

Criminal Database

1. Database Description:

1.1. Why This Database?

In any functioning nation with a cohesive law enforcement division, a criminal database is needed to document criminals and their crimes, recognize repeat offenders, and analyze crime trends. A criminal database must keep track of a number of things; any and all information about documented criminals, details of crimes committed, officers responding to crimes, as well as the proceeding court cases. A criminal database should ultimately not only make the process of storing and viewing criminal data more convenient, but also accelerate police work in every other aspect and give insight into future crimes.

1.2. Potential Users

Obviously, the majority of users for criminal databases will be those in law enforcement, whether they be field officers or office workers. For this reason, the database should foremost be designed for law enforcement workers to make it as efficient for them as possible. After law enforcement, some criminal databases are made available to the general public. However, since much of the stored information could be classified, the scope of the database should be limited for these users.

2. Summary of Data Requirements of Users

In a criminal database, the scope needs to be adjusted based on who is using the database in order to prevent the exposure of data not intended for the public eye. A

user such as an officer would be able to access any part of the database without restriction for the most part, except in special cases whenever a certain officer is disallowed from participating in a specific case. Situations like this may arise whenever the offender or victim of a case has a personal relation to an officer. On top of this, a system of logging the IDs of officers who access the system should also be implemented for accountability purposes. Only those with access to the full scope of the majority of the database need to be logged, so only those working for the police force.

The general public will also have access to a limited amount of data in the database, namely only being able to access records of criminals of resolved cases, and resolved case files. There are a number of reasons for these restrictions, all in the interest of protecting both the public and witnesses.

3. Function Requirements

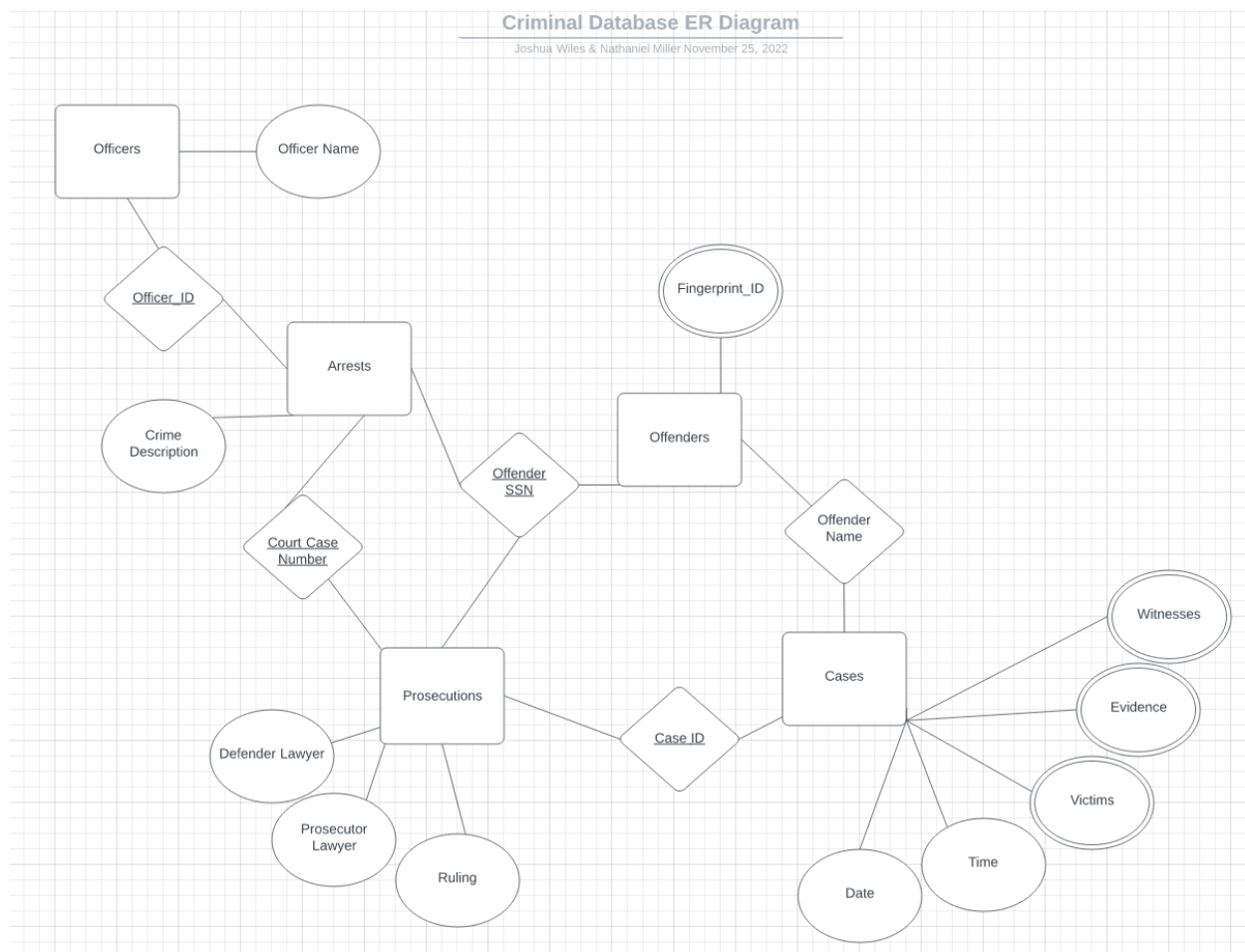
The officers need to be able to input the names and identities of known criminals. They then need to be able to input the crimes that the individual has committed and link that with their identity. They then need to be able to link the criminal to any arrests and prosecutions that have occurred.

