Universitat Politècnica de Valéncia

Escuela Técnica Superior de Ingeniería Informática

Project Task 7: Crime Investigation

Databases and Information Systems

Mouhoub Walid, Fernández Pérez Rubén, Hovmand Eva Kathrine, Haas Patrick

Autumn 2023

Professor: Soledad Valero

Contents

List of Figures

List of Tables

Chapter 0

UML Class Diagram

0.1 Task Description

First we present the given task description with some minor adjustments from our side in **bold**. **Furthermore, additional adjustments after the first submission are indicated in red.**

> We are going to design the database of a police investigation department. We must store information about the department's employees, whose SSN is unique and it is always known; the passport number, which is also unique; the employee number, which identifies the employee, the name, telephone number, and category. Some workers are investigators, and others are forensic experts.
>
> The following data on the crimes to be investigated are also kept: identification code, type, date of registration of the crime, description, and place. **The place is stored as a longitude and latitude, as a crime potentially could take place in a secluded area without a specific address. The longitude and latitude has to be provided.** There is also information on other persons, for whom a code is stored to identify them, year of birth, and address; a general description of further known details of the person are also stored. It is necessary to record whether there is a relationship between persons, indicating what type of relationship (parent, sibling, cousin, friend, co-worker, etc.). **The database does not store information about all existing people.**

For each crime, it is possible to know which persons have been victims, indicating, in that case, the effect of the crime on them. **A victim always has an effect.**
The persistence of who is the victim, witness, and suspect in a crime are in this case considered fixed roles after a trial and judgment have been made. Therefore, one person can only take one of these roles in one crime, and the role cannot be changed. The suspect in this case should therefore be considered more as a convict. The choice of using fixed roles was made to keep a more simplistic approach to the assignment. However, this does not perfectly reflect the reality in which a person might change role in a crime or might take more than one role as an investigation progresses or if a case gets reopened. When there is a fatality, and the corpse has been found, the corpse data **(Data about the dead person itself)** are recorded, and a forensic scientist is later assigned to perform the autopsy. When the autopsy is finished, the forensic scientist fills out a report on the result. For each crime, it is also stored, when known, which person or persons are suspects. It is also stored which persons have witnessed the crime, **and the witness always has an interrogation.** Also the the investigator who has conducted the interrogation, the date of the interrogation, and the statement of the witness has to be stored. **It was chosen that only witnesses gets an interrogation.**

For each crime, the evidences that are found are stored. Each piece of evidence is numbered with a unique number for each crime. A piece of evidence can be of different types: a photo from which the investigator who took it is saved; a fingerprint, which is associated, when possible, with a person; a recording or other types of evidence. For each piece of evidence, different types of analysis can be performed. Each type of analysis has a code that identifies it, a name, and a description. **For every analysis of an evidence, a report** is made after analyzing a piece of evidence which consists of the resultant data and the conclusion. **The forensic expert, who has conducted the analysis, also writes the report.**

0.2 Additional Requirements

As demanded by the task description we add the following elements:

In the real world, certain crimes may undergo a trial, each identified by a specific trial number. It is essential to record the trial date and its outcome. Each trial requires the presence of a judge, who is employed within the justice

department. Like the investigators or forensic experts, a judge possesses a distinctive identification, which is his Social Security Number (SSN). Furthermore, we also want to store a judge number, a name, and an email address and the date of his/her employment. **These additions was chosen because they greatly resonates with the previously described concept of a persons involvement/role in a crimes is fixed, meaning that the judgement has been made and the people who was involved is either convicted, witnessing or deemed a victim.**

