Introduction

- in today’s competitive environment, data (or information) and its efficient management is the most critical business objective of an organization
- database system is a tool that simplifies the above tasks of managing data and extracting useful information in a timely fashion
- databases have become integral component of our everyday life

Data

- defined as known fact that can be recorded and that have implicit meaning
- raw or isolated facts from which required information is produced
- distinct pieces of information, usually formatted in special way
- binary computer representations of stored logical entities - can exist in variety of forms

Data

- single piece of data represents single fact about something in which we are interested
- for an academic organization, it may be the fact that Calin CENAN employee (or social security) number is 106519, or that largest course attendance it is at DataBases Course or that E-Mail address of one of key lecturer is Calin.Cenan@cs.utcluj.ro

Data

- usually there are many facts (data) to describe something of interest to us
- data is also known as plural of datum, which means single piece of information
- in practice, data is used as both singular and plural form of word
- term is often used to distinguish machine readable information from human readable (textual) information

Enterprise (organizational) data

- *Operational data* - stored in operational systems throughout organization (both internal and external) systems.
- *Reconciled data* - stored in data warehouse and data store
  - intended as single, authoritative source for all decision support applications
- *Derived data* - stored in each of data mart
  - limited and summarized data warehouse, selected, formatted and aggregated for end-user decision support applications

Information

- https://www.youtube.com/watch?v=nqwx2XFb1fQ
- What’s the difference ?
- One is my name. The other is not.

Information

- is processed, organized or summarized data
- collection of related data that when put together, communicate meaningful and useful message to recipient who uses it, to make decision or to interpret data to get meaning
- data are processed to create information, which is meaningful to recipient
- *who is attending DataBases course*

Data Warehouse

- collection of data designed to support management in decision-making process
- subject-oriented, integrated, time-variant, non-updatable collection of data used in support of management decision-making processes and business intelligence

DataBase Data Warehouse

- student Floriana grade on DataBases topic
- student Floriana credits
- how many students have between 27 and 37 credits
- management decision
  - more examinations
  - no more examinations

Data Warehouse

- present coherent picture of business conditions at single point of time
- unique kind of database, which focuses on business intelligence, external data and time-variant data
- process, where organizations extract meaning and inform decision making from their informational assets through the use of data

Metadata

- also called *data dictionary* is data about data; also called *system catalog*
- data that describes objects in database and makes easier for those objects to be accessed or manipulated
- describes database structure, constraints, applications, authorization, sizes of data types, …
- often used as integral tool for information resource management

Data dictionary (also called information repositories)

- mini DataBase Management Systems that stores metadata, attribute names and definitions
- repository of information about database that documents data elements of database
- integral part of DataBase Management Systems (DBMS)
- aid database administrator in management, user view definitions as well as their use

Data dictionary

- descriptions of schema
- information on physical database design, storage structures, file & record sizes
- database users, responsibilities & access rights
- descriptions of database transactions and relationships to users and data items referenced by them
- usage statistics such as frequencies of queries and transactions and access counts to different portions of database
- *relationships* among *entities*

Data Item or Fields

- smallest unit of data that has meaning to its user, called also *data element*
- represented in database by value
- names, telephone numbers, bills amount, address, …
- molecules of database (there are atoms and sub-atomic particles composing each molecule (bits and bytes)), but they do not convey any meaning so are of little concern to users
- may be used to construct other, more complex structures

Record

- collection of logically related fields or data items, with each field possessing fixed number of bytes and having fixed data type - consists of values for each field
- occurrence of named collection of zero, one, or more than one data items or aggregates; data items are grouped together to form records
- grouping of data items can be achieved through different ways to form different records for different purposes
- are retrieved or updated using programs

Files

- collection of related sequence of records
- in many cases, all records are of same record type (each record having an identical format)
- if every record has same size file is said to be *fixed-length records*
- if different records have different sizes, is said to be *variable-length records*

- data dictionary contains the following:
  - Entities
  - Attributes
  - Relationships
  - Keys

Entities

- real physical object or an event; the user is interested in keeping track of
- any item about which information is stored
- for example Calin CENAN is real living person and employee of Technical Univeristy of Cluj-Napoca - is entity for which company (sinu) is interested in keeping track
- Dacia Logan model car CJ-10-SQL is real physical object and an entity
- employee owning car (and Access to Observator parking) it is an event and it is another entity

Entity Set

- collection of entities of same type, for example “all” of company’s employees
- record describes entity and file describes *entity set*

Attributes

- property or characteristic (field) of an entity
- name, personal identification number (PIN = Cod Numeric Personal), position, … all are his attributes
- height, weight, color of eyes, sense of humor are characteristics but not attributes (are not stored in database)
- values in all fields are attributes

Relationships

- associations or ways that different entities relate to each other
- for example the relationship between employees and departments, Rodica POTOLEA is member of Computer Science dept. and Liviu MICLEA is member of Automation dept.

Keys

- data item (or field) computer uses to identify record in database system
- single attribute or combination of attributes of entity set that is used to identify one or more entities, instances of set
- various types of keys:
  - Primary Key
  - Super key
  - Alternate Key

Primary Key

- used to uniquely identify record - also called entity identifier, for example, PIN in EMPLOYEE file, MIN-NO in CAR file
- when more than one data item is used to identify record, it is called *concatenated key*, for example, both FirstName, LastName and Initial …

Super key

- includes any number of attributes that possess uniqueness property (not just the minimal number of such attributes like for PK)
- for example, if we add additional attributes to primary key, resulting combination would still uniquely identify an instance of entity set
- Primary Key is minimum super key

Alternate Key

- or also called secondary key used to identify all records, which have certain property
- attribute or combination of attributes that classifies entity set on particular characteristic
- Primary Key it is an Alternate Key chosen by DBA to actually identify entities
- PIN vs. FirstName + LastName+ Initial vs. …
- choose PIN because it is integer number & shorter

- Relationships could be of following types:
- one-to-one (1:1) relationship
- one-to-many (1:m) relationships
- many-to-many (n:m) relationships

- for example, one department have many faculty staff, it is a one-to-many (1:m) relationships

DataBase

- database is defined as collection of logically related data stored together that is designed to meet the information needs of an organization
- organized in such a way that computer program can quickly select desired pieces of data

DataBase
can further be defined as

- collection of interrelated data stored together without harmful or unnecessary redundancy
- serves multiple computer program applications in which each user has his own view of data (data is protected from unauthorized access by security mechanism and concurrent access to data is provided with recovery mechanism)
- stores data independent of programs and changes in data storage structure or access strategy do not require changes in accessing programs or queries
- has structured data to provide foundation for growth and controlled approach is used for adding new data, modifying and restoring data

DataBase
https://en.wikipedia.org

- organized collection of data, stored and accessed electronically

DataBase
https://en.wikipedia.org

- DataBase Management System (DBMS) is software that interacts with end users, applications, and database itself to capture and analyze data; allows definition, creation, querying, update, and administration of databases
- database generally stored in DBMS-specific format
- often term "database" used to loosely refer to any of DBMS, database system or application associated

Database Definition

- **database -** collection of data, typically describing activities of one or more related organizations
- **DataBase Management System**, or **DBMS -** software designed to assist in maintaining and using collections of data
- software that manages and controls access to database
- **database application -** program that interacts with database at some point in its execution

Database

- **database -** collection of data, typically describing activities of one or more related organizations
- **database -** shared collection of logically related data and its description, designed to meet the information needs of an organization.

