LSSL

La Salle Super League

2023-2024

laSalle

UNIVERSITAT RAMON LLULL

INTRODUCTION

La Salle Super League (LSSL) is a start-up company that wants to manage future team sports competitions. To do this they want to collect all the data that is generated around sports, specifically football. They ask database students to design and implement a database according to the requirements and functionalities described in this document.

The following set of requirements has been divided into four sections, therefore the project will be developed in teams of four people in order to distribute the work equally among the members. The sections will be strongly interrelated, and it is therefore vital that students maintain good communication and extensive knowledge of the work carried out by their peers. If this is not the case, the project will not be able to be carried out, as multiple incompatibilities may arise when pooling the work of each of the members, which will ultimately lead to costly additional work.

The project is divided into three stages.

STAGE 1: DESIGN OF THE LSSL DATABASE

The following section describes in detail the requirements of the LSSL database. The database design (conceptual model and relational model) should start with the information included in this section.

1) Competitions

The **LSSL** wants to manage the main women's and men's football competitions. The competitions will involve clubs in which each season they will compete against each other to see who is the champion (the one ranked first). The clubs are the protagonists of the sports show, and the database must reflect in detail the information related to each club, including name, players, coach, and everything essential for an efficient management of the football competitions. Players are part of a club's squad for a certain period of time with a specific squad number that may vary throughout the player's time at the club. The coach is also part of the club for a period of time. For both the player and the coach we want to store the first name, surname, date of birth, city of birth, country of birth and gender. In the case of players, we also need to save their preferred position, and in the case of coaches, their preferred tactics and training style.

Clubs may be affiliated to other clubs. For example, a junior club can be affiliated to a main club or clubs from women's and men's competitions can be affiliated to each other. Clubs are established in one city, and unlike other types of sporting competitions, clubs do not change cities. The clubs have facilities where they play matches and have shops where they sell merchandising products.

The most important element of the competitions are the matches in which two teams play each other, one as away team and one as home team. The database should comprehensively capture information relating to each match, including details such as date, time, stadium, results, scores, cards, goals and other key statistics. The coaches involved in the match and all players from both clubs must be recorded. For each player who is involved in the match from the start or when a change is made during the match, we want to know how many goals, shots, and assists he/she has made, how many yellow and red cards he/she has received, and in which position he/she has played. The matches are officiated by referees in a neutral manner enforcing the rules. From the referees we want to know their first and last name, where and when they were born, their gender and what certification they have. Referees, when judging a match, can do so in different roles (head referee, linesman, fourth official, var).

2) Merchandising

One of the major sources of income for clubs is the sale of their products, such as shirts, accessories, and other related items, which are characterised by attributes such as name, cost, officiality, size, special feature, and the club to which the product is related. From the products it is known what type they are and for which activity they are mainly designed. The product and activity types have descriptive texts so that the buyer knows what he/she is buying.

Products can be categorised according to whether they are clothing, footwear, or accessories. Each one has its own characteristics, for example, in the case of footwear we must keep its use, properties, and types of closure. For clothing, the most suitable season to wear it and the material; and for accessories we must keep the most relevant characteristic.

The products are stored and displayed in shops that may or may not belong to a club. The shops have a very attractive description in order to attract customers. It is known how many units of the products exist in the shops. All sales made in the shops are recorded, the date and time of the sale, how many products were sold, the discount applied and the credit card used for the sale are known. For the credit card, the expiry date and the provider are known.

The shops are managed by shopkeepers, who are responsible for coordinating daily operations, maintaining stock control, managing sales, and ensuring that each product is properly labelled. From the shopkeepers we want to know their personal details (name, surname, city where they live...), number of days off, and the shift and role they have in the shop where they work. On match days some shopkeepers move temporarily to the stadium shops and may have a different role and shift than in their home shops.

3) Communication

The **LSSL** recognises the importance of communication in connecting with fans and publicising football competitions. Information management on media, TV channels, radios,

social networks and programmes is essential to maintain a vibrant and efficient media presence.

Communication focuses on media groups and the LSSL social network. Media groups are characterised by a name, description, the name of the headquarters, and the city where the headquarters is physically located. Media groups can have radio stations and TV channels. For radio stations we need to keep the name, description, and broadcasting frequency. For TV channels, on the other hand, the title, and the broadcasting quality (SD, HD, FHD...).

Radio stations have programmes all day long, of which we need to save the name, description, schedule, and whether or not they have an associated podcast. Television channels also have programmes that are characterised by their name, description, name of the production company and broadcasting schedule. The programmes are hosted by communicators who are people who, apart from the typical data, have to keep their specialisation. Communicators participate in different programmes with a certain role, e.g. a person can be the director of one programme and a commentator in another.

