Databases

dr. Jörg Verstraete

Polish Academy of Sciences
Systems Research Institute

jorg.verstraete@ibspan.waw.pl
http://www.mini.pw.edu.pl/~verstraetej/
Contents

- Introduction
- Relational database systems
- Database design
- Normalisation
- SQL
- Data
- OLTP systems, transactional processing, locks
- Stored procedures, triggers, functions, …
- GIS systems
- Data warehouse, OLAP, Business intelligence
- Java based client applications and database APIs
- …
Practical matters

• Lectures
  – Friday, 10.00 (+Wednesday 23/03, 15/06)
    • room 102
• Labs
  – Friday, 12.00 / Friday, 14.00 / Friday, 16.00
    • room 302
• Evaluations
  – 3-4 assignments to be prepared during the labs
  – exam
  – Final grade depends on total points
Practical matters

• E-mail
  – jorg.verstraete@ibspan.waw.pl

• Website
  – http://www.mini.pw.edu.pl/~verstraetej
    • Announcements
    • Slides, tasks, solutions, files, links, ...
Today

• Lecture
  – General introduction in database concepts
  – Terminology
  – Models

• Labs
  – MS SQL Server 2014
    • user interface
    • practical implementation of concepts
    • environment for SQL labs, focus on generic SQL
Database

• Definition
  - Set of **positive** facts concerning the real world, from the **universe of discourse**, stored in a **persistent** manner.
  - Universe of discourse
  - Positive information
  - Persistent

For most databases, the content can be imagined as a set of tables containing the description of real world facts

<table>
<thead>
<tr>
<th>Building name</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Building</td>
<td>Room</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Chemistry</td>
<td>Room</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Only existing rooms are stored
Database - example

- Real world information is filtered and checked
- Many databases are possible for one organisation: different universes of discourse
Database system and DBMS

• Database System
  – “Cost-effective method for storing, organising, retrieving and managing data” (J.C. Shepherd)

• Database Management System (DBMS)
  – Computer system for managing a database
    • Hardware
    • Software
    • Data
  – Two layer architecture
    • Internal layer: interfaces with physical devices / hardware
    • Logical layer: data
DBMS

• Database Management System
  – Typical representation
    • Set of disk files or partitions containing data
    • Appropriate organization to facilitate and speed up information retrieval and update
  – Specializations
    • Distributed databases
    • Spatial databases
    • Multimedia databases
    • Data warehouses
    • Data mining applications
Database system and DBMS

- **Database Management System (DBMS)**
  - Interacts between user applications and database to provide access to the data
  - Supports administrative tasks of DBA, which includes creating and manipulating databases.

  - **components**
    - DBMS, application software, tools, communication software, data-warehousing, data analysis software, report generators

  - **examples**
    - SQL Server, Oracle, MySQL, PostGRES, DB2, Sybase, ...
DBMS

• Main functionality
  – Database definition
  – Database manipulation
  – Database construction

• Secondary functionality
  – Sharing data
  – Securing data
  – Optimization
  – Administration and verification
Users

- Data administrator (DA)
  - Responsible for data
    - design of the database, strategies, user profiles, security, processing, etc.

- Database administrator (DBA)
  - Responsible for the database
    - consistency, performance checking, performing backup/restore,

- Software developer

- End user
  - simple: access via application (limited complexity)
  - advanced: access via database language (full complexity)
Database and DBMS

- Data processing teams
- DBA
- Software developer
- End users

Utility programs (e.g. security manager)

Application programs (e.g. sales applications)

DBMS

Database

Database
## Why a database?

