# INFORMATION SYSTEMS IN HEALTH CARE

Michel Kana, Ph.D.

**Lesson** 5 – Winter Term 2014

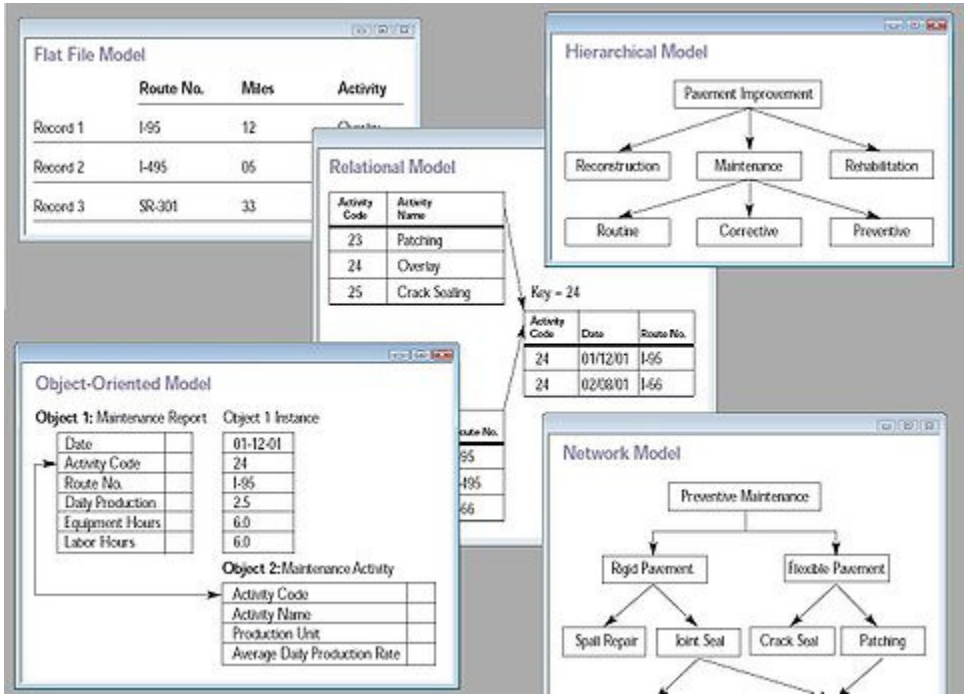
# Schedule

1. **Database theory**
2. **Database design**
3. **SQL**

# Database theory

- *A database is an information system for storing, processing and retrieving data.*

- Hierarchical database
- Network database
- **Relational database**
- Object-oriented database




Zdroj. Wikipedia.org

# Relational database

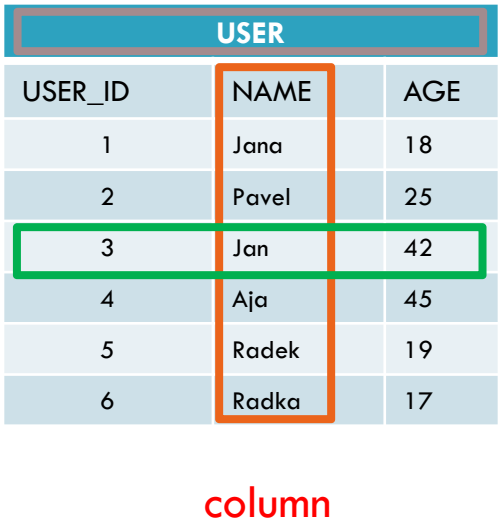



Table name

row

# Relational database

| USER | |
|---|---|
| USER_ID | Number |
| NAME | Text |
| AGE | Number |

USER (USER_ID number, NAME text, AGE number)