Database example

- university database might contain information about:
  - entities such as students, faculty, courses, and classrooms
  - relationships between entities, such as students' enrollment in courses, faculty teaching courses, and the use of rooms for courses

Database: Functional Definition

- Database is a stored data collection having the following characteristics:
  - assures **data independence**
  - assures **accesses** (possibly shared access) to large volumes of data

DataBase Management System (DBMS)

- software designed to assist in maintaining and utilizing large collections of data
- software system that enables users to define, create, maintain, and control access to database
- software package used to store, manage, and analyze data

DataBase Management System (DBMS)

- alternative to using a DBMS is to use ad hoc approaches that do not carry over from one application to another
  - store data in files and write application-specific code to manage it

Database application

- computer program that interacts with database by issuing an appropriate request (typically an SQL statement) to DBMS
- **SQL – Structured Query Language** is common interface between language, technology and database
- Architecture
  - Client – Server
  - Web application

Database Application Program

- maintain database and generate information
- written in programming language
- opposed to file-based approach, physical structure and storage of data are now managed by DBMS

Database – Formal Definition

- A database is an *ordered collection* of *related data elements* intended to meet *the information needs* of an organization and designed to be (possible) *shared* by multiple users

Ordered collection

- collection of data elements, not just a random assembly of data structures
- put together deliberately with proper order
- linked together in the most logical manner

Related data elements

- not disjointed structures without any relationships among them
- related among themselves
- pertinent to particular organization

Information needs

- collection of data elements in database is there for a specific purpose
- to satisfy and meet information needs of organization

DataBase

- designed, built and populated with data for specific purpose
- has intended group of users and some preconceived applications in which users are interested
  - has some source from where data is derived
  - some degree of interaction with events in real world
  - audience that is actively interested in contents
- can be of any size and of varying complexity
- created & maintained by DataBase Management System

DataBase
consist of

1. Data items
2. Relationships
3. Constraints
4. Schema

DataBase consist of

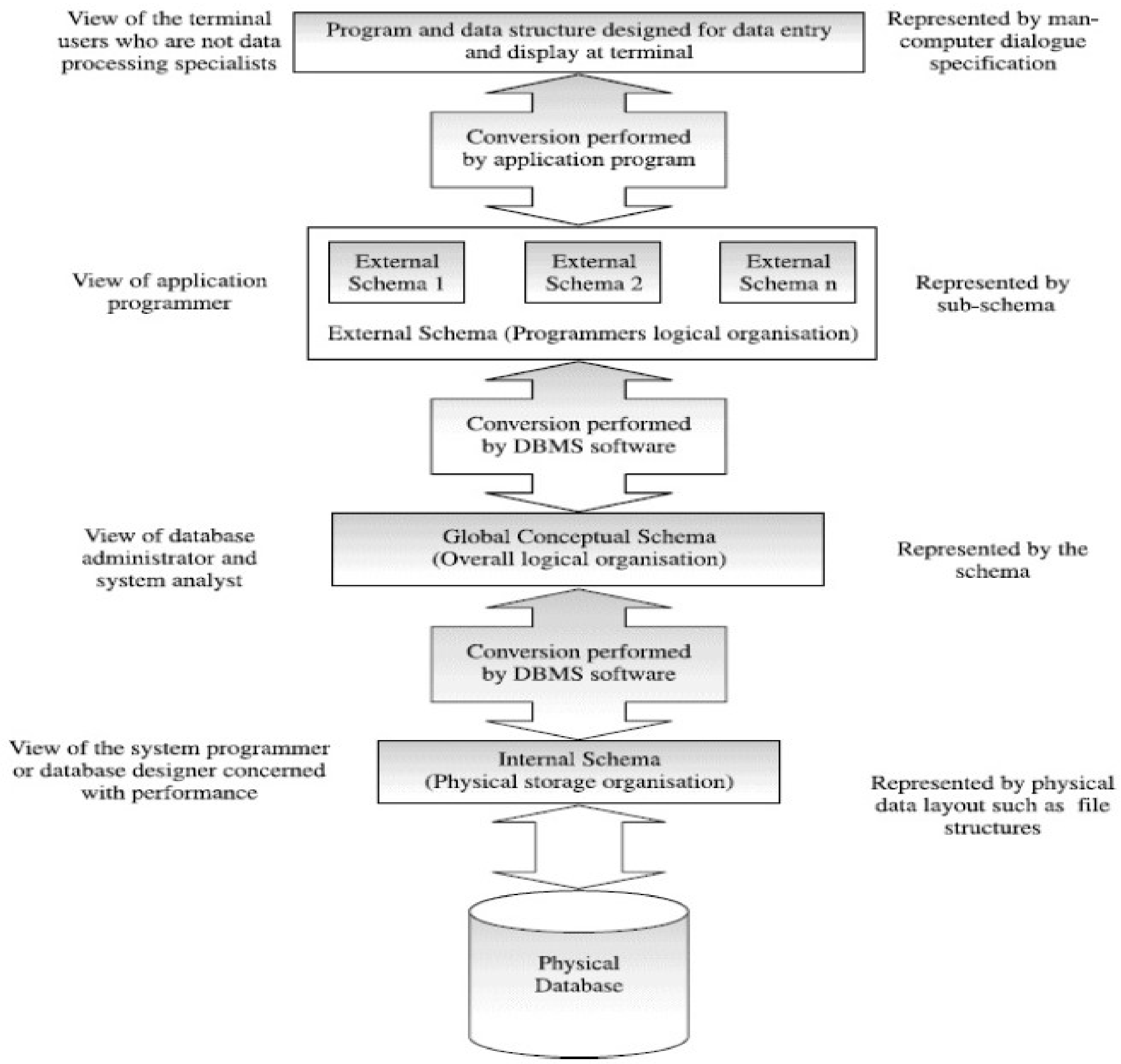
- *Data* (or *data item*) is distinct piece of information
- *Relationships* represent correspondence between various data elements
- *Constraints* are predicates that define correct database states
- *Schema* describes organization of data and relationships within database

DataBase Schema

- defines various views of database for use of various system components of DBMS and for application security
- separates physical aspect of data storage from logical aspects of data representation