0.3 UML Class Diagram

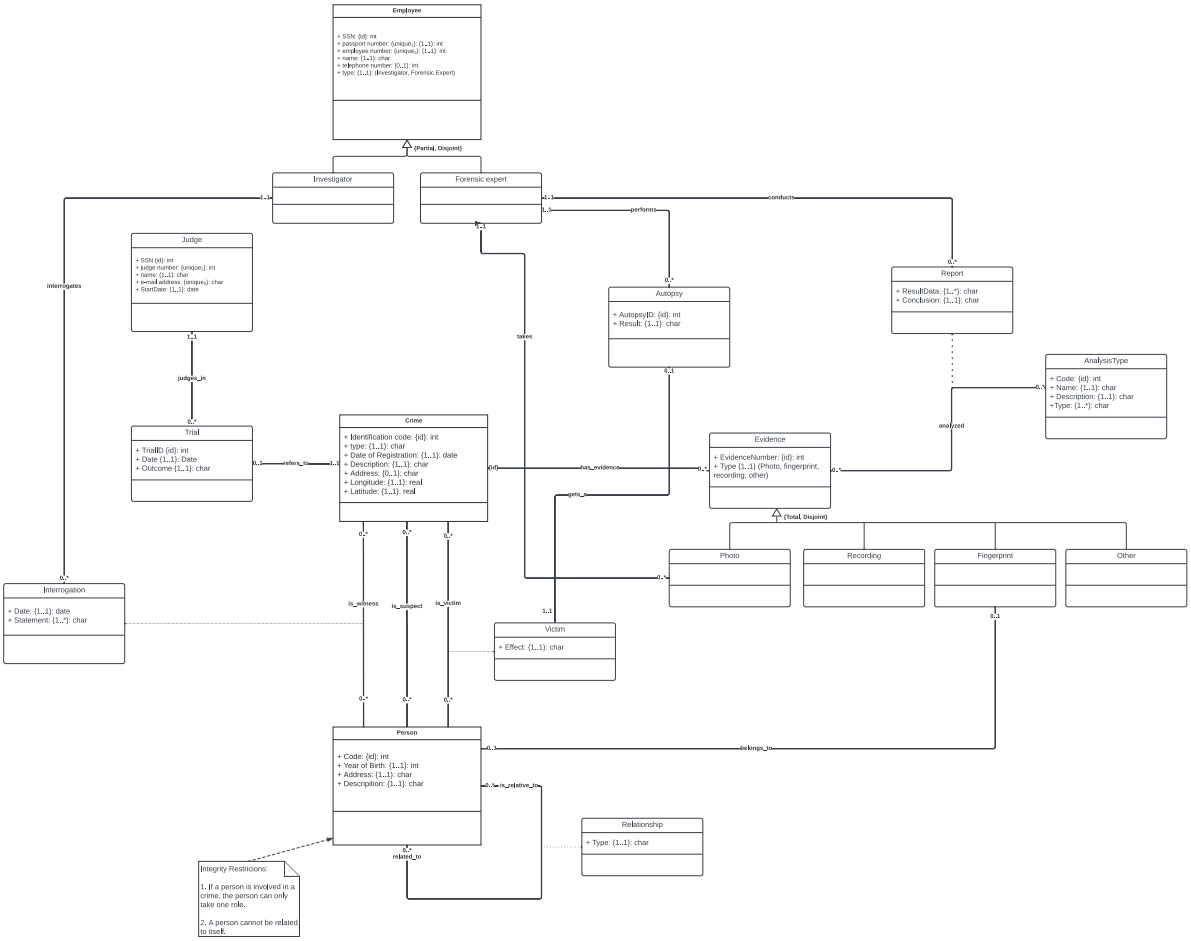

Figure 0.1: The obtained UML Class Diagram.

Chapter 1

Relational design and implementation

1.1 Translation of conceptual schema to the relational schema

1.1.1 Autopsy

```
Autopsy (AutopsyId: int, Code: int, Identification code: int, SSN: int,
Result: char)
PK:{AutopsyId}
UNI:{Code, Identification code}
NNV:{Code, Identification code, SSN}
FK:{Code, Identification code} → Victim(Code, Identification code)
FK:{SSN} → Forensic expert(SSN)
```

1.1.2 Person

```
Person (Code: int, Year of Birth: int, Address: char, Description: char)

PK:{Code}
NNV:{Year of Birth, Address, Description}
```

1.1.3 is_suspect

```
is_victim (Code: int, Identification Code: int)
PK:{Code, Identification Code}
FK:{Code} → Person(Code)
```

FK:{Identification Code} → Crime(Identification Code)

1.1.4 Relationship

Relationship (is_relative_to: int, related_to: int, Type: char)
PK:{is_relative_to, related_to}
NNV:{Type}
FK:{is_relative_to} → Person(Code)
FK:{related_to} → Person(Code)
constraint: is_relative_to must be different to related_to

1.1.5 Victim

Victim (Code: int, Identification Code: int, Effect: char)
PK:{Code, Identification Code}
FK:{Code} → Person(Code)
FK:{Identification Code} → Crime(Identification Code)
NNV:{Effect}

1.1.6 Evidence

Evidence (EvidenceNumber: int, Identification code: int)
PK:{EvidenceNumber, Identification code}
FK:{Identification code} → Crime(Identification code)

