DATABASES / DATABASES I

2024/2025 – Fall Semester

Practical Classes Workbook (Part 3)

Introduction to the SQL Language

TABLE OF CONTENTS

3. Introduction to the SQL Language

SQL has two languages:

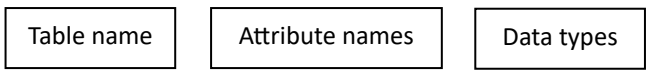
- **DDL** (Data Definition Language), which includes the CREATE, ALTER and DROP commands, used to create the structure of a database.
- **DML** (Data Manipulation Language), used for querying, inserting, updating and deleting data from a database.

Data Definition Language

Create

We have already seen how to create a database using the "Create" statement, now let's see how to use it to create a table. The base syntax is the following:

```
CREATE TABLE PEOPLE(
                    PersonID NUMERIC,
                    Name VARCHAR(30),
                    Email VARCHAR(30),
                    DateOfBirth DATE
                    );
```



It is possible to add other specifications in the CREATE TABLE statement:

1. **Primary keys** (single or compound)

```
CREATE TABLE People (
    PersonID NUMERIC PRIMARY KEY,
    Name   VARCHAR(30));


CREATE TABLE Invoices (
    InvoiceID NUMERIC,
    InvoiceLineID INT,
    CONSTRAINT pk_invoices
          PRIMARY KEY (InvoiceID, InvoiceLineID));
```

2. **Mandatory attributes**
3. **Alternate keys** –> can also be simple or compound, are unique like primary keys, but can assume null values.

```
CREATE TABLE People (
    PersonID NUMERIC PRIMARY KEY,
    NIF    INT UNIQUE);


CREATE TABLE Parking_Slots (
    SlotID NUMERIC PRIMARY KEY,
    Color VARCHAR(20) NOT NULL,
    Number INT NOT NULL,
    CONSTRAINT slot_color_nr UNIQUE (Color, Number));
```

Returning to the previous example discussed in Data Modeling -> Relational Data Model -> Conversion from Conceptual to Relational Model -> One to One Relationship Model.

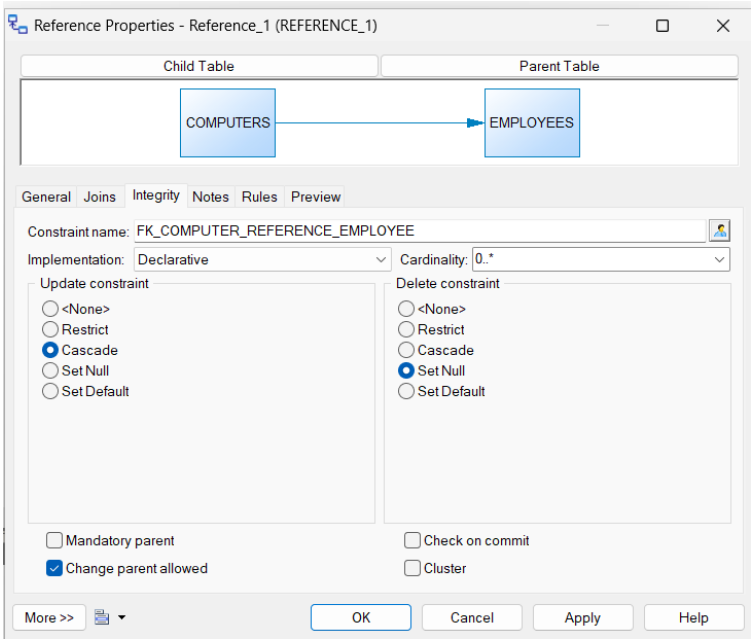
The script for the table will be something like this:

```
CREATE TABLE COMPUTERS (
    IDComputer NUMERIC PRIMARY KEY,
    IDEmployee   NUMERIC UNIQUE,
    FOREIGN KEY (IDEmployee) REFERENCES EMPLOYEES(IDEmployee)
        ON DELETE SET NULL -- When an employee is deleted, the reference is set to NULL
        ON UPDATE CASCADE -- When the IDEmployee in EMPLOYEES is updated, the reference is updated
        );
```

ON DELETE SET NULL ensures that if an employee is deleted, the IDEmployee in COMPUTERS is set to NULL, leaving the computer without an owner, i.e, when a parent row (in the referenced table) is deleted, the foreign key value(s) in the child table are set to NULL. This action effectively breaks the link between the child and parent rows, but it does not delete the row in the child table itself.