DataBase
three independent levels

1. Physical storage organization or internal schema layer
2. Logical organization or global conceptual schema layer
3. External schema layer for programmers' logical organization


View of the terminal users who are not data processing specialists
Program and data structure designed for data entry and display at terminal
Represented by man-computer dialogue specification
Conversion performed by application program
View of application programmer
External Schema 1
External Schema 2
External Schema n
External Schema (Programmers logical organisation)
Represented by sub-schema
Conversion performed by DBMS software
View of database administrator and system analyst
Global Conceptual Schema (Overall logical organisation)
Represented by the schema
Conversion performed by DBMS software
View of the system programmer or database designer concerned with performance
Internal Schema (Physical storage organisation)
Represented by physical data layout such as file structures
Physical Database

DataBase
three independent levels

1. *internal schema* defines how and where data are organized in physical data storage
2. *conceptual schema* defines stored data structure in terms of the database model used
3. *external schema* defines view of database for particular users

- DataBase Management System provides for accessing database while maintaining required correctness and consistency of stored data

DataBase Management System

- also called database system, or database engine is a generalized software system for manipulating databases
- basically a computerized record-keeping system; which stores information and allows users to add, delete, change, retrieve and update that information (Create, Read, Update, Delete – CRUD operations)
- provides for simultaneous use of database by multiple users

DataBase Management System

- also a collection of programs that enables users to create and maintain database; general purpose software system that facilitates process of defining (specifying data types, structures and constraints), constructing (storing data on media) and manipulating (querying to retrieve specific data, updating to reflect changes and generating reports from data) for various applications

DataBase Management System typically has three components

1. Data description language (DDL) - allows users to define database: specify data types & data structures, constraints on data; translates schema into object schema, thereby creating logical and physical layout of database
2. Data manipulation language (DML) and query facility - allows users to insert, update, delete and retrieve data; provides general query facility through Structured Query Language
3. Software for controlled access of database - provides controlled access to database; preventing unauthorized user trying to access database, providing concurrency control to allow shared access, recovery control system to restore database to previous consistent state following failure, …

DataBase Administrator

- someone at senior level in organization understands data and needs
- whose job is to decide what data should be stored in database and establish policies for maintaining and dealing with that data
- what information is to be stored, identifies entities of interest, decides content of database at abstract level
- process known as *logical* or *conceptual database design* - overall understanding of system

DataBase Administrator

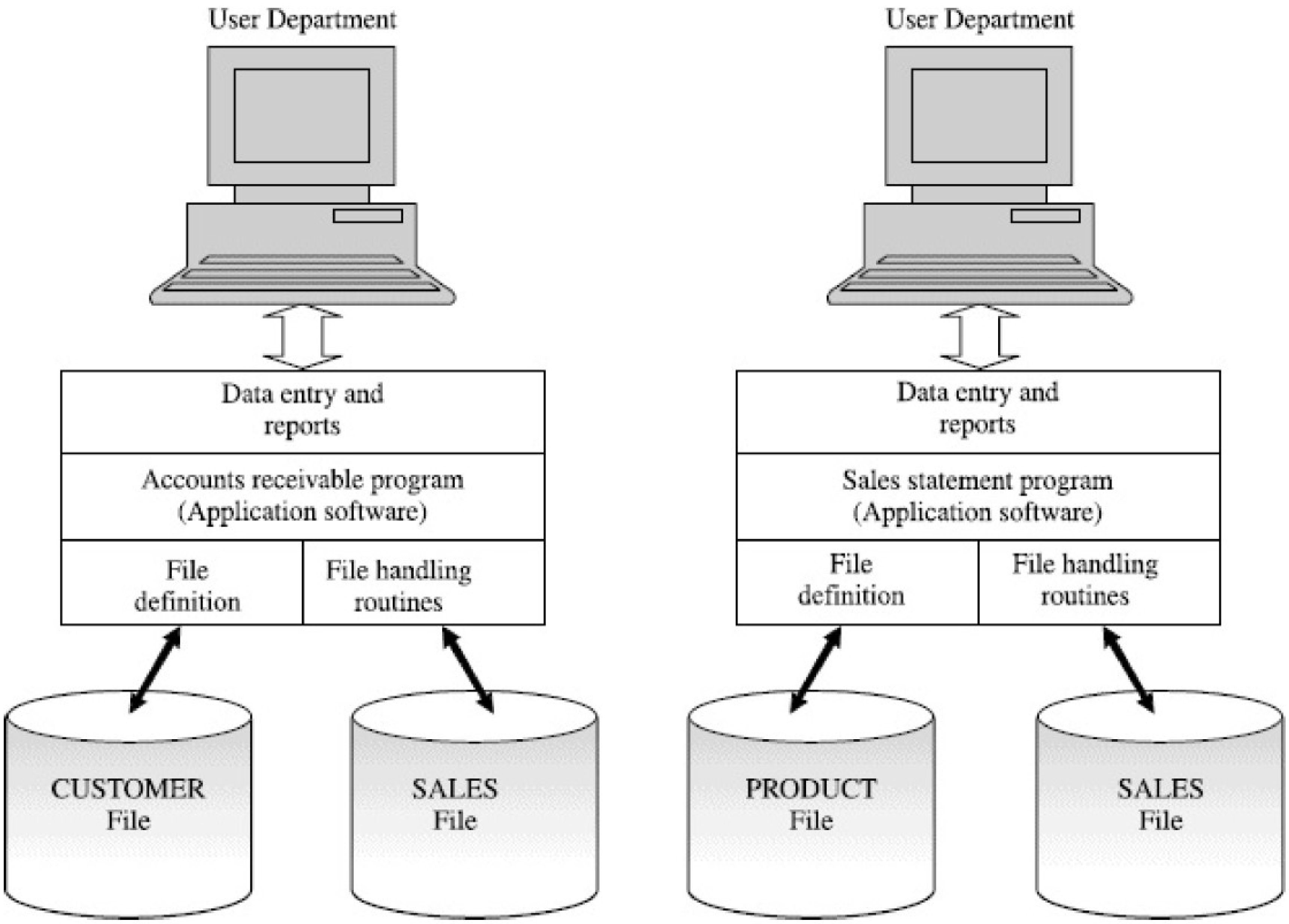
- provides necessary technical support for implementing policy decisions of databases; responsible for overall control of system at technical level; oversees and manages all resources; responsible for authorizing access, for coordinating and monitoring its use and for acquiring software and hardware resources as needed; accountable for security system, appropriate response time and ensuring adequate performance

DataBase Administrator Functions and Responsibilities

- defining conceptual schema and database creation
- storage structure and access-method definition
- granting authorization to the users
- physical organization modification
- routine maintenance
- job monitoring

