Introduction to Database System

Data vs Information

- Data are raw facts.
  - the facts have not yet been processed to reveal its usefulness to the user.
- Information is the result of processing raw data to reveal its meaning.
  - Data processing can be:
    - as simple as organizing data to reveal patterns or
    - as complex as making forecasts
  - To reveal meaning, information requires context.

- Raw data must be properly formatted for storage, processing, and presentation.
- Today, production of accurate, relevant, and timely information is the key to good decision making – a *key to business survival in a global market*.
- Data management is a discipline that focuses on the proper generation, storage, and retrieval of data

Database and the DBMS

- A database is a shared, integrated computer structure that stores a collection of:
  - End-user data, that is, raw facts of interest to the end user.
  - Metadata, or data about data, through which the end-user data are integrated and managed.
    - The metadata provide a description of the data characteristics and the set of relationships that link the data found within the database.
- A database management system (DBMS) is a collection of programs that manages the database structure and controls access to the data stored in the database

Role and Advantages of the DBMS

- The DBMS serves as the intermediary between the user and the database. See fig 1.
- The DBMS hides much of the database's internal complexity from the application programs and users.
- Important advantages:
  - The DBMS enables the data in the database to be shared among multiple applications and users.
  - The DBMS integrates the many different users' view of the data into a single all-encompassing data repository.

- Other advantages:
  - Improved data sharing.
  - Improved data security – provides a framework better enforcement of data privacy and security policy.
  - Better data integration
  - Minimized data inconsistency (*data inconsistency happens when the same data appears in different places*.)
  - Improved data access - supports ad hoc queries
  - Improved decision making – generate better information
  - Increased end-user productivity

Types of Databases

- Databases can be classified according to:
  - the number of users,
  - the database location(s), and
  - the expected type and extent of use.
- Number of users
  - ***Single-user*** (desktop) database supports only one user at a time – runs on a personal computer.
  - ***Multiuser database*** – support multiple users at the same time.
    - Less then 50 users or a specific department – workgroup database
    - More than 50 users and the entire organization use it – enterprise database

- Location of the database:
  - *Centralized* database is in a single location.
  - *Distributed* database supports data distributed across several different sites.
- The expected type and extent of use:
  - *Operational* (transactional or production) database support a company's day-to-day operation.
  - *Data warehouse* storing data used to generate information required to make tactical or strategic decision, e.g. pricing decision, sales forecast, market positioning, etc.

- Database can also be classified to reflect the degree to which the data are structured.
  - Unstructured data – data that exists on it original state when it was collected.
  - Structured data – data formatted to facilitate storage, use, and generation of information.
- See fig 2: features of known DBMSs.

Files and File Systems

- Good reasons for studying file systems:
  - An understanding of the relatively simple characteristics of file systems makes the complexity of database design easier to understand.
  - An awareness of the problems that plagued file systems can help you avoid those same pitfalls with DBMS software.
  - If you intend to convert an obsolete file system to a database system, knowledge of the file system's basic limitations will be useful.

- During the early days of computerization, the computer files within the file system were like the manual files.
- See fig 3. sample customer file
- The conversion of the manual file system to computer file system was technically complex – a new kind of professional, the *data processing specialist* had to be hired.
- During this time specialized vocabulary were developed. See table 1.

- As needs increases more files were created - the birth of a small file system. See fig 4.
- Each of these file system used its own application to store, retrieve, and modify data.
- An individual or department who commissioned its creation owns the file.
- As file systems grew, the demand of the DP specialist's programming skills also grew, and more DP programmers were hired – the birth of the Data Processing Department managed by a data processing manager.

Problems with File System Data Management

1. The first and obvious problem is that even the simplest data-retrieval task requires extensive programming – programmers had to specify what must be done and how it is to be done.
2. Does not support ad hoc queries.
3. Making changes in an existing structure can be difficult in a file system environment.
    - Any changes to a file structure, no matter how minor, requires modifications to all programs that use the data in the file.

- Changes in the structure, requires the following steps
  a. Reads a record from the original file.
  b. Transforms the original data to conform to the new structure's storage requirements.
  c. Writes the transformed data into the new file structure.
  d. Repeats steps b to d for each record in the original file.

5. As the number of files in the system grows, system administration becomes more difficult.
6. Security features are difficult to implement and often omitted.

