### Exam prep!

with Jonas Duregård

# ER-modelling – bonus question!

Make an ER-diagram for a part of a university database dealing with booking lectures.

The database should contain teachers with names and id-numbers, and rooms with names.

It should be possible for teachers to book rooms at specified times. A teacher can book the same room on multiple different times.

# FD's (HT2014, slightly modified)

Question 2. Suppose we have relation R(A, B, C, D, E, F, G) with keys ABC, ACD and ACG, and 10 p functional dependencies  $A \to E$ ,  $AB \to D$ ,  $ABC \to F$ ,  $ABC \to G$ ,  $CD \to G$ ,  $E \to F$ ,  $G \to B$ .

#### a) Find all (minimal) keys of R

- b) i) State, with reasons, which of the FDs listed above violate BCNF. (1p)
  - ii) Decompose relation R to BCNF. Show each step in the normalisation process, and at each step indicate which functional dependency is being used. Indicate keys and references for the resulting relations. (4p)

# SQL Queries, from VT2013

Tracks(<u>trackId</u>, title, length) length > 0Artists(<u>artistId</u>, name) Albums(<u>albumId</u>, title, yearReleased) TracksOnAlbum(<u>album</u>, <u>trackNr</u>, track)  $album \rightarrow Albums. albumId$  $track \rightarrow Tracks.trackId$ (album, track) unique trackNr > 0Participates(<u>track</u>, <u>artist</u>)  $track \rightarrow Tracks.trackId$  $artist \rightarrow Artists.artistId$ Users(<u>username</u>, email, name) email unique *Playlists*(<u>user</u>, playlistName)  $user \rightarrow Users.username$ InList(<u>user</u>, playlist, <u>number</u>, track)  $(user, playlist) \rightarrow Playlists.(user, playlistName)$  $track \rightarrow Tracks.trackId$ *PlayLog(user, time, track)*  $user \rightarrow Users.username$  $track \rightarrow Tracks.trackId$ (user, time) unique

(i) (2p) Write an SQL query that lists all artists appearing on any album released this year(2013).

(ii) (2p) Write an SQL query that lists, for each user, how many playlists that user has.

C) Write an SQL query that finds the title, length and album title of the longest track in the database. If the track appears on more than one album, list the album where it appeared first. If more than one track of the same length qualifies, list the one that was released first, as given by the album it appears on. If there is still a tie, list all such tracks.

# Relational algebra 1 (from 2016-8)

Players(<u>team,number</u>,name,age).

Write a relational algebra expression that returns the name and age of every player whose age is higher than the age of player number 11 in the team named "IFC". (4p)

# Relational algebra: Bonus question!

Players(team,number,name,age)

Write an expression for all players (team and number) that have a globally unique name

#### And now a few words from ours sponsors

At this point you're probably thinking "Wow, it sure is nice of Jonas to organize this extra exam prep session, if only there was something we could do for him"

Well – there is! You can answer the course survey (after it opens that is)

It's the first time I'm giving this course, and your opinions (positive and negative) help me improve it

So please answer the course survey!

# Constraints and triggers (VT2016)

#### 5 Constraints and triggers (12p)

Constraints and triggers are techniques that can be used to control properties of databases. As a rule of thumb, constraints are the simplest solution, but not always possible. In the following cases, use either constraints (added to a table) or triggers (on a table), whichever you find to be the simplest solution.

**a**. Assume you want to create a table that should have at most one row. How can you guarantee this? Write an example CREATE TABLE statement, together with a trigger if you need one.

**b**. Assume you have a table Teachers(name, phoneNumber) that lists the teachers currently in charge for some class. Assume the table already has some rows. How can you guarantee that this table will never become empty i.e. never have 0 rows? Show the exact CREATE TABLE statement and/or trigger.

c. Consider a table with the schema Distances(fromCity,toCity,distance) that stores distances between cities. Since the distance from Y to X is always the same as the distance from X to Y, it would be redundant to store them both. How can you guarantee that the table never stores the distance from Y to X if it already has the distance from X to Y? Show the exact CREATE TABLE statement and/or trigger.

## Constraints - quickround

- What is the best way to ensure that a column is e.g. the sum or the average of something else?
  - Answer: Use a view! If a column value is completely decided by other values, there is no point in allowing the value to be modified at all, so placing it in a view and computing it from other values makes sense.

## Constraints – Bonus question!

- Suppose we have: Teams(teamName)
  Players(team, number, name, age)
- What is the best way to ensure the following: "If a team name is modified, the team value of all players in that team should be modified as well"
  - A view, constraint or trigger?
    - Answer: A constraing, specifically a reference constraint in Players with ON UPDATE CASCADE

### Bonus question

- Which of these will you do after the exam?
  - Answer the course survey
  - Party like it's 1999
  - Both, but not at the same time

#### JSON (From 2016-8, sortof) 6 XML (8p)

A **rose tree** is a tree whose every node has a label and branches to zero or more rose trees. Rose trees can be used for many purposes, for instance, syntax trees and family trees (showing descendants but ignoring spouses). Here is an example of a family tree, starting with a node for the present king of Sweden and showing his children and grandchildren:



JSON Schemaa. Design a DTD for representing rose trees and nothing but rose trees. The labels of the branches nodes should carry a value that can be any string (#PCDATA).<br/>(3p)JSON Documentb. Show an XML element representing the above example tree, and which is<br/>valid according to your DTD. (3p)JSON PathC) Write a JSON Path expression that finds the names of all descendants<br/>of Madeleine

## JSON – Bonus question!

a) Write a JSON Schema for objects that have a "name" property and either "job" or "school", but never both (all values should be strings)

 b) Extend the schema so you can have an optional salary (a number), but only if "job" is defined