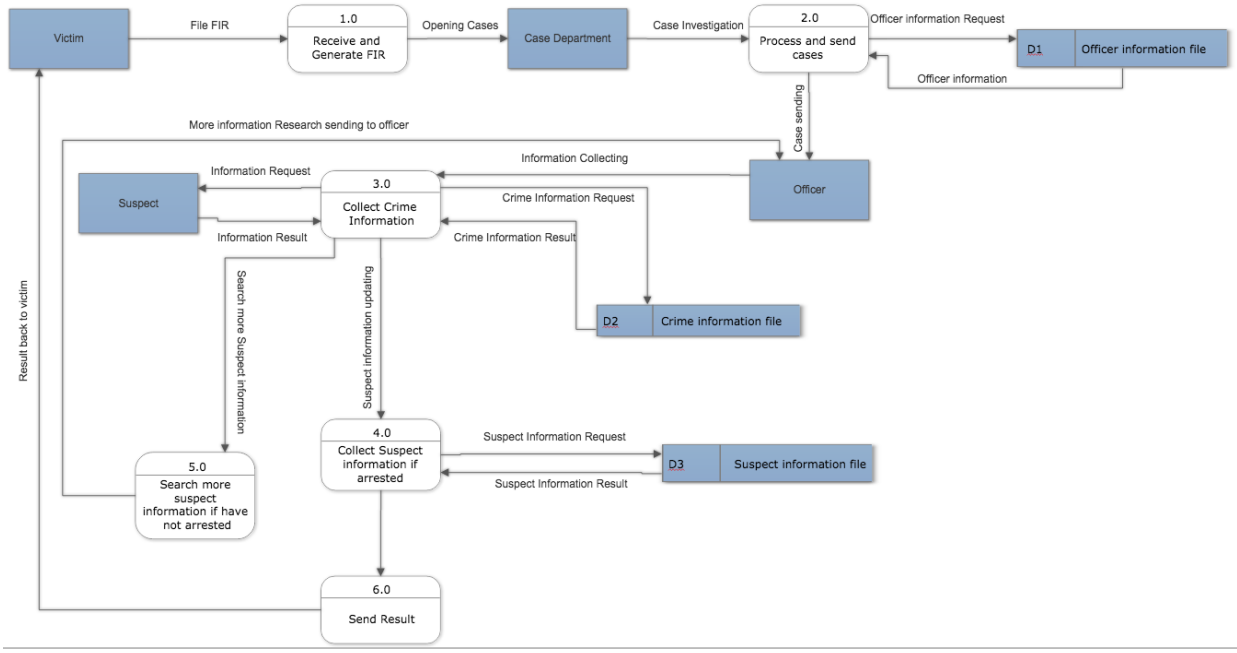
1. Each victim has been assigned at least one offense ID, and when the victim cannot be identified, the victim is given a default name such as Unknown Victim 1, Unknown Victim 2, etc.;
2. Each criminal has been assigned at least one offense ID, and when the criminal cannot be identified, the criminal is given a default name such as Unknown Criminal 1, Unknown Criminal 2, etc.;

3. Since we have the address of the crime location in the form of X and Y coordinates in the neighborhood where the crime was committed, we have created a table titled Crime Location that shows only the address of the crime; this differs from Address which shows the home address of the victim as well as the police officer and police station that responded to the crime; and,
4. Whether a victim is identified or not identified, if the Address of the victim is not identified then a default address is assigned to that victim whose Address ID is 3.

