

EXAMEN DE FIN D'ÉTUDES SECONDAIRES – Sessions 2024

QUESTIONNAIRE

Date :	23.09.24	Horaire :	08:15 - 11:15	Durée :	180 minutes	
Discipline :	AMINF	Type :	écrit	Section(s) :	CI	
					Numéro du candidat :	

Documents provided

- `Answers.dotx`: to be used to insert your answers in the predefined structure
- `Translation.drawio`: contains the CDM to be translated
- `Database.sql`: MySQL database with a sample data set
- `Functions.pdf`: List of known SQL functions

Preparation and delivery

In your working directory (to be defined by each school), you will find a folder called **EXAMEN_AMINF**. Rename this folder by replacing the current name with your exam code (example notation: **LXY_CI_07**). All your files should be saved in this folder, which will be called **your folder** afterwards!

Then open the file **Answers.dotx** inside your folder, adapt the header by adding your candidate number and the date. Then save the document in **your folder** (example of notation: **LXY_CI_07_Answers.docx**).

At the end of the exam, create a PDF version of your answers file (example of notation: **LXY_CI_07_Answers.pdf**) and check:

- that the PDF file contains all your solutions,
- that the screenshots are readable.
- that all SQL queries are provided, one query per file.

Remember to save your file regularly!

You will be evaluated solely on the contents of the PDF file!

Question 1 – Establishing a CDM

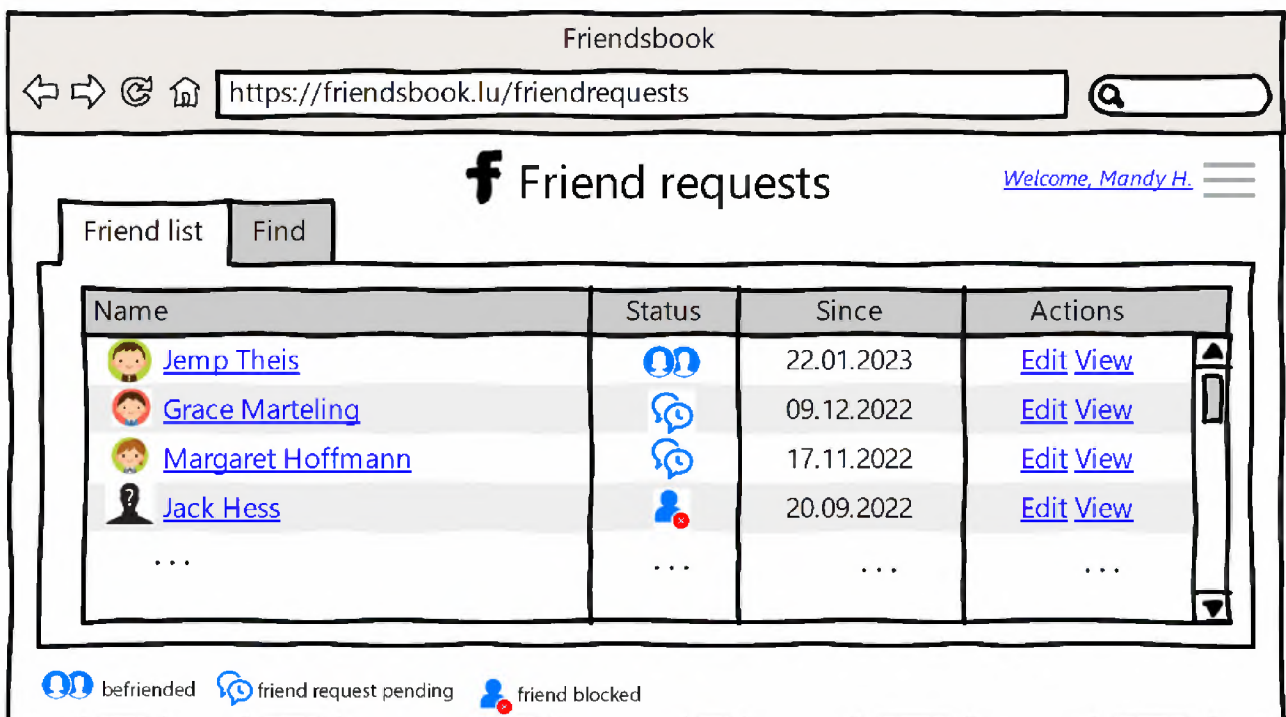
[20p]

Create a *Conceptual Data Model* (CDM) using the software draw.io for a web application named “Friendsbook” based on information from the following mock-ups and descriptions. Upon completion, include a screenshot on a white background without grid lines in your answers document.