ON UPDATE CASCADE ensures that when the primary key of an employee in the EMPLOYEES table is updated, the corresponding foreign key value in the COMPUTERS table is automatically updated to reflect the change. This maintains referential integrity by ensuring that the relationship between the employee and their computer remains consistent, without leaving outdated or mismatched foreign key values in the COMPUTERS table.

Steps: Double click on the Reference > Integrity



4. Foreign keys

```
CREATE TABLE Ratings(IDRATING int, IDCRITIC int, IDMOVIE int,
                     STARS int, RATINGDATE date,
                     PRIMARY KEY (IDRATING, IDCRITIC, IDMOVIE),
                     FOREIGN KEY (IDCRITIC) REFERENCES CRITICS(ID),
                     FOREIGN KEY (IDMOVIE) REFERENCES MOVIES(ID)
                     );
```

Drop

The "Drop" is used to delete databases, tables, columns, constraints…

To delete a database the command is simple: `DROP DATABASE databasename;`

A similar syntax is used to delete a table: `DROP TABLE table_name;`

Regarding columns and constraints, an additional command is needed – "Alter table".

Alter

After a table is created, it is possible to update its specifications: update datatypes, add/remove columns, add/remove constraints… To do these operations, the "Alter table" command is used either with "Drop", "Add" or "Alter". Some examples of the use of these statements can be seen below.

```
ALTER TABLE PERSONS
DROP COLUMN Phone;
```

To drop a column

```
ALTER TABLE PERSONS
ALTER COLUMN Phone VARCHAR(13);
```

To change a datatype

```
ALTER TABLE PERSONS
ADD CONSTRAINT pk_persons PRIMARY KEY (IDPeople);
```

To add a primary key constraint

```
ALTER TABLE PERSONS
ADD CONSTRAINT fk_country
      FOREIGN KEY (IDCountry)REFERENCES COUNTRIES (IDCountry);
```

To add a foreign key constraint

```
ALTER TABLE PERSONS
ADD CONSTRAINT gender_constraint
CHECK (Gender IN ('Female', 'Male', 'Other'));
```

To add a check constraint (like standard checks in PD)

```
ALTER TABLE PERSONS
ADD CONSTRAINT unique_email UNIQUE (Email);
```

To add a unique constraint (alternate key)

```
ALTER TABLE PERSONS
DROP CONSTRAINT gender_constraint;
```

To drop a constraint

Exercises

10. Consider a database that is intended to store information on the various cocktails produced at a bar which contain the following tables:

- Cocktails (CocktailName, Price)
- Ingredients (IngredientName, UnitCost, AlcoholPercentage)
- Recipes (Cocktail, Ingredient, Units)

Underlined attributes are the primary keys. The "Cocktail" and "Ingredient" attributes in the Recipes table are foreign keys to the Cocktails and Ingredients tables, respectively. "Units" represents the amount of each ingredient that goes into the composition of a cocktail.

i. Manually create a new script and write the DDL statements to create a new database and the tables described above. Call it "*Cocktails_CreateDB_ex10*".

11. Consider the relations R (W, X, Y, Z) and S (A, B) and the following DDL statements:

```
CREATE TABLE R(W INT, X INT, Y INT, Z INT, PRIMARY KEY(W, X);
CREATE TABLE S(A INT, B INT, PRIMARY KEY(A));
```

i. Write a DDL expression that guarantees there are no repeated values in column Z but it may be NULL. Consider the above statements have already been executed.

ii. Solve the exercise above, but now by updating the first "Create table" statement.

iii. Write a DDL expression that guarantees the column Y is a foreign key that points to S.A. Consider the above statements have already been executed.

iv. Solve the exercise above, but now by updating the first "Create table" statement.

12. Below is the (partial) business model of an air travel booking site that stores information on the various existing flights and bookings made.

