B0B36DBS, BD6B36DBS: **Database Systems**

http://www.ksi.mff.cuni.cz/~svoboda/courses/172-B0B36DBS/

Practical Classes 6 and 7

SQL: Data Querying

Author: **Martin Svoboda**, martin.svoboda@fel.cvut.cz

Tutors: **J. Ahmad**, **R. Černoch**, **M. Řimnáč**, **M. Svoboda**, **G. Šourek**

27. 3. 2018

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 (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)

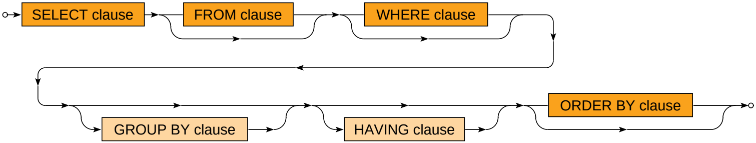
Schedule (course, teacher, semester, day, time, room)
course ⊆ Course (code), teacher ⊆ Teacher (id), room ⊆ Room (number)

Room (number, building, capacity)

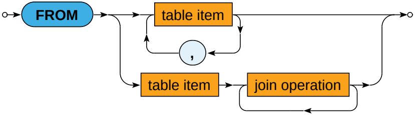
Enrollment (student, semester, code, result)
student ⊆ Student (id), code ⊆ Course (code)

Select Queries

SELECT statement (simplified)

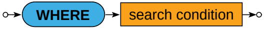


FROM clause

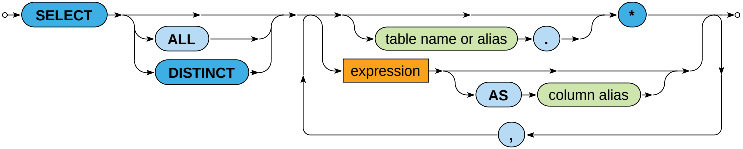


Select Queries

WHERE clause



SELECT clause



Exercise 1

Express the following SQL query

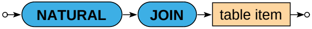
- **Teachers from department *KSI***

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

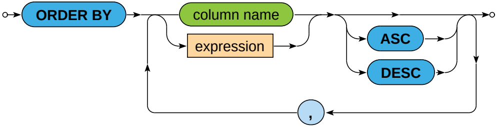
Department (name, chair)
chair $\subseteq$ Teacher (id)

Select Queries

Natural JOIN operation



ORDER BY clause



Exercise 2

Express the following SQL query

- **Study results of a student with identifier *4301* from the previous semester (*171*)**
  - 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

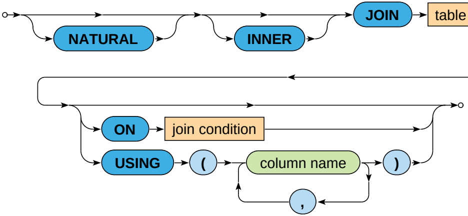
Student (id, name, address)

Course (code, title, annotation)

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

Select Queries

Inner JOIN operations



Exercise 3

Express the following SQL query

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

Teacher (id, name, phone, department)
department ⊆ Department (name)

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

Exercise 4

Express the following SQL query

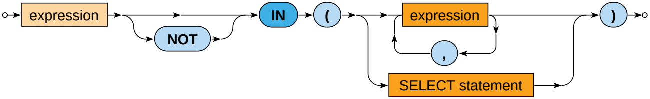
- **Codes and titles of all courses that are taught on *Mondays* or *Fridays* during this semester (*172*)**

Course (code, title, annotation)

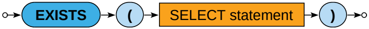
Schedule (course, teacher, semester, day, time, room)
course ⊆ Course (code), teacher ⊆ Teacher (id), room ⊆ Room (number)

