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 ⊆ Department(name)

**Department**(name, chair)
  chair ⊆ Teacher(id)

**Course**(code, title, annotation)

**Dependency**(course, requisite)
  course ⊆ Course(code), requisite ⊆ Course(code)

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

**Room**(number, building, capacity)

**Enrollment**(student, semester, code, result)
  student ⊆ Student(id), code ⊆ Course(code)
Exercise 1

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

  \[
  \textbf{Teacher}(\text{id}, \text{name}, \text{phone}, \text{department}) \\
  \text{department} \subseteq \text{Department}(\text{name}) \\
  \textbf{Department}(\text{name}, \text{chair}) \\
  \text{chair} \subseteq \text{Teacher}(\text{id})
  \]
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

\[
\begin{align*}
\text{Student} & \left( id, \text{name}, \text{address} \right) \\
\text{Course} & \left( \text{code}, \text{title}, \text{annotation} \right) \\
\text{Enrollment} & \left( \text{student, semester, code, result} \right) \\
\text{student} & \subseteq \text{Student}(id), \text{code} \subseteq \text{Course}(code)
\end{align*}
\]
Exercise 3

Express the following query in SQL

- Names of teachers from all departments that have **Tomas Skopal** as a department chief

```sql
Teacher(id, name, phone, department)
  department ⊆ Department(name)
Department(name, chair)
  chair ⊆ 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)

\[
\begin{align*}
\text{Course} & (\text{code, title, annotation}) \\
\text{Schedule} & (\text{course, teacher, semester, day, time, room}) \\
\text{course} & \subseteq \text{Course} (\text{code}), \text{teacher} \subseteq \text{Teacher} (\text{id}), \text{room} \subseteq \text{Room} (\text{number})
\end{align*}
\]
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 ⊆ Course(code), teacher ⊆ Teacher(id), room ⊆ 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

\[
\text{Student} (\text{id}, \text{name}, \text{address}) \\
\text{Enrollment} (\text{student}, \text{semester}, \text{code}, \text{result}) \\
\text{student} \subseteq \text{Student}(\text{id}), \text{code} \subseteq \text{Course}(\text{code})
\]
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 ⊆ Department(name)
Schedule(course, teacher, semester, day, time, room)
  course ⊆ Course(code), teacher ⊆ Teacher(id), room ⊆ Room(number)
Enrollment(student, semester, code, result)
  student ⊆ Student(id), code ⊆ 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

\[
\text{Student}(\text{id}, \text{name}, \text{address}) \\
\text{Teacher}(\text{id}, \text{name}, \text{phone}, \text{department}) \\
\quad \text{department} \subseteq \text{Department}(\text{name}) \\
\text{Schedule}(\text{course}, \text{teacher}, \text{semester}, \text{day}, \text{time}, \text{room}) \\
\quad \text{course} \subseteq \text{Course}(\text{code}), \text{teacher} \subseteq \text{Teacher}(\text{id}), \text{room} \subseteq \text{Room}(\text{number}) \\
\text{Enrollment}(\text{student}, \text{semester}, \text{code}, \text{result}) \\
\quad \text{student} \subseteq \text{Student}(\text{id}), \text{code} \subseteq \text{Course}(\text{code})
\]
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 ⊆ Department(name)

Schedule(course, teacher, semester, day, time, room)
  course ⊆ Course(code), teacher ⊆ Teacher(id), room ⊆ 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 ⊆ Student(id), code ⊆ Course(code)