- A flight has a departure and a destination airport. Each airport is located in a particular city of a particular country.
- A departure is an execution of a flight that occurs on a particular date and time and is associated with a specific aircraft.
- Each departure may be operated by one or more airlines and each airline operating it will have their own designation for that departure (e.g. flight 356 is operated by TAP and BA assuming the designations TP356 and BA356 respectively). One airline can operate multiple departures.
- Each aircraft has a model and belongs to a particular manufacturer and category. A manufacturer has multiple aircrafts, and one category may also be represented by multiple aircraft.
- Passengers, characterized by name, email and passport, can make reservations for a certain departure through one of the operating airlines by selecting their seat.
- Each aircraft has a maximum number of seats.

i. Draw the entity-relationship model whose functional context is described above. It might be needed to add additional attributes.

ii. Write the DDL statement that would ensure in your model that two different passengers cannot select the same seat on the same departure. Consider that the DDL script to create the database and its objects was already executed.

iii. Write the DDL statement that would ensure in your model that an aircraft is not being used on two departures with the same date and time of departure. Consider that the DDL script to create the database and its objects was already executed.

Data Manipulation Language

The **Select** command is always used when we want to retrieve information from a database. **The result of an SQL query is always a table**.

The syntax of a basic query is the following:

```
SELECT <column list>
FROM <table list>
WHERE <conditions>
```

<table list> - tables in the database or tables resulting from other Select statements.
<column list> - attributes belonging to the table list or expressions built on top of them.
<conditions> - logical expressions

To select all attributes from a table we use:

```
SELECT *
FROM table_name
```

If the Where condition is not applied, then all rows will be selected.

All the SQL theory can be seen in the slides of the theoretical classes, available in Moodle.

Exercises

13. Use the scipt created in exercise 10.i (this script is the solution for exercise 10.i and is available on Materials – "10_*Cocktails_CreateDB.sql*") and execute the SQL script "*Cocktails_data.sql*" (also available on Materials) to insert data in the tables. Write and execute the following queries.
    i. Using an SQL expression, list the names of ingredients that cost less than 3€.
    ii. Using an SQL expression, list the names of ingredients that are not used in any recipe.
    iii. Using an SQL expression, list the names of ingredients that cost less than 3€ or are not used in any recipe.
14. Delete the TVShows database if you already have one and restore it using "*TVShows_*" available on Materials. This will create the TVShows database we have been using but contains data in the tables. Write and execute queries to deliver the requested results. After each exercise there is an image with the expected result.
    i. List the name of the TVShows which name does not start with the word "The".

	TITLE
1	My Own Kind of Series
2	Who are you?

    ii. List all the scenes filmed in studio "123 Studios".

	IDSCENE	IDEPISODE	SCENE_NR	DATEFILMED	SCENARIO	STUDIO	LOCATION	LANDSCAPETYPE
1	1	1	1	2017-03-12 00:00:00.000	Dad's House	123 Studios	NULL	NULL
2	2	1	2	2017-03-12 00:00:00.000	Dad's House	123 Studios	NULL	NULL
3	4	2	1	2017-04-13 00:00:00.000	School	123 Studios	NULL	NULL
4	5	2	2	2017-04-13 00:00:00.000	School	123 Studios	NULL	NULL
5	6	2	3	2017-03-12 00:00:00.000	Dad's House	123 Studios	NULL	NULL

iii. List the IDs of the actors who never participated in a TVShow.

	IDPerson
1	7

iv. List the name and rating of the top 3 TVShows ordered by rating (higher to lower).

	TITLE	RATING
1	The therapist	8,6
2	The cauliflower	8,4
3	The portuguese way	7,1

v. List the name and rating of all TVShows ordered ascendingly by rating. Exclude the TVShows with no rating.

	TITLE	RATING
1	Who are you?	5,3
2	The portuguese way	7,1
3	The cauliflower	8,4
4	The therapist	8,6

vi. How many scenes were filmed on a beach (LandscapeType = "Beach")? Rename the result column to "Beach Scenes".

	Beach Scenes
1	3

vii. List the name of the directors who directed episodes from the TVShow "The cauliflower". Rename the result column to "Director" and show only unique values.

	Director
1	John
2	Pauline

viii. What is the average duration of the episodes in season 1 of "The cauliflower"? Rename the result column to "Avg Duration".

	Avg Duration
1	39

ix. List the name of the actors and how many scenes they participated in (rename this column in the result). Exclude the actors with no scenes. Order the result by the number of scenes they participated in (highest first).

	Name	Total nr Scenes Played
1	Frances	8
2	Anthony	3
3	Beatrice	2

x. List the name of the actors and how many scenes they participated in (rename this column in the result). Include actors with no scenes (their total scenes will be 0).

	Name	Total nr Scenes Played
1	Frances	8
2	Anthony	3
3	Sam	0
4	Beatrice	2

xi. List the names of the actors who never participated in a TVShow.

