INST327 - Database Design and Modeling

Final Project Submission

Group 7

Samantha Gessner

Miracle Agyapong

Travis Johnson

Sammi Lacovara

Kiehwan Park

Introduction:

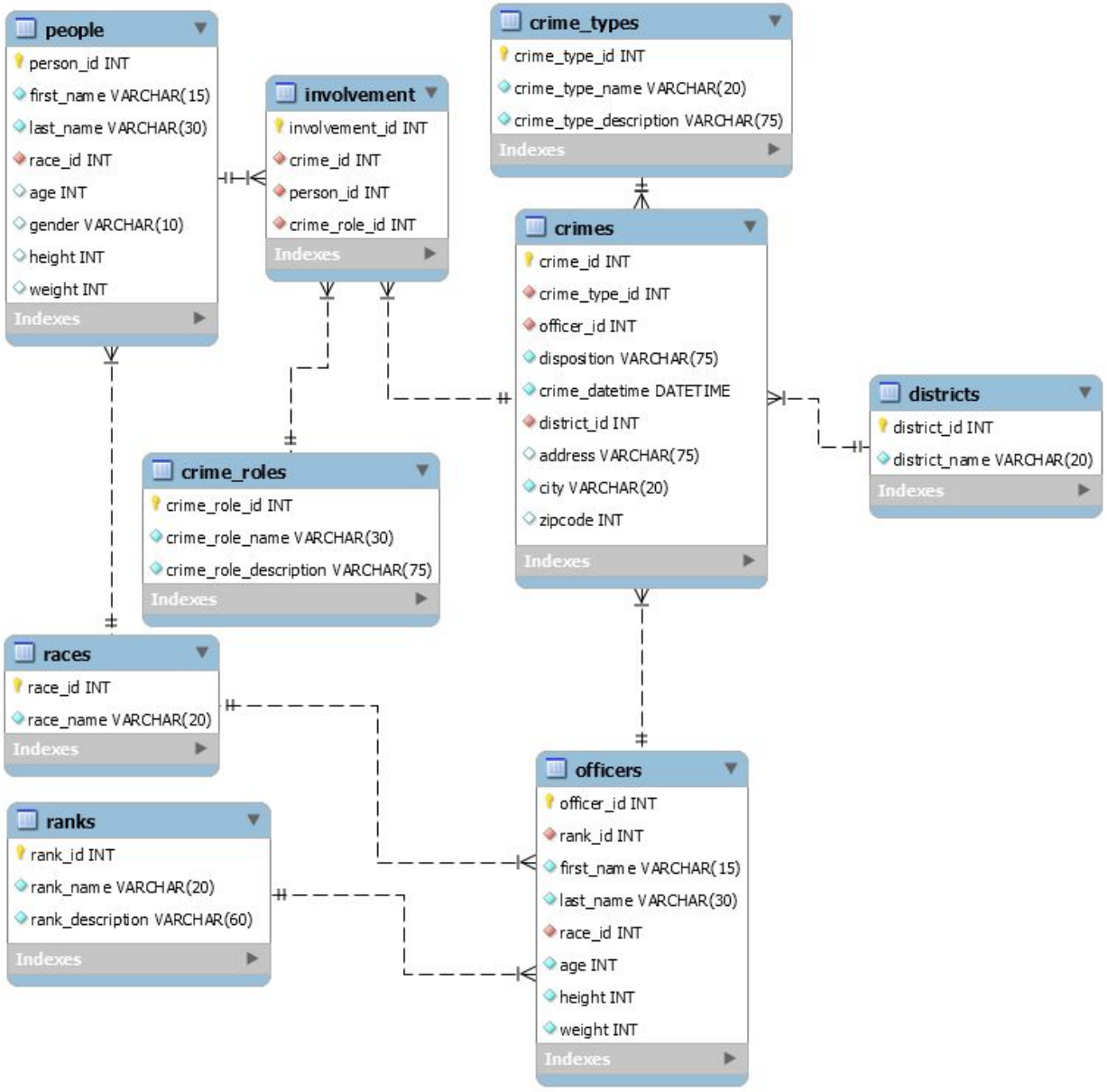
We believe creating a database on crimes in Prince George's County will be beneficial to residents and non-residents of Prince George's County. This section mentions our database, problem domain, and motivation.

Our database focuses on crimes in Prince George's County (PGC). We believe this database could be beneficial to residents or potential residents of PGC, especially students at the University of Maryland. We initially decided to generate reports about crimes within PGC within the last three months but later on decided to filter it down to five days. The database covers most major crimes, including theft, murder, etc. In addition, includes information on crimes, officers, witnesses, and district number. Some of the data we used in the database is fake because some data is restricted from public use, for safety. However, all the crimes were real, and their locations are accurately represented. We chose this topic because it has an impact on the area we currently live in, and possible areas for residents/newcomers to avoid. It will also help those looking at the database to be alert about certain areas. For example, if there are many thefts in Hyattsville, users could prevent their things from being stolen by not leaving valuable items in plain view in their cars. We made a couple of changes to the database from our proposal; however, they will be further discussed throughout this report.

