FROM RELATIONAL TO OBJECT DATABASE MANAGEMENT SYSTEMS

V. CHRISTOPHIDES

Department of Computer Science & Engineering
University of California, San Diego
ICS - FORTH, Heraklion, Crete

I) INTRODUCTION
What is a DBMS?

- **Database**: Logically coherent collection of “computerized” data
  - Stored Data: Persistent facts with some inherent meaning
  - Accessible Data: Extracted and derived facts for a specific purpose
- **Database Management System**: General purpose software that facilitates the process of defining, constructing and manipulating database for various applications. It offers the following services:
  - Data Persistence
  - Disk Management
  - Data Sharing
  - Data Reliability
  - Ad hoc Queries

DBMS Applications

- **Administration & Management**
  - Bank Accounts
  - Company Stocks & Personal
  - Airline Reservation
  - University Courses and Notes
  - Library Books Borrowing
- **Media & Net-centric Applications**
  - Engineering (CAD/CAM/CIM/CAE)
  - Scientific Applications
    - Genome Databases, Environmental Applications (GIS)
  - Telecommunications + Databases
    - Network management, Telemedicine, Info Brokering
  - Multimedia, Entertainment, Visualization
    - Interactive, Virtual Reality based
  - Software Design and Management
  - Data Warehouses
  - Web Site Management
  - Electronic Commerce
DBMS vs. other types of software

- Effective Manipulation of Persistent Data
  - recovery from failures
  - concurrent access
  - security & integrity controls
  - data independence physical & logical & user

- Efficient Access to Large Data Volumes
  - buffer management
  - indexing/clustering
  - query optimization
  - distribution

Example: Phone Directory

- One file:
  - first and last name, address, profession, telephone number

- Two type of access:
  - Direct by first/last name (white pages)
  - Sequential by profession headings (yellow pages)

- Implementation without a DBMS:
  - Hashing on first/last name (key) and linked lists of profession headings
  - Change of the access mode implies PROGRAM RERWITING

- Implementation with a DBMS:
  - Choice of a physical structure
  - Choice of a logical structure
  - Application programs are build on the top of data logical structuring
The ANSI-SPARC Architecture

- **External Level:**
  - What actually seen by users (not necessary the whole database)

- **Conceptual Level:**
  - Modeling of the real-word data, independent from the underlying DBMS

- **Internal Level:** How data is physically stored
  - **Logical-schema:** Logical organization of data using the DBMS data model (tables, etc.)
  - **Physical-schema:** Physical organization of data in the secondary memory (files, records, etc.)

➲ The ability to modify a scheme definition in one level without affecting a scheme definition in a higher level is called **data independence**

Three Level Architecture

![Diagram of three-level architecture](image)
The Data Independence issue

- **Physical** (for optimization)
  - Changes of physical structures don’t affect data logical organization
  - EXAMPLE: Addition/Suppression of indices in a DBMS

- **Logical** (for maintainability)
  - Modifications of data logical structure should not affect programs
  - EXAMPLE: New Relations/Attributes in the logical schema of a DBMS

- **User** (for flexibility)
  - Multiple views on the same logical organization of data
  - EXAMPLE: Secretary and Professor views on a University database

✔ Data independence is achieved by *inter-level mappings*
✔ Queries are translated between levels automatically

Mappings in DBMS
DBMS Components

- **Data Collections:**
  - Minimizing data redundancy
  - Enabling multi-user access

- **People:**
  - Database Designer
  - Database Administrator
  - Application Programmers
  - End-Users

- **Hardware:**
  - Processor(s)
  - Main Memory
  - Secondary Storage (e.g., disks, CD-ROMs, etc.)

- **Software:**
  - The Storage Manager: buffer and file manager
  - The Transaction Manager: locking, logging, and transaction commitment (ACID)
  - The Query Processor: parsing, optimization, execution
  - Various Automated Tools: Development and Design aids, Report writers, etc.

Typical Architecture of a DBMS

```
<table>
<thead>
<tr>
<th>Schema Modifications</th>
<th>Queries</th>
<th>Data Modifications</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&quot;Query&quot; Processor</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Storage Manager</td>
<td>Transaction Manager</td>
</tr>
<tr>
<td></td>
<td>Data &amp; Metadata</td>
<td></td>
</tr>
</tbody>
</table>
```
Application Development: Backend vs. Frontend

- Backend ➔ the DBMS
- Frontend ➔ Applications which run on top of the DBMS
- Examples of vendor-provided applications or tools
  - Query language processor
  - Business graphics sub-systems
  - Spreadsheet
  - Statistical packages
  - Application generators
  - CASE Tools
  - Report writer
  - Web-DBMS Gateways
- Examples of typical utilities to help DBAs in various tasks
  - Load routines
  - Unload/Reload routines
  - Reorganization routines
  - Statistic routines
  - Analysis routines

Application Development: Client/Server Architecture

- Partition of programs between client and server processes, communicating via queries
- Hierarchical partition of functions
  - Data at the servers are shared by several clients
  - Graphical interfaces at the end-user workstations
  - Communication through standardized protocols
  - Distribution of application programs in order to minimize transfer costs
Client/Server Architecture: First Generation

**SERVER**

- **DBMS**
  - NT, UNIX, NOVELL
  - GCOS, VMS, MVS

**CLIENTS**

- Windows
- NT
- UNIX

Queries \rightarrow Results

---

Client/Server Architecture: Second Generation

- **Stored Procedures**
  - Procedure accomplish a service function on data

- **Service-oriented** architecture rather than query-oriented
  - Distribution of data manipulation

- **Extensibility and Scalability**
  - Possibility to have several, eventually redundant, servers
  - Possibility to have private data on clients
Client/Server Architecture: Third Generation

- Integration of the Web with the client-server
  - Use Web browsers for a standard presentation to the client
  - Possibility to run small client application (applets)
  - Significant portability (Virtual Private Network, Intranet, Internet)
- Three(or Multi)-tiered Architectures
  - Databases with stored procedures
  - Shared application services
- Hypermedia support
  - Various media types which can be extended (text, image, video)
  - Navigation between documents and applications

Database Technology Timeline

<table>
<thead>
<tr>
<th>Simple Data Management</th>
<th>Global Enterprise Management</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-relational</td>
<td>Enterprise-capable Relational</td>
</tr>
<tr>
<td>Early 80s</td>
<td>Internet Computing</td>
</tr>
<tr>
<td>Late 80s</td>
<td></td>
</tr>
<tr>
<td>Early 90s</td>
<td></td>
</tr>
<tr>
<td>Late 90s - 21st C</td>
<td></td>
</tr>
<tr>
<td>Simple OLTP</td>
<td>Packaged &amp; Vertical Applications</td>
</tr>
<tr>
<td>Simple transactions, on-line backup &amp; recovery</td>
<td></td>
</tr>
<tr>
<td>Stored procedures, triggers</td>
<td></td>
</tr>
<tr>
<td>Scaleable OLTP, parallel query, partitioning, cluster support, row-level locking, high availability</td>
<td></td>
</tr>
<tr>
<td>Support for all types of data, extensibility, objects</td>
<td></td>
</tr>
<tr>
<td>Middleware (messaging, queues, events) Java, CORBA, Web interfaces</td>
<td></td>
</tr>
</tbody>
</table>