Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Übungen zur Vorlesung Informatik II (D-BAUG) FS 2017
D. Sidler, F. Friedrich
http://lec.inf.ethz.ch/baug/informatik2/2017

Solution to exercise sheet # 11 **22.5.2017 – 30.5.2017**

Problem 11.1. ER Modeling

1. People and Traveling

   Create an ER model of the following mini-world:

   - A person has a name, an age and an email address.
   - People like to visit foreign cities.
   - Cities have a name and are located in a specific country.
   - In order to travel to these cities, people always travel in groups.
   - A travel group can have multiple participants, but only one destination.
   - A trip to a city requires a start and an end date.

2. Library

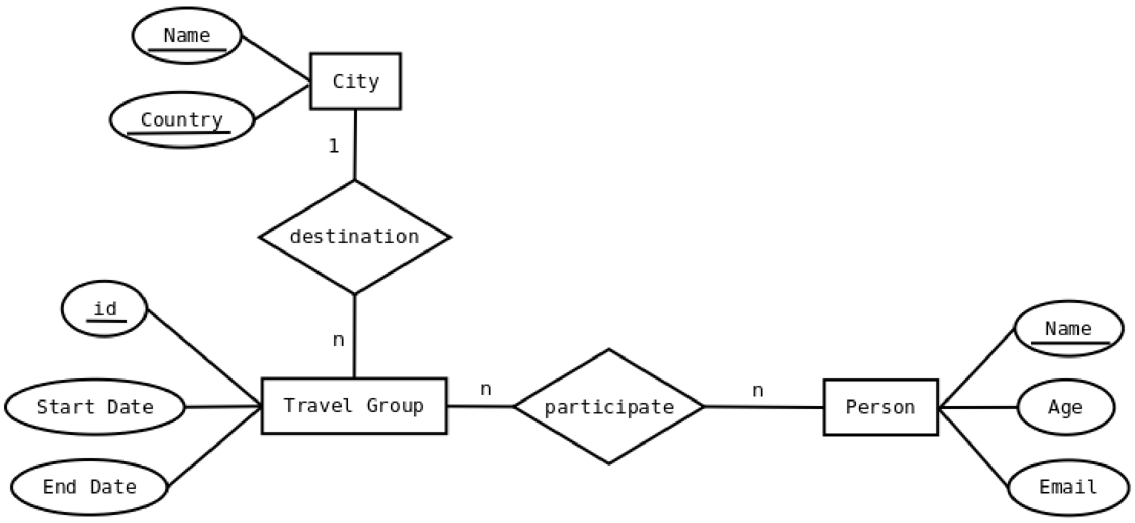
   Assume you are creating a system to manage a library. Give an ER model with the following properties:

   - The library contains one or several books.
   - Every book is located at a specific location in a shelf and is identified by the copy number and the ISBN number.
   - In addition, a book has a publication year, a title and an author.
   - Books are published by publishers.
   - A publisher has a name and as a location.
   - Each reader needs to provide his/her family name, his/her first name and his/her city of residence in order to register at the library.
   - Readers borrow books.
   - Upon borrowing a book, a return date is stored.