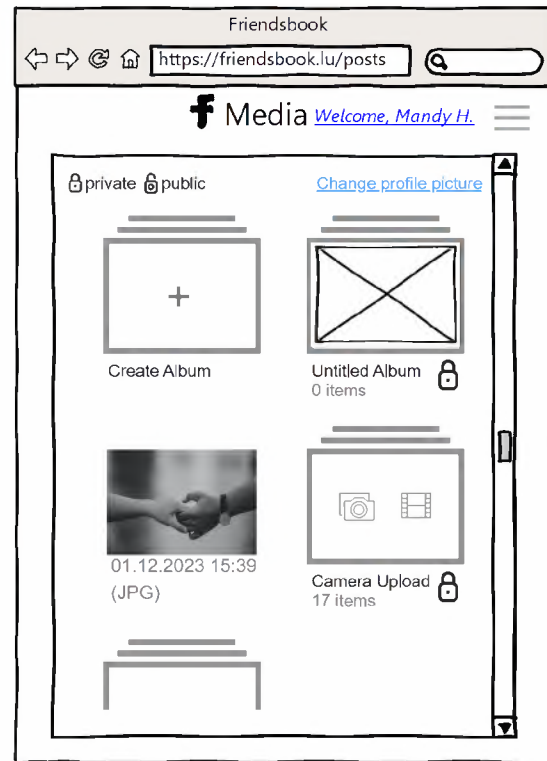
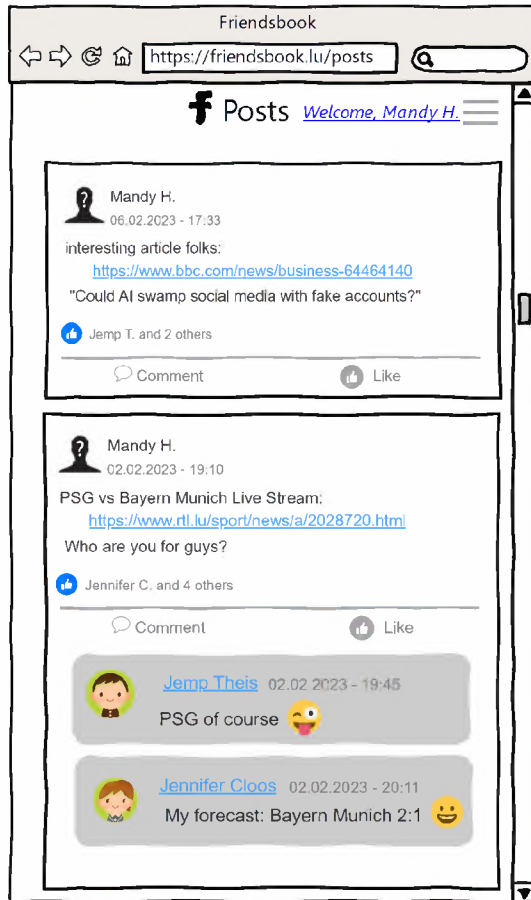
Before being able to use Friendsbook, a user must first create an account. The email address, password, first name, last name, date of birth, and account creation date are recorded. For each user account, there must be a status indicating whether the account is active, provisional, or permanently blocked.



The application allows users to search for other users. User A can send a friend request to user B. The two users are considered friends only when user B has accepted the request. From that moment, both users can view each other's profiles, including posts and other media. However, a friendship between two users can end at any time. For this, user A can remove another user B from their friends' list or block them. Thus, user B will no longer be able to send friend requests or messages to user A. The system must be able to record the history of new friendships, including friendship terminations, with a precise date indication.



At any time, a user can publish a post to communicate something with their friends. Each post must be uniquely identified and can have optional content. Additionally, the creation date must be recorded. The author's friends can then comment or like the post.



