

Projekt BD1 - System policyjny przechowujący informację o przestępstwach.

Brygida Silawko, Kornelia Błaszczuk

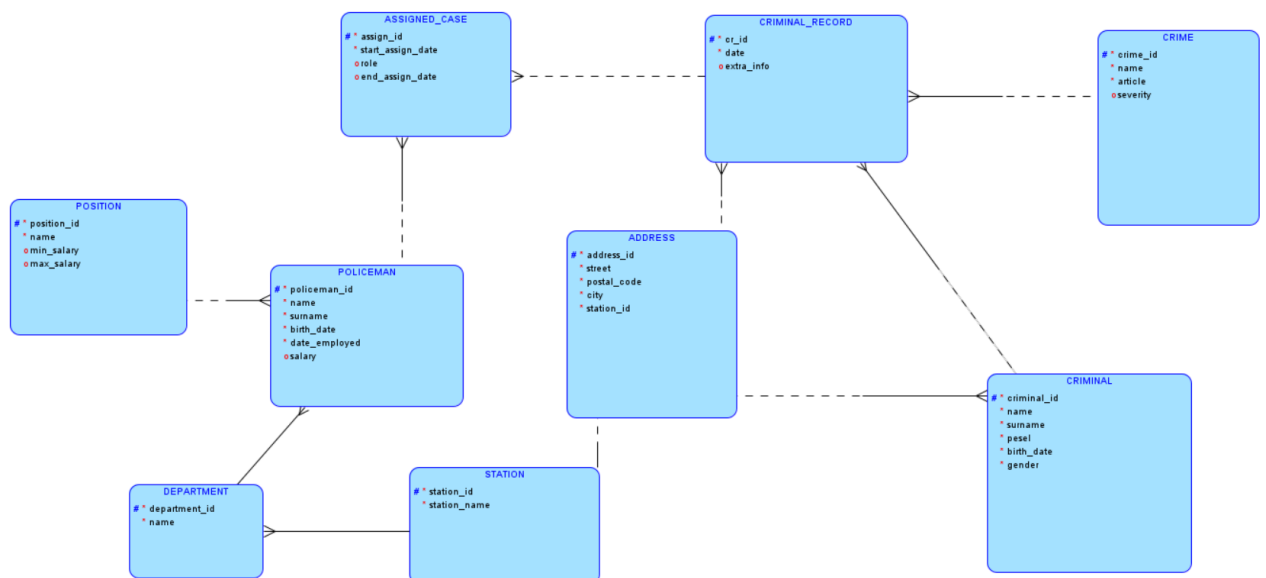
1 Założenia początkowe

W naszym projekcie stworzyliśmy bazę danych dla policji. Ma ona na celu obsługę prowadzonych śledztw, przestępstw, prowadzonych spraw oraz kadry pracowniczej.

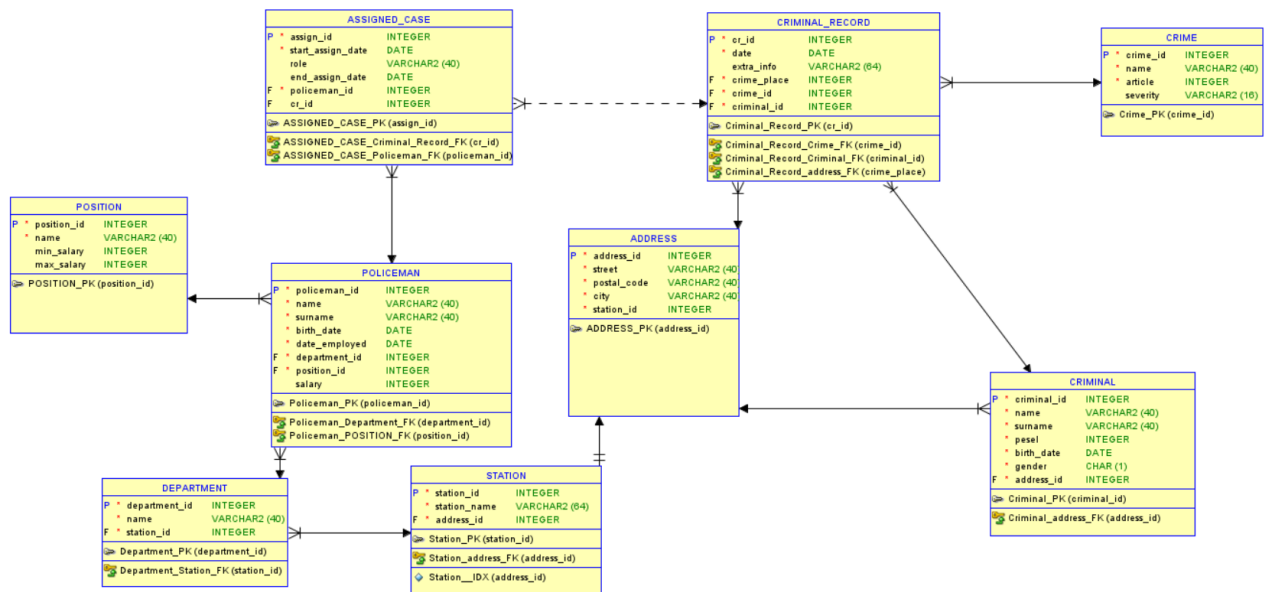
Przy podziale tabel szczególny nacisk położyliśmy na stworzenie przejrzystej struktury. Ponadto, struktura systemu umożliwia łatwą skalowalność i przyszłą rozbudowę o dodatkowe moduły.

