BS DEGREE IN INFORMATICS ENGINEERING Academic year: 2022/2023 - 2nd year, 2nd term Subject: File Structures and Databases First Assignment's Report: Relational Design and Impl.	uc3m | Universidad Carlos III de Madrid

Lecturers:			
Group:	**89**	**Lab User**	
Student:	**Lucía García-Ochoa Agüero**	**NIA:**	**100472088**
Student:	**Alfonso Pineda**	**NIA:**	**100472157**
Student:		**NIA:**	

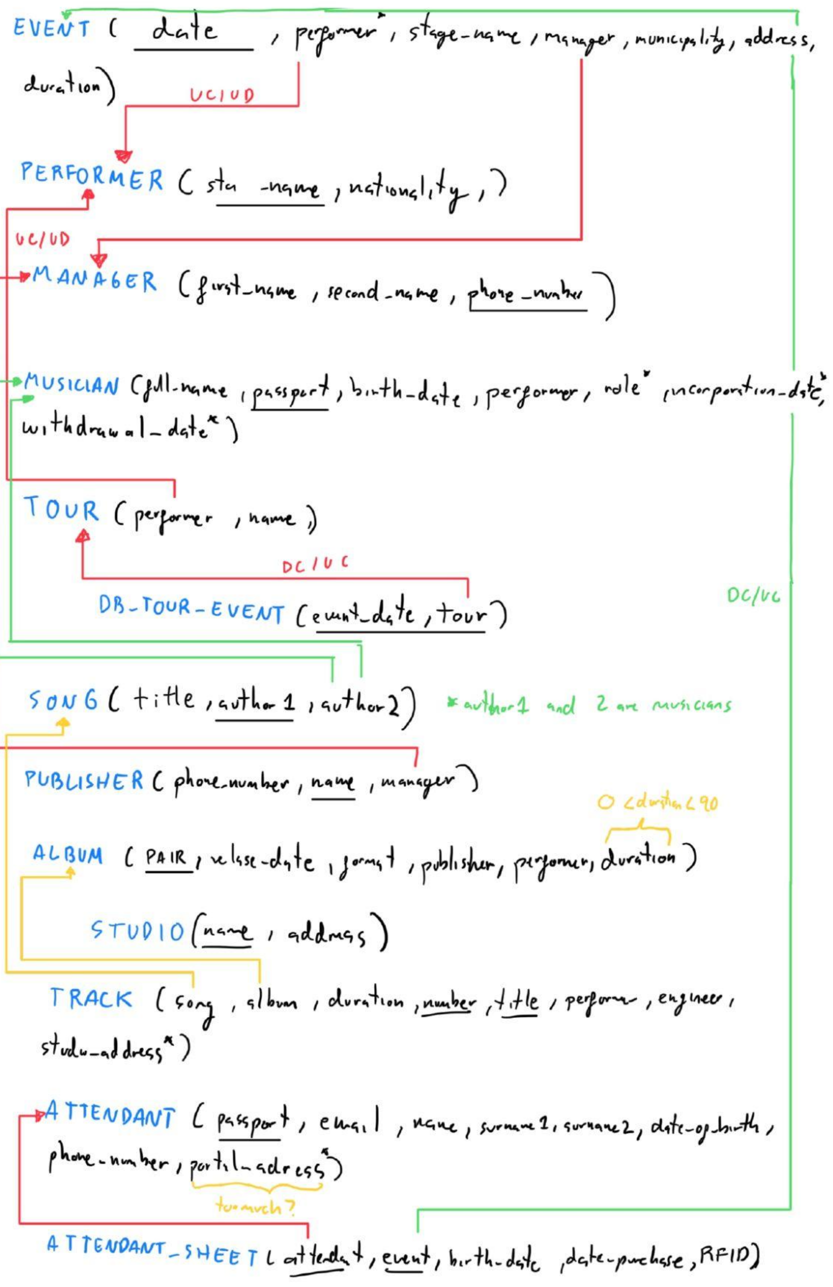
1 Introduction

The first problem consists of designing a relational design of a database based on a given prompt which we show in the next section, after that we were asked to create the tables in SQL/PL, which we did in the creation_script where we created every table keeping the design in mind. The final problem and the hardest one is to adapt a given database, a disastrous one, to ours. We couldn't finish this part because of lack of time, members and mostly bad time management, but we did what we could as good as time gave us a chance to.