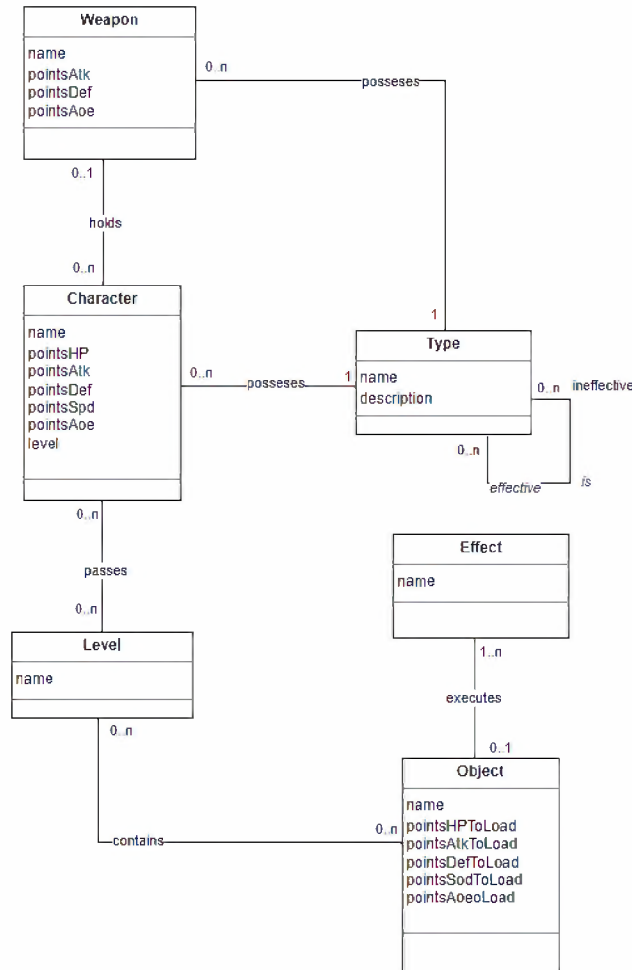
A user can upload media files to their account. Only the user's friends can view these media files. In the initial version of the application, only photos and videos are supported. The upload date must be recorded. However, the application should be designed in such a way that it can support additional media types (e.g., PDF) in the future.

The user can set a specific photo as their profile picture. Multiple media files can be grouped into an album by a user. However, a media file can only be represented in one album at a time. An album has a name and a creation date. For each album, it is possible to define whether it is private or public. Private albums can only be viewed by their creator.

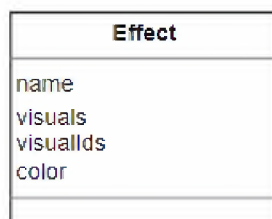
Question 2 – LDM and Comprehension

[9+3 = 12p]

- a. Translate the CDM below into a LDM using the provided file. Please adhere to the conventions covered. Use draw.io for your work and insert a screenshot (on a grid-less white background) of your LDM into your solution document. **[9p]**



- b. Consider the screenshot below which shows a more elaborated Effect class. It includes the visuals to display and the color of the effect. An effect can have multiple visuals. Explain the problem(s) related to normalisation. Identify the normal form(s) that are not being respected. **[3p]**



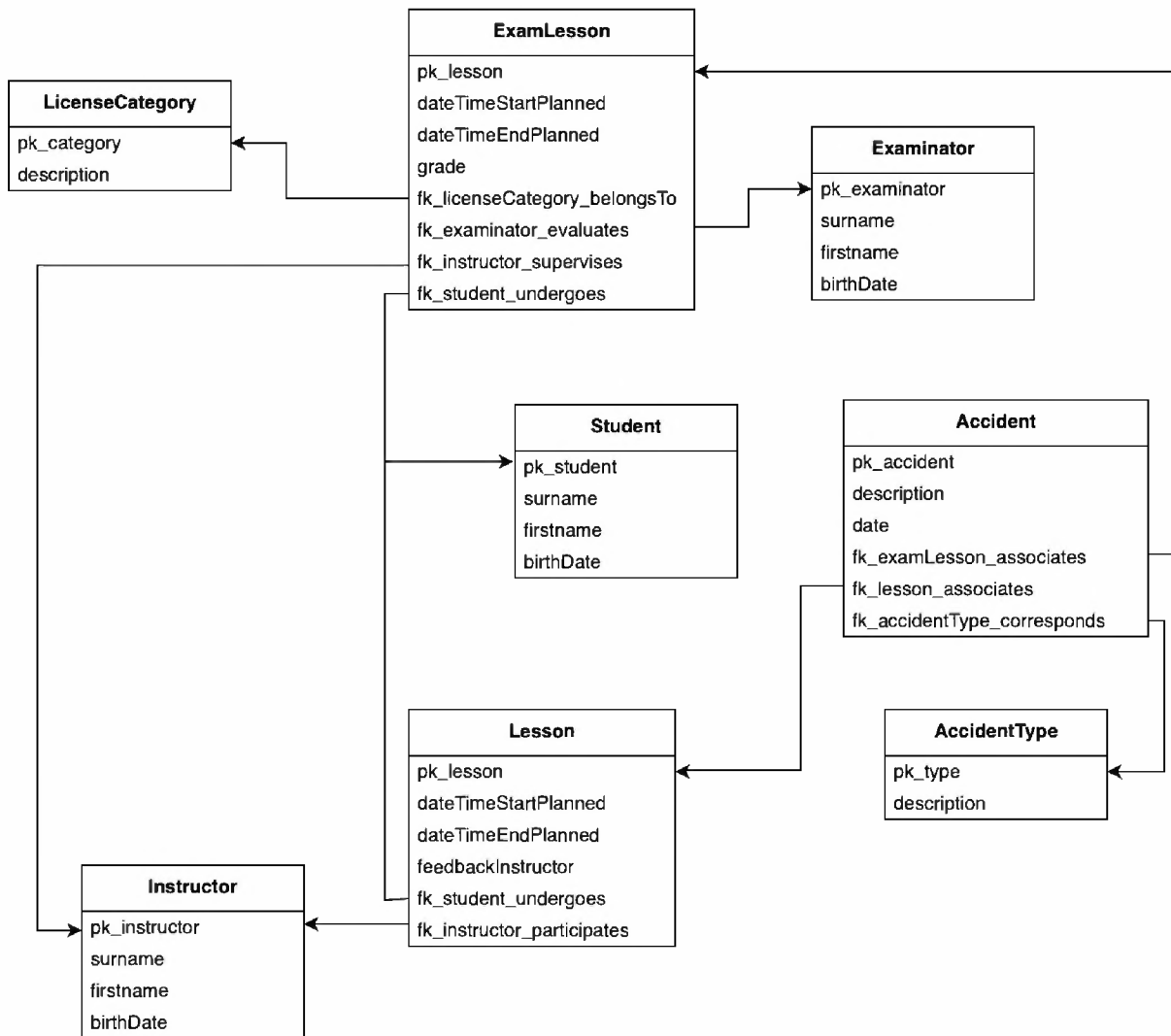
Question 3 – SQL Queries and Comprehension