# Relational database – primary key

**A primary key** is a column or a group of column that uniquely identify the data in a table row.

| USER | | |
|---|---|---|
| USER_ID | NAME | AGE |
| 1 | Jana | 18 |
| 2 | Pavel | 25 |
| 3 | Jan | 42 |
| 4 | Aja | 45 |
| 5 | Radek | 19 |
| 6 | Radka | 17 |

| INVOICE | | |
|---|---|---|
| INVOICE_ID | USER_ID | VALUE |
| 1 | 1 | 400 |
| 2 | 1 | 100 |
| 3 | 6 | 800 |
| 4 | 5 | 300 |
| 5 | 6 | 200 |
| 6 | 4 | 400 |

# Relational database – foreign key

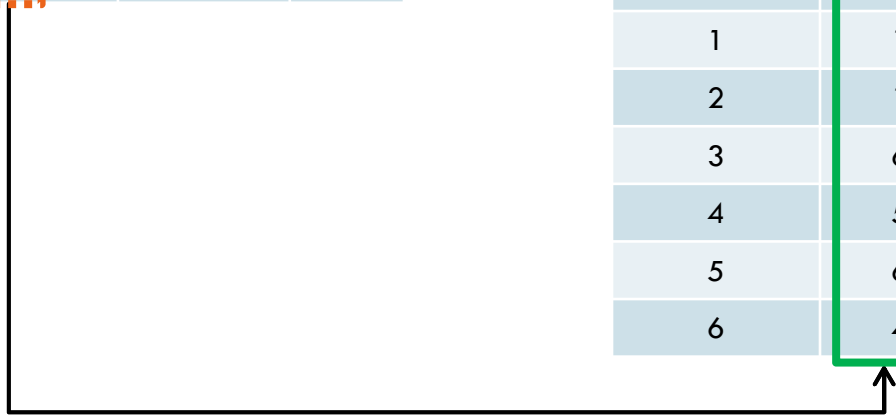
**USER**

| USER_ID | NAME | AGE |
|---|---|---|
| 1 | Jana | 18 |
| 2 | Pavel | 25 |
| 3 | Jan | 42 |
| 4 | Aja | 45 |
| 5 | Radek | 19 |
| 6 | Radka | 17 |

**A foreign key** links one or more columns of a table with columns of a foreign table.

**INVOICE**

| INVOICE_ID | USER_ID | VALUE |
|---|---|---|
| 1 | 1 | 400 |
| 2 | 1 | 100 |
| 3 | 6 | 800 |
| 4 | 5 | 300 |
| 5 | 6 | 200 |
| 6 | 4 | 400 |



# Database design

- Indentify tables, columns, primary keys and foreign keys
- Data requirements for a hospital health care system:
  - Team A
    - Each patient can have one or more insurances
  - Team B
    - There are two types of patient health records: department health records and ambulatory health records
  - Team C
    - A health records folder contains one or more health records of the same patient
    - A patient can have several health records folders
  - Team D
    - A health record folder can be borrowed by a department, a doctor or another hospital
    - There is a history about who borrowed which folders

# Database design

- Team A
  - Each patient can have one or more insurances
    - PATIENT (PATIENT_ID, PATIENT_NAME, PATIENT_AGE, PATIENT_GENDER)
    - INSURANCE (INSURANCE_ID, INSURANCE_NAME)
    - PATIENT_INSURANCE (PATIENT_ID, INSURANCE_ID)
- Team B
  - There are two types of patient health records: department health records and ambulatory health records
    - RECORD_TYPE (RECORD_TYPE_ID, RECORD_TYPE_NAME)
    - RECORD (RECORD _ID, PATIENT_ID, RECORD _TYPE_ID, RECORD _DATE)
- Team C
  - A health records folder contains one or more health records of the same patient. A patient can have several health records folders
    - FOLDER (FOLDER_ID, FOLDER_NAME, PATIENT_ID)
    - FOLDER_RECORD (FOLDER _ID, RECORD _ID)
- Team D
  - A health record folder can be borrowed by a department, a doctor or another hospital
    - BORROWER_TYPE (BORROWER_TYPE_ID, BORROWER_TYPE_NAME)
    - BORROWER (BORROWER_ID, BORROWER _TYPE_ID, BORROWER _NAME)
  - There is a history about who borrowed which folders
    - HISTORY (HISTORY_ID, BORROWER_ID, FOLDER_ID, HISTORY_DATE)

