

This is the pdf version of the assignment, which is going to be visible on Ans! Please use Ans to work together and to submit your answers.

**Group:** Enroll in a 3-member group on Brightspace by the **21.02.2025**. Groups with fewer than 3 students will be merged together after the deadline. To form a group, we encourage you to discuss in-person with your colleagues, use the Brightspace discussion forum, or if you wish to be assigned a group randomly, please send an email to: [databases@liacs.leidenuniv.nl](mailto:databases@liacs.leidenuniv.nl) with the subject “[Group assignment] Random group”.

In this group assignment, you will design and use your own database in SQLite. The general idea is to demonstrate the use of a relational database, following the 4-steps database design process:

1. Requirements list (statements in natural language about the data organization)
2. ER Diagram
3. Translation to SQL-DDL
4. Example queries

You are **free to choose the topic** of your database, e.g., dancing classes, shops, scientific collections, but you are **not allowed** to use the same examples as in the lectures or exercise classes – all of these are already published so please skim through them. It is mandatory to use the **same style** as in the lectures. Please, follow the design techniques discussed in class or provide valid reasons if you deviate from them; following a different course online is not a valid reason. It is mandatory to use **SQLite**. Lastly, your solutions do not need to be long but aim for **completeness** and **correctness**.

A maximum of **100 points** can be earned with this group assignment. All group members can access and work on the same assignment in Ans, but one group member hands it in on behalf of the whole group. You can access your assignment up until the deadline, i.e., **6th of April, 23:59**. In case you have questions regarding the assignment, please post them on Brightspace.

**Interview:** on the **14th of April, 11:00-12:45** (re-take date TBD, will be posted in the Brightspace Timetable), we will check your understanding of this submission by conducting 10-minute interviews. Closer to the date, we will post a spreadsheet with your group’s time-slot. The interview is mandatory yet it is not graded. We wish to make sure each team member has a good understanding of the solution submitted, you get to train your explanation skills which is relevant for your future job! In case a team (member) fails to show understanding during the interview session, we will provide you with a “resit” opportunity, but then your maximum assignment grade is 5.5. Note that this resit only applies to the oral evaluation. There is no retake possibility for the group assignment.

**Warning:** All submissions are automatically scanned for plagiarism. Fraud will not be tolerated and will always be reported to the Board of Examiners.

**Tip:** We encourage you to work on the group assignment on a weekly basis, following the course material.

- 1.[16p ] Write a list of 10 requirements for your database. These are statements in natural language about the data organization.
- 2.[24p ] Translate your requirements to an ER Diagram.  
Make sure to include at least 1 example of each of the following:
  - i. IS-A hierarchy with COVER/OVERLAP constraint
  - ii. Key constraint on a relationship
  - iii. Weak entity set
  - iv. Self-Join with roles

*Note: if you cannot satisfy one or more of the requirements above, please update your list of requirements in the previous exercise.*

3.[26p ] Translate the ER Diagram to SQL-DDL.

Add example data to your solution using SQL INSERT value statements. Around 5 rows per table is sufficient.

*Note: Check out the material on Brightspace on how to translate self-joins to SQL-DDL.*

Could you capture all the constraints? If you cannot capture some constraints, explain why.

4.[34p ] Implement example queries for your database. Make sure to include one example of each of the following:

- i. GROUP BY ... HAVING
- ii. Set operation (UNION, INTERSECT, EXCEPT)
- iii. Nested query or correlation
- iv. Aggregation (MAX, AVG, SUM, COUNT)
- v. Query over multiple tables (JOIN)
- vi. LIKE (string matching)

Please specify distinct queries for each category. Otherwise, we will only score one of the categories.

For each query, please briefly explain what it computes. Without an explanation, you will not earn any points for that query.