2 Relational Design

This section is subdivided into three subsections:

- Relational Schema: the complete design (as it should be, despite the restrictions of the tool that will be used to implement the design). The design must be notated with the *relational graph notation* (as explained in class).


EVENT (date , performer*, stage-name , manager , municipality, address, duration)
UC/UD
PERFORMER (sta -name , nationality ,)
UC/UD
MANAGER (first_name , second_name , phone_number)
MUSICIAN (full-name , passport , birth-date , performer , role* , incorporation-date*, withdrawal-date*)
TOUR (performer , name)
DC/UC
DB-TOUR-EVENT (event_date , tour)
DC/UC
SONG (title , author 1 , author2)
* author1 and 2 are musicians
PUBLISHER (phone-number , name , manager)
0 < duration < 90
ALBUM (PAIR , release-date , format , publisher, performer, duration)
STUDIO (name , address)
TRACK (song , album , duration , number , title , performer , engineer , studio-address*)
ATTENDANT (passport , email , name , surname 1 , surname 2 , date-of-birth , phone-number , postal-adress*)
too much ?
ATTENDANT_SHEET (attendant , event , birth-date , date-purchase , RFID)

I know there are more semantics and I got more written but I don't have more time to write them nor the required organization in them to do it.

- Implicit semantics: semantic presuppositions that are not found in the explicit description, but which are required to complete the relational design. Again, a tabular format is recommended for describing these presuppositions, such as the following:

Presp_id	Stage	Mechanism	Description
I_1	Design	Primary key	Performers are identified by their stage name
I_2	Design	Primary key	Managers are identified by their phone number because their name could be repeated
I_3	Design	Primary key	Tour should be identified by the performer and the name of the tour (in the design it doesn't but the creation of the tables it does)
I_4	Design	Primary key	Songs are identified by their name and first author (in the design it doesn't but the creation of the tables it does)

Table 1: Implicit semantics incorporated into the relational graph

- Non-observed explicit semantics: each of the explicit presuppositions (stated in the problem description) that could not be included in the relational graph, will be identified (with a label, such as S1, S2, …) and described in this section. Tabular format is recommended, as shown next:

Presp_id	Description
S_1	Phone numbers have 9 digits (at least, at most)
S_2	Duration of Album can't be more than 90 mins (duration < 90) should not be in the design
S_3	The attendant can't have less than 18 years old

Table 2: Non-observed explicit semantics

3 Relational Statics Implementation in SQL (DDL)

This section must include the creation of each table. In addition to the code (*NEWcreation.sql* script) for creating tables (valid syntax in PL/SQL), you should include the correspondent subsections referring to the excluded semantics that are re-incorporated, the newly incorporated implicit semantics, and the explicit semantics that were observed but are now excluded. All these sections will be accomplishing by fulfilling the correspondent table (see tables 3, 4 and 5). Any of these tables is empty (in case), the table should be omitted and replaced by a phrase such as "Has not been reported."

This part of the report cannot be filled at all because we couldn't populate the database properly (we could only populate one table), therefore we don't have a NEWcreation.sql

Re-incorporated semantics: (identifiers referred to those assigned in table 1)

Presp_id	**Solution Description**
S_1	field size is 9; a constraint (*constraint_name*) CHECK (phone≥100000000) is added to the table <*table_name*>
…	…

4 Workload (DML)

This section will describe the uploading of the workload (*NEWload.sql* script) from the tables provided (and described in the statement). To this end, we will analyze the problem of populating the tables with the workload. The solution will be described, with emphasis on:

- The specific order of tables to dump data into them (reasoned).
- The problems that arise (obligatory field value, inconsistencies in the original data, etc...) and the solutions adopted to overcome them.

1. insert into performer(stage_name, nationality) select distinct band, band_nation from fsdb.artists join fsdb.recordings on (band = performer);
    a. This line is to insert select into the performers table all the bands and soloists of the database, we verified that there are no extra performers in fsdb.livesingings. If there were we should insert select again but with an exclusion clause with the existing performers in the table.
2. insert into musician(passport, full_name, birth_date, performer, role_, incorporation_date, withdrawal_date) select distinct passport, musician, birthdate, band, role, start_date, end_date from fsdb.artists join fsdb.recordings on (band = performer) where passport is not null;
    a. The purpose of this line is to insert select those artists previously inserted in the lane before, but we got a problem because there are repeated passports so we should sort the selection so there was no repetition but I don't have more time