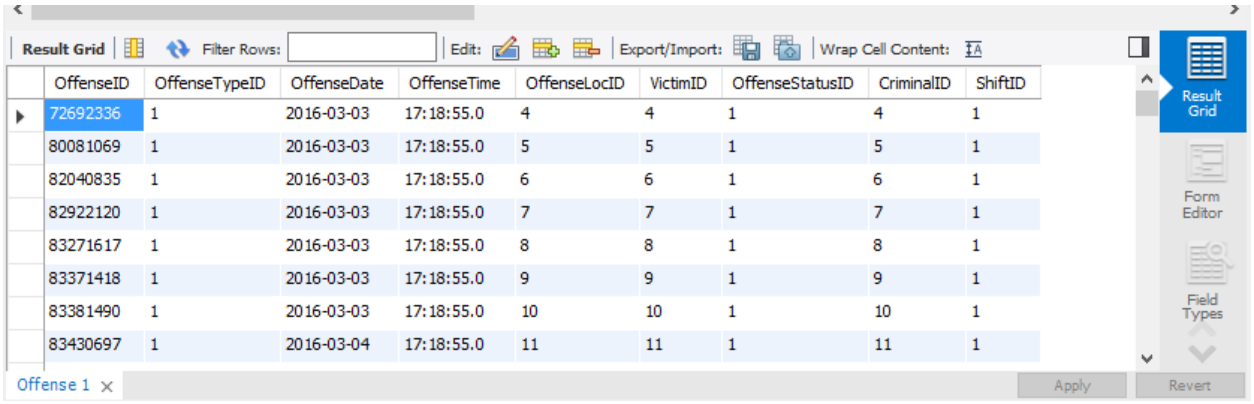
Appendix B - Context Diagram



Appendix C - Level 0 Data Flow Diagram



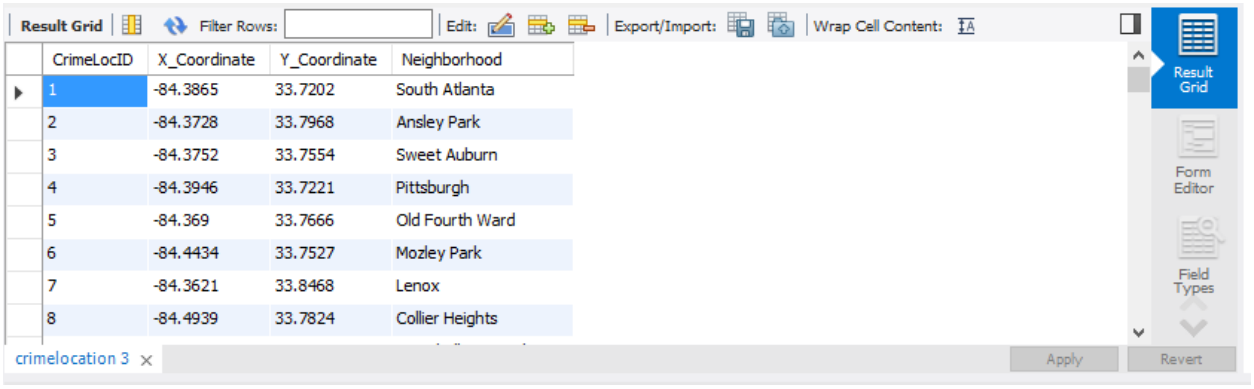
Appendix D – Examples of Physical Table and Sample Data Screenshots



Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

OffenseID	OffenseTypeID	OffenseDate	OffenseTime	OffenseLocID	VictimID	OffenseStatusID	CriminalID	ShiftID
72692336	1	2016-03-03	17:18:55.0	4	4	1	4	1
80081069	1	2016-03-03	17:18:55.0	5	5	1	5	1
82040835	1	2016-03-03	17:18:55.0	6	6	1	6	1
82922120	1	2016-03-03	17:18:55.0	7	7	1	7	1
83271617	1	2016-03-03	17:18:55.0	8	8	1	8	1
83371418	1	2016-03-03	17:18:55.0	9	9	1	9	1
83381490	1	2016-03-03	17:18:55.0	10	10	1	10	1
83430697	1	2016-03-04	17:18:55.0	11	11	1	11	1