Structural and Data Dependence

- *Structural dependence* exists when access to a file is dependent on its structure
  - Changes, for example adding a field would require the steps mentioned in slide #18
  - The application programs are affected by the change in the file structure.
- *Structural independence* exists when it is possible to make changes in the file structure without affecting the application program's ability to access the data

- The system exhibits *data dependence* when programs that access the data are modified because of the change of the file's storage characteristics, e.g., change in the data type.
- *Data independence* exists when it is possible to make changes in the data storage characteristics without affecting the application program's ability to access the data.
- Data dependence makes the file system extremely cumbersome from the point of view of a programmer and database manager.

Field Definitions and Naming Conventions

Data Redundancy

- The file system's structure makes it difficult to combine data from multiple sources and its lack of security renders the file system vulnerable to security breaches.
  - Storage of the same data in different locations – islands of information
  - Each location stores different versions of the data.
  - This results data redundancy which when become uncontrolled posses several problems.

- Uncontrolled data redundancy results:
  - Data inconsistencies – different and conflicting versions of the same data appear in different places.
  - Data anomalies – data redundancy promotes an abnormal condition because users are forced to do field value changes in many different locations.

Database Systems

- Database system consists of logically related data stored in a single logical data repository.
  - Logical in the sense that from the users' perspective, the database is in one location but can be physically distributed to different locations.
- It is not only the data that is stored, it also includes the relationships between those structures and the access paths to those structures — all in a central location.
- The current generation of DBMS software also takes care of defining, storing, and managing all required access paths to those components.
- See Fig 5. Comparison of file system and Database system

The Database System Environment

- *Database system* refers to an organization of components that define and regulate the collection, storage, management, and use of data within a database environment.
- From management's point of view, database system is composed of five major parts. See fig. 6.

Database Systems Component

- **Hardware** - refers to all of the system's physical devices.
- Software:
  - ***Operating system*** *software* manages all hardware components and makes it possible for all other software to run on the computers.
  - ***DBMS software*** manages the database within the database system.
  - ***Application programs and utility software*** are used to access and manipulate data in the DBMS and to manage the computer environment in which data access and manipulation take place

- **People** - includes all users of the database system.
  - ***System administrators*** oversee the database system's general operations.
  - ***Database administrators***, also known as DBAs, manage the DBMS and ensure that the database is functioning properly.
  - ***Database designers*** design the database structure - the database architect.
  - ***Systems analysts and programmers*** design and implement the application programs – includes data entry screens, reports and procedures.
  - ***End users*** are the people who use the application programs to run the organization's daily operations.

- P**rocedures** are:
  - the instructions and rules that govern the design and use of the database system.
  - used to ensure that there is an organized way to monitor and audit both the data that enter the database and the information that is generated through the use of that data.
- **Data** - covers the collection of facts stored in the database.

DBMS Functions

- A DBMS performs several important functions that guarantee the integrity and consistency of the data in the database.
- These functions include:
  - ***Data dictionary management***.
    - The DBMS stores definitions of the data elements and their relationships (metadata) in a data dictionary.
    - provides data abstraction, and it removes structural and data dependency from the system.
    - See fig 7. Data definition in IBM DB2

- Data storage management.
  - The DBMS creates and manages the complex structures required for data storage, relieving us from the difficult task of defining and programming the physical data characteristics.
- Security management. The DBMS creates a security system that enforces user security and data privacy.
- Multiuser access control. The DBMS ensures that multiple users can access the database concurrently without compromising the integrity of the database.

- Backup and recovery management. The DBMS systems provide special utilities that allow the DBA to perform routine and special backup and restore procedures.
- Data integrity management. The DBMS promotes and enforces integrity rules, minimizing data redundancy and maximizing data consistency.
- Database access languages and application programming interfaces.
  - Provides data access through a query language.
  - Provides application programming interfaces to procedural languages
- Database communication interfaces. Current DBMS accepts end-user request from multiple, different network environments.

Disadvantages of DBMS

- ***Increased costs.*** Database systems require sophisticated hardware and software and highly skilled personnel
- ***Management complexity***. Database systems interface with many different technologies and have a significant impact on a company's resources and culture.
- ***Maintaining currency***
- ***Vendor dependence***.
- ***Frequent upgrade/replacement cycles***.