This is very similar to 6.iii but now we want to return the names and not the IDs of the actors.

	Name
1	Sam

xii. List the name of the technicians who performed more than one function in at least one scene.

	NAME
1	Lucy

15. Delete the MusicFestivals database if you already have it restore it using the "*ex7_MusicFestivals_CreateDB_AddData.sql*" file available on Materials. This is the Music Festivals database created in exercise 8 but contains data. Write and execute queries to deliver the requested results. After each exercise there is an image with the expected result.

i. List the name of the artists who have only played in concerts as the invited artist.

	NAME
1	Milhanas
2	Rapaz Ego

ii. Using an (inner) join, list the names of the artists who have played at least one concert as the main artist.

	NAME
1	A Garota Não
2	Amor Electro
3	Capicua
4	Capitão Fausto
5	Cassete Pirata
6	Dead Combo
7	Fogo fogo
8	Gisela João
9	Luísa Sobral
10	Stereossauro
11	Valter Lobo

iii. Using a left join, list the name of the artists who haven't played in any concerts.

	NAME
1	Salto
2	Ana Lua Caiano

iv. Now, use a right join to deliver the same result as above (in iii). What must change in the query?

v. How many times did each artist perform as the main artist? Consider all concerts registered and list all artists, including the ones with 0 performances as main artist. In the result, show the artist name, ID, and the number of times he/she performed (renamed). Order the results from highest to lowest.

	IDARTIST	NAME	Performances as main artist (overall)
1	1	Capitão Fausto	3
2	2	Capicua	2
3	9	Cassete Pirata	2
4	11	Gisela João	2
5	12	Valter Lobo	1
6	13	Dead Combo	1
7	15	Stereossauro	1
8	7	Luísa Sobral	1
9	3	A Garota Não	1
10	4	Amor Electro	1
11	5	Fogo fogo	1
12	6	Milhanas	0
13	8	Salto	0
14	10	Ana Lua Caia...	0
15	14	Rapaz Ego	0

vi. How many times did each artist perform as the main artist in the festival "Bons Sons" (regardless of the edition number)? Include only artists who have performed at least once at this festival as the main artist. In the result, show the artist name, ID and the number of times he performed (renamed). Order the results from highest to lowest.

	IDARTIST	NAME	Performances as main artist in BS
1	1	Capitão Fausto	3
2	2	Capicua	2
3	3	A Garota Não	1
4	4	Amor Electro	1
5	5	Fogo fogo	1
6	7	Luísa Sobral	1
7	9	Cassete Pirata	1
8	11	Gisela João	1
9	12	Valter Lobo	1
10	13	Dead Combo	1

vii. How many times did each artist perform as the main artist in the festival "Bons Sons" (regardless of the edition number)? Include only artists who have performed more than once at this festival as the main artist. In the result, show the artist name, ID and the number of times he performed (renamed). Order the results from highest to lowest.

	IDARTIST	NAME	Performances as main artist in BS
1	1	Capitão Fausto	3
2	2	Capicua	2

viii. How many times did each artist perform in the festival "Bons Sons" (regardless of the edition number and if they were the main or invited artist)? Include only artists who have performed at least once at this festival. In the result, show the artist name, ID and the number of times he performed (renamed). Order the results from highest to lowest.

	IDARTIST	NAME	Performances at Bons Sons
1	1	Capitão Fausto	3
2	2	Capicua	2
3	11	Gisela João	2
4	14	Rapaz Ego	2
5	12	Valter Lobo	1
6	13	Dead Combo	1
7	3	A Garota Não	1
8	4	Amor Electro	1
9	5	Fogo fogo	1
10	6	Milhanas	1
11	7	Luísa Sobral	1
12	9	Cassete Pirata	1

ix. How many times did each artist perform as the main artist in the festival "Bons Sons" (regardless of the edition nr)? Include all artists, even the ones who never performed at this festival. In the result, show the artist name, ID and the number of times he performed (renamed). Order the results from highest to lowest.