Result Grid | Form Editor | Field Types | Offense 1 | Apply | Revert



Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

CrimeLocID	X_Coordinate	Y_Coordinate	Neighborhood
1	-84.3865	33.7202	South Atlanta
2	-84.3728	33.7968	Ansley Park
3	-84.3752	33.7554	Sweet Auburn
4	-84.3946	33.7221	Pittsburgh
5	-84.369	33.7666	Old Fourth Ward
6	-84.4434	33.7527	Mozley Park
7	-84.3621	33.8468	Lenox
8	-84.4939	33.7824	Collier Heights

Result Grid | Form Editor | Field Types | crimelocation 3 | Apply | Revert

Appendix E - MySQL Queries

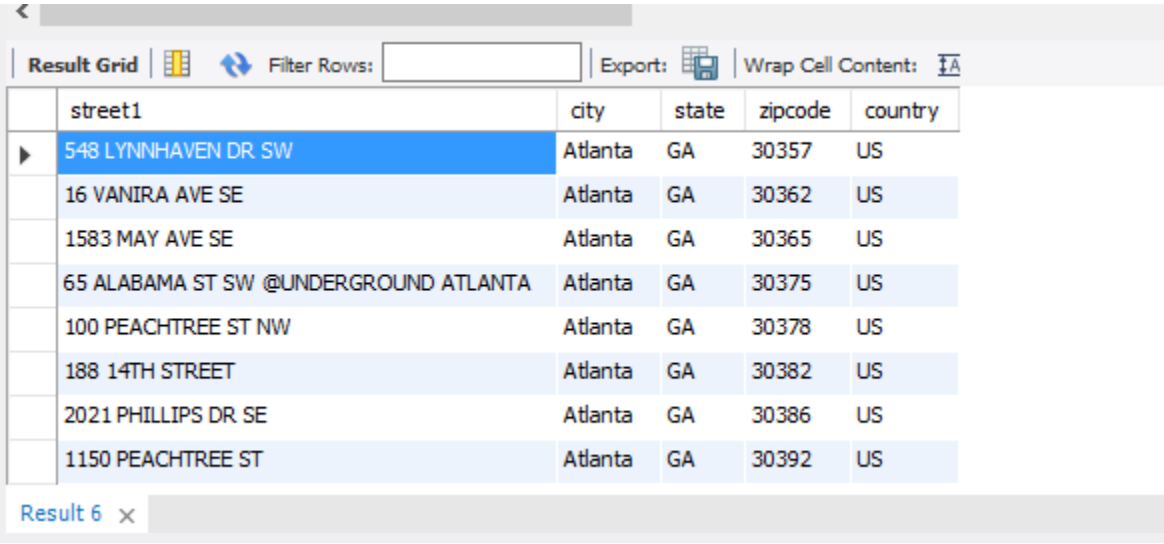
1. Total active crimes in Atlanta.

```
select count(offenseID)  from databaseproject.offense as a, databaseproject.crimestatus as b
where a.OffenseStatusID = b.CrimeStatusId and b.CrimeStatusType = 'A'
```

2. Finding the address of the victim whose crime status is not complete but whose file is still closed.

```
select a.street1,a.city,a.state,a.zipcode,a.country from databaseproject.address a ,
 databaseproject.victim b,databaseproject.offense c,databaseproject.crimestatus d
where a.AddressId = b.VictimAddressId
and c.VictimID = b.VictimID
and c.OffenseStatusID = d.CrimeStatusId
and d.CrimeStatusType = 'CI'
```

