

Database Systems

Autumn 2025

Exercise 2

Hand-in: 26.10.2025 (11:59 pm)

Solving the Exercises: The exercises must be solved in groups of three people. Use the notations introduced in the lecture. Please choose the same partners for exercises and project. The DMI plagiarism guidelines apply for this lecture.

Submission Information: Please upload your solutions BEFORE the deadline to ADAM using the team hand-in feature. Solutions that are handed in too late cannot be considered. The use of the exercise template, located in "Exercise Material", is mandatory.

Task 1: Data Integrity — Exercise database (8 points)

You are given the following excerpt of the relational schema of the exercise database, together with an exemplary tuple. Both primary and foreign keys are denoted.

Exercise	(<u>ExID</u> , No, Semester, Lecture) {45, 2, 'Autumn 2025', 'Database Systems'}
contains	(<u>ExID</u> , <u>TaskID</u> , Sequence) {45, 33, 1}
Task	(<u>TaskID</u> , Points, Difficulty, Text, <u>AuthorID</u>) {33, 20, 2, 'You are given the following ...', 4}
Author	(<u>AuthorID</u> , LastName, FirstName, Title) {4, 'Fagin', 'Ronald', 'PhD'}
consists_of	(<u>SuperTaskID</u> , <u>SubTaskID</u> , Sequence) {33, 34, 1}

Note that the *TaskID* attribute of the *contains* relation only refers to super-tasks (i. e., not to sub-tasks) and *consists_of* is not recursive.

Add a new relation *Lecture* and a new relation *Lecturer*. A lecture has a title, credit points, semester week hours and a lecturer. A lecturer has a first name, a last name, a title and an ID.

Create a SQL script for the extended relational schema. Cover all primary and foreign key constraints. Moreover, specify the following integrity constraints using the methods presented in the lecture for ensuring integrity, i. e., primary and foreign keys, constraints, assertions and triggers:

- The title of the lecture has to be unique and may not be altered if any exercise is available for the lecture.

- For a lecture, no more than 10 credit points may be awarded.
- Lecturers may give multiple lectures.
- A lecture may include several exercises. An exercise always belongs to exactly one lecture.
- Before a new author is entered into the system, it should be checked that no other author with the same first name, last name and title is present.

Task 2: Foreign Keys

(6 points)

- a) For the relation *Lecture*, the following foreign key is defined:

```
CREATE TABLE Lecture(
    ...
    FOREIGN KEY(fk_lecturer) REFERENCES Lecturer(
        pk_lecturer)
)
```

“Simulate” this foreign key using (i) assertions, and (ii) triggers.

Describe your solution first in textual form, then write the definition. For the solution, it is sufficient to specify the trigger as done in the lecture slides.

(4 points)

- b) On the relation *Student* and *Exam*:

Student	(<u>SID</u> , Name, Birthday, <u>FkExam</u>)
	⟨ ‘24-384-284’, ‘Jon Snow’, ‘1993-09-23’, ‘34523’ ⟩
Exam	(<u>PkExam</u> , Date, <u>LectureID</u>)
	⟨ ‘34523’, ‘2024-02-23’, ‘234113’ ⟩

The following foreign key is defined:

```
CREATE TABLE Student(
    ...
    FOREIGN KEY(FkExam) REFERENCES Exam(PkExam) ON
    DELETE SET NULL
)
```

“Simulate” this foreign key using triggers. Describe your solution first in textual form, then write the definition. For the solution, it is sufficient to specify the trigger as done in the lecture slides.

(2 points)

Task 3: Advanced Integrity Constraints

(14 points)

You are operating an online casino, where customers can gamble in different forms of mini games, with credits. The database to manage this operation has the following relations:

User (UID, Name, Street, Country, Mail, Age)
 ⟨ 101, 'John Doe', '123 Maple', 'USA', 'jd@mail.com', 35 ⟩

Credit (CID, UID, CreditScore)
 ⟨ 201, '2024-10-10 14:00', 500.00 ⟩

Regulation (RID, Country, RegulationCode)
 ⟨ 301, 'USA', 'U2584' ⟩

Game (GameID, GameName, MinBet, MaxBet, HouseEdge)
 ⟨ 402, 'Roulette', 1.00, 1000.00, 0.027 ⟩

Bet (BetID, UID, GameID, BetAmount, BetDate, BetTime, Outcome)
 ⟨ 501, 101, 401, 50.00, '2024-10-10', '14:30', 'win' ⟩

Session (SessionID, UID, Date, LoginTime, LogoutTime, Duration)
 ⟨ 601, 101, '2024-10-10', '14:00', '15:00', 60 ⟩

a) Due to regulations you have add SQL integrity constraints: The amount of negative credit users can have at any point is mandated according to the users recidency.

- Users, which are citizens of countries with regulation code U2109 are allowed to have at most a negative of 1000 credits.
- Users, which live in countries with regulation code U2304 are allowed to have at most 10% of their highest amount bet.

(5 points)

b) Due to newly implemented legislation in countries with the regulation code M5475, it is now mandatory to track the total daily online time for users aged 14 to 18 (inclusive), by their name and country of residence. To meet this requirement, you have decided to create a materialized view that will automatically refresh whenever changes occur in any of the underlying tables.

Specify the SQL query that creates this materialized view.

(3 points)

c) Due to performance constraints in your infrastructure, you need to modify the update logic for the materialized view. Instead of refreshing it every time a change occurs in any of the underlying tables, you want to restrict updates to occur only when a country that enforces the regulation code M5475 is modified, added, or removed.

While you are allowed to change how and when the materialized view is refreshed, the view itself must remain unchanged. Specify your adjustments and the SQL logic to achieve this behavior.

(6 points)

Task 4: Deferrability

(2 points)

When does it make sense for assertions to specify the deferrability as DEFERRABLE INITIALLY DEFERRED? Give an example.