# SQL

Structured Query Language is a declarative language that serves the following purposes:

- Data structure definition
  - **CREATE**
  - **ALTER**
  - **DROP**
- Data manipulation
  - Storing data
    - **INSERT**
  - Actualizing data
    - **UPDATE**
  - Deleting data
    - **DELETE**
  - Querying data
    - **SELECT**
- Database administration
  - Managing access rights
    - **GRANT**
  - Managing transactions
    - **START TRANSACTION, COMMIT, ROLLBACK**

# SQL – Data Definition Language

- Database definition
  - **CREATE DATABASE db_name;**
- Table definition
  - **CREATE TABLE table_name (column_name1 column_type1, column_name2 column_type2, … PRIMARY KEY (column_name1, …) );**
  - Data types: **INT, VARCHAR(length), DATE**
  - Primary key: **PRIMARY KEY**
- Create the database **BMT_DB**
- Team A – Table definition
  - **PATIENT** (PATIENT_ID, PATIENT_NAME, PATIENT_AGE, PATIENT_GENDER)
  - **INSURANCE** (INSURANCE_ID, INSURANCE_NAME)
  - **PATIENT_INSURANCE** (PATIENT_ID, INSURANCE_ID)
- Team B – Table defintion
  - **RECORD_TYPE** (RECORD_TYPE_ID, RECORD_TYPE_NAME)
  - **RECORD** (RECORD _ID, PATIENT_ID, RECORD _TYPE_ID, RECORD _DATE)
- Team C – Table definition
  - **FOLDER** (FOLDER_ID, FOLDER_NAME, PATIENT_ID)
  - **FOLDER_RECORD** (FOLDER _ID, RECORD _ID)
- Team D – Table definition
  - **BORROWER_TYPE** (BORROWER_TYPE_ID, BORROWER_TYPE_NAME)
  - **BORROWER** (BORROWER_ID, BORROWER _TYPE_ID, BORROWER _NAME)
  - **HISTORY** (HISTORY_ID, BORROWER_ID, FOLDER_ID, HISTORY_DATE)

# SQL – Data Definition Language

- Create the database **BMI_DB**
  - **CREATE DATABASE BMI_DB;**
- Team A
  - CREATE TABLE **patient** ( patient_id INT PRIMARY KEY, patient_name VARCHAR(80), patient_age INT, patient_gender INT );
  - CREATE TABLE **insurance** ( insurance_id INT PRIMARY KEY, insurance_name VARCHAR(160));
  - CREATE TABLE **patient_insurance** ( patient_id INT, insurance_id INT, PRIMARY KEY (patient_id, insurance_id) );
- Team B
  - CREATE TABLE **record_type** ( record_type_id INT PRIMARY KEY, record_type_name VARCHAR(30) );
  - CREATE TABLE **record** ( record_id INT PRIMARY KEY, patient_id INT, record _type_id INT, record _date DATE );
- Team C
  - CREATE TABLE **folder** ( folder_id INT PRIMARY KEY, folder_name VARCHAR(80), patient_id INT );
  - CREATE TABLE **folder_record** ( folder_id INT, record_id INT, PRIMARY KEY (folder_id, record_id) );
- Team D
  - CREATE TABLE **borrower_type** ( borrower_type_id INT PRIMARY KEY, borrower_type_name VARCHAR(80) );
  - CREATE TABLE **borrower** ( borrower_id INT PRIMARY KEY, borrower _type_id INT, borrower _name VARCHAR(160) );
  - CREATE TABLE **history** ( history_id INT PRIMARY KEY, borrower_id INT, folder_id INT, history_date DATE );