FILE-ORIENTED SYSTEM VS. DATABASE SYSTEM

- File-oriented systems were an early attempt to computerize manual filing system
- systems performed normal record-keeping functions, they were called *data processing (DP) systems*
- with assistance of DP department, files were used for number of different applications by user departments
- with programs which accomplishes specific task of practical value in business (called *application program or software*), developed & designed to meet specific needs of particular requesting department or user group


User Department
Data entry and reports
Accounts receivable program (Application software)
File definition
File handling routines
CUSTOMER File
SALES File
User Department
Data entry and reports
Sales statement program (Application software)
File definition
File handling routines
PRODUCT File
SALES File

- it can be seen from the above examples that there is significant amount of duplication of data storage in different departments

Disadvantages of File-oriented System

- Data redundancy (or duplication)
- Data inconsistency (or loss of data integrity)
- Program-data dependence
- Inadequate data manipulation capabilities
- Excessive programming effort
- Limited data sharing
- Poor data control
- Security problems

Data redundancy (or duplication):

- each department used their own independent application programs and special files of data; resulted into duplication of same data
- is wasteful and requires additional or higher storage space, costs extra time and money, and ***requires increased effort to keep all files up-to-date***

Data inconsistency (or loss of data integrity)

- data redundancy also leads to *data inconsistency* (or loss of *data integrity*)
- since either data formats or values may be inconsistent, may no longer agree
- over period of time, such discrepancies can cause serious degradation in quality of information contained in data files and can also affect accuracy of reports and performance of bussines

Program-data dependence

- physical structure, storage of data files and records are defined within each application program
- program contains detailed file description for files
- as a consequence, any change for file structure requires changes to file description for all programs that access file
- difficult to even locate all programs affected by such changes; subject to error when making changes

Poor data control

- there is no centralized control at data element
- common for data field to have multiple names defined by various departments and depending on file
- could lead to different meanings in different context
- leads to poor data control, resulting in big confusion

Limited data sharing

- each application has its own private files and users have little opportunity to share data outside their own applications
- to obtain data from several incompatible files in separate systems will require major programming effort

Inadequate data manipulation capabilities

- file-oriented systems do not provide strong connections between data in different files and therefore its data manipulation capability is very limited

Excessive programming effort

- high interdependence between program and data in file-oriented system
- new applications often requires number of other data fields that may not be available in existing file; programmer had to rewrite code for definitions for needed data fields from existing file as well as definitions of all new data fields
- each new application required that programmers essentially start from scratch by designing new file formats and descriptions and then write file access logic for each new program

Security problems

- since applications programs are added to file-oriented system in an ad hoc manner, it was difficult to enforce such security system

Data dependence

- physical structure and storage of data files and records are defined in application code
- changes to existing structure are difficult to make
- all programs that access file must be modified to conform to new file structure

DataBase Approach

- previously mentioned limitations of file-based approach can be attributed to
- (1) definition of the data is embedded in application programs, rather than being stored separately and independently
- (2) there is no control over access and manipulation of data beyond that imposed by application programs

Data Abstraction

- definition of data is separated from application programs - internal definition and users see only external definition
- can change internal definition without affecting users, provided that external definition remains
- if new data structures are added or existing structures are modified, then application programs are unaffected, provided that they do not directly depend upon what has been modified

Driving Force for Database Systems

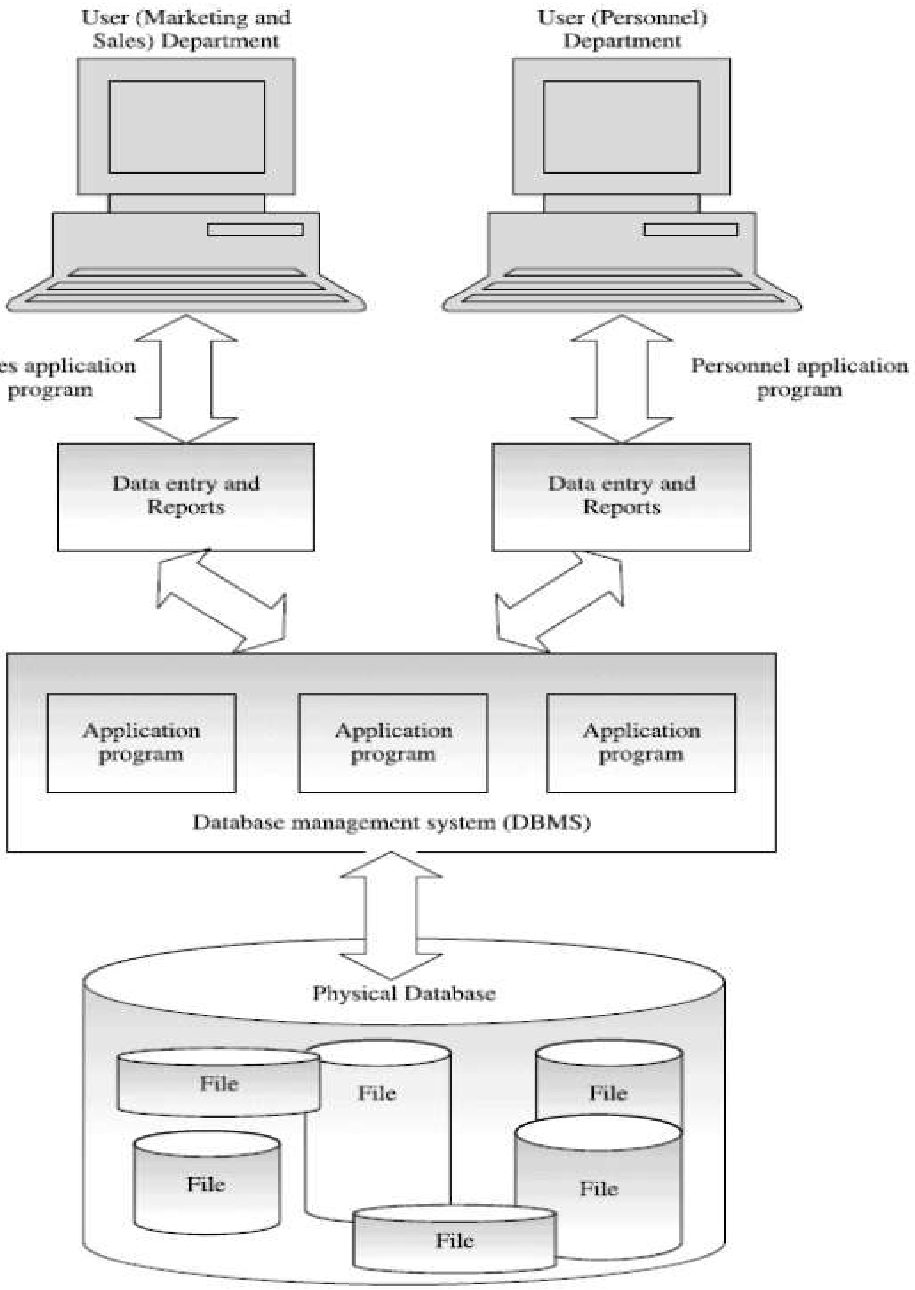
- Information as a Corporate Asset
  - ***autovit.ro sell 4.7 mil. Euro***
- Explosive Growth of Computer Technology
- Escalating Demand for Information
- Inadequacy of Earlier Data Systems

problems inherent in file-oriented systems make using database system very desirable

