

Courses B0B36DBS, A7B36DBS: **Database Systems**

Practical Classes 05 and 06:

SQL: DML

Martin Svoboda

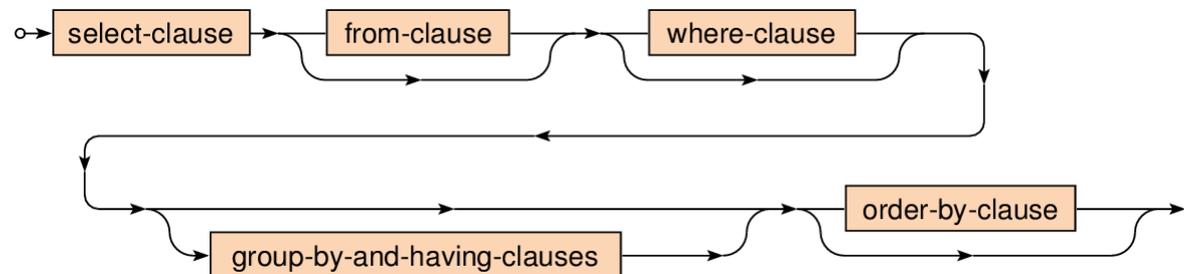
21. and 28. 3. 2017

Faculty of Electrical Engineering, Czech Technical University in Prague

Select Queries

- **SELECT statements in a nutshell**

- **SELECT** clause: columns to be included in the result
- **FROM** clause: tables with data to be queried
- **WHERE** clause: condition a row must satisfy
- **GROUP BY** clause: attributes to be used for grouping
- **HAVING** clause: condition a group of rows must satisfy
- **ORDER BY** clause: criteria to be used for sorting



Exercises

- Assume the following relational database schema

Student(id, name, address)

Teacher(id, name, phone, department)
department \subseteq Department(name)

Department(name, chair)
chair \subseteq Teacher(id)

Course(code, title, annotation)

Dependency(course, requisite)
course \subseteq Course(code), requisite \subseteq Course(code)

Schedule(course, teacher, semester, day, time, room)
course \subseteq Course(code), teacher \subseteq Teacher(id), room \subseteq Room(number)

Room(number, building, capacity)

Enrollment(student, semester, code, result)
student \subseteq Student(id), code \subseteq Course(code)

Exercise 1

- Express the following query in SQL
 - **Teachers from department *KSI***

Teacher(id, name, phone, department)
department \subseteq Department(name)

Department(name, chair)
chair \subseteq Teacher(id)

Exercise 2

- Express the following query in SQL
 - **Study results of a student with identifier *4301* within the previous semester (161)**
 - Return course codes, names, and the actual results
 - Order the rows according to the actual study results and then also course names in descending order

Student(id, name, address)

Course(code, title, annotation)

Enrollment(student, semester, code, result)

student \subseteq Student(id), code \subseteq Course(code)

Exercise 3

- Express the following query in SQL
 - **Names of teachers from all departments that have *Tomas Skopal* as a department chief**

Teacher(id, name, phone, department)
department \subseteq Department(name)

Department(name, chair)
chair \subseteq Teacher(id)

Exercise 4

- Express the following query in SQL
 - **Codes and titles of all courses that are taught on *Mondays or Fridays* during this semester (162)**

Course(code, title, annotation)

Schedule(course, teacher, semester, day, time, room)

course \subseteq Course(code), teacher \subseteq Teacher(id), room \subseteq Room(number)

Exercise 5

- Express the following query in SQL
 - **Codes and titles of all courses that are not taught on *Mondays* and nor on *Fridays* this semester (162)**

Course(code, title, annotation)

Schedule(course, teacher, semester, day, time, room)

course \subseteq Course(code), teacher \subseteq Teacher(id), room \subseteq Room(number)

Exercise 6

- Express the following query in SQL
 - **Students without any enrolled course this year (semesters *161* and *162*)**
 - Return student names and addresses

Student(id, name, address)

Enrollment(student, semester, code, result)

student \subseteq Student(id), code \subseteq Course(code)

Exercise 7

- Express the following query in SQL
 - **Names of students who have enrolled in at least one course that is taught by at least one teacher from department *KSI* during this semester (162)**

Student(id, name, address)

Teacher(id, name, phone, department)
department \subseteq Department(name)

Schedule(course, teacher, semester, day, time, room)
course \subseteq Course(code), teacher \subseteq Teacher(id), room \subseteq Room(number)

Enrollment(student, semester, code, result)
student \subseteq Student(id), code \subseteq Course(code)

Exercise 8

- Express the following query in SQL
 - **Names of students that are enrolled only in courses taught only by teachers from department *KSI* during this semester**
 - Assume only students with at least one enrolled course

Student(id, name, address)

Teacher(id, name, phone, department)
department \subseteq Department(name)

Schedule(course, teacher, semester, day, time, room)
course \subseteq Course(code), teacher \subseteq Teacher(id), room \subseteq Room(number)

Enrollment(student, semester, code, result)
student \subseteq Student(id), code \subseteq Course(code)

Exercise 9

- Express the following query in SQL
 - **Names of teachers who have time conflicts in their schedules for the next semester (171)**
 - Two events are in a conflict if...
 - they have overlapping times, but also
 - when there is less than 10 minutes for a break / 45 minutes for a transfer in case of events scheduled within the same / in different buildings respectively
 - Each event is 90 minutes long

Teacher(id, name, phone, department)
department \subseteq Department(name)

Schedule(course, teacher, semester, day, time, room)
course \subseteq Course(code), teacher \subseteq Teacher(id), room \subseteq Room(number)

Room(number, building, capacity)

Exercise 10

- Express the following queries in SQL
 - **Overall and average capacity of all rooms**
 - **Overall and average capacity of all rooms for each individual building**

Room(number, building, capacity)

Exercise 11

- Express the following query in SQL
 - **Overall numbers of enrolled students and average achieved results for courses from semester *161***
 - Return course titles
 - Include only courses with at least 10 enrolled students
 - Return the courses in a descending order according to the average results

Course(code, title, annotation)

Enrollment(student, semester, code, result)

student \subseteq Student(id), code \subseteq Course(code)