1.1.7 AnalysisType

AnalysisType (Code: int, Name: char, Description: char, type: char)
PK:{Code}
NNV:{name, description, type}

1.1.8 Report

Report (EvidenceNumber: int, Identification code: int, Code: int, ResultData: char, Conclusion: char, SSN: int)
PK:{EvidenceNumber, Identification code, Code}
FK:{EvidenceNumber, Identification code} → Evidence(EvidenceNumber, Identification code)

FK:{Code} → AnalysisType(Code)
FK:{SSN} → Forensic expert(SSN) NNV:{ResultData, Conclusion, SSN}

1.1.9 Photo

Photo (EvidenceNumber: int, Identification code: int, SSN: int)
PK:{EvidenceNumber, Identification code}
FK:{EvidenceNumber, Identification code} → Evidence(EvidenceNumber, Identification code)
FK:{SSN} → Forensic expert(SSN)
NNV:{SSN}

1.1.10 Recording

Recording (EvidenceNumber: int, Identification code: int) PK:{EvidenceNumber, Identification code}
FK: {EvidenceNumber, Identification code} → Evidence(EvidenceNumber, Identification code)

1.1.11 Fingerprint

Fingerprint (EvidenceNumber: int, Identification code: int, Code: int)
PK:{EvidenceNumber, Identification code}
UNI:{Code}
FK: {EvidenceNumber, Identification code} → Evidence(EvidenceNumber, Identification code)
FK:{Code} → Person(Code)

1.1.12 Other

Other (EvidenceNumber: int, Identification code: int) PK:{EvidenceNumber, Identification code}
FK: {EvidenceNumber, Identification code} → Evidence(EvidenceNumber, Identification code)

1.1.13 Interrogation

Interrogation(Date: date, Statement: char, Identification code: int, Code: int, SSN: int)
PK: {Identification code, Code}
NNV: {Date, Statement, SSN} FK: {Identification code} → Crime(Identification code)
FK: {Code} → Person(Code)
FK: {SSN} → Investigator(SSN)

1.1.14 Trial

Trial(TrialID: int, Date: date, Outcome: char, Identification code: int, SSN: int)
PK: {TrialID}
NNV: {Date, Outcome, Identification code, SSN}
UNI: {Identification code}
FK: {Identification code} → Crime(Identification code)
FK: {SSN} → Judge(SSN)

1.1.15 Judge

Judge(SSN: int, Judge number: int, Name: char, E-mail address: char, StartDate: date)
PK: {SSN}
UNI: {Judge number}
NNV: {Name, Startdate}
UNI: {E-mail address}

1.1.16 Crime

Crime(Identification_code:int,type:char,Date of Registration:date,Description:char,Longitu

PK: {Identification_code}
NNV: {DateOfRegistration,type,Description,Longitude,Latitude}

1.1.17 Employee

```
Employee(SSN:int,passportnumber:int,employee_number:int,name:char,telephone:int,type:char)

PK: {SSN}
NNV: {name}
UNI: {passport_number,employee_number}
```

1.1.18 Investigator

```
Investigator(SSN:int)
PK: {SSN}
FK: {SSN} → Employee(SSN)
```

1.1.19 Forensic Expert

```
Forensic Expert(SSN:int)
PK: {SSN}
FK: {SSN} → Employee(SSN)
```

1.2 List of integrity constraints

1. The specialization of "Employee" is Partial, Disjoint.
2. The specialization of "Evidence" is Total, Disjoint.
3. If a person is involved in a crime, this person can only take one role.
4. A person cannot be related to himself/herself.

Chapter 2

Justification of normalization

2.1 1st NF

We justify that our relational schema is in the 1st Normal Form, as none of the relations contains attributes that store sets, lists, or records of data. This is evident, for example, in the 'Person' relation where the attribute 'Address', present in the conceptual and logical design, has been divided into the attributes: 'Street', 'ST.number' and 'Postalcode' in the physical design. For attributes like 'Telephone number' in the 'Employee' relation or 'e-mail' in the 'Judge' relation, it was decided that an employee could only have one recorded telephone number and similarly that a judge could only have one e-mail. This was chosen to avoid lists of recorded telephone numbers and e-mails, thereby ensuring the normalization of the 1st form. This was chosen for the sake of simplicity, although this, to a lesser extent, reflects reality, in which it is common to have multiple telephone numbers and multiple e-mails. If it for example had been decided that multiple telephone numbers should be stored for an employee, a new relation would have been added with the combination of the employee ID and the telephone number as the primary key. This would ensure normalization of the 1st form, while also reducing redundancy in the database.