They need to be able to upload the fingerprints gotten from a criminal and link them with the criminal. They also need to be able to create cases, and add evidence to those cases.

Beginning Of Task 2:

[LINK TO GITHUB REPOSITORY](#)

ER Diagram:



Explanation:

In the ER Diagram for our Criminal Database, we used a couple of different notations and relationships to describe the way that data interacted within the database. In between the Offenders table and the Arrests table, we used “one or many” to connect the data because a single offender could be arrested multiple times, but they would not

have multiple instances of themselves in the Offenders table. Similarly, this was used between the Offenders and Prosecutions table, an Offender would only appear once on the Offenders table but could appear on the Prosecutions table multiple times if they went to court for more than one offense.

Relationships:

A kind of relationship also present in the database is that between the Offenders table and the Cases table. In a single criminal case, there can be multiple offenders for a certain crime, but there can also be only one.

On top of that, there exists a relationship between the Arrests table and the Officers table. In this case, a single Officer can make more than one arrest and an arrest can be carried out by multiple Officers at once. It is also possible for a single Officer or multiple Officers to arrest multiple individuals at once as well, all being filed under the same arrest.

Relational Schemas:

| | | | |
|-----------|------------|------|----------------|
| Offenders | <u>SSN</u> | NAME | FINGERPRINT_ID |
|-----------|------------|------|----------------|