[28p]

Import the file Database.sql into Workbench. This will create a database corresponding to the LDM below. Use the dataset inserted into this database for testing your SQL queries. You may also add additional records.



Elaborate on your responses and insert the SQL code along with a screenshot of the query results into your document. Also create a .sql file for each query in your folder.



Note: Use meaningful aliases when using aggregation functions.

Query 1. Develop the SQL code to return the last name, first name, and grade for students who have failed an exam, that is, they obtained a grade below 45. For each of those exams, provide the planned start date and time they failed this exam on. Sort by grade in descending order.

[3p]

Query 2. Develop the SQL code to display the number of attempts for each student (see screenshot for column names) who has attempted the exam at most once. Sort the result in descending order of number of attempts. **[5p]**

Full Name	Number of Attempts
Müller Anna	1
Schmit Tom	1
Wagner Sophie	1
Becker Laura	1
Huber Martin	1
Schneider Peter	1
Jacobs Kevin	0
Klein Michael	0
Weber Christine	0

Query 3. Develop the SQL code to return the surname and first name of students who have taken exams in all available license categories. **[5,5p]**

Query 4. Develop the SQL code to return a list of instructors (I) (see screenshot for column names) and their students (S). Show only those where the surname of the instructor begins with the surname of the student. **[4,5p]**

Surname (S)	First name (S)	Surname (I)	First name (I)
Schneider	Peter	Schneider	Michael
Wagner	Sophie	Wagner	Christine
Becker	Laura	Becker	Laura
Klein	Michael	Klein	Thomas
Fisch	Jennifer	Fischer	Sarah

Query 5. Develop the SQL code to return a list of all instructors (see screenshot for column names and sorting order) as well as the number of lessons and exams given. **[5,5p]**

pk_instructor	surname	firstname	lesson_count	exam_count
10	Schmit	Tom	0	0
6	Fischer	Sarah	0	1
8	Weber	Jennifer	1	1
3	Meyer	Peter	1	2
7	Huber	Martin	2	0
9	Müller	Anna	2	1
5	Klein	Thomas	3	1
4	Becker	Laura	3	4
2	Wagner	Christine	4	0
1	Schneider	Michael	4	1

Query 6. **[1,5+1,5+1,5=4,5p]**

a. Express the following query in a sentence:

```
SELECT pk_type, T.description
FROM AccidentType AS T
LEFT JOIN Accident ON pk_type = fk_accidentType_corresponds
WHERE fk_accidentType_corresponds IS NULL;
```

b. Is the alias *T* important in this query? Explain!

c. Explain the following problem statement in a sentence or two:

```
SELECT UPPER(surname), GROUP_CONCAT(DISTINCT pk_category)
FROM Student
INNER JOIN ExamLesson ON fk_student_undergoes = pk_student
INNER JOIN LicenseCategory ON fk_licenseCategory_belongsTo = pk_category
WHERE grade >= 45
GROUP BY pk_student
```