End of Presentation

Fig 1: Interaction between user and the database

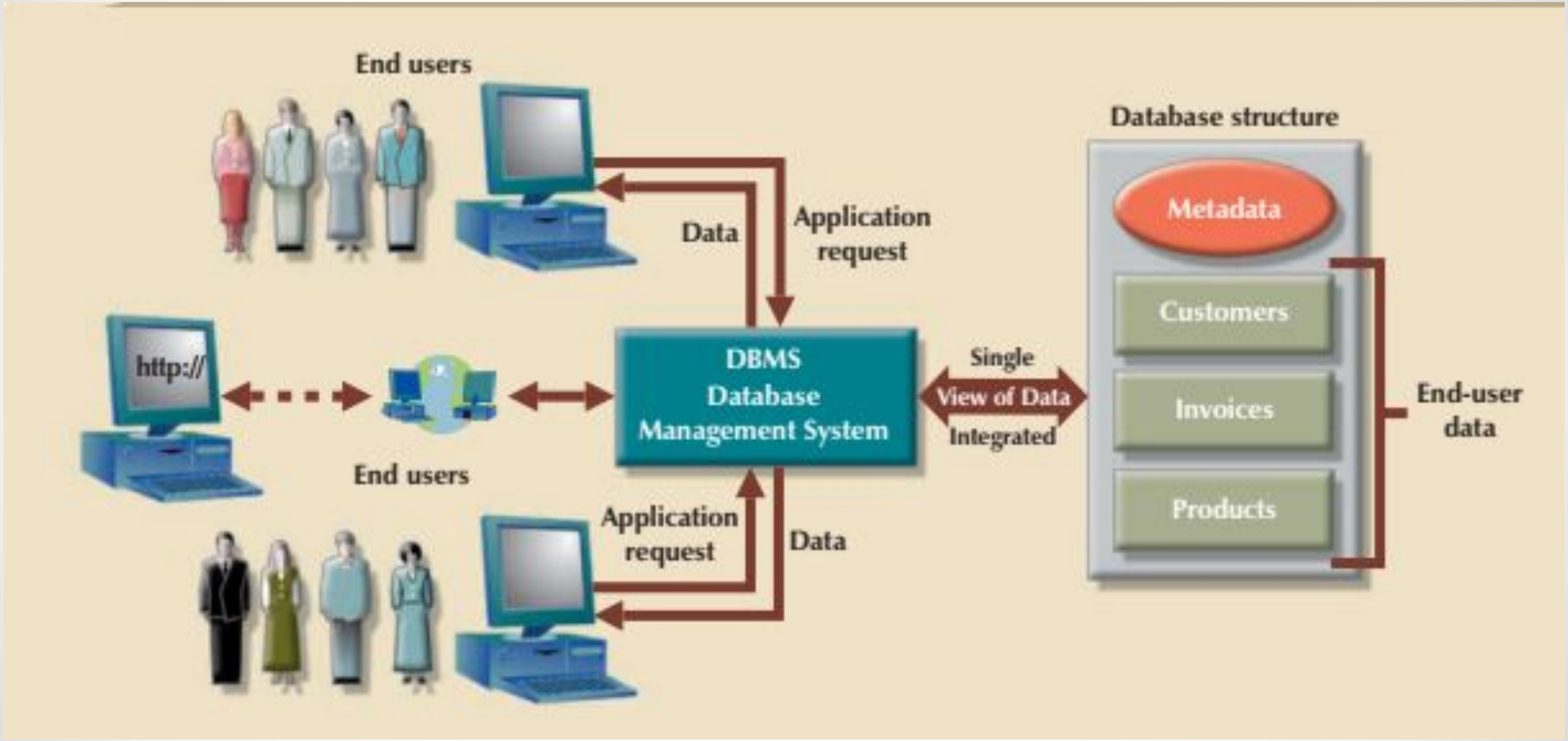


Fig 2: Features of known DBMSs

PRODUCT	NUMBER OF USERS			DATA LOCATION		DATA USAGE		XML
	SINGLE USER	MULTIUSER		CENTRALIZED	DISTRIBUTED	OPERATIONAL	DATA WAREHOUSE	
		WORK-GROUP	ENTER-PRISE					
MS Access	X	X		X		X		
MS SQL Server	X[2]	X	X	X	X	X	X	X
IBM DB2	X[2]	X	X	X	X	X	X	X
MySQL	X	X	X	X	X	X	X	X*
Oracle RDBMS	X[2]	X	X	X	X	X	X	X

* Supports XML functions only. XML data is stored in large text objects.