- Data redundancy (or duplication)
- Data inconsistency (or loss of data integrity)
- Program-data dependence
- Inadequate data manipulation capabilities
- Excessive programming effort
- Limited data sharing
- Poor data control
- Security problems

database approach

- database system consists of logically related data stored in a single data dictionary
- change in way end user data are stored, accessed and managed
- emphasizes integration and sharing of data throughout organization
- overcome disadvantages of file-oriented system; eliminate problems related with data redundancy and data control by supporting integrated and centralized data structure


User (Marketing and Sales) Department
User (Personnel) Department
Sales application program
Personnel application program
Data entry and Reports
Data entry and Reports
Application program
Application program
Application program
Database management system (DBMS)
Physical Database
File
File
File
File
File
File

Database System Environment

- refers to organization of components that define and regulate collection, storage, management and use of data within database
- consists of:
  1. Data
  2. Hardware
  3. Software
  4. Users (People)

Data

- most important component of database system
- database are both *integrated* and *shared* in system (users can access same piece of data concurrently (at same time))

Hardware

- physical devices, servers and client computers

Software

- application programs together with operating system
- DataBase Management System (DBMS) - comprises all requests from users to access database
- application software to solve specific common problems
- written in programming language such as C, C++, Visual Basic, Java, COBOL, Ada, Pascal, php, Python, using SQL embedded

Users

- people interacting with database system
  - DataBase Administrators (DBAs)
  - database designers
  - application programmers who write database application programs
  - end users who interact with system

DataBase Administrators

- database are corporate resources that must be managed like any other resource
- Data Administrator
- DataBase Administrator

Data Administrators

- responsible for management of data resource, including database planning; development and maintenance of standards, policies and procedures; and conceptual/logical database design
- consults with and advises senior managers, ensuring that direction of database development will ultimately support corporate objectives

DataBase Administrators

- responsible for physical realization of database, including physical database design and implementation, security and integrity control, maintenance of operational system, and ensuring satisfactory performance of applications for users
- DBA more technically oriented than role of DA
- requiring detailed knowledge of target DBMS and system environment
- in some organizations there is no distinction

Database Designers

- in large database design projects, we can distinguish between two types of designer:
- logical database designers
- physical database designers

Logical Database Designers

- concerned with identifying data, relationships and constraints
- must have thorough and complete understanding of organization’s data and any constraints on this data (sometimes called **business rules)**
- these constraints describe main characteristics of data as viewed by organization

Logical Database Designers

- must involve all prospective database users in development of data model
- split work into two stages:
- conceptual database design, which is independent of implementation details, such as target DBMS, application programs, program languages, or any other physical considerations
- logical database design, which targets specific data model, such as relational

Physical Database Designers

- decides how logical database design is to be physically realized
- mapping logical database design into set of tables and integrity constraints
- selecting specific storage structures and access methods for data to achieve good performance
- designing any security measures required on data

Application developers

- application programs to provide required functionality for end-users must be implemented
- work from specification produced by systems analysts
- programs contains statements that request DBMS to perform some operation on database
- programs written in programming language

End-Users

- are the “clients” of database, which has been designed and implemented and is being maintained to serve their information needs; can be classified according to way they use the system:
- Naïve users
- Sophisticated users

Naïve End-Users

- typically unaware of DBMS
- access database through specially written application programs that attempt to make operations as simple as possible
- invoke database operations by entering simple commands or choosing options from menu
- means that they do not need to know anything about database or DBMS

Sophisticated End-Users

- familiar with structure of database and facilities offered by DBMS
- may use high-level query language such as SQL to perform required operations
- may even write application programs for their own use

Advantages of DBMS
due to centralized management and control

- Minimal data redundancy
- **Program-data independence**
- Efficient data access
- Improved data sharing
- Improved data consistency
- Improved data integrity
- Improved security
- Increased productivity of application development
- Enforcement of standards
- Economy of scale
- Balance of conflicting requirements
- Improved data accessibility and responsiveness
- Increased concurrency
- Reduced program maintenance
- Improved backup and recovery services
- Improved data quality

Minimal data redundancy

- views of different user groups are integrated
- unnecessary duplication of data are avoided
- facts are ideally recorded in only one place in database
- data storage requirement is effectively reduced
- sometimes there are sound business and technical reasons for maintaining multiple copies of same data, for example, to improve performance, relationships and so on…
- in database system redundancy can be controlled - DBMS is aware of it, assumes responsibility for propagating updates and ensuring that multiple copies are consistent

Program-data independence

- separation of data from application programs is called data independence
- allows for changes at one level of database without affecting other levels - changes are absorbed by mappings between levels
- allows organization's data to change and evolve (within limits) without changing application programs that process the data

Efficient data access

- DBMS use variety of sophisticated techniques to store and retrieve data efficiently

Improved data sharing

- data belonging to entire organization (all departments)
- existing application programs can share data
- new application programs can be developed on existing data - to share same data and add only data that is not currently stored
  - rather then having to define all data requirements again

Improved data consistency

- corollary to redundancy
- when data is duplicated and changes made at one site are not propagated to other site, it results into inconsistency
- database supplies incorrect or contradictory information to its users
- redundancy is removed or controlled, chances of having inconsistence data is also removed and controlled - inconsistencies are avoided

Improved data integrity

- means that data contained in database is both accurate and consistent
- integrity usually expressed in terms of *constraints,* which are consistency rules that database system should not violate
- database system ensures that adequate checks are incorporated
- Price > 0, 1 <= Month <= 12, ensure that if there is reference to certain object, that object must exit
- for example, user is not allowed to transfer fund from nonexistent saving to checking account

Improved security

- protection of database from unauthorized users
- proper (centralized) access procedure is followed

Increased productivity of applications

- DBMS provides many of standard functions that application programmer would normally have to write in file-oriented application
  - provides all low-level file-handling routines
  - provide high-level environment consisting of productivity tools, such as forms and report generators, to automate some of activities of database design and simplify development of database applications
- results in increased productivity of programmer and reduced development time and cost

Enforcement of standards

- with central control defines and enforces necessary standards
- standards can be defined for data formats to facilitate exchange of data between systems, naming conventions, display formats, report structures, terminology, documentation standards, update procedures, access rules and so on
- including business rules

Economy of scale

- permits consolidation of data and applications
- thus reduces amount of wasteful overlap between activities of data-processing personnel in different departments
- combined low cost budget is required (instead of accumulated large budget)
- reduces overall costs of operation and management

