
Databases Lab 04

This is the final lab of Databases. This lab focuses on improving your skills in dealing with SQL queries. There are tasks for both DDL and DML statements. It is recommended that you familiarize yourself with the assignments to allow a more effective participation in Laboratory 4. If you have questions or need any support, help each other, or use the forum in our moodle room.

Submission Deadline

Deadline to upload the solutions for all tasks is Saturday, 11:59 pm before the lab date.

General Information

The following tasks are to be worked on in fixed teams of two. Each team member must be able to explain all solutions. Please submit only one solution for each team of two. The submission must be a PDF file in our Moodle room with the name and matriculation number. Solutions must be in digital format with intermediate steps and detailed explanations (no handwritten scans). You can use any tool or drawing program of your choice to create the diagrams. If you have questions or need support, use the forum in our Moodle room and help each other.

Contents

Submission Deadline	1
General Information	1
1. Assignment 1: Chemistry database	1
2. Assignment 2: Geography database	2
3. Assignment 3: SQL-statements for the COMPANY example from Elmasri also used in the lecture	2
4. Assignment 4: Transactions	3

1. Assignment 1: Chemistry database

Download the script “chemistry.sql” from Moodle and execute the script in PostgreSQL.

1. Write a transaction to insert two new elements in table ELEMENTS and a new connection in table COMPOUNDS into the database. If an error occurs during one of the insertions, all changes should be reversed. In addition, the transaction should be logged in table CHANGELOG.
2. Write a transaction to add
 - a new lab (ChemLab6 located at Building C, Room 101) into table LABORATORIES,
 - a new researcher (Sophia Neumann) assigned to this lab into table RESEARCHERS, and
 - two new reactions discovered by this new researcher into table REACTIONS.

If an insertion fails, all changes should be reversed. Additionally, log each change with a detailed description in table CHANGELOG.

3. Create the view V_RESEARCHERS_DETAILS to display information about all researchers in detail, including their full name, the name of their lab and the location of the lab. The view should combine data from the tables RESEARCHERS and LABS.
4. Try to insert, delete, and update tuples in the view V_RESEARCHERS_DETAILS. Which operations (INSERT, DELETE, and UPDATE) can be executed and which not? Explain your answer.

2. Assignment 2: Geography database

Download the script “geography.sql” from Moodle and execute the script in PostgreSQL. Write SQL queries for the following questions.

1. What is the capital of Germany?
2. Write an SQL query to list all cities in the USA.
3. Write an SQL query to find the capitals and populations of all countries with names beginning with the letter “C”.
4. List all rivers that are longer than 4000 km.
5. Identify the highest mountains in descending order of height.
6. List all cities with a population over 5 million in descending order of population.
7. Add a new language called “Swahili” and a new country “Kenya” with CountryID 19. Then, link the language and the country in table COUNTRYLANGUAGES. Write a query displaying all information about the country “Kenya” and the language “Swahili” that checks the completeness of the data.
8. Create the view V_LARGEST_CITY_AND_HIGHEST_MOUNTAIN that shows for each

country the name of the country, the name of the most populated city, the population of this city, the name of the highest mountain and the height of this mountain.

9. Try to insert, delete, and update tuples in the view

V_LARGEST_CITY_AND_HIGHEST_MOUNTAIN. Which operations (INSERT, DELETE, and UPDATE) can be executed and which not? Explain your answer.

3. Assignment 3: SQL-statements for the COMPANY example from Elmasri also used in the lecture

Let’s revisit the company example.

1. Create a SQL statement which creates the view V_PROJECT1. View V_PROJECT1 has the project name, controlling department name, number of employees, and total hours worked per week on the project for each project.
2. Create a SQL statement which creates the view V_PROJECT2. View V_PROJECT2 that has the project name, controlling department name, number of employees, and total hours worked per week on the project for each project with more than one employee working on it.
3. Assume the view V_SENIORS with following code is given.

```
1 CREATE VIEW V_SENIORS AS
2 SELECT *
3 FROM employee
4 WHERE salary > 45000
5 WITH CHECK OPTION;
```

SQL

- Explain the purpose of views in general and the function of the view V_SENIORS.
- Is it possible to modify the tuples in view V_SENIORS?
 - First think about it in theory.

- ▶ Second try it out in a database:
 - Update the salary of one employee in view V_SENIORS and in base table EMPLOYEE.
 - Insert some new employees in view V_SENIORS and in base table EMPLOYEE.
 - Delete some tuples in view V_SENIORS and in base table EMPLOYEE.

4. Assignment 4: Transactions

For this assignment, you need two distinct sessions (connections) to your database. In PostgreSQL's pgAdmin you can just open two instances of either the CLI psql or the query tool, which will act as independent session. The two database sessions are denoted as session1 and session2, resp. For every task include your statements and the corresponding output in session1 and session2 in a table. Make sure everything is in chronological order. Example:




Timestamp	Session 1	Session 2
0	<pre>1 INSERT INTO TAB1...;</pre>  <ul style="list-style-type: none"> Result screenshot <pre>1 SELECT * FROM TAB1;</pre>  <ul style="list-style-type: none"> Result screenshot 	-
1		<pre>1 SELECT * FROM TAB1;</pre> 

Table 1: Sessions

1. In session1 run a command creating a table named TAB1 with two attributes:
 - id with data type integer, primary key
 - n with data type integer

When is table TAB1 visible in session2?

2. In session1 insert the following tuples into TAB1 within one transaction:

```
1 (1,1), (2,2), (3,3)
```

What content of TAB1 is displayed in session2 before and after you commit your changes in session1?

3. In session1 update the value of n to 33 for the tuple with id 3 (without committing). Afterwards rollback that transaction.
 - What value of n (id=3) is displayed in session1 before and after the rollback?
 - What value of n (id=3) is displayed in session2 before and after the rollback?