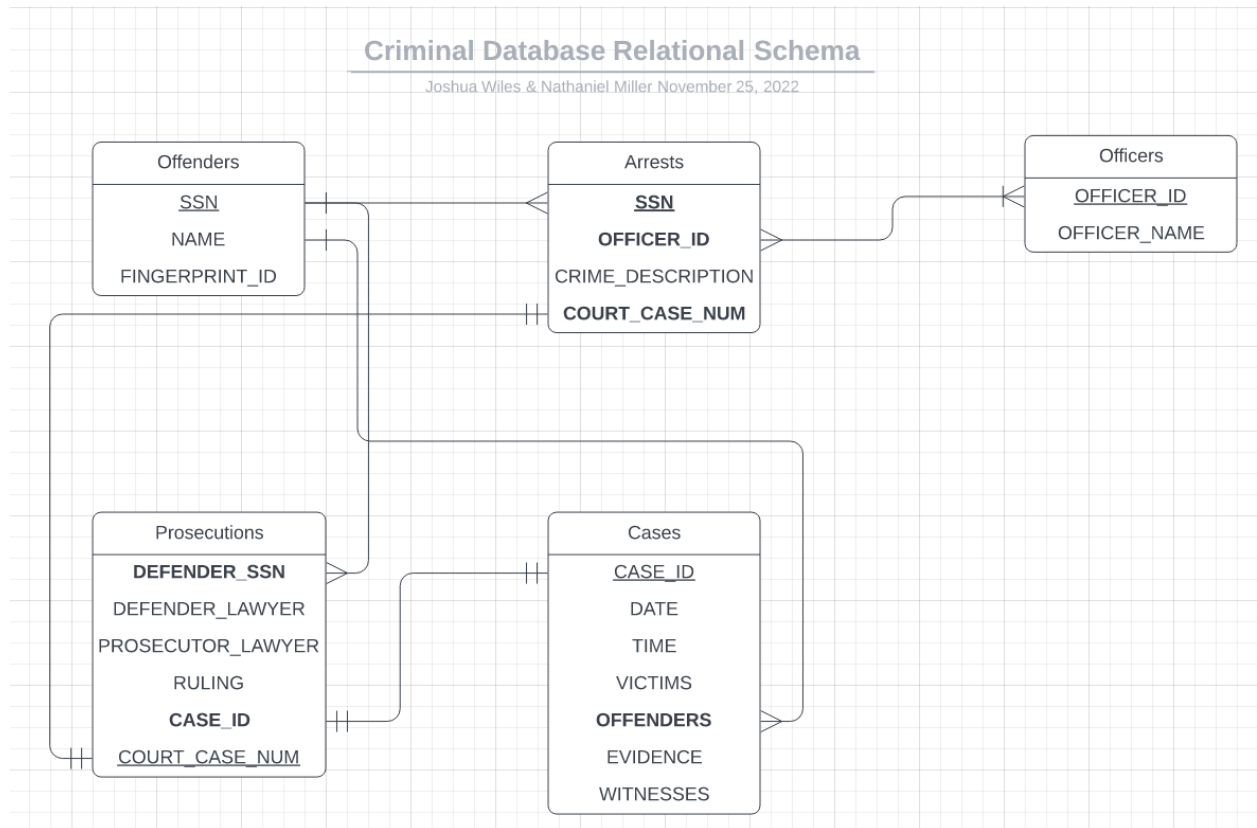
| | | | | |
|---------|------------|------------|-------------------|----------------|
| Arrests | <u>SSN</u> | OFFICER_ID | CRIME_DESCRIPTION | COURT_CASE_NUM |
|---------|------------|------------|-------------------|----------------|

| | | | | | | |
|-------------|--------------|-----------------|-------------|--------|---------|----------------|
| Prosecution | DEFENDER_SSN | DEFENDER_LAWYER | PROS_LAWYER | RULING | CASE_ID | COURT_CASE_NUM |
|-------------|--------------|-----------------|-------------|--------|---------|----------------|

| | | | | | | | |
|-------|----------------|------|------|---------|-----------|----------|-----------|
| Cases | <u>CASE_ID</u> | DATE | TIME | VICTIMS | OFFENDERS | EVIDENCE | WITNESSES |
|-------|----------------|------|------|---------|-----------|----------|-----------|

| | | |
|----------|-------------------|--------------|
| Officers | <u>OFFICER_ID</u> | OFFICER_NAME |
|----------|-------------------|--------------|

Schema Diagram:



Web App Description:

The app gives access to the main 4 tables, being the Offenders, the Arrests, the Prosecutions, and the Cases. Each of the four main tables are accessible from the home page as well as a search bar that takes in two separate inputs. The entire site is built with basic HTML and CSS with all redirects being to PHP-generated websites.

In the search bar an HTML form was used that leverages a text area and drop-down menu. From the drop-down, a user can select a table to search from and a keyword can be entered to search for any instances in the table. After entering all of the data for the search, pressing the submit button sends the data to a PHP file that creates an SQL select statement with said data. Once the query is executed, the user is

redirected to a new page where the PHP automatically builds an HTML table to display the retrieved data.

For the other main feature of the site, as mentioned previously, a user can choose one of the main four tables from the homepage. Upon selecting one of the four tables, a PHP file sends a query to the database requesting all data from the selected table. Similarly to the search bar, the user is redirected to a new page where an HTML table is automatically generated to house all of the data from the query.

Our website functionally is simple as it displays data without too many advanced searching options or features. We found this approach to be realistic for the user scope of a police database. The average user of the database, a police officer, likely wouldn't be doing anything too sophisticated like data entry or advanced queries. It's likely that all data entry would be handled by someone with a more intimate knowledge of the database system. Additional things that could be added to the database would be a login system that would connect to the officer table in the database that would require a password and officer ID to access the more sensitive parts of the database. There could also be an option to view the database without a password, although it would hide certain information, like the SSNs of offenders. We didn't find it necessary to delve too deep into these extra features in order to keep everything simple and prioritize the most important functions of the database.