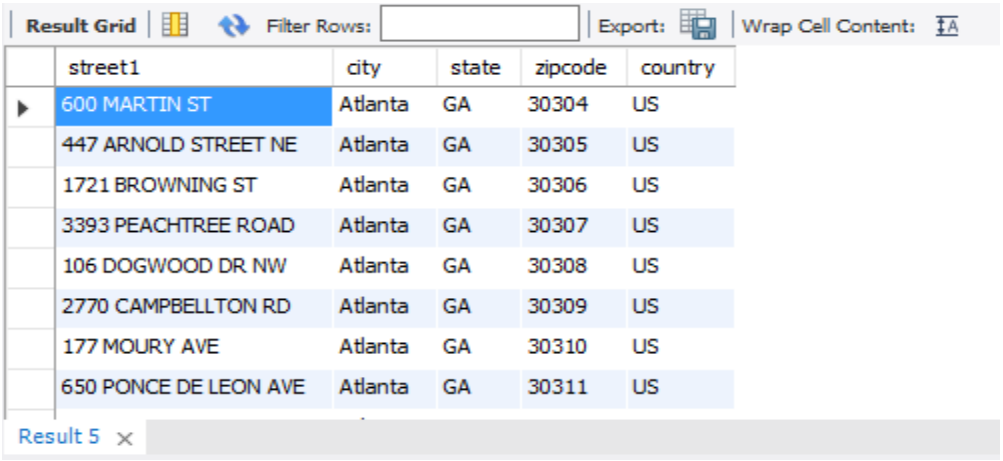
Total 8 Rows Returned as Output



3. Finding the address of the victim whose crime status is active.

```
select a.street1,a.city,a.state,a.zipcode,a.country from databaseproject.address a ,
 databaseproject.victim b,databaseproject.offense c,databaseproject.crimestatus d
where a.AddressId = b.VictimAddressId
and c.VictimID = b.VictimID
and c.OffenseStatusID = d.CrimeStatusId
and d.CrimeStatusType = 'A';
```

Total 175 rows Returned –



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

street1	city	state	zipcode	country
600 MARTIN ST	Atlanta	GA	30304	US
447 ARNOLD STREET NE	Atlanta	GA	30305	US
1721 BROWNING ST	Atlanta	GA	30306	US
3393 PEACHTREE ROAD	Atlanta	GA	30307	US
106 DOGWOOD DR NW	Atlanta	GA	30308	US
2770 CAMPBELLTON RD	Atlanta	GA	30309	US
177 MOURY AVE	Atlanta	GA	30310	US
650 PONCE DE LEON AVE	Atlanta	GA	30311	US

Result 5

4. Finding the address of the victim whose crime status is completed and whose file is closed.

```
select a.street1,a.city,a.state,a.zipcode,a.country from databaseproject.address a ,
 databaseproject.victim b,databaseproject.offense c,databaseproject.crimestatus d
where a.AddressId = b.VictimAddressId
and c.VictimID = b.VictimID
and c.OffenseStatusID = d.CrimeStatusId
and d.CrimeStatusType = 'CC';
```

Total – 13 Rows Returned as Output

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

street1	city	state	zipcode	country
1137 AVON AVE SW	Atlanta	GA	30320	US
BUCKEAD AVE./GRANDVIEW AVE.	Atlanta	GA	30321	US
195 ARIZONA AVE NE	Atlanta	GA	30322	US
58 SAUNDERS ST NE	Atlanta	GA	30323	US
1145 PRINCESS AVE SW	Atlanta	GA	30324	US
1530 RALPH D ABERNATHY BLVD SW	Atlanta	GA	30325	US
14 BELMONTE CT SW	Atlanta	GA	30326	US
585 MCWILLIAMS RD SE	Atlanta	GA	30327	US