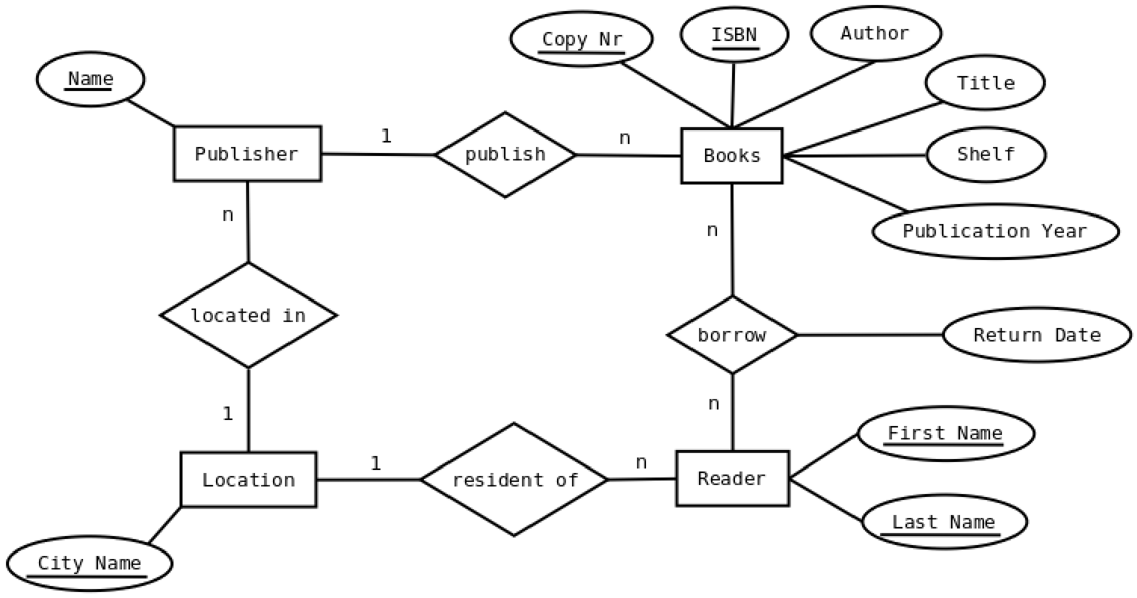
Submission link: https://codeboard.ethz.ch/inf2baugex11t01

Solution of Problem 11.1.

1. People and Traveling



2. Library



Problem 11.2. Relational Model

1. ER to Relational Model

   For each of the ER models from Problem 11.1, create a corresponding relational schema.

2. Relational Algebra

   Given the following relational schema:

   READER (<u>RDNR</u>, Firstname, Lastname, City, Birthdate)
   BOOK (<u>ISBN</u>, Title, Author, NumberOfPages, PublicationYear, PublisherName)
   PUBLISHER (<u>PublisherName</u>, PublisherCity)
   COPY (<u>ISBN</u>, <u>CopyNumber</u>, Shelf, Position)
   BORROW (<u>ReaderNr</u>, <u>ISBN</u>, <u>Copy</u>, ReturnDate)

   Write the following queries in relational algebra:

   - Which are the last names of all the readers living in Zurich?
   - Which books (Title, Author) have their publisher located in Zurich?
   - Which books (Title, Author) have been borrowed by the reader “John Doe”?

Submission link: https://codeboard.ethz.ch/inf2baugex11t02

Solution of Problem 11.2.

1. ER to Relational Model

   CITY (<u>CityName</u>, <u>CounrtyName</u>)
   TRAVELGROUP (<u>GroupId</u>, CityName, CounrtyName, StartDate, EndDate)
   PERSON (<u>PersonName</u>, Age, Email)
   PARTICIPATE (<u>GroupId</u>, <u>PersonName</u>)

   LOCATION (<u>CityName</u>)
   PUBLISHER (<u>PublisherName</u>, CityName)

READER (<u>FirstName</u>, <u>LastName</u>, CityName)
BOOKS (<u>ISBN</u>, <u>CopyNr</u>, ShelfLocation, PublicationYear, PublisherName, Title, AuthorName)
BORROW (<u>ISBN</u>, <u>CopyNr</u>, <u>FirstName</u>, <u>LastName</u>, ReturnDate)

2. Relational Algebra

- $$\pi_{\text{LastName}}(\sigma_{\text{City}='\text{Zurich}'}(\text{READERS}))$$

- $$\pi_{\text{Title,Author}}(\sigma_{\text{BOOK.PublisherName}=\text{PUBLISHER.PublisherName}}(\text{BOOK} \times \sigma_{\text{City}='\text{Zurich}'}(\text{PUBLISHER})))$$

  or (equivalently)

  $$\pi_{\text{Title,Author}}(\text{BOOK} \bowtie \sigma_{\text{City}='\text{Zurich}'}(\text{PUBLISHER}))$$

