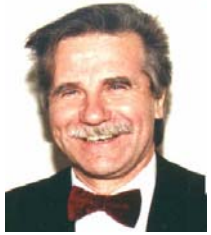
Database and Knowledge-base Systems

Range: 3+1, 5 credits, finished by examination

Lecturer: Prof. Ing. Jiří Bíla, DrSc.

(bila@vc.cvut.cz)

(Vice-Rector for External Relations, CTU)



Other teachers:

ing. Vladimír Hlaváč
(hlavac@fs.cvut.cz)

doc. ing. Ivo Bukovský, Ph.D.
(ivo.bukovsky@fs.cvut.cz)

Database and Knowledge-base Systems

First part of the subject: **Databases**

taught in range 3+1 "in a block"

Lectures – on Wednesday, at 14:15, room 308

Exercises (training) – after lectures, as needed

Coffee break – inside lecture (approx. 15:45)

Exercises – MS Access – personal database

– SQL – client-server system

– last one – first mid-term classification

Computer classroom T4:C1-**308** (2001).

Literature for the 1st part

Web page of this course:

www.fs.cvut.cz/prt/DKS/

Links to on-line books on this page:

Pallaw, Vijay Krishna: Database Management Systems. Readable only from the school campus in Dejvice.

Isrd Group: Introduction to Database Management Systems. On-line on Google books, look by Tata McGraw Hill code: radqcdrkrxbqb .

Many terms are explained on the Wikipedia:

http://en.wikipedia.org/wiki/Database/

Data

Data is what we will save in our database

Information: response to the question

Amount of information

Redundancy

Noise

Data encoding (ASCII, binary)

ASCII – see http://www.lookuptables.com/

UTF-8 – standard in new databases

Data meaning (denotation) depends on the database

Example: Physics: 42 inches, 42 meters, 42 PSI, 42 kW

Record

Basic method how to store data.

Example:

Wolf	**John**	**314**	**2737**	**volfjaro**

Record represents information about a real object.

Consistency

Consistency and Redundancy

Consistency check function

Database – how to organize records

Database Management System (DBMS)

DBMS is system, which separates data and application.

DBMS typically provides:

- Data redundancy
- Physical data independence
- Logical data independence
- High level language

Database with pointers

Pointer is typically part of the record

- often bound with item of record
- if position for a pointer is free, there will be special symbol or value (text)
- it can create a structure, so if we can access one record, there is a way, how to reach each other record (even if could be rather complicated to find it)
- typical structure could be „no structure“ (general structure, it creates **network model**), or **hierarchical** (tree).

Network database

Example: Database with contacts to companies

(two parts of the record:
- what we can do, - list of other companies)

Masonry	Victor Bruck	01476 25654
Carpenter	John Smith	01476 31175
Decorator	Anne White	01476 23857
Penguin, Barkston		*(pointer)*
Brown Castle, Grantham		*(pointer)*
Rubicon, Ropslay		*(pointer)*
Whitestone, Ancaster		*(pointer)*

Network database

Example: Originally proposed World Wide Web structure

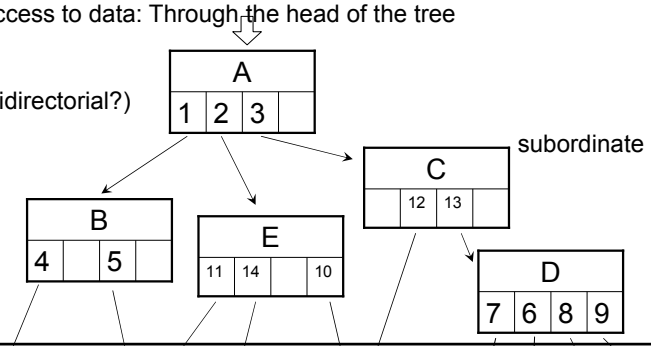
Hierarchical database

Tree contains data & pointers

There could be defined number of pointers in a record

Access to data: Through the head of the tree

(bidirectorial?)



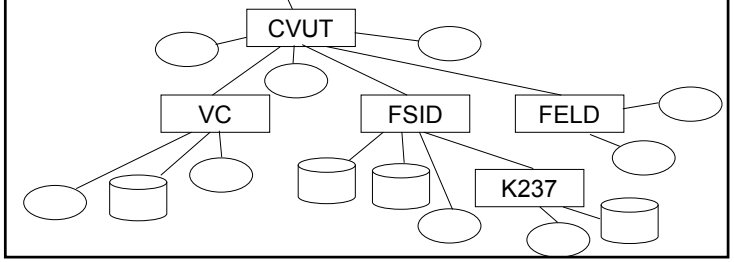
Ex.: X.400 and NDS (Novell Netware)

Tree head is a country (only logical)