# SQL – Data Manipulation Language

- Storing data
  - **INSERT INTO table_name (column_name1, column_name2, …)**
    **VALUES (value_column1, value_column2, …);**
- Team A – store a patient with 1 health insurance and a patient with 2 health insurances
  - **PATIENT** (PATIENT_ID, PATIENT_NAME, PATIENT_AGE, PATIENT_GENDER)
  - **INSURANCE** (INSURANCE_ID, INSURANCE_NAME)
  - **PATIENT_INSURANCE** (PATIENT_ID, INSURANCE_ID)
- Team B – store a department health record and an ambulatory health record for both patients
  - **RECORD_TYPE** (RECORD_TYPE_ID, RECORD_TYPE_NAME)
  - **RECORD** (RECORD _ID, PATIENT_ID, RECORD _TYPE_ID, RECORD _DATE)
- Team C – store 2 health records folders for each patient
  - **FOLDER** (FOLDER_ID, FOLDER_NAME, PATIENT_ID)
  - **FOLDER_RECORD** (FOLDER _ID, RECORD _ID)
- Team D – store 1 doctor and two loans of a health records folder for each patient
  - **BORROWER_TYPE** (BORROWER_TYPE_ID, BORROWER_TYPE_NAME)
  - **BORROWER** (BORROWER_ID, BORROWER _TYPE_ID, BORROWER _NAME)
  - **HISTORY** (HISTORY_ID, BORROWER_ID, FOLDER_ID, HISTORY_DATE)

# SQL – Data Manipulation Language

- Team A
  - INSERT INTO **patient** (patient_id, patient_name, patient_age, patient_gender) VALUES (1, 'Jakub Janda', 25, 'male');
  - INSERT INTO **insurance** (insurance_id, insurance_name) VALUES (1, 'Všeobecná zdravotní pojišťovna');
  - INSERT INTO **patient_insurance** (patient_id, insurance_id) VALUES (1, 1);
- Team B
  - INSERT INTO **record_type** (record_type_id, record_type_name) VALUES (1, 'oddělení'), (2, 'ambulantní');
  - VALUES INSERT INTO **record** (record _id, patient_id, record _type_id, record _date) VALUES (1, 1, 1, now());
- Team C
  - INSERT INTO **folder** (folder_id, folder_name , patient_id) VALUES (1, 'Složka záznamů pro Jakuba Jandu', 1);
  - INSERT INTO **folder_record** (folder _id, record _id) VALUES (1, 1);
- Team D
  - INSERT INTO **borrower_type** (borrower_type_id, borrower_type_name)
    VALUES (1, 'oddělení'), (2, 'lékař'), (3, 'nemocnice');
  - INSERT INTO **borrower** (borrower_id , borrower _type_id, borrower _name ) VALUES (1, 2, 'MUDr. Jan Malík');
  - INSERT INTO **history** (history_id , borrower_id, folder_id, history_date) VALUES (1, 1, 1, now());

# SQL – Data query

- Querying data
  - **SELECT** column_name1, column_name2, …
    **FROM** table_name
    [**WHERE** conditions]
    [**GROUP BY** column_name1, column_name2, …]
    [**HAVING** conditions]
    [**ORDER BY** column_name1, column_name2, …]
    [**LIMIT** row_count];

**PATIENT** (PATIENT_ID, PATIENT_NAME, PATIENT_AGE, PATIENT_GENDER)

**INSURANCE** (INSURANCE_ID, INSURANCE_NAME)

**PATIENT_INSURANCE** (PATIENT_ID, INSURANCE_ID)

**RECORD_TYPE** (RECORD_TYPE_ID, RECORD_TYPE_NAME)

**RECORD** (RECORD _ID, PATIENT_ID, RECORD _TYPE_ID, RECORD _DATE)

**FOLDER** (FOLDER_ID, FOLDER_NAME, PATIENT_ID)

**FOLDER_RECORD** (FOLDER _ID, RECORD _ID)

**BORROWER_TYPE** (BORROWER_TYPE_ID, BORROWER_TYPE_NAME)

**BORROWER** (BORROWER_ID, BORROWER _TYPE_ID, BORROWER _NAME)

**HISTORY** (HISTORY_ID, BORROWER_ID, FOLDER_ID, HISTORY_DATE)

- **Team A**
  - Total number of patients
  - Insurance of patient Jakub Janda
  - Folders borrowed by doctor MUDr. Malík
- **Team B**
  - Total number of male patients
  - Ambulantory health records of patient Jakub Janda
  - All patients without folders
- **Team C**
  - Folders which were borrowed by a hospital
  - Oldest patient
  - Last date when a folder of patient Klára Bíla was borrowed
- **Team D**
  - Number of health insurances for patient Ludmila Černa
  - All health records for patient Jana Hezka
  - All loans of health records for patient Švejk which were made by doctors and hospitals