- $$\pi_{\text{BOOK.Title,BOOK.Author}}(\sigma_{\text{BOOK.ISBN}=\text{BORROW.ISBN}}(\text{BOOK} \times \sigma_{\text{BORROW.ReaderNr}=\text{READER.RDNR}}(\text{BORROW} \times \sigma_{\text{FirstName}='\text{John}' \wedge \text{LastName}='\text{Doe}'}(\text{READER}))))$$

  or (equivalently)

  $$\pi_{\text{BOOK.Title,BOOK.Author}}(\text{BOOK} \bowtie \text{BORROW} \bowtie_{\text{BORROW.ReaderNr}=\text{READER.RDNR}} \sigma_{\text{FirstName}='\text{John}' \wedge \text{LastName}='\text{Doe}'}(\text{READER}))$$

Problem 11.3. SQL Queries

ETH offers a hosted mySQL Database service for all students, which you can access within the ETH network. Thus, you do not need to install your mySQL on your own machine.

To create a database, log into your NETHZ account at `https://password.ethz.ch` and go to the *My Services* tab. Here you can select *mySQL* (see Figure 1). On the next page you have to set a password for your database. After the database has been created, write down the hostname of the database server (e.g., *mysqlweb1.ethz.ch*) and remember the password you have chosen. The database name corresponds to your NETHZ username.

Using the PhpMyAdmin tool you can administer the database. Log in to `https://phpmyadmin.ethz.ch/` using your NETHZ username and the password defined above and click on *MySQL V4*.

Test your database

To test your database, we will create a simple 'Person' table in which we will store the first name, last name and email address of a person (Step 1). Next, we will populate (Step 2) and query the database (Step 3).

You will learn SQL in the second database lecture. For now, you can copy the statements provided below.

Step 1: To create a table for storing data about people, we use the 'CREATE TABLE' command.

```
CREATE TABLE person (
id  INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
firstname  VARCHAR(30) NOT NULL,
lastname VARCHAR(30) NOT NULL,
email  VARCHAR(50) NOT NULL
)
```

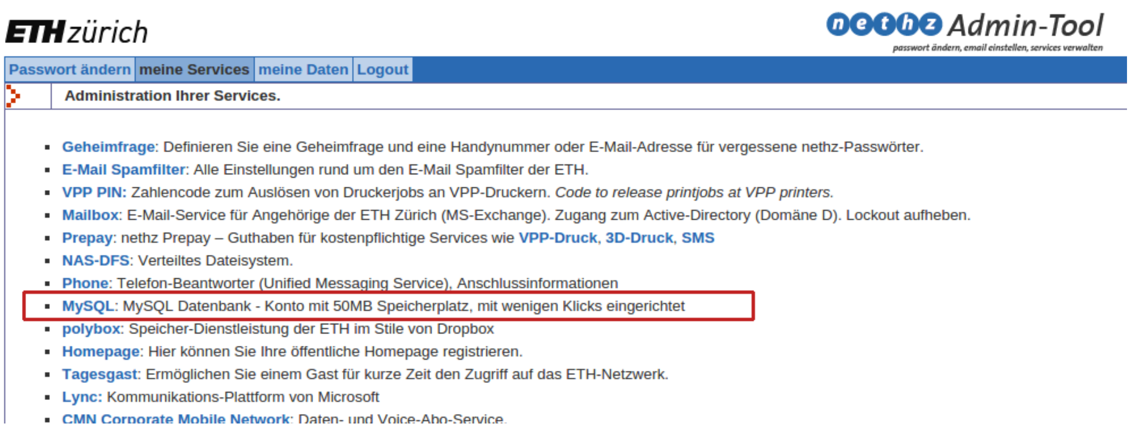
Step 2: Insert values into the newly created table is done via the 'INSERT INTO' command.

```
INSERT INTO person ( id, firstname , lastname , email)
VALUES
(NULL, 'Donald', 'Duck', 'donald.duck@disney.com'),
(NULL, 'Mickey', 'Mouse', 'mickey.mouse@disney.com');
```

Step 3: Query the database to get the email address of "Mickey Mouse".

```
SELECT email
FROM person
WHERE firstname = 'Mickey' AND lastname='Mouse';
```

