Project BD1 - Police System for Storing Crime Information

Brygida Silawko, Kornelia Błaszczuk

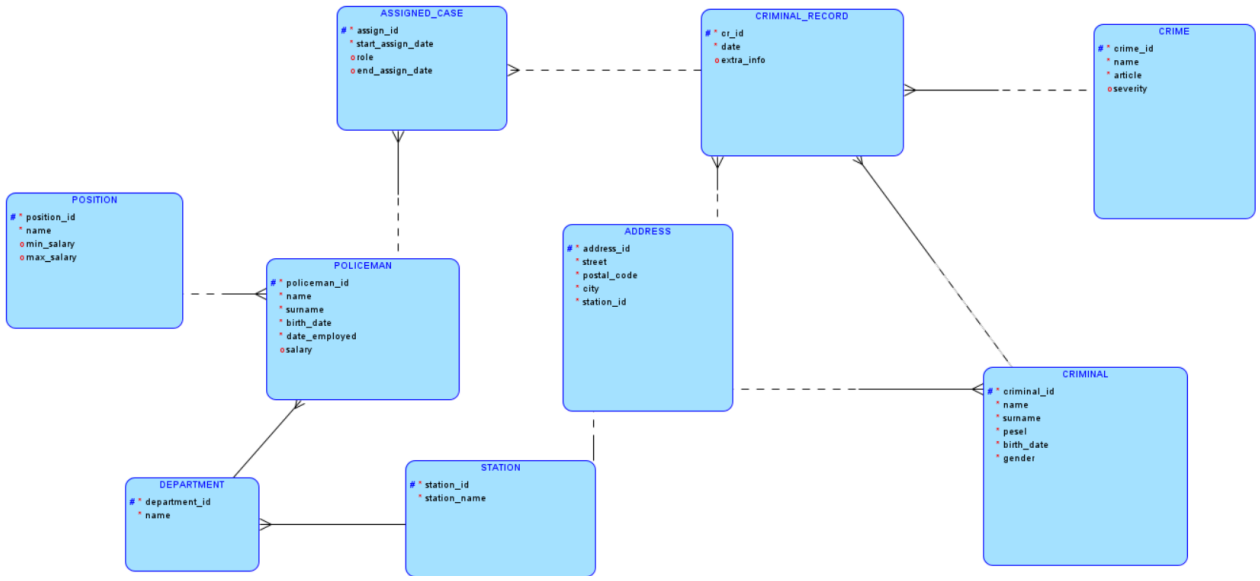
1 Initial Assumptions

In our project, we created a database for the police. Its purpose is to manage ongoing investigations, crimes, active cases, and police personnel.

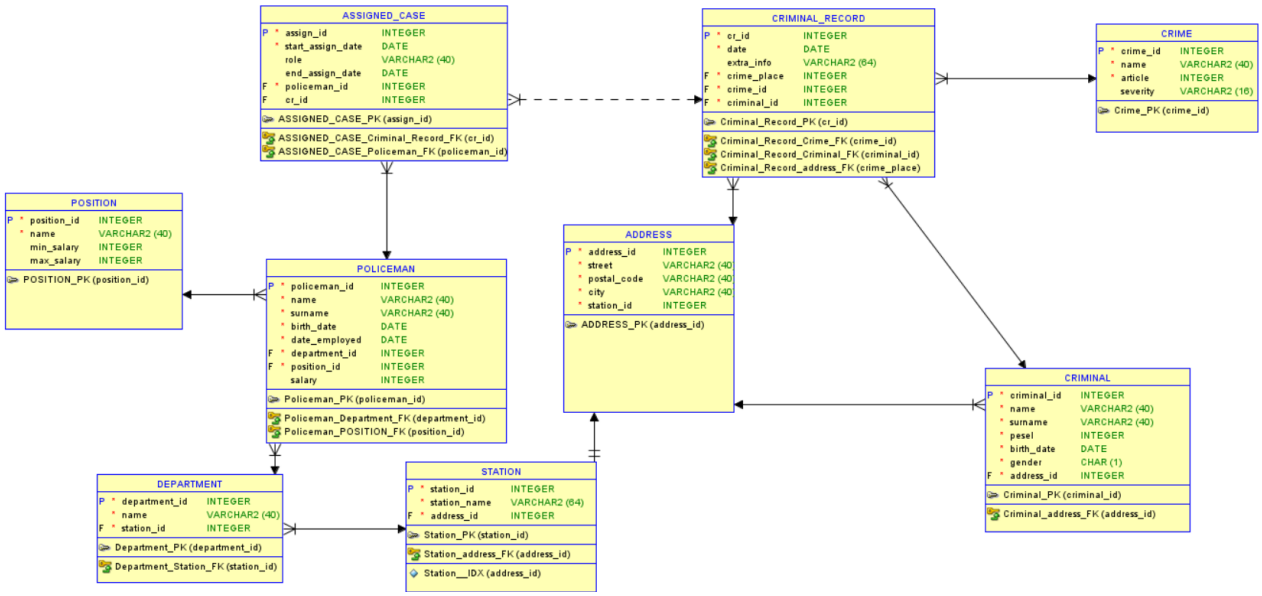
When designing the tables, we placed special emphasis on creating a clear structure. Additionally, the system structure allows for easy scalability and future expansion with additional modules.

