Information Systems

WS 2005, JKU Linz
Course 1: Introduction

Gábor Bodnár

URL: `http://www.risc.uni-linz.ac.at/education/courses/ws2005/is/`

Organizational items

- Please register via the KUSSS system.
- The lecture notes are available at the course home page.
- There will be a written exam at the end of the semester.
- There is no lecture on 31.10.2005.

About the Course

What is an information system?

It is an automated or manual system comprising people, machines and/or methods organized to collect, store, process, transmit, disseminate or display data which represent information for some target user group.

Therefore data management plays an important role in information systems.

The course is an introduction to the fundamentals of database theory, with an outlook towards on-line applications, and to XML technologies.

Objectives of the Course

- To learn the basic concepts, problems and solution techniques in relational data modeling.

- To learn the fundamentals of SQL.

- To understand the basic problems and solutions in concurrent systems.

- To learn the elementary facts about XML, and XML related technologies (XML Schema, XPath, XSLT, etc.).

Course Overview

Relational Data Modeling

- Entity-Relationship model, ER diagrams.
- The relational model, the relational algebra.
- Functional dependencies, normalization.
- Indexing, hashing.
- SQL.

Course Overview

On-line applications

- Transaction processing in concurrent systems.
- Data warehouses, OLAP.
- Information retrieval systems (search engines).

Course Overview

XML

- XML basics, namespaces.
- XML Schema (document validation).
- XPath, XSLT (accessing parts of XML documents, document transformations).
- XQuery.

Introduction to Database Systems

A *database management system* (DBMS) consists of:

- a collection of structured, interrelated, persistent and shared data.
- a set of application programs to access, update and manage the data.

The two most important criteria for DBMSes in practice:

- reliability (robustness, concurrency and security management),
- efficiency (data manipulation speed, application program development support).

Example

The administration of a university has to store and maintain (among countless other things):

- Data of students, teachers, courses.
- Relations between the above entities in each semester.
- New data arising from these relations: e.g. grades.
- Interrelations between the data (e.g. courses that build on each other, etc.).

The maintenance of the data and the upgrades of the information system should be relatively easy.

Problems of Data Management

- Data redundancy and inconsistency.
- Data integrity: ensuring that the data on the system fulfills certain constraints, crash recovery.
- Flexible data access: support for a large class of queries, efficient data retrieval.
- Data isolation: the data may be around in different files and formats.
- Concurrency management, security.

Data Abstraction

We distinguish three abstraction levels:

Physical level It deals with the storage and the retrieval of the data using the file system support of the underlying OS.

Conceptual level It allows the database administrator to define what data is stored in the system and what are the relationships between the data.

View level It describes a subset of the data in the database for a particular set of users.

Example–Physical Level

- A record with fixed structure of fixed length for each field is stored just by concatenating the data of the fields in the file storing the records.
- Records can be retrieved by precomputed addresses.
- A character string of fixed length is stored as concatenation of ASCII codes.
- An unsigned integer is stored on four bytes as a binary value.

Example–Conceptual Level

The database should store:

- Last names and IDs of students, where the names are character strings of maximal length 30 and the ID is an unsigned integer.
- Course names: strings of 50 character maximal length, and course IDs, which are unsigned integers.
- Moreover, if a student with ID S takes a course with ID C then a pair of IDs: (S, C).

There is an additional constraint that an (S, C) pair can only appear in the database if there is already a student with ID S and a course with ID C.

Example–View Level

A teacher wants to get the following information from the database:

- The list of courses he teaches (times, locations, etc.)
- For each course the list of subscribed students.

The dean on the other hand wants to see only the following:

- The list of teachers with the courses assigned to them.
- The number of students subscribed for a course.
- The total number of students taking a course offered by the faculty and the total number of subscriptions.

Data Models

Data models are conceptual tools to describe: data, data relations, data constraints, data semantics.

Object-based data models:

- Describe data on the conceptual and view levels.
- Provides flexible structuring capabilities.
- Data constraints can be specified explicitly.

Examples: Entity-Relationship model, object-oriented model, functional model.

Data Models

Record based data models:

- Describe data on the conceptual and view levels.
- Used with fixed record structure databases.
- SQL is a typical query language for record based data models.

Examples: Relational model, hierarchical model.

Data Independence

The overall design of the database is called the database *scheme*.

The data of a snapshot of the database at a particular point in time is called an *instance* of the database.

The concept of *data independence* expresses that changes of the database scheme at a given level should not affect schemes at higher levels (physical $<$ conceptual $<$ view).

Thus we can talk about:

- physical data independence,
- logical data independence.

Standard Terminology in DBMSes

DDL Data definition language, to describe the conceptual (or logical) scheme of the database.

DML Data manipulation language, to form queries for the database as well as to modify the database. There are non-procedural and procedural DMLs.

Database Manager The program that provides an interface between the conceptual and the physical levels.

Database administrator The person controlling the DBMS.

Database users Can be: application programmers or naive users.

Overall structure of a DBMS

File manager Usually belongs to the underlying OS.

Database manager Discussed on the last slide.

Query processor The interpreter of the query language.

DDL compiler Converts DDL statements into database metadata.

Data directory The database metadata, describing the logical scheme in some physical representation.

Data files The files that store the data held by the DB.

Indices Files that accelerate data retrieval from data files.

Overall structure of a DBMS