Problem 11.4. SQL Queries

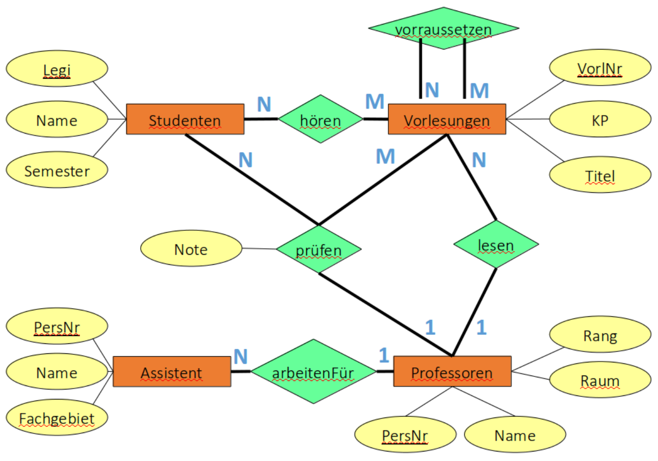


Administration Ihrer Services.

- **Geheimfrage**: Definieren Sie eine Geheimfrage und eine Handynummer oder E-Mail-Adresse für vergessene nethz-Passwörter.
- **E-Mail Spamfilter**: Alle Einstellungen rund um den E-Mail Spamfilter der ETH.
- **VPP PIN:** Zahlencode zum Auslösen von Druckerjobs an VPP-Druckern. *Code to release printjobs at VPP printers.*
- **Mailbox**: E-Mail-Service für Angehörige der ETH Zürich (MS-Exchange). Zugang zum Active-Directory (Domäne D). Lockout aufheben.
- **Prepay**: nethz Prepay – Guthaben für kostenpflichtige Services wie **VPP-Druck**, **3D-Druck**, **SMS**
- **NAS-DFS**: Verteiltes Dateisystem.
- **Phone**: Telefon-Beantworter (Unified Messaging Service), Anschlussinformationen
- **MySQL**: MySQL Datenbank - Konto mit 50MB Speicherplatz, mit wenigen Klicks eingerichtet
- **polybox**: Speicher-Dienstleistung der ETH im Stile von Dropbox
- **Homepage**: Hier können Sie Ihre öffentliche Homepage registrieren.
- **Tagesgast**: Ermöglichen Sie einem Gast für kurze Zeit den Zugriff auf das ETH-Netzwerk.
- **Lync**: Kommunikations-Plattform von Microsoft
- **CMN Corporate Mobile Network**: Daten- und Voice-Abo-Service

Abbildung 1: NETZ Administration

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

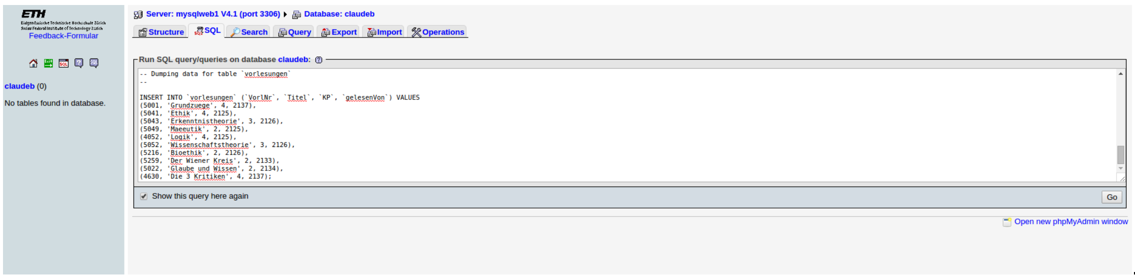


Given the university schema from the lecture, provide the SQL queries to answer the following questions:

1. Which is the average semester of all students?
2. Which students are enrolled in a higher semester than the average?
3. Return all the names and Id-Numbers of all the professors, assistants and students. In the case of professors and assistants, the Id-Number is the PersNr, and in the case of students, it is their Legi. Sort the result based on the Id-Number in descending order.
4. Provide the names of all students which visit at least one lecture of Sokrates.
5. Provide the name of all professors which give at least one lecture along with the total number of points (KP) of all their courses.
6. Which are the courses which require the lecture 'Grundzuege'? (We only consider direct successors! Bonus question: Can SQL deal with transitivity? If yes, how?)
7. Which student is the most popular? You can assume that students know each other from common courses. The more people know a particular student, the more popular he/she is. Sort the resulting list by the student's popularity.

