Table of Contents

1 Introduction to Information Systems ......................................................... 2
  1.1 Introduction .............................................................................................. 2
  1.2 Architecture of Information systems ......................................................... 2
  1.3 Classification of Data Models ................................................................. 4
  1.4 Relational Data Model (Overview) ....................................................... 8
  1.5 Conclusion ............................................................................................... 12
1 Introduction to Information Systems

1.1 Introduction

In this course, we define an Information System as any software application dealing with persistent information (i.e. persistent data) in computer's memory.

Persistent Data exist independently of a user's session.
In other words, persistent data objects are created by a particular user and available for all others at any time.

1.2 Architecture of Information systems

Primitively speaking, we can see an Information System as a combination of three main components:
Application Layer implements purpose-oriented data processing algorithms and a particular user interface. There is no common solutions which are used for designing different information systems. For example, a project management system, electronic library, financial transaction system, computer-based training system, etc. require essentially different data processing algorithms and user interfaces.

Data Management Layer provides data structures and operations to be utilised by the application layer. Formally, a combination of data structures and data processing operations supported by the Data Management Layer is called a Data Model. We can speak about information systems based of a file management system, on a relational data model, on World-Wide Web, etc. In other words, information systems may be based on different data models.
It should be especially noted that essentially different information systems may be based on one and the same data model. For example, a collaborative document authoring system and an university information system might well utilize one and the same relational data model.

1.3 Classification of Data Models

Data management principles are essentially different depending on nature of information to be processed. Thus, we can speak at least about well-structured data and ill-structured data.
We speak about well-structured data if structure of data objects may be separated from its actual content. In other words, there are a number of data objects having identical structure (meta-structure). In this case, any data object is just an instance of a previously defined data type. Well-structured data management is based on so-called **database technology**:

- **database schema** contains a description of all data types which are of interest to users.
- **database** contains instances of the data types previously defined in the database schema.

To build a Database:

- 1. Database Administrator has to accurately define a particular Database Schema;
- 2. Application layer programmers apply a collection of operators which are supported by the DBMS, to retrieve, store, or modify data.

Note that data objects are created as instances of previously defined data types, i.e. by filling predefined forms with actual data.
Thus, a DBMS maintains two different but closely connected languages:

- 1. **Data Description Language (DDL)** is applied to define database schemata;
- 2. **Data Manipulation Language (DML)** is a collection of operators to create, access and modify Databases.

We speak about *ill-structured* data if structure of a data object cannot be separated from its actual content.
In other words, each data object may have an unique data structure which is coded as a part of its actual content.
In order to work with Ill-structured Data,

- 1. **Data Objects** are coded (authored) using special data encoding (mark-up) language
- 2. **References to other objects** are inserted directly into the object code (embedded links)
- 3. **Special software** interprets such object code

For example, World Wide Web (WWW) utilizes HTML as the data encoding language and standard software packages (Internet Browsers) to interpret the code. Such systems are normally referred to as **first generation hypermedia** systems.

**Second generation Hypermedia** supports meta-languages, i.e. **general mechanisms** which can be used to define application oriented data encoding paradigms. For example, eXtensible Markup Language (XML) and Cascading Style Sheets (CSS). Meta-language allows to manage any information in a platform- and vendor-neutral way, and then easily share this information with any other application.
Second generation Hypermedia operates with higher-level data structures called Hypermedia Composites. Hypermedia Composites may be seen as structured collections of data objects and other composites. Composites facilitate collaborative authoring, usability and integrity of hypermedia systems.

Thus, we can see a comparison of these two data management techniques as follows:

1.4 Relational Data Model (Overview)

Relational Data Model allows to operate with a database as with collection of two-dimensional tables called relations.
Tuples (i.e. Data Objects) are related by means of so-called foreign key/primary key relationships.

A relational database schema may be seen as a definition of a number of relations (tables). Each relation is defined as a unique name with a list of domains which corresponds to columns of the relation.

For example:

- **Customer(C#, Cname, Ccity, Cphone)**
- **Product(P#, Pname, Price)**
- **Transaction(C#, P#, Date, Qnt)**
To insert (put) new tuples into a particular relation declared in a current data base schema:

- Step1: A user (an application programm) chooses a relation (say, Customer);
- Step2: The user (programm) prepares a new tuple of the relation on the screen or in the computer's memory.
- Step3: The DBMS put a new tuple into the relation (data base).

Relational Data Model automatically supports so-called **Referential Integrity**: 

"A foreign key can have only two possible values, either the relevant primary key value or Null-Value. No other values are permitted"
The most popular relational query language is called *SQL (Structured Query Language)*. The basic operation in SQL is called mapping, which transforms values from a database to user requirements. This operation is syntactically represented by the SQL block.

SQL blocks may be nested in order to implement more complex queries. Consider for example the following query:

"Get names of such customers who bought the product CPU ".

<table>
<thead>
<tr>
<th>C#</th>
<th>Cname</th>
<th>City</th>
<th>Cphone</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Codd</td>
<td>London</td>
<td>2263035</td>
</tr>
<tr>
<td>2</td>
<td>Martin</td>
<td>Paris</td>
<td>5555910</td>
</tr>
<tr>
<td>3</td>
<td>Deen</td>
<td>London</td>
<td>2234391</td>
</tr>
</tbody>
</table>

---

The most popular relational query language is called **SQL (Structured Query Language)**. The basic operation in SQL is called mapping, which transforms values from a database to user requirements. This operation is syntactically represented by the SQL block.
SQL blocks may operate with multiple relations to select data from such different relations. Consider for example the following query:

"Get customer names, dates and numbers of product units for transactions dealing with the product number 1".

1.5 Conclusion

Existing Data Modeling Techniques:

- File Management:
- Database Technology:
  - Relational Data Model
Introduction to Information Systems

- Network Data Model
- Object-Oriented Data Model

**Knowledge Representation:**
- Deductive (Logic) Data Model
- Functional Data Model
- Frame-Based Data Model

**Hypermedia:**
- Node-Link Data Model
- Logic Hypermedia Composites
- Semantic Hypermedia Composites