Select Queries

IN expression

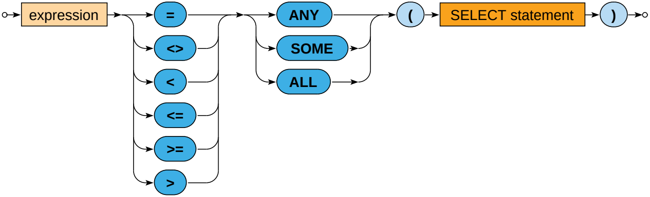


EXISTS expression



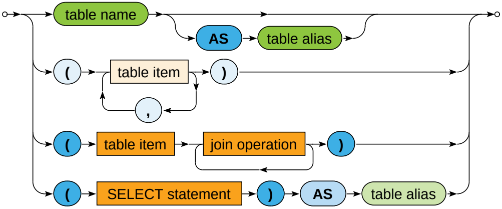
Select Queries

ANY / **SOME** / **ALL** quantifier expressions



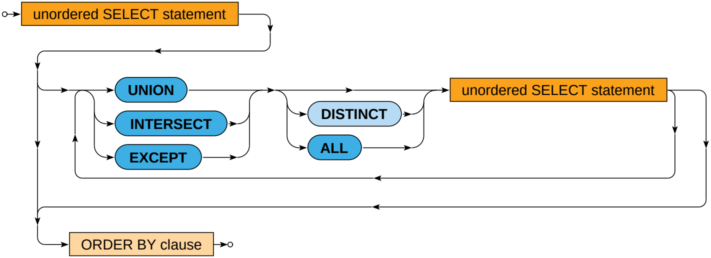
Select Queries

Table references in FROM clause



Select Queries

SELECT statement



Exercise 5

Express the following SQL query

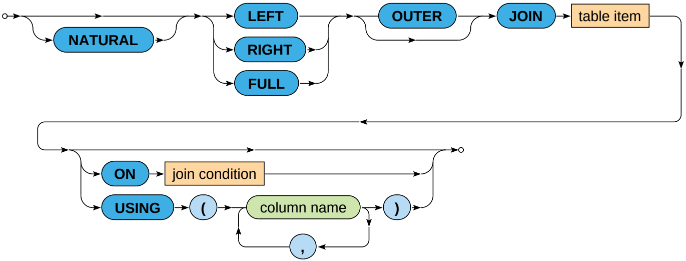
- **Codes and titles of all courses that are not taught on *Mondays* and nor *Fridays* during this semester (*172*)**

Course (code, title, annotation)

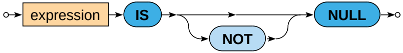
Schedule (course, teacher, semester, day, time, room)
course ⊆ Course (code), teacher ⊆ Teacher (id), room ⊆ Room (number)

Select Queries

Outer JOIN operations



IS NULL comparison expression



Exercise 6

Express the following SQL query

- **Students without any enrolled course this year (semesters *171* and *172*)**
  - 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 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 (*172*)**

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 SQL query

- **Names of students who are enrolled only in courses that are taught only by teachers from department *KSI* during this semester (*172*)**
  - 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

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 9

Express the following SQL query

- **Names of teachers who have time conflicts in their schedules for the next semester (*181*)**
  - 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

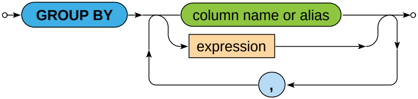
Teacher (id, name, phone, department)
department ⊆ Department (name)

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

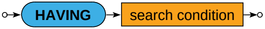
Room (number, building, capacity)

Select Queries

GROUP BY clause

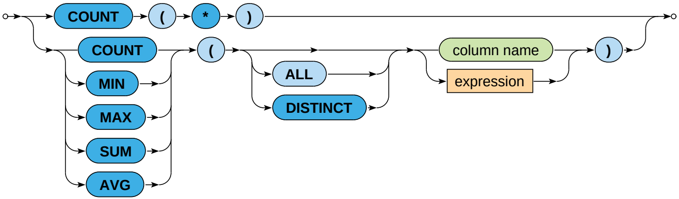


HAVING clause



Select Queries

Aggregate functions



Exercise 10

Express the following SQL queries

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

Room (number, building, capacity)

Exercise 11

Express the following SQL query

- **Overall numbers of enrolled students and average achieved results for courses from the previous semester (*171*)**
  - 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 ⊆ Student (id), code ⊆ Course (code)