2nd level: Organization

3rd level: Organization unit (this one could be recoursed)

last level – Leaves : Each leaf has common name



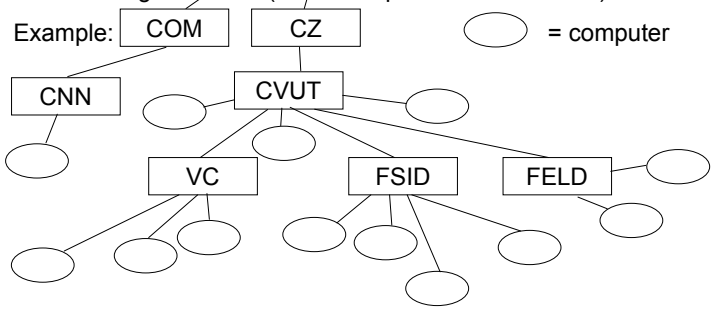
Ex.: ip nameserver

Tree head is so called DNS root zone (13 servers)

2nd level: top level domains (.com, .edu, countries)

3rd level: Organizations (could be split into subdomains)

Example:



Relational database

Each line in the table is a relation

Each line describes properties of some „real“ object (i.e. payment, book borrow, tv program)

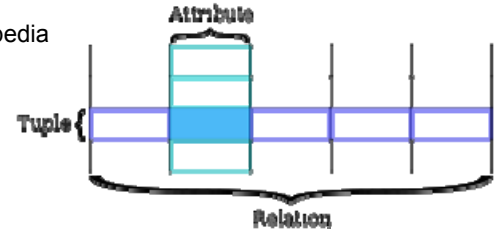
Obviously there are relations between tables

Surname	*Firstname*	*Room*	*Phone*
Volf	Jaromir	314	2737
Bukovský	Ivo	433	2529
Novák	Martin	318	2573

Consistency = each line represents „real“ object

Relational database theory

Source: Wikipedia



Terminology:

tuple is an ordered list of elements. In **set** theory, an (ordered) n-tuple is a sequence (or ordered list) of n elements, where n is a positive integer *(they are „ordered“ in our case)*.

In SQL, we use term „row“ (and „column“ for attribute)

Column type

The format, type and **size** (how many letters, or range of value) for each column is defined

If inserted text is longer, it is cut

If text is shorter, the current length is stored into table

If number is out of defined range, error is returned

Typical type is number, integer number, many variant of a string (text), and date in different formats.

Primary key

Use: for unique identification of each line

If a column contain primary key, there are not the same value in two rows

Typical solution is integer number with the „autoincrement“

id	*Surname*	*Firstname*	*Room*	*Phone*
1	Volf	Jaromír	314	2737
2	Bukovský	Ivo	433	2529
3	Novák	Martin	318	2573

Foreign key

For connecting two tables (to create relation), the best way is to include the primary key value from the depended table to other tables.

Example: List of examination can contain only ID of examiner. When a student try to list all exams, the examiner ID will be replaced by real name from the table (student will not see the examiner ID). If student will register for the particular exam, the students ID will be inserted to the term. The exam table then contains only ID of examiner and of the students, not the real names.

Codd rules

In the 1985 Edgar E. Codd express 12 rules, which define, if a database works as relational

More info on the Wikipedia:

http://en.wikipedia.org/wiki/Codd_rules

Note. Database security.

Relational database example

This is the most frequent type of database.

Typical representation on PC:

DBase (1979)

FoxBase, FoxPro (1989, MS 1992)

Paradox (1985)

MS Access (1992)

Data sharing can be solved by means of a LAN.

#file sharing, record lock, dbase (.dbf) file structure.

Relational database example

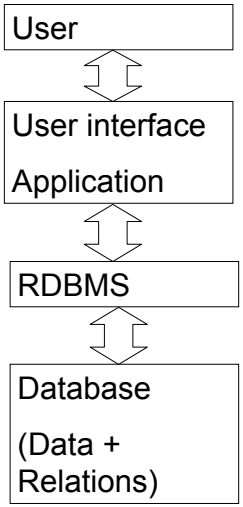
The most of typical database solutions now use SQL.

SQL – Structured Query Language (SEQUEL, 1970)

- not exactly relational database
- in implementation, data are not stored in tables (independent records are used)
- there are no defined position of data (rows) we get by query, if we need some order, we need to ask to sort
- Database management system is independent on user interface, it works by the Client-Server method even on the same computer
- Scalable – it is easy to use for a “big” application

Client - Server

(how it works)



Note: Catalog.

How a Database Works

External view

Conceptual scheme
- table contents
- relation between tables
- prepared functions

Internal level

Relational model - example

see dks1a.ppt

how to describe:

UML = Unified Modeling Language

E – R diagram

RM = relational model (for example, in MS Access)

Database design

- Data analysis
- Requirement analysis (input, output)
- Data structure (tables, relations)

Note – we can add a column or even a table, but we cannot change later the concept of database (structure).

Examples

tasks to be solved:

Car rental service

Book library reservation

Small on-line shop

Hairdresser or dentist reservation

Task #1

Create relational database on computers in computer classroom 308, using MySQL server and any client.

Task #2

Create the same database, using MS Access 2013.