Balance of conflicting requirements

- knowing overall requirements of organization (instead of requirements of individual users), resolves conflicting requirements of various users and applications
- structure the system to provide overall service that is best for organization
- chose best structure and access methods to get optimal performance for response-critical operations, while permitting less critical applications to continue to use database (with relatively slower response)

Improved data accessibility&responsiveness

- provide query languages and or report writers that allow users to obtain required information
  - without requiring programmer to write some software to extract this information from database

Increased concurrency

- DBMSs manage concurrent databases access and prevents problems of loss of information or loss of integrity

Reduced program maintenance

- in file-oriented environments, descriptions of data and logic for accessing data are built into individual application programs; as a result, changes to data formats and access methods inevitably result in need to modify application programs
- in database environment, data are more independent of application programs

Improved backup&recovery services

- DBMS provides facilities for recovering from hardware or software failures through its back up and recovery subsystem
- database is restored to state it was before

Improved data quality

- database system provides number of tools and processes to improve data quality

HISTORICAL PERSPECTIVE OF DATABASE SYSTEMS

- during 1960s, US President, J.F. Kennedy initiated project “Moon Landing”
- project expected to generate large volume of data and there was no system available at that time - file-based system was unable to handle such voluminous data
- DataBase systems were first introduced during this time to handle such requirements

- North American Aviation (now known as Rockwell International), which was prime contractor for project, developed a software known as Generalized Update Access Method (GAUM) to meet voluminous data processing demand of project
- based on concept that smaller components come together as parts of larger components, and so on, until final product (man on moon) is assembled
- structure confirmed to up-down tree and was named ***hierarchical*** *structure*

- in mid-1960s, the first general purpose DBMS was designed by Charles Bachman at General Electric, and was called *Integrated Data Store* (*IDS*) - formed basis for ***network*** *data model*
- Bachman was first recipient of computer science equivalent of Nobel Prize, called *Association of Computing Machinery* (*ACM*) *Turing Award,* for work in database area (1973)

- network data model was standardized by *Conference of Data Systems Languages* (*CODASYL*), comprising representatives of US government and world of research, and business - it strongly influenced database systems
- CODASYL formed *Data Base Task Force* (*DBTG*) in 1967 to define standard specifications for an environment that would allow database creation and data manipulation

- IBM joined the North American Aviation to develop GAUM into what is known as *Information Management System* (*IMS*) DBMS, released in 1969
- More than 50 years later, old IBM slogan, “The world depends on it,” still holds true. Many of the largest corporations in the world rely on the system to run their everyday business. Indeed, more than 95 percent of the top Fortune 1000 companies use IMS to process more than 50 billion transactions a day and manage 15 million gigabytes of critical business data. And IBM continues to add new features to IMS to adjust to the changing IT world.

Tale of Vern Watts IMS inventor http://vcwatts.org/ibm_story.html

- from 1956, until 2009 Vern Watts worked on one single company, IBM, and only one single project, IMS database
- like …?

Tale of Vern Watts IMS inventor http://vcwatts.org/ibm_story.html

- from 1956, until 2009 Vern Watts worked on one single company (IBM) and only one single project IMS database
- like …?
- Er Bimbo de Oro (The Golden Boy), L'Ottavo Re di Roma (The Eighth King of Rome), Er Pupone (The Big Baby), Il Capitano (The Captain), and Il Gladiatore (The Gladiator), Francesco Totti spent his entire career at Roma

- IMS restricted to management of hierarchies of records to allow use of serial storage devices (magnetic tape were market requirement at that time)
- still the main hierarchical DBMS for most large mainframe computer installations

- SABRE system for making airline reservations was jointly developed by American Airlines and IBM around same time - today is being used to power popular web-based travel services

- hierarchical model - structured data as directed tree with root at top and leaves at bottom
- network model - structured data as directed graph without circuits, slight generalization that allowed to represent certain real-world data structures more easily
- during 1970s hierarchical and network DBMS were developed to cope with increasingly complex data structures that were extremely difficult to manage with conventional file processing methods
- approaches are still being used by organizations and are called *legacy systems*.

- drawbacks with hierarchical and network:
  - queries against data were difficult to execute, normally requiring program written by expert programmer who understood what could be a complex navigational structure of data
  - data independence is very limited so that programs are not insulated from changes to data formats
  - widely accepted theoretical foundation is not available

- in 1970, Edgar Codd, at IBM’s San Jose Research Laboratory, wrote landmark paper (Codd, E. F. (1970). "A relational model of data for large shared data banks". Communications of the ACM. 13 (6): 377) proposing new data representation framework called ***relational data model*** and non-procedural ways of querying data
- received widespread commercial acceptance and diffusion during 1980s, sparked development of several DBMSs based on relational model (starting with Oracle), along with rich body of theoretical results that placed database field on a firm foundation

- in relational model, all data are represented in the form of tables and simple language called *Structure Query Language* (*SQL*) is used for data retrieval
- simplicity of relational model, possibility of hiding implementation details completely from programmer and ease of access for non-programmers, solved major drawbacks of first-generation DBMSs
- Codd won 1981 ACM’s Turing Award for his work

- relational model was not used in practice initially because of its perceived performance disadvantages and remained academically interesting for users
- in 1980s, IBM initiated *System R* project that developed techniques for construction of an efficient relational database system
- resulted in relational model consolidating its position as dominant DBMS paradigm

- this led to development of fully functional relational database product, called Structured Query Language / Database System
- in late 1980s, early 1990s, SQL was standardized and was adopted by American National Standards Institute (ANSI) and International Standard Organizations (ISO)

- now there are several relational DBMSs for both mainframe and PC environments for commercial applications, such as Ingress from Computer Associates, Informix, ORACLE, IBM DB2, Access and FoxPro from Microsoft, Paradox from Corel Corporation, InterBase from Borland and R-Base
- PostgreSQL – academic, evolved at the University of California, Berkeley in 1982
- MySQL, Microsoft SQL Server, Access

- in late 1990s, new era of computing started, such as client / server computing, data warehousing, and Internet applications
- advances were made in many areas of database systems
- complex data types and multimedia data (including graphics, sound images and video) became increasingly common; an *object-oriented database* (*OODBMS*) and *objected-relational databases* (*ORDBMS*) were introduced during this period to cope with these

- the late 1990s also saw rise of Internet, three-tier client-server architecture, and demand to allow corporate databases to be integrated with Web applications
- the late 1990s saw development of **XML (eXtensible Markup Language),** which has had a profound effect on many aspects of IT
- XML 1.0 ratified by W3C - XML becomes integrated with DBMS products and native XML databases are developed

- + 2000 DB vendors provide for newer data types
- Spatial databases introduced
- Databases support ERP (Enterprise Resource Planning), DW (Data Warehouse), DM (Data Mining) applications
- Big Data
- No SQL