# SQL – Data query

**PATIENT** (PATIENT_ID, PATIENT_NAME, PATIENT_AGE, PATIENT_GENDER)

**INSURANCE** (INSURANCE_ID, INSURANCE_NAME)

**PATIENT_INSURANCE** (PATIENT_ID, INSURANCE_ID)

**RECORD_TYPE** (RECORD_TYPE_ID, RECORD_TYPE_NAME)

**RECORD** (RECORD _ID, PATIENT_ID, RECORD _TYPE_ID, RECORD _DATE)

**FOLDER** (FOLDER_ID, FOLDER_NAME, PATIENT_ID)

**FOLDER_RECORD** (FOLDER _ID, RECORD _ID)

**BORROWER_TYPE** (BORROWER_TYPE_ID, BORROWER_TYPE_NAME)

**BORROWER** (BORROWER_ID, BORROWER _TYPE_ID, BORROWER _NAME)

**HISTORY** (HISTORY_ID, BORROWER_ID, FOLDER_ID, HISTORY_DATE)

- **Team A**
  - Total number of patients

    ```
    select count(*) from patient
    ```
  - Insurance of patient Jakub Janda

    ```
    select insurane.insurance_name from insurance, patient, patient_insusurance
    where insurance.insurance_id = patient_insurance.insurance_id and    patient_insurance.patient_id =
    patient.patient_id and    patient.patient_name = 'Jakub Janda‘
    ```
  - Folders borrowed by doctor MUDr. Malík

    ```
    select folder.folder_name from folder, borrower, history
    where folder.folder_id = history.folder_id
    and    history.borrower_id = borrower.borrower_id
    and borrower.borrow_name = 'MUDr. Malik‘
    ```

# SQL – Data query

**PATIENT** (PATIENT_ID, PATIENT_NAME, PATIENT_AGE, PATIENT_GENDER)

**INSURANCE** (INSURANCE_ID, INSURANCE_NAME)

**PATIENT_INSURANCE** (PATIENT_ID, INSURANCE_ID)

**RECORD_TYPE** (RECORD_TYPE_ID, RECORD_TYPE_NAME)

**RECORD** (RECORD _ID, PATIENT_ID, RECORD _TYPE_ID, RECORD _DATE)

**FOLDER** (FOLDER_ID, FOLDER_NAME, PATIENT_ID)

**FOLDER_RECORD** (FOLDER _ID, RECORD _ID)

**BORROWER_TYPE** (BORROWER_TYPE_ID, BORROWER_TYPE_NAME)

**BORROWER** (BORROWER_ID, BORROWER _TYPE_ID, BORROWER _NAME)

**HISTORY** (HISTORY_ID, BORROWER_ID, FOLDER_ID, HISTORY_DATE)

- **Team B**
  - Total number of male patients

    ```
    select * from patient where gender='male'
    ```
  - Ambulantory health records of patient Jakub Janda

    ```
    select record.* from record, record_type, patient
    where record.record_type_id = record_type.record_type_id
    and    record.patient_id = patient.patient_id
    and record_type.record_type_name = 'ambulantní‘
    and patient.patient_name = 'Jakub Janda‘
    ```
  - All patients without folders

    ```
    select patient.patient_name, count(*) from folder, patient
    where folder.patient_id = patient.patient_id
    group by patient.patient_name
    having count(*)=0
    ```