Fig. 3: file system – CUSTOMER file

C_NAME	C_PHONE	C_ADDRESS	C_ZIP	A_NAME	A_PHONE	TP	AMT	REN
Alfred A. Ramas	615-844-2573	218 Fork Rd., Babs, TN	36123	Leah F. Hahn	615-882-1244	T1	100.00	05-Apr-2008
Leona K. Dunne	713-894-1238	Box 12A, Fox, KY	25246	Alex B. Alby	713-228-1249	T1	250.00	16-Jun-2008
Kathy W. Smith	615-894-2285	125 Oak Ln, Babs, TN	36123	Leah F. Hahn	615-882-2144	S2	150.00	29-Jan-2009
Paul F. Olowski	615-894-2180	217 Lee Ln., Babs, TN	36123	Leah F. Hahn	615-882-1244	S1	300.00	14-Oct-2008
Myron Orlando	615-222-1672	Box 111, New, TN	36155	Alex B. Alby	713-228-1249	T1	100.00	28-Dec-2008
Amy B. O'Brian	713-442-3381	387 Troll Dr., Fox, KY	25246	John T. Okon	615-123-5589	T2	850.00	22-Sep-2008
James G. Brown	615-297-1228	21 Tye Rd., Nash, TN	37118	Leah F. Hahn	615-882-1244	S1	120.00	25-Mar-2009
George Williams	615-290-2556	155 Maple, Nash, TN	37119	John T. Okon	615-123-5589	S1	250.00	17-Jul-2008
Anne G. Farriss	713-382-7185	2119 Elm, Crew, KY	25432	Alex B. Alby	713-228-1249	T2	100.00	03-Dec-2008
Olette K. Smith	615-297-3809	2782 Main, Nash, TN	37118	John T. Okon	615-123-5589	S2	500.00	14-Mar-2009

C_NAME = Customer name
C_PHONE = Customer phone
C_ADDRESS = Customer address
C_ZIP = Customer zip code

A_NAME = Agent name
A_PHONE = Agent phone
TP = Insurance type
AMT = Insurance policy amount, in thousands of $
REN = Insurance renewal date

Table 1: Basic File Terminology

TERM	DEFINITIONS
Data	Raw facts, such as telephone number a customer name, birth date, etc.
Field	A character or group of characters (alphabetic or numeric) that has a specific meaning. A field is used to define and store data.
Record	A logically connected set of one or more fields that describes a person, place, or thing.
File	A collection of related records. Example, a file that contain the records of students who are currently enrolled.