2.2 2nd NF

The 2nd Normal Form is reached when two conditions are met. Firstly, the relational schema has to be in 1st NF, which we have justified in the previous section. Secondly, the relational schema should not contain any partial dependencies. To ensure that all non-prime attributes have a full functional dependency to the primary key of the given relation, we identify the relations where we find a many-to-many relationship. This is

found for example in the relationships between 'Crime' and 'Person', in this case there are multiple roles a person can take in a crime, but common for all roles is that there is a many-to-many relationship between the two relations. For all the many-to-many relationships we need to create a new relation 'R'. The new relation will have a foreign key to each of the primary keys of 'Crime' and 'Person'. The two foreign keys will then create a composite primary key for 'R'. For all relations with composite primary keys we consider if there are any no primary attributes that is dependant on only one of the attributes that is composing the primary key. We Justify that the relational schema reaches 2nd NF as it contains no partial dependencies.

2.3 3rd NF

Lastly we justify that the relational schema is in the 3rd Normal Form, as all the relations have attributes that are functionally depend on the primary key. This means that there are no transitive dependencies, as there are no non-prime attributes dependant on another non-prime attribute.

Chapter 3

Oracle relational schema created with the Data Modeler-tool

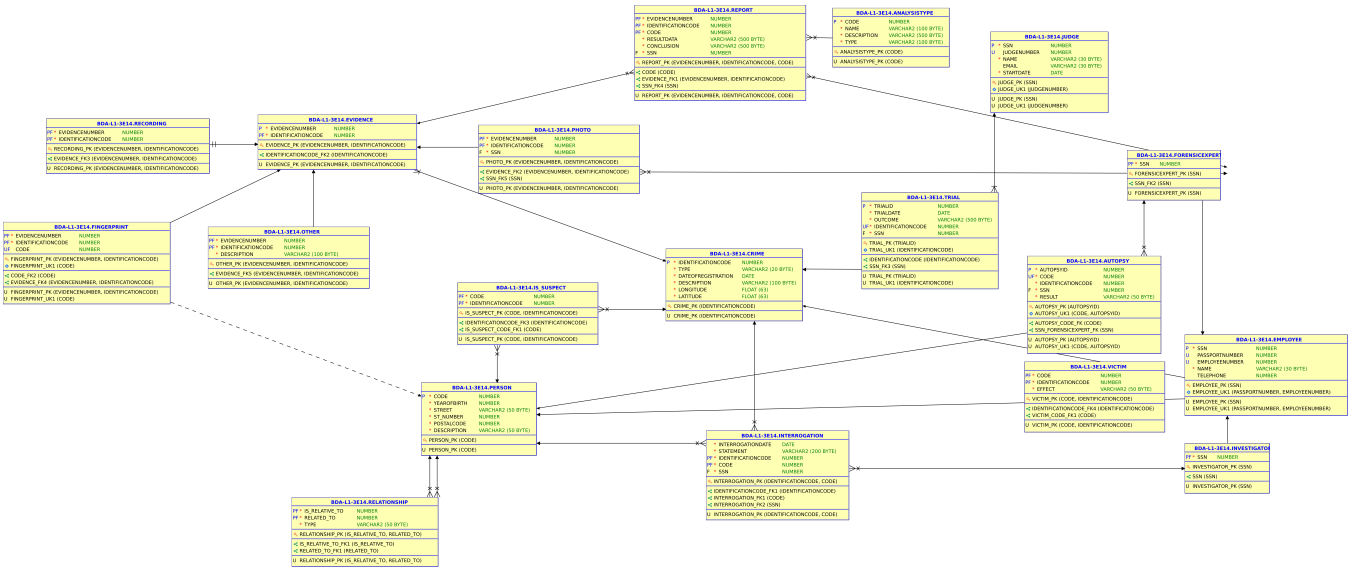

Figure 3.1: The obtained relational schema.

Chapter 4

Description of the data loading process

4.1 Login credentials for the Oracle server

```
Usario:  BDA-L1-3E14
Contraseña:  idontknow
```

4.2 Data loading process

For loading the data into our beforehand created database, we used the provided GUI of Oracle SQL Developer since it provides an easy functionality to create transactions to correctly insert data into our tables even if they have a 1:1 relation. If we would have written the SQL statements ourselves, we would have to create transactions since 1:1 relations are problematic.

Bibliography