<table>
<thead>
<tr>
<th>Flat files (TXT, XML, CSV, ...)</th>
<th>Database</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Performance</strong></td>
<td>Special structures (indexes) speed up information retrieval</td>
</tr>
<tr>
<td>Limited: slow to find the data a user is looking for.</td>
<td>Radix trie, Patricia Trie</td>
</tr>
<tr>
<td>Queries</td>
<td>Any query can be answered (e.g. all the clients who bought product A last week, but never ordered more than 50 items of product B)</td>
</tr>
<tr>
<td>Only simple queries</td>
<td></td>
</tr>
<tr>
<td>! regular expressions</td>
<td></td>
</tr>
<tr>
<td>Concurrency</td>
<td>Constant updates can be performed on the fly. Transactional processing ensures consistency</td>
</tr>
<tr>
<td>Difficult to concurrently apply changes on behalf of many users</td>
<td></td>
</tr>
<tr>
<td>Security</td>
<td>Complicated permission system secures the data</td>
</tr>
<tr>
<td>Anyone with access to the file can modify/damage it</td>
<td></td>
</tr>
<tr>
<td>Robustness</td>
<td>Various backup strategies possible, e.g. incremental backup</td>
</tr>
<tr>
<td>Difficult to make a backup of a file that is constantly modified</td>
<td></td>
</tr>
</tbody>
</table>

Most applications need to store data permanently. Whenever multiple users share the data, a database is needed.
Database categories

- Historical
  - Operational models
    - inverted list, hierarchical model, network model, ...
  - Structural models
    - semi-relational, relational, ...
  - Semantic models
    - RM/T, object-relational, object-oriented, ...
  - Other models
    - spatio-temporal, deductive, fuzzy and uncertain, ...
Database categories

• Based on how data is managed
  – DBMS engine embedded in a database application
  – DBMS server application that works on behalf of database applications and is accessible over the network
Traditional file based approach

• Example

[Diagram showing file server connected to databases through network communication]

- Sales
  - DBMS engine
- Inventory
  - DBMS engine
- Accounting
  - DBMS engine

All database applications access the database through an embedded DBMS engine, a software library.

The database in this case is just storage, e.g. a collection of files.
File-based approach

- Applications access the database using embedded libraries:
  - data is typically transferred over the network for updates/queries
  - poor scalability for bigger databases
    - possible network congestion
    - higher requirements to clients (CPU, …)
  - limited security
    - files can be copied and or modified by the users
    - network problems may result in partial updates and/or damaged file structure
Client-server approach

- Example

Server or server cluster

- e.g. MS Windows, UNIX, Linux, ...

SQL statements (local or over network)

- Clients interact with the database server, a server side application that fulfills requests by the applications and modifies/queries the data on their behalf.

- Sales
- Inventory
- Accounting

- Database server (DBMS engine)

- Database
Client-sever approach

- Applications access the database by communicating with the database server:
  - Only queries and query results are transferred over the network.
  - Improved scalability for bigger databases
    - limited increase in network
    - increased demand in computing resources is server side as processing happens on the server
  - Security
    - server can impose security constraints and permissions
    - server can insure data integrity
# File-based and client-server DBMS

<table>
<thead>
<tr>
<th></th>
<th>File based (historically mostly)</th>
<th>Client-server</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Examples</strong></td>
<td>• MS Visual FoxPro</td>
<td>MS SQL Server, ...</td>
</tr>
<tr>
<td></td>
<td>• dBase</td>
<td>Oracle &gt;5 (1985!), ...</td>
</tr>
<tr>
<td></td>
<td>• MS Access</td>
<td>mySQL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Apache HBase</td>
</tr>
<tr>
<td><strong>No. of concurrent users</strong></td>
<td>± 1-50 users</td>
<td>1-millions of users</td>
</tr>
<tr>
<td><strong>Data integrity</strong></td>
<td>Limited (network errors can corrupt data)</td>
<td>Advanced mechanisms to maintain integrity (transactional processing, ...)</td>
</tr>
<tr>
<td><strong>Security</strong></td>
<td>Limited</td>
<td>Increased (database user accounts, detailed permissions, data files not directly accessible to end users and client applications)</td>
</tr>
</tbody>
</table>
Web based access

- Reduced hardware requirements
- Easier maintenance (up to date browser client)
- Usually reduced total cost of ownership
- No need to install applications on client workstations

For all these reasons, many database applications are developed as web applications: access to the database is provided via an application that runs in a web browser