Fig 4: A simple file system

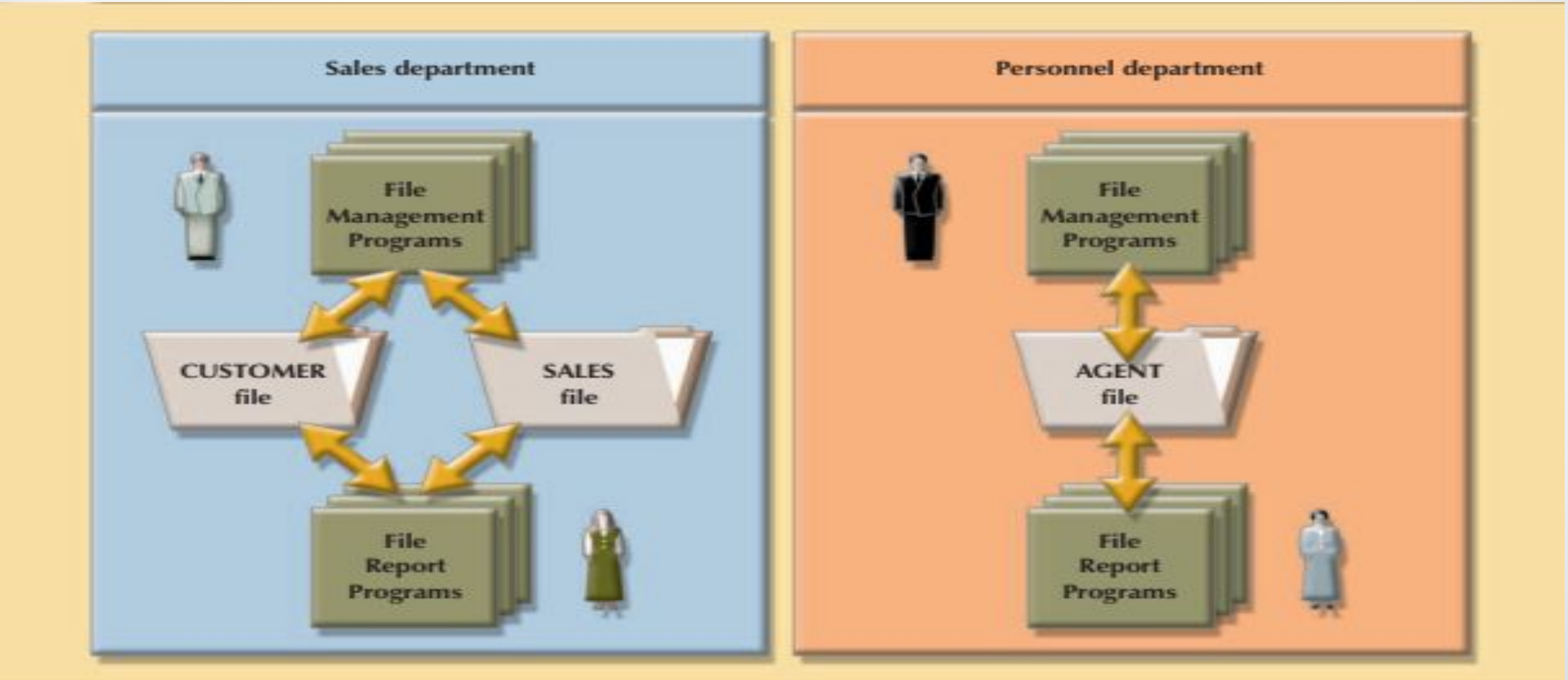


Fig 5. Comparison file system vs database system

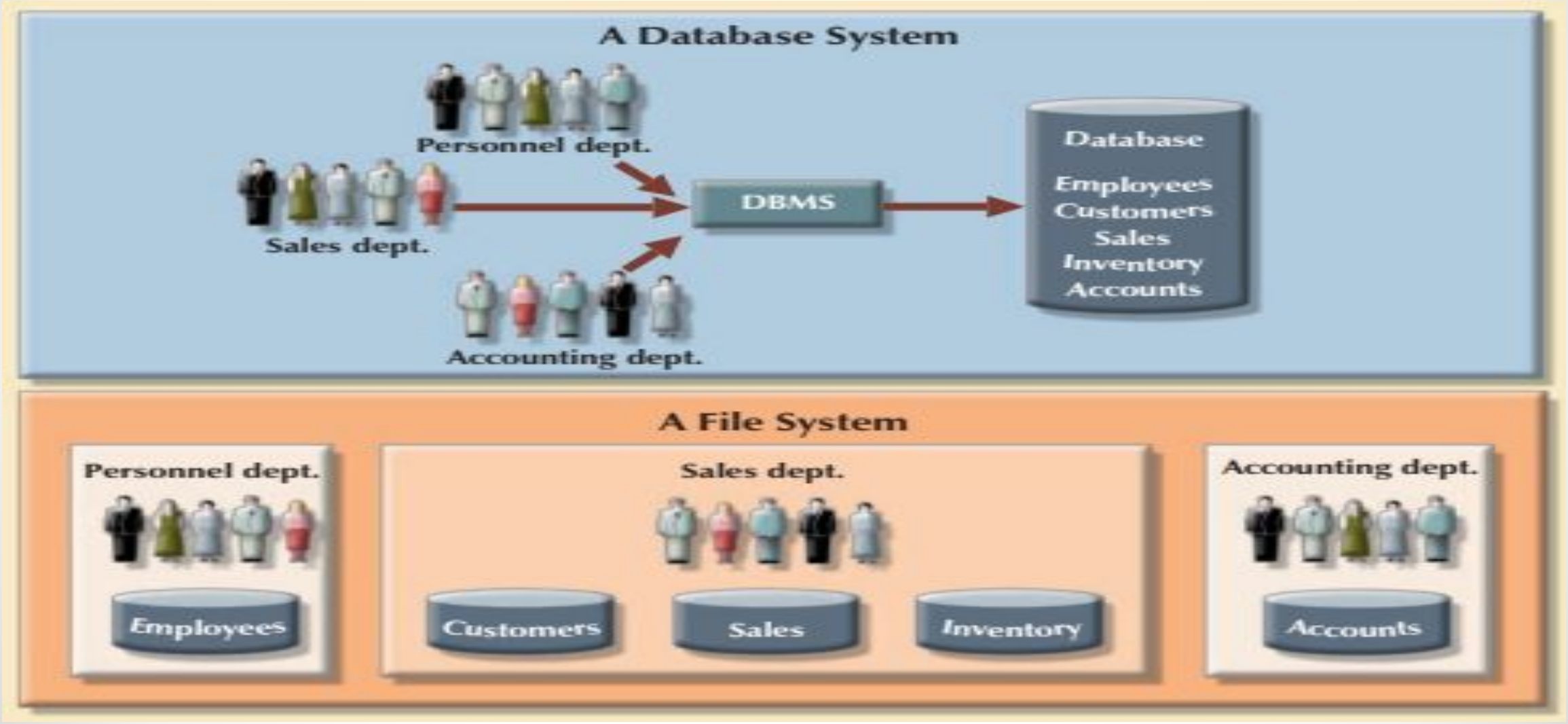