	IDARTIST	NAME	Performances as main artist in BS
1	1	Capitão Fausto	3
2	2	Capicua	2
3	3	A Garota Não	1
4	4	Amor Electro	1
5	5	Fogo fogo	1
6	7	Luísa Sobral	1
7	9	Cassete Pirata	1
8	11	Gisela João	1
9	12	Valter Lobo	1
10	13	Dead Combo	1
11	14	Rapaz Ego	0
12	15	Stereossauro	0
13	10	Ana Lua Caiano	0
14	8	Salto	0
15	6	Milhanas	0

More exercises

16. The following relational model is used in the Transit Authority and contains information about the citizens living in the country (PEOPLE) - identified by their citizen card (CITIZEN_CARD) - and the motor vehicles in circulation (CARS) - identified by their PLATE. The model also contains a relationship with information about the owners of each vehicle (OWNS) - it is assumed that each vehicle has only one owner and each person can have 0 to many vehicles. The model also stores information about parking tickets passed to vehicles (FINES) - each fine has a unique identifier (ID_FINE) and, among other attributes, an attribute that saves the postcode where the fine was issued (POSTCODE_FINE).

    PEOPLE (CITIZEN_CARD, NAME, PHONE, ADDRESS, POSTCODE)

    CARS (PLATE, COLOR, BRAND, MODEL, YEAR_PRODUCTION)

    OWNS (PLATE, CITIZEN_CARD)

    FINES (ID_FINE, PLATE, DATE, POSTCODE_FINE)

i. Obtain the total number of fines in 2014 for each of the postcodes of the fine.

ii. Obtain the citizen card number and name for all persons who have received the maximum number of fines passed to one person. The result can contain more than one person.

iii. Obtain a listing of the names and postcodes of residence of all persons who have been fined in their area of residence (i.e. in their postcode of residence) and who have not been fined in areas outside their area of residence.

17. Consider a database of an image tagging site:

    Members (MemberID, Name, Age)

    Images (ImageID, year)

    Tags (MemberID, ImageID)

    The **Members** relation contains one record for each member registered on the site; **Images** contains the metadata of the images - ImageID (PK) and year the picture was taken; **Tags** keep

information about users who have been tagged in images. Primary keys are underlined. Non-Key Attributes May Have NULL Values.

i. Write an SQL query that returns the names of all members that were tagged in images from year 2011 and year 2014. Return the result in alphabetical order.

ii. Write an SQL query that returns, for each member, the total number of times that he/she has been tagged in 2015 images. Sort the results alphabetically by name. You should ensure that you include in the results all members, even those not tagged in that year.

18. Consider the following relational schema for an airline database (primary keys are underlined):

    FLIGHTS (FlightNumber, CityOfOrigin, CityOfDestination)

    DEPARTURES (FlightNumber, Date, TypeOfAirplane)

    PASSENGERS (IDPassenger, Name, Address)

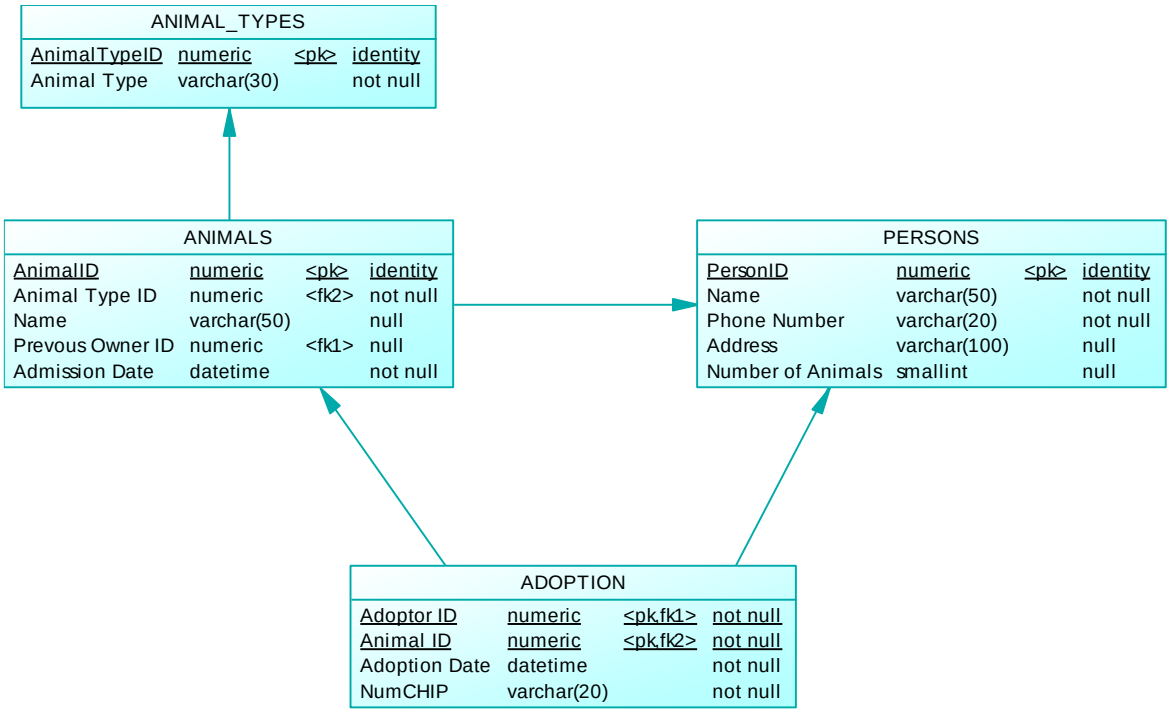
    RESERVATIONS (IDPassenger, FlightNumber, Date, Seat Number)