- in response to increasing complexity of database applications “new” systems have emerged
- object-oriented DBMS and object relational DBMS
- Oracle (Java)
- MS SQL Server (.NET)
- new generation NoSQL DBMS

Data Models

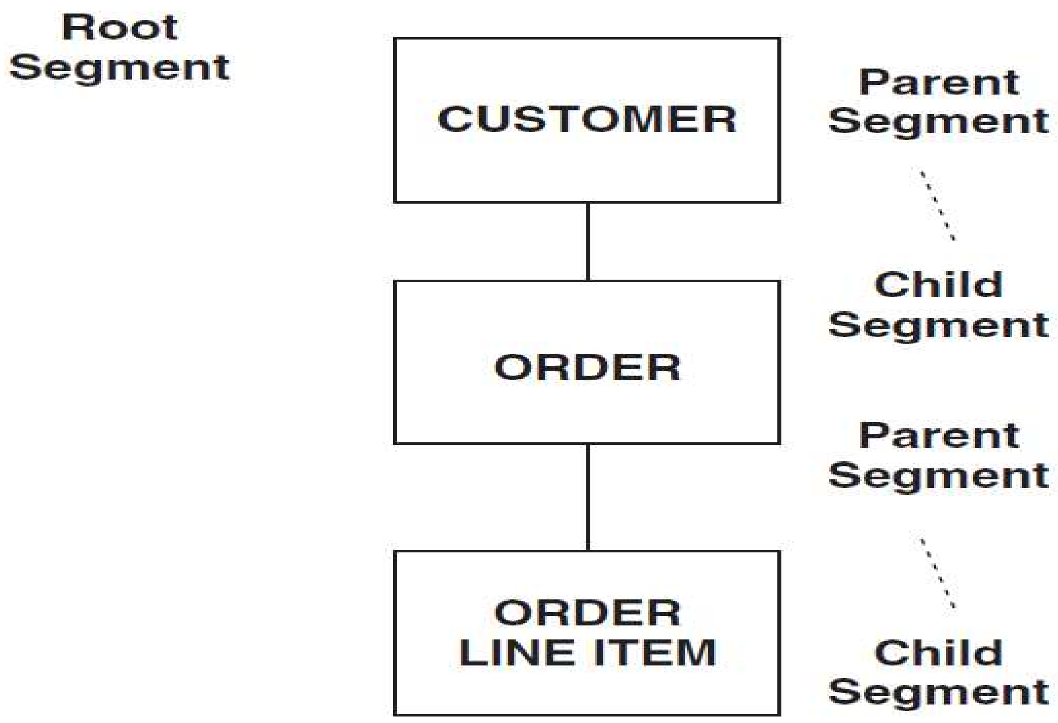
Data Models

- represents data requirements of an organization
- can diagrammatically show data model with symbols and figures
- data for an organization reside in database
- when designing database - first create data model
- would represent real-world data requirements
- would show arrangement of data structures

Data Models

- Hierarchical
- Network
- Relational
- Object – Relational

Hierarchical



Relationship links through physical pointers

Hierarchical

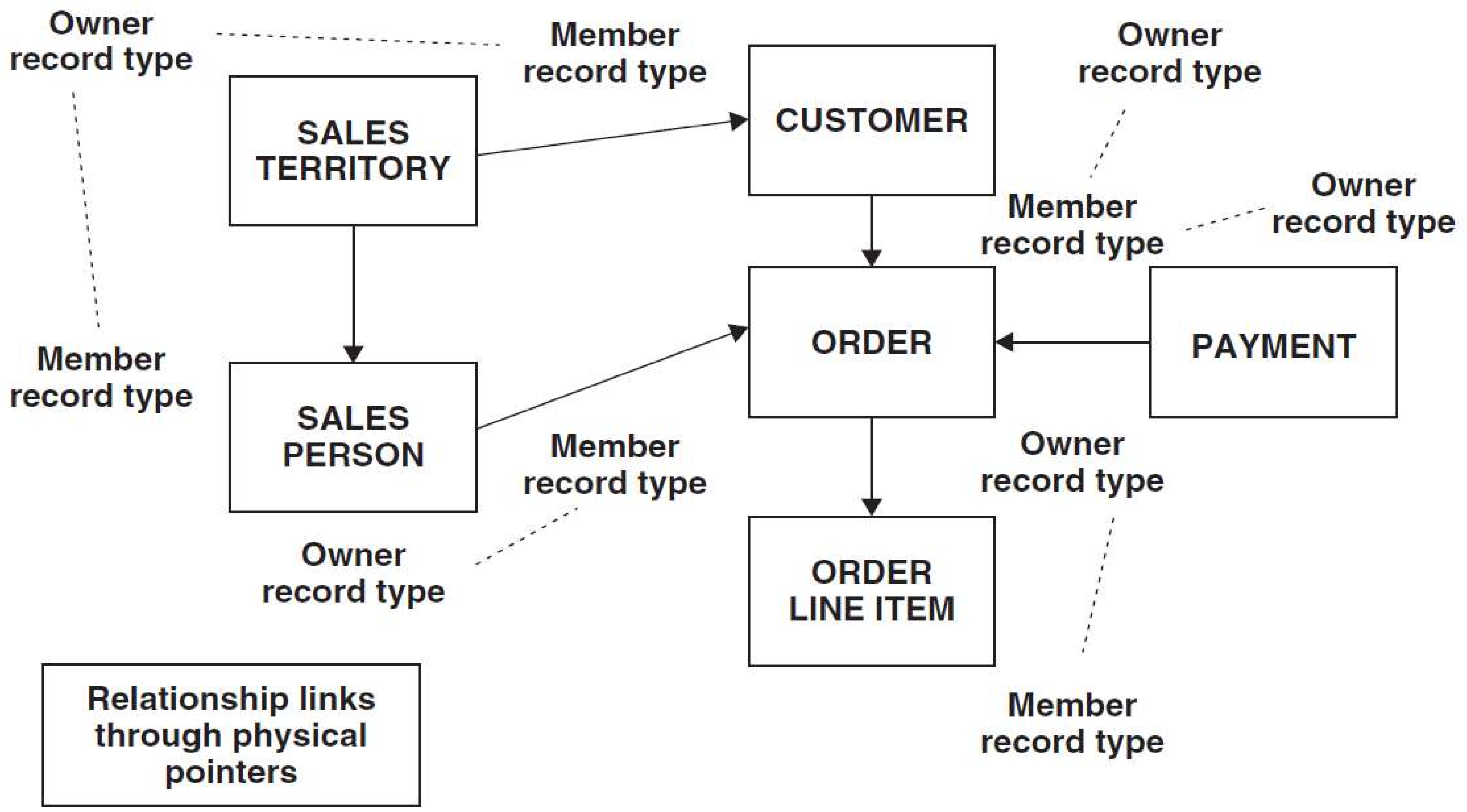
- *Levels* - each data structure representing business object is at one of hierarchical levels
- *Parent-Child Relationships* - relationship between each pair of data structures at levels next to each other is a parent-child relationship
- CUSTOMER is parent data segment whose child is ORDER data segment
- to separate orders into phone orders and mail orders - CUSTOMER may have PHONE ORDER and MAIL ORDER as two child segments