# SQL – Data query

**PATIENT** (PATIENT_ID, PATIENT_NAME, PATIENT_AGE, PATIENT_GENDER)

**INSURANCE** (INSURANCE_ID, INSURANCE_NAME)

**PATIENT_INSURANCE** (PATIENT_ID, INSURANCE_ID)

**RECORD_TYPE** (RECORD_TYPE_ID, RECORD_TYPE_NAME)

**RECORD** (RECORD _ID, PATIENT_ID, RECORD _TYPE_ID, RECORD _DATE)

**FOLDER** (FOLDER_ID, FOLDER_NAME, PATIENT_ID)

**FOLDER_RECORD** (FOLDER _ID, RECORD _ID)

**BORROWER_TYPE** (BORROWER_TYPE_ID, BORROWER_TYPE_NAME)

**BORROWER** (BORROWER_ID, BORROWER _TYPE_ID, BORROWER _NAME)

**HISTORY** (HISTORY_ID, BORROWER_ID, FOLDER_ID, HISTORY_DATE)

- **Team C**
  - Folders which were borrowed by a hospital

```
select folder.* from folder, borrower, borrower_type, history
where folder.folder_id = history.folder_id
and    history.borrower_id = borrower.borrower_id
and    borrower. borrower_type_id = borrower_type. borrower_type
and    borrower_type. borrower_type_name = 'nemocnice‘
```

  - Oldest patient

```
select * from patient order by patient_age desc limit 1
```

  - Last date when a folder of patient Klára Bíla was borrowed

```
select max(history.date) from history, folder, patient
where history.folder_id = folder.fodler_id
and    folder.patient_id = patient.patient_id
and    patient.patient_name = 'Klara Bila‘
```

# SQL – Data query

**PATIENT** (PATIENT_ID, PATIENT_NAME, PATIENT_AGE, PATIENT_GENDER)

**INSURANCE** (INSURANCE_ID, INSURANCE_NAME)

**PATIENT_INSURANCE** (PATIENT_ID, INSURANCE_ID)

**RECORD_TYPE** (RECORD_TYPE_ID, RECORD_TYPE_NAME)

**RECORD** (RECORD _ID, PATIENT_ID, RECORD _TYPE_ID, RECORD _DATE)

**FOLDER** (FOLDER_ID, FOLDER_NAME, PATIENT_ID)

**FOLDER_RECORD** (FOLDER _ID, RECORD _ID)

**BORROWER_TYPE** (BORROWER_TYPE_ID, BORROWER_TYPE_NAME)

**BORROWER** (BORROWER_ID, BORROWER _TYPE_ID, BORROWER _NAME)

**HISTORY** (HISTORY_ID, BORROWER_ID, FOLDER_ID, HISTORY_DATE)

- **Team D**
  - Number of health insurances for patient Ludmila Černa

```
select count(*) from insurance, patient
where insurance.insurance_id = patient.insurance_id
and patient.patient_name = 'Ludmila Cerna‘
```

  - All health records for patient Jana Hezka

```
select * from record, patient
where record.patient_id = patient.patient_id
and     patient.patient_name = 'Jana Hezka'
```