It is a good idea to try out a lot of examples including the solutions of the above with the real database. To do so, you can install the university schema from the lecture as described below. A detailed description how to set up a database has been provided in exercise 3. We will now populate this database with the university schema.

1. Log in to your database at `https://phpmyadmin.ethz.ch/`. You can only access this page from within the ETH network. A VPN connection might be required if you are working remotely.
2. Download the university schema from the course webpage `http://lec.inf.ethz.ch/baug/informatik2/2017/code/Unidb.txt`.
3. In the database administration page, select your database (your NETHZ name) on the left side and then click on the 'SQL' tab. Copy and paste the content of the university schema into the text field and click 'Go'.

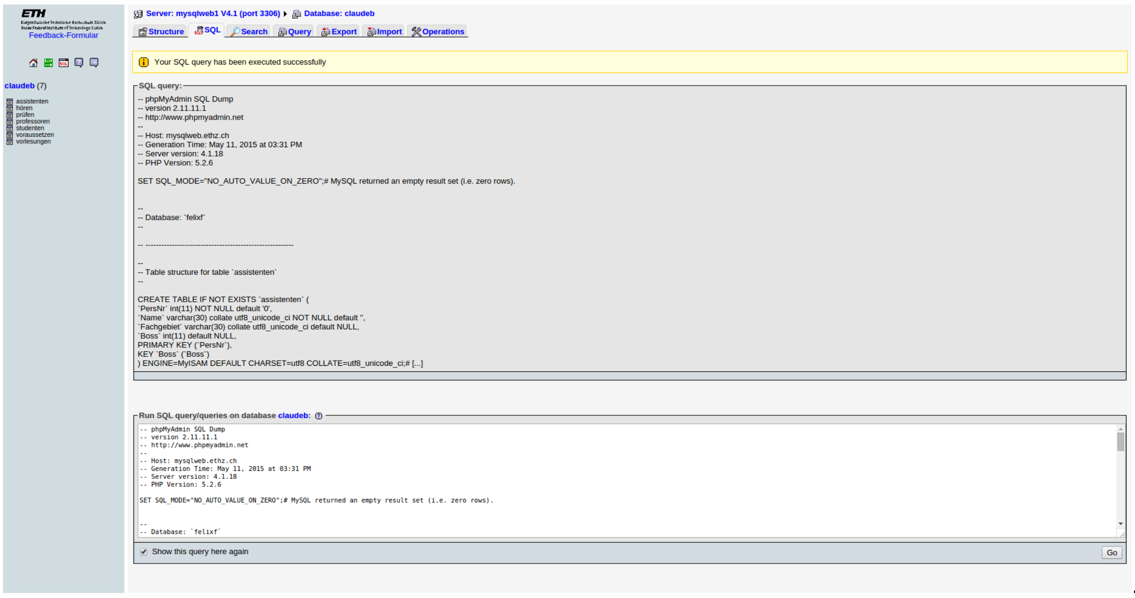

Feedback-Formular
Server: mysqlweb1 V4.1 (port 3306)
Database: claudeb
Structure
SQL
Search
Query
Export
Import
Operations
claudeb (0)
No tables found in database.
Run SQL query/queries on database claudeb:
Show this query here again
Go
Open new phpMyAdmin window

```
-- Dumping data for table `vorlesungen`
--

INSERT INTO `vorlesungen` (`VorlNr`, `Titel`, `KP`, `gelesenVon`) VALUES
(5001, 'Grundzuege', 4, 2137),
(5041, 'Ethik', 4, 2125),
(5043, 'Erkenntnistheorie', 3, 2126),
(5049, 'Maeeutik', 2, 2125),
(4052, 'Logik', 4, 2125),
(5052, 'Wissenschaftstheorie', 3, 2126),
(5216, 'Bioethik', 2, 2126),
(5259, 'Der Wiener Kreis', 2, 2133),
(5022, 'Glaube und Wissen', 2, 2134),
(4630, 'Die 3 Kritiken', 4, 2137);
```

.