Communication is the central axis of the LSSL social network. This network is made up of users where we must store their nickname, date of creation of the user, whether they are a verified user or not, and the city in which they live. Users make posts on the social network consisting of a text with the date of creation of the post. Also, each post has the number of likes and the number of reposts. Posts can have replies, which are posts associated with a specific post. Posts can have images associated with them, which are displayed next to the text of the post. For images, we must save the title and the path where the file is physically stored. Users can enter a series of tags when making a post by typing hash before the tag name. For each tag we must know the name, the description, and the status of the trend (initial, viral, controversy...).

Radio and television programmes have a user on this social network to disseminate the content of their programmes. The communicators who participate in the programmes also have a user on the LSSL network.

4) Advertising

Another source of income for clubs and the LSSL is advertising. Customers play a key role in this context. Each customer who engages in advertising activities must be identified with a profile that includes information such as customer ID, name, basic address, and available budget. Customers contract campaigns to advertise their products or services. Each advertising campaign should be characterised by a unique identifier, the name, objective of the campaign, product(s) being advertised, the start and end dates, as well as the allocated budget and the target audience of the campaign. This information allows to manage and track the effectiveness of each advertising campaign. It should be noted that the products advertised may be the products of the LSSL clubs.

Audiences define the audience segments targeted by the campaigns. The demographic and interest details associated with each audience are essential for targeting advertising

strategies and optimising their results. The data that should be stored for each audience is age range and primary interest.

Advertisements (ad) represent the specific advertising content that will be delivered during a campaign. Each ad has a unique ID, title, description, format, status, and date of creation. In addition, they may be associated with specific categories that define their content or purpose. Categories group the ads according to common characteristics. Each category can have an identifier, a name and a description, providing an organisational structure for your advertisements and making them easier to manage. Categories can have subcategories, and subcategories can have subcategories.

Advertisements are disseminated through placements. Each placement has a unique ID, a date, a start and end time, a cost and whether it is an exclusive location or not. Placements are of a particular type, each type of placement has its own description. Placement can be located in shops, stadiums, radio programmes, television and even LSSL network publications.

STAGE 2: PHYSICAL MODEL IMPLEMENTATION

The following section describes in detail the requirements for the implementation (physical model) of the LSSL database. The physical model of the database shall be based on the relational model developed in stage 1.

The physical model shall be implemented in the ORACLE database.

1) Selection of data types

The first step in the implementation of the physical model is to choose the data types for the columns based on the relational model. For this, it is necessary to select for each column the most suitable data type based on the requirements of step 1. Each team member will propose the types (varchar, date, integer...) of the attributes of each table of the module (which he/she supervises). The whole team will work together to agree on a final list of types for all table attributes. The justification for the selection of the data types will be described in the report to be delivered.

2) Coding of the physical model

Based on the relational model and the data type selection, each team member will program an SQL script to create the tables of the model he/she supervises. The result of this step is a single text file with .sql extension that will contain all the creation scripts for all the tables in the database.

The script must be programmed in such a way that it can be executed several times **without errors** and the physical model **will be created automatically each time the script is executed.**

3) Importing the database

The aim of this step is to import data into the database. Therefore, there are several CSV files to be imported into the database. Each file will be imported by the team member who is in charge of the module to which the file belongs.

The name and content of the CSV files cannot be modified in any way.

Once the CSV files are imported, the team will generate data for those tables that are empty after importing the CSV files.

The CSV files to be imported are located in the Project section of the eStudy.

STAGE 3: QUERIES AND TRIGGERS

In this section we will implement queries that will obtain important information that can be used within the application that manages the **LSSL**. It will also be necessary to develop triggers to automate some procedures that need to take place in the database mainly to guarantee the consistency of its data.

The queries proposed in this section can be solved in various ways with different degrees of optimisation, depending on the design of the physical model and its implementation, as well as the design decisions made in the first place. However, there are better ways to obtain the same results than others, so your goal in solving them is to answer what the statement asks for by trying to **reduce the complexity of query resolution and the additional tools used**, such as unnecessary subqueries, incorrectly formed JOINs or extra functions.

Some of the triggers can be solved without the creation of any auxiliary table, but some of them have the main purpose of storing the information of certain activities carried out by the users of **La Salle Super League**. For this reason, the *WarningsList* table can be created with the following command:

```
CREATE TABLE WarningsList (
    affected_table VARCHAR(30) NOT NULL,
    error_message VARCHAR(255) NOT NULL,
    id_reference VARCHAR(255) NOT NULL,
    date_warning DATE NOT NULL,
    user_warning VARCHAR(255) NOT NULL
);
```

To each section we indicate about the data format to insert into the table, and in which cases we have to work with them, so remember that it is not a requirement to complete all the triggers but a dynamic specification.