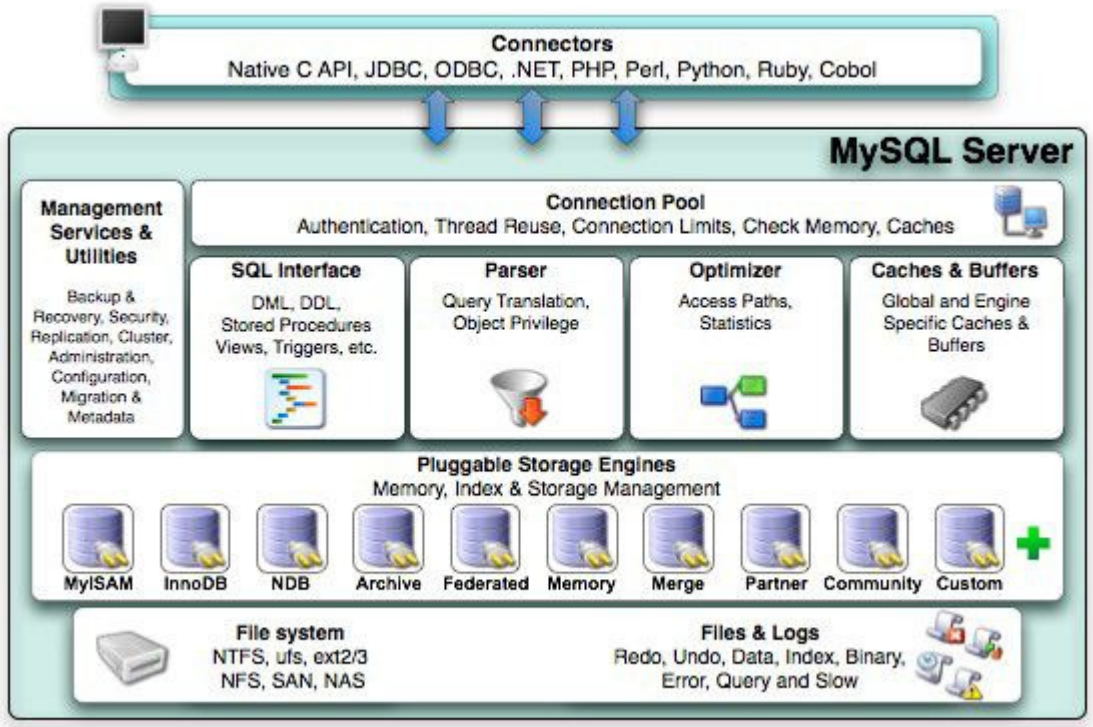
  - All loans of health records for patient Švejk which were made by doctors and hospitals

```
select * from history, borrower, borrower_type, patient, folder
where history.borrower_id = borrower.borrower_id
and    borrower.borrower_type_id = borrower_type.borrower_type_id
and    history.folder_id = folder.folder_id
and    folder.patient_id = patient.patient_id
and    borrower_type.borrower_type_name in ('doctor', 'hospital')
and    patient.patient_name = 'Svejk'
```

# Database Systems

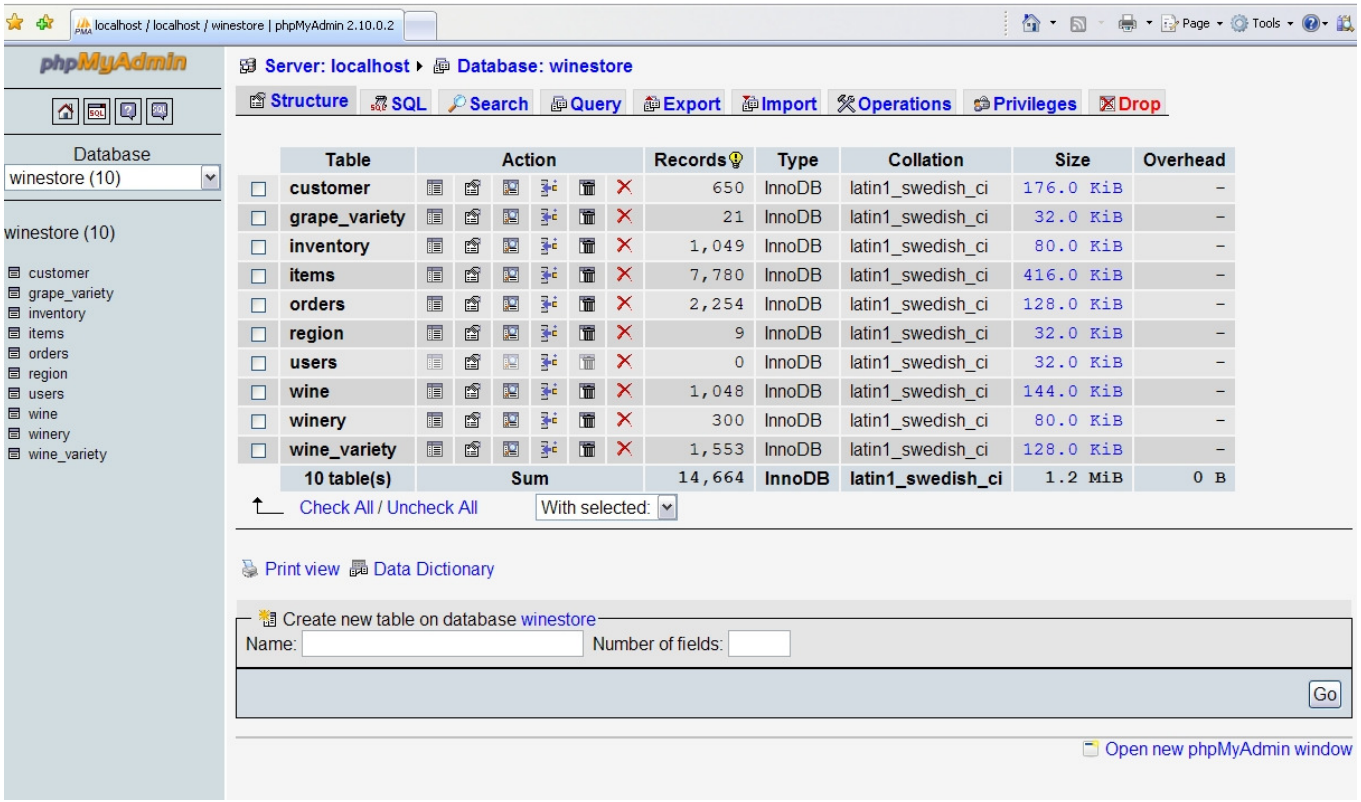
- Commercial database systems
  - Oracle
  - Informix Dynamic Server
  - Microsoft SQL Server
  - IBM DB2
  - Sybase, Teradata, SQL Anywhere
- Open source database systems
  - MySQL
  - PostgreSQL
  - Firebird
  - Ingres

