BOB36DBS, BD6B36DBS: Database Systems

http://www.ksi.mff.cuni.cz/~svoboda/courses/182-B0B36DBS/

Practical Classes 6 and 7 SQL: Data Querying

Author: Martin Svoboda, martin.svoboda@fel.cvut.cz Tutors: Ahmad, Černoch, Kostov, Řimnáč, Svoboda, Šourek

26. 3. 2019

Czech Technical University in Prague, Faculty of Electrical Engineering

Database Schema

Assume we have the following schema of a relational database for a simple **student information system**

Student (<u>id</u>, name, address) Teacher (<u>id</u>, name, phone, department) department ⊆ Department (name)

Department (<u>name</u>, chair)

chair \subseteq Teacher (id)

Course (code, title, annotation)

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

```
Schedule (course, <u>teacher</u>, <u>semester</u>, <u>day</u>, <u>time</u>, 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)

SELECT statement (simplified)



FROM clause





WHERE clause



SELECT clause



Express the following SQL query

Teachers from department KSI

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

Natural JOIN operation



ORDER BY clause



Express the following SQL query

- Study results of a student with identifier 4301 from the previous semester (181)
 - Return course codes, names, and the actual results
 - Sort 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)

Inner JOIN operations



Express the following SQL query

• Names of teachers from all departments that have *Tomas Skopal* as their chief

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

Express the following SQL query

• Codes and titles of all courses that are taught on *Mondays* or *Fridays* during this semester (182)

Course (code, title, annotation)

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

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

IN expression



EXISTS expression

↔ EXISTS → () → SELECT statement → () → •

ANY / SOME / ALL quantifier expressions



Table references in FROM clause



SELECT statement



Express the following SQL query

 Codes and titles of all courses that are <u>not</u> taught on Mondays and nor Fridays during this semester (182)

Course (<u>code</u>, title, annotation)

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

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

Outer JOIN operations



IS NULL comparison expression



Express the following SQL query

- Students without any enrolled course this year (semesters 181 and 182)
 - Return student names and addresses

 $\begin{array}{l} \mbox{Student} \left(\underline{id}, name, address \right) \\ \mbox{Enrollment} \left(\underline{student}, \underline{semester}, \underline{code}, result \right) \\ \mbox{student} \subseteq \mbox{Student} (id), code \subseteq \mbox{Course} (code) \end{array}$

Express the following SQL query

 Names of students who are enrolled in at least one course that is taught by at least one teacher from department KSI during this semester (182)

```
Student (id, name, address)
```

Teacher (<u>id</u>, name, phone, department) department ⊂ Department (name)

```
Schedule ( course, <u>teacher</u>, <u>semester</u>, <u>day</u>, <u>time</u>, 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 )
```

Express the following SQL query

- Names of students who <u>are</u> enrolled <u>only</u> in courses that <u>are</u> taught <u>only</u> by teachers from department KSI during this semester (182)
 - Assume only students with at least one enrolled course
 - Also assume that for each course with at least one enrollment there exists at least one schedule event in a given semester

```
\begin{array}{l} \label{eq:student} \textbf{Student} \left( \underline{id}, name, address \right) \\ \hline \textbf{Teacher} \left( \underline{id}, name, phone, department \right) \\ department \subseteq Department (name) \\ \hline \textbf{Schedule} \left( \textbf{course}, \underline{\textbf{teacher}}, \underline{\textbf{semester}}, \underline{day}, \underline{time}, room \right) \\ course \subseteq Course ( code ), teacher \subseteq Teacher ( id ), room \subseteq Room ( number ) \\ \hline \textbf{Enrollment} \left( \underline{\textbf{student}}, \underline{\textbf{semester}}, \underline{code}, result \right) \\ \textbf{student} \subseteq \textbf{Student} ( id ), code \subseteq Course ( code ) \\ \end{array}
```

Express the following SQL query

- Names of teachers who have time conflicts in their schedules for the next semester (191)
 - Two events are in a conflict if...
 - they have overlapping times, but also
 - when there is less than 15 minutes for a break / 60 minutes for a transfer in case of events scheduled in rooms within the same building / in different buildings respectively
 - Assume that 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 )
```

GROUP BY clause



HAVING clause



Aggregate functions



Express the following SQL queries

- Average capacity and number of all rooms
- Overall capacity of rooms in each individual building

Room (<u>number</u>, building, capacity)

Express the following SQL query

- Overall numbers of enrolled students and average achieved results for courses from the previous semester (181)
 - Return course titles
 - Include only courses with at least 10 enrolled students
 - Sort the courses according to the average results

Course (code, title, annotation)

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