

Übung Datenbanken und Informationssysteme 1 2023w



Lecturer:

Lisa Ehrlinger, Daniel Hofer, Josef Küng, Wolfram Wöß

SQL Part 2

Submission 07.12.2023, 13:00

Technical Guidelines

The course information system is implemented in an Oracle database.

Oracle SQL Developer

The client software "Oracle SQL Developer" is currently available for Windows, macOS, and Linux. The "Oracle SQL Developer" (e.g., Windows 64-bit with JDK included) has to be downloaded from the Oracle website and installed on your PC. In Windows, after unzipping the downloaded file, you can immediately start "sqldeveloper.exe" without any further installation procedure. (<https://www.oracle.com/tools/downloads/sqldev-downloads.html>)

Create a new database connection (green +-symbol at the upper left side of the window) and connect to the database:

Name (connection): choose a name for the connection by your own

Benutzername (user): infosys

Kennwort (password): infosys

Hostname (host): infosys.faw.jku.at

Port: 1521

Service-Name: infosys

After you are successfully connected to the database, you can create and execute SQL statements in the "Query Builder" frame.

Course Information System

The JKU stores data about courses of the SS 2030 in an information system with the following four relations: LVA (course), Person (lecturer), Abhaltung (appointment), and Raum (room).

LVANr is structured as follows: the first 3 digits correspond to the institute number and the first 4 digits correspond to the department number. The institutes with the number 311, 312, and 321 comprise the entire area of "Computer Science". Course types are VO (Vorlesung / lecture), UE (Übung / exercise), SE (Seminar / seminar), PR (Praktikum / practical course).

The relation "Abhaltung" (appointment) is based on the calendar day. Thus, for each appointment a course takes place, there is one entry in the table.

Relation name	Attribute	Type	Comment	English
LVA (course)	LVANr	varchar2(6)	312704	course number
	Name	varchar2(50)		course title
	Std	number(2)		weekly hours
	Typ	char(2)	VO, UE, SE, ...	type (lecture, ...)
Person (lecturer)	PersNr	varchar2(4)	Personal-Nummer	person id
	Name	varchar2(50)	Name	name
Abhaltung (appointment)	LVANr	varchar2(6)		course number
	PersNr	varchar2(4)		person id
	Tag	date	Kalender-Tag	date (calendar day)
	Von Stunde	number(2)		start hour
	Von Minute	number(2)		start minute
	Bis Stunde	number(2)		end hour
	Bis Minute	number(2)		end minute
	Raum Id	varchar2(8)		room id
Raum (room)	Raum Id	varchar2(8)	Raum-Nummer	room id
	Name	varchar2(30)	Raumbezeichnung	room name
	AnzPers	number(4)		number of persons
	Gebaeude	varchar2(20)		name of building

Exercises

Create and execute the following SQL statements. You have to submit the SQL statement as well as the result set (output) including the number of rows in the result set. Please consider that the layout of the output should be easy to read (one line for one row).

- 6.9. Create a list of all buildings and the room with the highest capacity in each of these buildings. Filter the list to include only buildings whose largest room has a capacity greater than 100 and sort the list by capacity in descending order. The result should contain the building (name) and the corresponding capacity of the largest room. (5 points)
- 6.10. Create a list of all courses (course number, course title, type, weekly hours) that are currently taking place on March 21, 2030 at 10:30 in the building "Managementzentrum". (4 points)
- 6.11. Create a list of all courses (course number, course title, appointments, duration) with a duration of 45 minutes in descending order by number of appointments. Limit the list to courses with 5-10 appointments. (6 points)
- 6.12. Create a list of all "Computer Science" lectures (course title, type, date, start time, end time, duration) with the longest (maximum) appointment duration in minutes. (6 points)
- 6.13. Create a list of all FAW courses (course number begins with "3127") starting at 14:30 with the corresponding number of appointments (course number, course title, start time, end time, number of appointments). Limit the list to courses with a maximum of 15 appointments in descending order according to the number of appointments. (3 points)
- 6.14. Create a list of all courses with two weekly hours (course title, course number, weekly hours, type) in the period 01-30 June, 2030 held by the FAW institute (course number begins with "3127") in alphabetical order by name and without duplicates. (3 points)

Create a view based on that query with the name "FAW_Courses". Attention: This operation cannot be executed in the Oracle database since you do not have the "create view" privilege. Destructive student(s) attacked the database server and therefore student privileges had to be limited to "select". (2 points)

Delete the View "FAW_Courses". (1 point)

- 6.15. Create a list of all course appointments held by "Wöß Wolfram" in June, 2030 including the lecturer's name, course number, course title, type, date, and room id. The output format is defined in the following paragraph. (2 points)

Assume that your query result is persisted in the database as table "CourseAppointments" with the columns lecturer, courseNo, courseTitle, type, date, roomId.

CourseAppointments ({lecturer, courseNo, courseTitle, type, date, roomId}, {courseNo → courseTitle type}).

- a) What is the primary key of table "CourseAppointments"? (1 point)
- b) In which normal form is table "CourseAppointments"? (1 point)
- c) Rename the course "Übung Informationssysteme 1" to "Übung Datenbanken und Informationssysteme 1". This operation cannot be executed in the Oracle database (see comments above in 6.14). (2 points)
- d) What are the consequences of that update concerning anomalies and consistency of the entire course database. (2 points)