Result 4

5. Top 3 neighborhoods in crime.

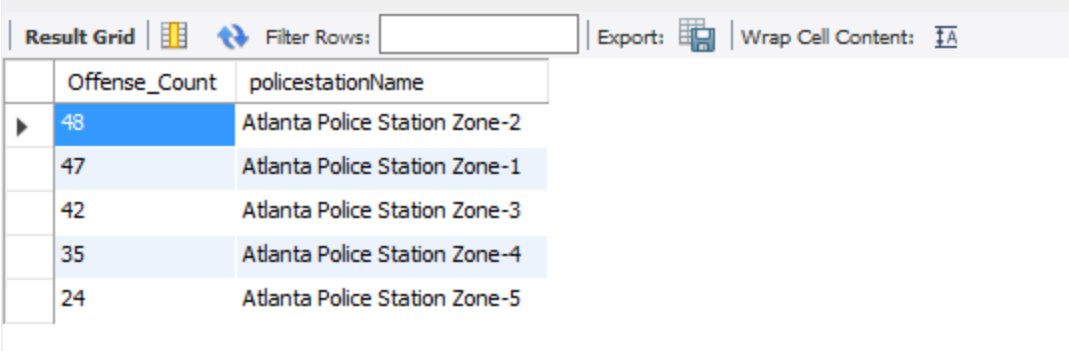
```
select count(OffenseID) as cnt, Neighborhood from
(select A.* , B.* from databaseproject.crimelocation as A, databaseproject.offense as B
where A.CrimeLocID = B.OffenseLocID) as T1
group by Neighborhood
order by cnt DESC
LIMIT 3
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

crime_count	Neighborhood
20	Downtown
13	Midtown
7	English Avenue

6. Count of number of crimes reported per police station in Atlanta.

```
select count(OffenseID) as cnt, policestationName from
(select B.OffenseID,policestationName from databaseproject.policestation as
A,databaseproject.offense as B,databaseproject.pso_mapping as C
 where A.PoliceStationId = C.PoliceStationID and C.offenseID = B.OffenseID
 ) as T1
group by policestationName
order by cnt DESC;
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Offense_Count	policestationName
48	Atlanta Police Station Zone-2
47	Atlanta Police Station Zone-1
42	Atlanta Police Station Zone-3
35	Atlanta Police Station Zone-4
24	Atlanta Police Station Zone-5

7. Top 3 police officers dealing with most offenses.

```
select policeofficer.PoliceOfficerFirstName, policeofficer.PoliceOfficerLastName,
count(pso_mapping.offenseID)
from pso_mapping,  policeofficer
where pso_mapping.PoliceOfficerID = policeofficer.PoliceOfficerId
group by pso_mapping.PoliceofficerID
order by (select count(pso_mapping.offenseID)) desc;
```

PoliceOfficerFirstName	PoliceOfficerLastName	count(pso_mapping.offenseID)
Jessica	Johnson	5
William	Smith	5
Thomas	Johnson	5

8. Get all the offense after 2016-04-15 with their descriptions.

```
select offense.OffenseID, offensetype.OffenseDescription
from offensetype, offense
where offense.OffenseDate > 2016-04-15 and
offense.OffenseTypeID = offensetype.OffenseTypeID;
```

OffenseID	OffenseDescription
83662023	LARCENY-NON VEHICLE
83662137	LARCENY-NON VEHICLE
90010061	LARCENY-NON VEHICLE
90010108	LARCENY-NON VEHICLE
90010133	LARCENY-NON VEHICLE
90010143	LARCENY-NON VEHICLE
90010741	LARCENY-FROM VEHICLE
90010745	LARCENY-FROM VEHICLE

9. Calculate the number of offenses that happen during daytime (we had three shifts: day, evening, and night).

```
select count(offense.offenseid)
from offense, shift
where offense.ShiftID in
(select shift.shiftid
from shift
where shift.ShiftName = "Day");
```

count(offense.offenseid)
123

10. List the neighborhoods where at least two crimes took place, ordered by locationID.

```
select crimelocation.crimelocid, crimelocation.neighborhood
from crimelocation
group by crimelocation.neighborhood
having count(crimelocation.crimelocid) > 1
order by crimelocation.crimelocid;
```

crimelocid	neighborhood
1	South Atlanta
4	Pittsburgh
5	Old Fourth Ward
7	Lenox
8	Collier Heights
9	Campbellton Road
10	The Villages at Carver

crimelocation 38