Dodatkowo, zaimplementowaliśmy aplikację napisaną w Pythonie, która pozwala na łatwą interakcję z bazą danych. Do połączenia użyliśmy biblioteki *oracledb*. Aby poprawić przejrzystość interfejsu terminalowego wykorzystaliśmy również inną bibliotekę *rich*.

2 Model ER



3 Model Relacyjny



4 Opis struktury

Baza danych skupia się na przechowywaniu informacji mających związek z prowadzonymi sprawami związanymi z popełnieniem przestępstwa. Aby tego dokonać potrzebne są również podstawowe informacje o policjantach oraz kryminalistach.

- prowadzone sprawy (w tym zakończone),
- przestępstwa popełnione przez kryminalistów,
- przestępstwo zapisane w prawie karnym,
- przestępcy,
- policjanci,
- stopnie służbowe dla policjantów,
- komendy, w tym ich adresy oraz departamenty policji.

5 Co zostało zaimplementowane?

5.1 Skrypty

- Skrypty do założenia schematu bazy danych,
- Skrypty do załadowania danych (używane z sekwencjami),
- Skrypty testowe - dla każdej z zaimplementowanych funkcji, procedur, wyzwalaczy dodano testy uwzględniające zachowania zgodne z założeniami, jak i te, które powinny zwrócić pewien rodzaj błędu. Dodatkowo, aby upewnić się, co do poprawnego działania bazy i jej schematu, dodano zapytania nietrywialne wykorzystujące różne rodzaj złączeń, filtrowanie, grupowanie danych.

5.2 Sekwencje

Dla każdej z tabel została stworzona sekwencja, aby usprawnić dodawanie nowych wpisów.

5.3 Wyzwalacze

1. Po zmianie daty zakończenia sprawy w tabeli spraw, wyświetlane są propozycje przestępstw, dla których nie rozpoczęło się jeszcze śledztwo.
2. Przy wprowadzaniu zmian w płacy policjanta, sprawdzana jest poprawność wykonywanych operacji - czy płaca mieści się w widełkach dla stanowiska.

5.4 Procedury

1. Wyświetlanie spraw aktywnych, które jeszcze nie zostały zakończone.
2. Podwyżka płac policjantów, w zależności od stażu pracy i liczby zakończonych spraw.

5.5 Funkcje

1. Liczba przestępstw dla danej osoby.
2. Liczba przestępstw w danym roku.

6 Podział zadań

W ramach realizacji projektu, zadania zostały podzielone między członków zespołu, w celu efektywnej realizacji wszystkich elementów systemu. Poniższa tabela przedstawia podział zadań oraz przypisanie odpowiedzialności.

Zadanie	Osoba odpowiedzialna
Model ER	Brygida Silawko, Kornelia Błaszczuk
Model relacyjny	Brygida Silawko, Kornelia Błaszczuk
Skrypty DDL	Kornelia Błaszczuk
Skrypty do załadowania danych	Brygida Silawko
Definicje sekwencji	Kornelia Błaszczuk
Definicje wyzwalaczy	Brygida Silawko, Kornelia Błaszczuk
Definicje procedur	Brygida Silawko, Kornelia Błaszczuk
Definicje funkcji	Brygida Silawko
Skrypty testujące	Brygida Silawko, Kornelia Błaszczuk
Aplikacja Python	Brygida Silawko, Kornelia Błaszczuk
Raport	Kornelia Błaszczuk

Tabela 1: Podział zadań w projekcie

7 Obsługa aplikacji w Pythonie

Aby przetestować aplikację, należy utworzyć plik `.env` (wzór znajduje się w pliku `.env.template`). Następnie uruchomimy główny skrypt Python, wykonując poniższą komendę w terminalu:

```
python main.py
```

Po uruchomieniu aplikacja zacznie działać zgodnie z jej przeznaczeniem. W przypadku problemów z uruchomieniem, upewnij się, że wszystkie zależności zostały zainstalowane, wykonując komendę:

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

8 Podsumowanie

W projekcie stworzono bazę danych, która umożliwia przechowywanie informacji związanych z przestępstwami, prowadzonymi sprawami oraz danymi pracowników policji. Skupiliśmy się na zapewnieniu odpowiedniej normalizacji danych oraz elastyczności systemu, aby umożliwić jego przyszłą rozbudowę.

Za pomocą aplikacji napisanej w Pythonie, użytkownicy mogą w prosty sposób zarządzać bazą danych, korzystając z przygotowanych funkcji, procedur oraz wyzwalaczy. Dzięki zastosowaniu nowoczesnych bibliotek, takich jak *oracledb* oraz *rich*, interakcja z systemem jest zarówno efektywna, jak i przyjazna dla użytkownika.

Projekt może stanowić solidną podstawę dla przyszłych usprawnień i rozwoju systemu, z możliwością integracji z dodatkowymi modułami oraz rozszerzaniem funkcjonalności w miarę potrzeb użytkowników.