4. The next page will inform you that the query has been executed. On the left side you should now see the individual table names.



Your SQL query has been executed successfully

```
-- phpMyAdmin SQL Dump
-- version 2.11.11.1
-- http://www.phpmyadmin.net
--
-- Host: mysqlweb.ethz.ch
-- Generation Time: May 11, 2015 at 03:31 PM
-- Server version: 4.1.18
-- PHP Version: 5.2.6

SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";# MySQL returned an empty result set (i.e. zero rows).


--
-- Database: `felixf`
--

-- --------------------------------------------------------

--
-- Table structure for table `assistenten`
--

CREATE TABLE IF NOT EXISTS `assistenten` (
`PersNr` int(11) NOT NULL default '0',
`Name` varchar(30) collate utf8_unicode_ci NOT NULL default '',
`Fachgebiet` varchar(30) collate utf8_unicode_ci default NULL,
`Boss` int(11) default NULL,
PRIMARY KEY (`PersNr`),
KEY `Boss` (`Boss`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;# [...]
```

```
-- phpMyAdmin SQL Dump
-- version 2.11.11.1
-- http://www.phpmyadmin.net
--
-- Host: mysqlweb.ethz.ch
-- Generation Time: May 11, 2015 at 03:31 PM
-- Server version: 4.1.18
-- PHP Version: 5.2.6

SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";# MySQL returned an empty result set (i.e. zero rows).


--
-- Database: `felixf`
```

.

Now you can test your solutions for question 1-7 with the real database.
Submission link: `https://codeboard.ethz.ch/inf2baugex11t04`

Solution of Problem 11.4.

1.
```
SELECT avg(s.Semester)
FROM Studenten s
WHERE 1
```

2.
```
SELECT *
FROM Studenten s
WHERE s.Semester > (
SELECT avg(s.Semester)
FROM Studenten s
WHERE 1)
```

3.
```
(SELECT PersNr as Id, Name FROM Professoren)
UNION
(SELECT PersNr as Id, Name FROM Assistenten)
UNION
(SELECT Legi as Id, Name FROM Studenten)
ORDER BY Id DESC
```

4.
```
SELECT DISTINCT s.Name
FROM Studenten s
JOIN hören h ON s.Legi = h.Legi
JOIN Vorlesungen v ON h.VorlNr = v.VorlNr
JOIN Professoren p ON v.gelesenVon = p.PersNr
WHERE p.Name = 'Sokrates'
```

5.
```
SELECT v.gelesenVon, p.Name, sum(v.KP)
FROM Vorlesungen v, Professoren p
WHERE v.gelesenVon = p.PersNr
GROUP BY v.gelesenVon, p.Name
```

6.
```
SELECT v2.Titel
FROM Vorlesungen v1
JOIN voraussetzen vor ON v1.VorlNr = vor.Vorgänger
JOIN Vorlesungen v2 ON vor.Nachfolger = v2.VorlNr
WHERE v1.Titel = 'Grundzuege'
```

Bonus: No SQL cannot handle transitivity. However, a lot of vendors offer custome extentions. Here the Oracle CONNECT BY clause:

```
SELECT Titel
FROM Vorlesungen
WHERE VorlNr in (
SELECT Vorgänger
FROM voraussetzen
CONNECTED BY Nachfolger = PRIOR Vorgänger
START WITH Nachfolger = (
SELECT VorlNr
from Vorlesungen
where Titel = '...' ));
```

7.
```
SELECT s.Legi, s.Name, count(*) as NumBekannter
FROM Studenten s, (
SELECT DISTINCT h1.Legi as Student, h2.Legi as Bekannter
FROM hören h1, hören h2
WHERE h1.VorlNr = h2.VorlNr AND h2.Legi <> h1.Legi
) b
WHERE s.Legi = b.Student
GROUP BY s.Legi, s.Name
ORDER BY NumBekannter DESC;
```