Furthermore, we implemented an application written in Python, which facilitates easy interaction with the database. We used the *oracledb* library for connectivity. To enhance the clarity of the terminal interface, we also used the *rich* library.

2 ER Model



3 Relational Model



4 Structure Description

The database focuses on storing information related to cases involving criminal offenses. To accomplish this, it also needs basic information about police officers and criminals.

- Ongoing cases (including closed cases),
- Crimes committed by criminals,
- Criminal offenses recorded in the penal code,
- Criminals,
- Police officers,
- Ranks for police officers,
- Police stations, including their addresses and departments.

5 Implemented Features

5.1 Scripts

- Scripts for database schema creation,
- Scripts for loading data (using sequences),
- Test scripts - for each implemented function, procedure, and trigger, we included tests that cover both expected behaviors and cases that should return errors. Additionally, we included non-trivial queries using various joins, filtering, and data grouping to ensure the database schema functions correctly.

5.2 Sequences

A sequence was created for each table to streamline the addition of new entries.

5.3 Triggers

1. When updating the end date of a case in the cases table, it checks whether the date is valid - ensuring it is not in the future and that the end date is later than the start date.
2. When modifying a police officer's salary, it checks whether the salary falls within the appropriate range for the given position.

5.4 Procedures

1. Display active cases that are not yet closed.
2. Increase police officers' salaries based on years of service and the number of cases closed.

5.5 Functions

1. Count the number of crimes committed by a given person.
2. Count the number of crimes in a given year.

6 Task Division

To ensure efficient completion of all system components, tasks were divided among team members. The table below outlines the task allocation and responsibilities.

Task	**Responsible Person**
ER Model	Brygida Silawko, Kornelia Błaszczuk
Relational Model	Brygida Silawko, Kornelia Błaszczuk
DDL Scripts	Kornelia Błaszczuk
Data Loading Scripts	Brygida Silawko
Sequence Definitions	Kornelia Błaszczuk
Trigger Definitions	Brygida Silawko, Kornelia Błaszczuk
Procedure Definitions	Brygida Silawko, Kornelia Błaszczuk
Function Definitions	Brygida Silawko
Test Scripts	Brygida Silawko, Kornelia Błaszczuk
Python Application	Brygida Silawko, Kornelia Błaszczuk
Report	Kornelia Błaszczuk

Table 1: Task Allocation in the Project

7 Python Application and Usage

The Python application allows users to easily access the database, offering a range of functions, procedures, and triggers for efficient system interaction. By utilizing modern libraries such as oracledb and rich, the application is both intuitive and efficient.

7.1 Database Interaction Options

Available options in the application include:

- Display a table.
- Count crimes committed by a specific criminal.
- Count crimes in a given year.
- Display crimes within a specified date range.

- Show active cases.
- Increase salary.
- Update case closure date.
- Add a crime.
- Add a criminal.
- Add an address.
- Add a police officer.
- Add a position.
- Update a police officer's salary.
- Exit.

7.2 Running the Application

To test the application, create a *.env* file (a template is available in *.env.template*). Then, run the main Python script using the following command in the terminal:

```
python main.py
```

To install dependencies, use:

```
pip install -r requirements.txt
```

8 Conclusion

This project created a database that stores information related to crimes, cases, and police personnel. The system ensures a well-structured and scalable design, enabling future enhancements.

The Python application facilitates easy database management. By integrating modern libraries like *oracledb* and *rich*, user interaction is both efficient and user-friendly.