# Database system MySQL




Source. mysql.com

# Database system MySQL - phpMyAdmin




Source. mysql.com

# Homework

- webstu.fbmi.cvut.cz – personal server
- databaze.fbmi.cvut.cz – server for this homework
- Account activation:
  1. FTP Connection (create your working directory):
     - Your PC -> ftp://databaze.fbmi.cvut.cz
     - Use your faculty login and password
  2. Open the admin web page
     - http://databaze.fbmi.cvut.cz/yourlogin
  3. Click on the link Vytvořit databázi for creating your database
  4. Your should receive an email on your faculty account with login details to MySQL

# Homework

- Create and insert illustrative data into the following tables in your faculty or personal database
  - **PATIENT** (PATIENT_ID, PATIENT_NAME, PATIENT_AGE, PATIENT_GENDER)
  - **INSURANCE** (INSURANCE_ID, INSURANCE_NAME)
  - **PATIENT_INSURANCE** (PATIENT_ID, INSURANCE_ID)
  - **RECORD_TYPE** (RECORD_TYPE_ID, RECORD_TYPE_NAME)
  - **RECORD** (RECORD _ID, PATIENT_ID, RECORD _TYPE_ID, RECORD _DATE)
  - **FOLDER** (FOLDER_ID, FOLDER_NAME, PATIENT_ID)
  - **FOLDER_RECORD** (FOLDER _ID, RECORD _ID)
  - **BORROWER_TYPE** (BORROWER_TYPE_ID, BORROWER_TYPE_NAME)
  - **BORROWER** (BORROWER_ID, BORROWER _TYPE_ID, BORROWER _NAME)
  - **HISTORY** (HISTORY_ID, BORROWER_ID, FOLDER_ID, HISTORY_DATE)
- Write SQL queries for the following tasks and run them against your database:
  - Folders with the highest number of ambulantory records for pacient Klara Novakova
  - Hospital with the highest number of rentals of folders for patients younger than 30 years old
  - Insurance with 100 or more folder rentals made by doctors
  - Records older than 10 days for pacients insured at VZP. For each record, display the total number of folders where the record appears, the total number of departments who rent those folders, the total number of rentals of those folders
- Send an export of your database together with the queries by Oct 26
- Install OpenMRS on your Laptop