The following section describes in detail the five sets of queries and triggers for the database. The queries and triggers must be executed on the database implemented in step 2. For each query, the solution must be provided, as well as an explanation of its design and a textual validation to ensure that the query answers the question. To validate the queries, the team can implement supporting queries to demonstrate that the proposed queries are a solution.

1) Competitions

Requests for competitions.

1. Find the clubs (id, name) of the city of Girona and the matches they have played and won as a local club giving the goals they have scored as a local club and the goals scored by the visiting team. Sort the result by the date of the matches.

2. Find the players of the clubs of the city of Barcelona (identifier, first name and surname concatenated), the name of their club who have left the club during the year 2022 and the date of dissolution of the contract with that club. Sort the results according to the contract end date and the player's name.

3. Shows the first and last name (concatenated) of the coaches, the name of the club they have coached, the start date and the end date they have been coaching at the club if they have been coaching at the club for more days than the average number of days of all coaches of all clubs.

4. Find the first and last names of referees with a FIFA certification and referees with a UEFA certification who have been involved in a match with a VAR-related position and that in the match the possession of the home team was the same as the possession of the away team. It is necessary to give the result ordered by the first and last name of the referees.

5. Find the number of matches of each competition and gender that is more than 500 matches and give the result ordered from highest to lowest by number of matches.

6. Find the second division clubs that have participated in the most competitions, indicating the number of competitions, in which they have always finished in the top half of the ranking table. Sort the result by the number of competitions descending and by the name of the club ascending.

7. We want to validate the consistency of the information of a match related to the ball possession of the home and away team, so that the sum of the two always adds up to 100. In case any of these two values of an existing match is modified and the information is inconsistent (does not add up to 100), we must add a row in the

WarningsList table indicating the affected table, a text related to the error, the identifier of the match, the current date and the user connected to the database. The relative error text must be: *home_possesion+away_possession is not equal to 100.*

2) Merchandising

Merchandising requests.

1. We are asked for the different number of products of clothing type 'Short' for each product size, sorting the result in descending order of the number of products for each size.

2. Give the name and cost of the products of size XL with a cost higher than 350 euros and the products of the activity 'Casual Wear' with a cost higher than 110 euros and the products that are accessories with *UV Protection* feature and cost higher than 55 euros. Sort the result by product name.

3. Give the identifier, name, description, and number of strokes used for the five most used activities in the products, ordering the result from highest to lowest by the number of strokes the products use for each activity.

4. Find the 10 credit cards that have made purchases with the highest total amount provided by VISA, ordering the result by the amount of purchases from highest value to lowest.

5. Find the shops located in Barcelona (name and description) that have a stock of products at least five times greater than the stock of products of the shop that has the least of all the shops in all the cities, ordering the result obtained by the total stock of the shop from highest to lowest.

6. Give the code, name and cost of the products costing more than 350 euros that have never been sold, sorting the result by product name.

7. When a sale of a product is made in a shop, a trigger will update the stock of this product in the shop where the sale has been made, the remaining stock of the product will be the number of units that have been sold in the sale made. If it is the case that there is not enough stock when the sale is made, the stock must be left with a value of 0 and an error message must be added to the *WarningsList* table.

3) Communication

Communication requests.

1. Find the total number of times each hashtag ending with "CF" has been used, taking into account only posts that have more likes than the average of all posts. Show the

hashtag and number of times that have been used in the posts sorted by the number of posts.

2. We are asked for the total number of TV channels in each city (if their name ends with "ona"). Show the name of the cities and the total number of channels, sorting the result by the name of the city.

3. Find the text and date of the post that has had the most responses.

4. Search for the 10 most used hashtags in the social media posts of registered users during the current year who have used at least one image in 'tiff' format. The result should be sorted by the number of times the hashtag has been used.

5. We are asked to make a ranking of the most influential users on social networks in a given year. We want to know the number of posts of users who have a post among the 500 posts with the most reposts on the network. Show the full name of the user, the total number of posts and the year. It is necessary to give the data sorted by year.

6. Displays the programme name, the programme type (TV or Radio) and the name length of the programmes whose name length is greater than the average length of the names of their programme type. Sorts the result by the length of the programme name and displays only the first 10.

7. In order to monitor the system for the publication of posts, we are asked to implement a mechanism that detects when a user adds a post with less than 100 related *likes*. With this situation, it will be necessary to add the issue to the *WarningsList* table.

4) Advertising

Advertising requests.