Fig. 6. Database System Environment

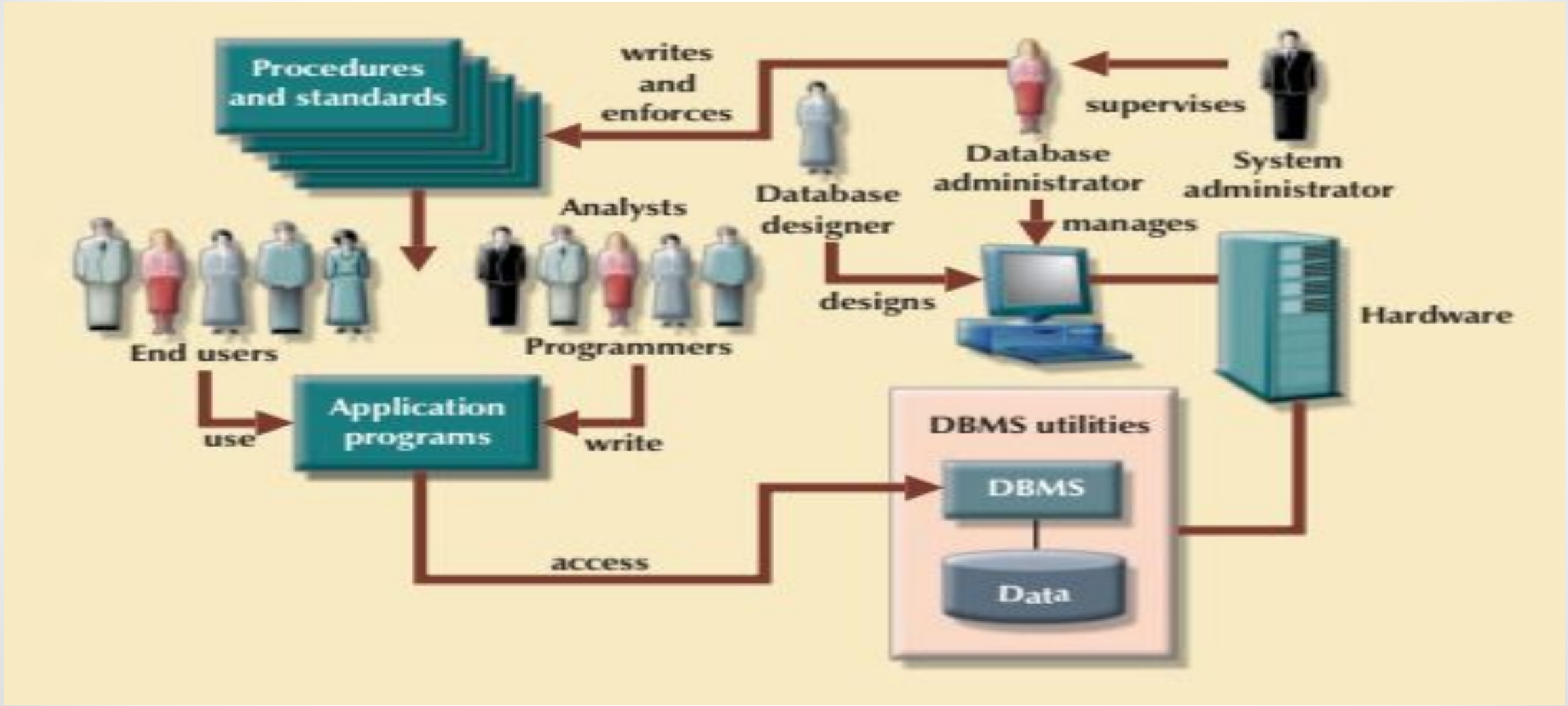
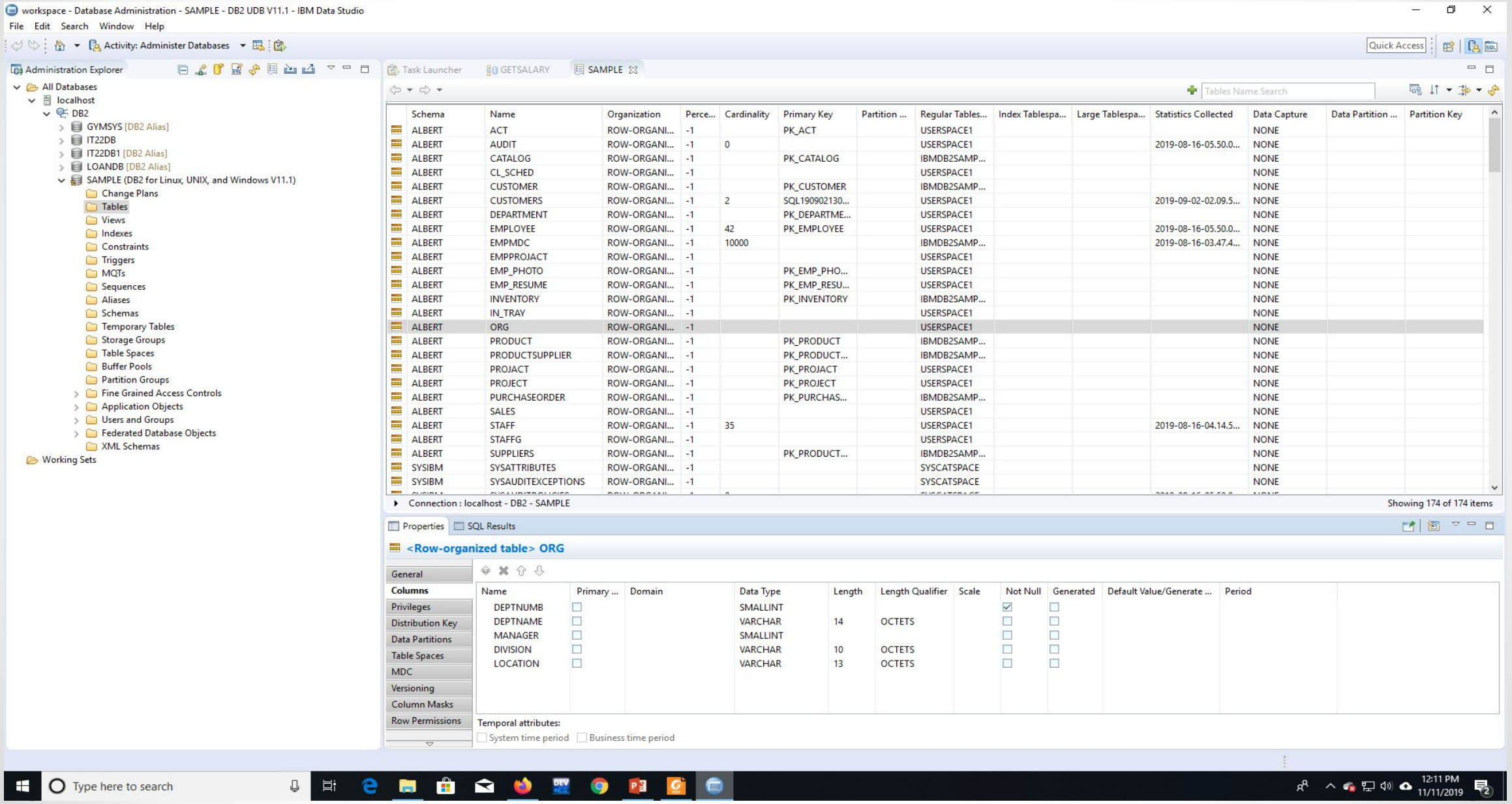


Fig. 7 Data definition in IBM DB2



Source:

Rob, Peter and Coronel, Carlos. *Database Systems : Design, Implementation and Management*, 7th Edition. Course Technology, Thomson Learning Inc. ©2007