    Express in SQL the following queries:

i. Return all cities that have direct flights to Madrid and Lisbon.

ii. Show the flight number and the date of all departures for which there are no reservations.

19. In the context of a database for the Society for Animal Protection (SPA) with information regarding the adoption of domestic animals, the following physical model was produced (developed in SAP Power Designer and instantiated to Microsoft SQL Server):



Remarks:

- The ANIMALS table keeps information about the animals registered in the SPA. Each Animal has the attributes Animal ID, Animal Type ID, Name, Previous Owner ID and Admission Date.
- The PERSONS table has the following information: Person ID, Name, Phone Number, Address, and Number of Animals (Number of animals the person currently has).
- ADOPTION table stores: Adoptor ID, Animal ID, Adoption Date, and NumCHIP (Animal Identification CHIP Number).
- The ANIMAL_TYPES table stores information about existing pet types - ID and Animal Type (e.g. cat, dog).

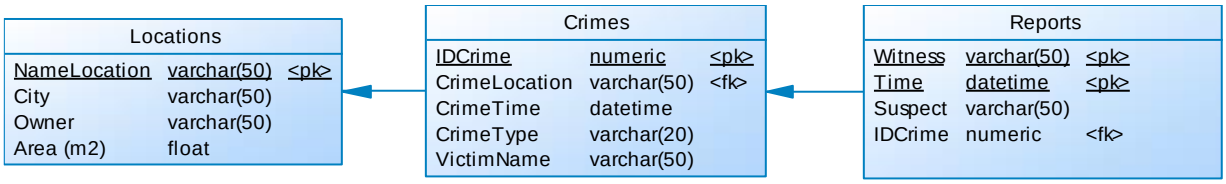
i. Infer the conceptual data model behind this physical data model and design it as an entity-relationship model (use SAP Power Designer symbology).

ii. Write an SQL command that allows you to calculate the total number of cats admitted to the SPA during 2015

iii. Write an SQL command that lists the names of the adopters who adopted at least one animal of each of the existing types.

iv. Write an SQL command that lists the names and addresses of all adopters who have made at least two adoptions.

20. Consider the relational model presented below which is a very simplified version of a database of crimes occurring at several locations (e.g. “Mosteiro dos Jerónimos”, “Torre de Belém”, “Museu da Electricidade”, “City Park”). These locations may be state owned or private. In each location there may be several crimes and for each crime there may be several different reports. A report always has a witness and a suspect associated.



i. Write a SQL query that returns the names of suspected murderers (“murder” is one of the types of crimes) that occurred in places where the owner is the "XPTO" company.

ii. Write a query that returns pairs of names A, B, where A figure as witness indicating B as suspect and B figure as witness indicating A as suspect for the same crime. For example, John may suspect that Peter murdered Alice and at the same time Peter suspects that John is the perpetrator of the same crime.

iii. Write a query that returns for each city and type of crime the total number of crimes (Number of Crimes) and the total number of crimes solved or about to be solved (Number of Crimes Solved): a crime is considered solved or about to be solved whenever there is at least one report of this crime, and all existing reports indicate the same suspect. You may have to write several SQL statements to get this answer by storing the resulting information from the intermediate steps.

iv. Write an SQL expression that returns the day/time, the type of crime and the name of the victim for all crimes committed in Lisbon.

v. Write an SQL expression that returns cities where no crime was committed.