1. Search the code and name of the campaigns of the year 2023 and their number of advertisements of those campaigns that the client is from the city of Madrid, ordering the result, from lowest to highest, by the number of advertisements of each campaign.

2. Search for the name of the active campaigns that have ads of a category whose name appears in a description of any ad.

3. Searches for the campaigns, whose code starts with 4, that have the clients with the largest budget. Displays the code, the total budget of the clients and the number of clients, if a client has a budget less than the average budget it will not be added to the total budget and will not be counted as a client of that campaign.

4. Find the name of the placement type and the total cost spent on ads that were in a campaign where the starting age of the audience is higher than the average starting age of the different audience types.

5. Find the id and name of the ads, which contain the word 'Friday' in their title, and have more customers from Girona than from other cities.

6. From the ads that have fewer placement than twice the average of all ads, find the id and title of the 5 ads that have more placements. It is necessary to order the result by the number of placements and the id of the ads.

7. If you change the client of an ad campaign that has more than five ads, you need to add an alert message to the *WarningsList* table like this: 'Updated client name from campaign with more than 5'.

5) Cross requests

Cross-requests.

1. Search for campaigns with the most products. Shows the name of the campaign and its number of products ordered by the number of products in descending order.

2. Search for the name of the shopkeepers and the name of their shop that is located in the city of Girona, if the name of the shop starts with G and the days of holidays of the shopkeepers are higher than the average number of days of holidays of all the shopkeepers, sorting the result in descending order by the number of days of holidays of the shopkeepers.

3. Find the name, city and country of the clubs that only have a single official product of the type "Polo Shirt", giving the result ordered by the name of the clubs from lowest to highest.

4. Find the name of the five campaigns that have used the most products and the name of the five campaigns that have used the least products, indicating also the number of products for each campaign and giving the result ordered by the number of products from the largest to the smallest.

5. Find the identifier of the ad placements that do not have any associated post, giving the result obtained ordered from largest to smallest by the identifier of the placements. Two different solutions are requested.

6. Find the number of people for each of its five existing types.

CONSIDERATIONS

This section describes the methodology to be followed by the database development team for a successful project.

Entity-relationship (conceptual) model design

The objective is to understand the requirements and represent them in the form of a **normalised entity-relationship** model following the syntax shown in the Database classes.

Each team member must read the requirements and the team must carry out a joint reflection to identify the main entities and relationships.

From that reflection, each database module will be distributed to each team member, who will be responsible for the module. Each team member will be responsible for at least one module.

Each team member will define part of the entity-relationship model taking into account the multiple relationships between the modules. The whole team will work together to agree on a final model. Therefore, all team members will know and understand in detail the final entity-relationship model.

The model must be created using a MIRO application following the syntax shown in the classes. Therefore, the delivered version cannot be created by hand.

Relational model design

The objective is to generate a **relational model** from the entity-relational model developed in the previous section.

Each team member will translate the module (for which he/she is responsible) into a relational model. The whole team will work together to agree on a final relational model that resolves the multiple relationships between the modules. **The final relational model must comply with up to the third normal form**. Therefore, all team members will know and understand the final relational model in detail.

The model must be created using a MIRO application following the syntax shown in the classes. Therefore, the delivered version cannot be created by hand.

Documentation

A template for reporting the project has been provided in Word format and all sections of the template are mandatory and must be filled in. It is necessary to insert the models in the document and explain the decisions taken in each module.

DELIVERY INSTRUCTIONS

The project (all three stages) must be developed by a team of four members. All team members must be from the same class group. The project documentation must be submitted **by 31 May 2023 at 23:59.**

The name of the submitted file will follow the following pattern: Group#.zip where # is the group number that will be assigned by the teachers. The ZIP file will contain these two folders:

- **Report**. This folder must include the report written by the team following the given template (you will find it in eStudy).
- **Models**. This folder will include the high-resolution images (png, jpg, pdf...) of the entity-relationship and relational models in order to enlarge them.
- **Scripts**. Two scripts will be included in this folder;
    - A script to create the physical model (e.g.: G1_physicalmodel.sql)
    - A script to import the CSV data (ex: G1_importation.sql)
    - A script with the queries and triggers (ex (G1_queries.sql). This script should follow the given template.

Considerations

- **If a file is missing, the submission will not be accepted.**
- **If the report template is not strictly followed, the submission will not be accepted.**
- **If a copy is detected, all members of the teams involved will fail the course, both those who have copied and those who have been copied. In addition, they will have to consult the university regulations to take into account other consequences.**
- **If AI tools are used in the resolution of the project, the report must include an explanation of what the AI was used for and what indications were given to obtain the results (screenshots of the process and interaction followed must be included). This is not a violation of academic honesty policies.**