Database Description:

Logical Design:

Our database includes four core, one linking, and five code tables. Listed below is our ERD with our entities and attributes:



Physical Database:

Moving from the logical design to the physical design required the team to pay close attention to details. Our entities became our tables: people, involvement, crime_types, crime_roles, crimes, districts, officers, ranks; and our attributes became our databases column names. We then specified primary keys, foreign keys, data types, and relationships. Using queries, we created the tables and relationships, as well as ensuring all of the table names/attributes met the requirements for the physical database (i.e. spaces are not allowed in table names when translating into SQL, however, we missed the 'crime roles' variable which

should have been ‘crime_roles’).

Sample Data:

Loading the sample data into the database was a long task, and we encountered many issues. We collected data from almost 50 total crimes throughout PG county, as well as made up some sample data such as names, heights (centimeters), weights (pounds), and races. We first entered all of the data into Google Sheets which allowed us to split up a lot of the work and not rely on one person to enter all the data into the database by themselves. After converting all of the files to csv, we started by importing the tables with no foreign keys. We had to make sure all information was imported prior to moving to the next step. Step-by-step, we followed the process indicated in the slide deck titled “Week 13 - ETL”. Every individual table in our database was represented by a table in Excel. We were effectively able to gather data in Excel as a team and then import that information in MySQL. Although some problems were encountered when entering the information into MySQL, as a team we were able to figure out what needed to be changed in order for everything to run smoothly.

Views/Queries:

The queries were written as partial pieces of information that may be useful to someone who uses the database. They are extensive (include all tables) and interactive with mostly more than one table.

View Name	2+ Tables	Filtering	Aggregate	Linking	Subquery
officer_info	X				
people_below_avg_age	X	X	X		X
victims_of_crimes	X	X		X	
district_crime_count	X		X		
crime_list_timeframe	X	X			

The first query is standard and is used to display officer information, including name, age, and their rank. The second query is a little more complex and gathers information on people (not officers) that are all below the average age of the general population. The third query is the most extensive spanning five tables and displayed all victims' names along with the crime they were victims of and the date/time of the crime. The fourth query is a standard aggregate query that gathers all the crimes, and counts them, while grouping them by their district. The final query is a standard crime list that shows the date/time of the crime, address, city, zip code, district, crime type, and disposition, but filters for only crimes within a specific time frame. The supplied time frame is April 1st - April 3rd, but can be changed if edited.

Changes From Original Design:

There were very little changes made from the original design, including data types and attribute requirements.

The only major change was moving the "disposition" table into the central core "crimes" table. We also had a small naming error in the "crime_roles" table. It was originally named "crime roles" instead of "crime_roles". Also, during data ingest, we realized our "rank_name" attribute in the "ranks" table was not long enough to hold the full rank names of officers, so it was extended from 10 characters to 20 characters.

- *Disposition table was integrated into crimes table*
- *crime_roles table title from "crime roles" to "crime_roles"*
- *Ranks table, rank_name type from varchar(10) to varchar(20)*

Lessons Learned:

Working with MySQL for the first time this semester was a difficult process for most of us in our group. This was a brand new idea and together, we had to figure out what worked and what did not work to create a database.

One of the most prominent lessons we learned was how to work in a group successfully. At the beginning of the assignment, we struggled with communication and expectations of one

another. We thought we had accurately layed out what we believed our individual parts were in this semester long project; however, there were many bumps along the way. Our way of working through our issues within the group was getting in touch with the professor and meeting with her privately with just our group in the library. She proposed we more thoroughly talk about expectations for individual projects and deliverables. We assigned each individual task to someone and clearly stated what needed to be done. This resolution helped us fix our communication problems and identify who needed to complete what aspect of the project. From this, we learned that communication among groups is a difficult task. We originally did not identify certain things about the deliverables that should have been talked about in group discussions. We learned that one way to effectively communicate and sort through expectations is assigning parts individually to everyone and making sure everyone gets a similar workload.

Other than this challenge, our group successfully worked on the project throughout the semester. Although we learned the difficulty of working with MySQL and the importance of getting every aspect right within it, this is a lesson that everyone in the class had learned as well. Our group worked effectively throughout the entire project, and every deliverable ran smoothly. If anyone was having an issue on their part, they reached out, and together as a group we solved it. Thus, the most prevalent lesson we learned throughout the semester was how to effectively work in a group with peers.

Strengths, Limitations, and Potential Future Work:

We had helped each other with good communication and fulfilled our ultimate goal. We showed good teamwork and that was main reason how we could finish our project well.

While doing our project, we had group meetings several times, and in every meeting, we planned what needed to be done and assigned the individual tasks for the next step. We believe this is was one of our many strengths throughout our project; we made an efficient plan, and everyone knew what they had to do. In our first meeting, we had asked for each other's schedules and availability for meetings, so that every member could catch up, participate, and contribute to the project. Another strength we possessed is that we effectively helped each other with tasks individuals struggled with. Although we had a little communication trouble in the middle of the project - which perhaps in itself acted as a potential limitation for our group, we

had clear communication with each other after we overcame these issues. Based on that, we always asked each other for help whenever someone was struggling with any problem. Thus, we could solve it very quickly through instant communication. This allowed us to work through our communication limitations through GroupMe and texting.

Besides the little trouble, we did not really have any obstacle in achieving our goal, and everything we planned had been done step-by-step. However, there is one thing we would change if we redid this project; since we had to the use of fake data, we had to make up more than half of the collected data. Although it is not a problem to create this fake data, it is inaccurate; thus, we think that for a future work, all needed data - such as names - should be readily available to the group creating the database.