CS4416 Project: 20 marks = 20%

This is a group project. A group must consist of either 4 or 5 students. Consider the relational database schema in **concerts_schema.sql**.

This is the schema of a relational database that contains data about artists (incl. bands), their songs, albums and concerts. It also contains data about fans and tickets they have purchased for concerts.

Assume that a **song_id** identifies a particular recording/version of a song. Note that the same artists may record the same song multiple times, and each recording would have a different **song_id**. Note also that the same **song_id** may appear on multiple albums.

Tasks 1

Propose a modified schema to eliminate data redundancy and support:

- multiple artists per album, song, and concert
- multiple favourite artists per fan
- shared tickets, i.e. multiple fans per **ticket_id**

Write all CREATE TABLE statements for the modified schema, including PRIMARY KEY and FOREIGN KEY constraints, and save them in **modified_concerts_schema.sql**.

In a separate file named **data.sql**, provide INSERT INTO statements for inserting some data into the tables of the modified schema. Please provide data for at least two artists, at least two albums, at least 2 concerts ad at least 2 fans who have purchased tickets for concerts.

Work together as a team on Task 1. You can then divide Tasks 2-6 among group members. Tasks 2-6 must be performed on the modified schema in **modified_concerts_schema.sql**.

Task 2

Draw an entity-relationship diagram (ERD) for the modified schema using crow's foot notation.

Task 3

Create a view for the modified schema. The view must use a join and both GROUP BY and HAVING clauses. Aim at demonstrating advanced SQL skills.

Task 4

Write one BEFORE and one AFTER trigger for the modified schema.

Task 5

Write a stored function that returns the total number of occupied seats for a given **concert_id**.

Task 6

Write a stored procedure that checks whether a given **song_id** is associated with a given **album_id**. If not, insert this association into the database. Additionally, adjust the song's release date if it is later than the album's.

Task 7: Report

Write a report with the following sections and save it as **report.pdf**.

Sections:

1. Contribution breakdown of each group member. If not provided, all members will receive equal marks. Significant differences in contributions will affect individual marks.
2. Description of the platform used (e.g., XAMPP on Windows 11). If not provided, the default assumption is XAMPP on Windows 11.
3. Description of the modifications made to the original schema and the reasons for these changes (Task 1).
4. Your ERD (Task 2) and explanation, including any assumptions.
5. Description of the view and the triggers (Tasks 3 and 4). Justify their usefulness.
6. Any assumptions you have made when writing the function and the procedure (Tasks 5 and 6).
7. Discussion on index requirements for efficient execution of the view, triggers, function, and procedure. Propose indexes and discuss use cases in which the proposed indexes are disadvantageous.

Submission

Submit the four files:

a. **modified_concerts_schema.sql** (Task 1)
b. **data.sql** (Task 1)
c. **code.sql** containing the CREATE statements for your view, triggers, function and procedure (Tasks 3-6)
d. **report.pdf** (Task 7)

as separate attachments on Brightspace **by 6 pm on Friday week 12**. One group member submits on behalf of the group.

Marking

- Task 1 + Report Section 3 — 5 marks
- Task 2 + Report Section 4 — 3 marks
- Task 3 + Report Section 5 — 2.5 marks
- Task 4 + Report Section 5 — 2.5 marks
- Task 5 + Report Section 6 — 2.5 marks
- Task 6 + Report Section 6 — 2.5 marks
- Report Section 7 — 2 marks

Penalties

- -2 marks for incorrect file submission or including unnecessary code.
- -5 marks for late submissions, which will be accepted until the end of Week 13.