Hierarchical

- *Root Segment* - data segment at top level of hierarchy is known as root data segment (as in an inverted tree)
- *Physical Pointers* - orders of particular customer linked in implementation of hierarchical data model by means of physical pointers or physical storage addresses embedded in database
- forward or backward pointers / physical pointers link parent – child and records of same segment type

- hierarchical data model represents well any business data that inherently contains levels one below other - in real world, most data structures do not conform to hierarchical arrangement
- customers placing orders and making payments, salespersons being assigned, and salespersons being part of sales territories – data elements cannot be arranged in hierarchy
- relationships cross over among data elements as though they form a network

Network


Owner record type
Member record type
SALES TERRITORY
CUSTOMER
Owner record type
Member record type
Owner record type
Member record type
ORDER
PAYMENT
SALES PERSON
Member record type
Owner record type
Owner record type
ORDER LINE ITEM
Member record type
Relationship links through physical pointers

Network

- *Levels* - no hierarchical levels exist in network; lines in network data model simply connect appropriate data structures wherever necessary without restriction of connecting only successive levels as in the hierarchical model
- *Record Types* each data structure is known as a record type
  - CUSTOMER record type represents data content of all customers
  - ORDER record type represents data content of all orders

Network

- *Relationships* - expresses relationships between two record types by designating one as the owner record type and the other as the member record type
- each member type with its corresponding owner record type is known as set
- set represents relationship between an owner and a member record type

Network

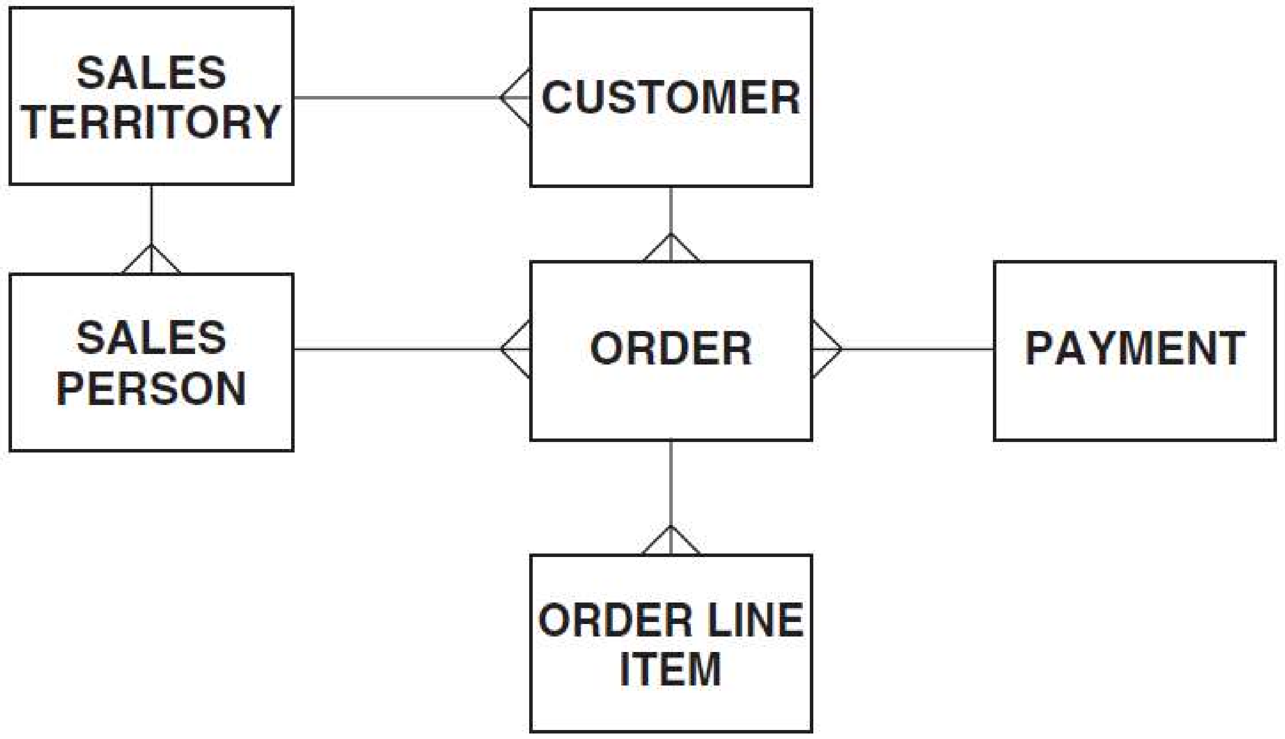
- *Multiple Parents* - for ORDER there are two parents or owner records, namely, CUSTOMER and PAYMENT
- for one occurrence of CUSTOMER, one or more occurrences of ORDER exist
- for one occurrence of PAYMENT there are one or more occurrences of ORDER
- by definition, hierarchical data model cannot represent this kind of data arrangement with two parents for one child data structure

Network

- *Physical Pointers* – related occurrences of two different record types in network model are connected by physical pointers or physical storage addresses embedded within database
- physical pointers link occurrences of an owner record type with corresponding occurrences of member record type
- within each record type itself individual occurrences may be linked to one another by means of forward and backward pointers

- in both of these models need physical pointers to connect related data occurrences
- have to rewrite physical addresses in data records every time you reorganize data, move data to different storage
- relational model establishes connections between related data occurrences by means of logical links implemented through foreign keys

Relational



Relationship links NOT through physical pointers, but by foreign keys

Relational

- *Levels* - no hierarchical levels are present; lines simply indicate relationships between appropriate data structures wherever necessary without restriction of connecting only successive levels (as in hierarchical)
- *Relations or Tables* - model consists of relations; a relation is two-dimensional table of data observing relational rules - for example:
  – CUST relation represents data content of all customer
  – ORDER relation represents data content of all orders

Relational

- *Relationships* - consider relationship between CUSTOMER and ORDER
- for each customer one or more orders may exist
- customer occurrence must be connected to all related order occurrences
- foreign key field included in ORDER data structure
- in each of order occurrences relating to certain customer, foreign key contains identification of that customer

Relational

- *Relationships* - consider relationship between CUSTOMER and ORDER
- when look for all orders for particular customer, you search through foreign key field of ORDER and find those order occurrences with identification of that customer in foreign key field
- *No Physical Pointers* - unlike hierarchical or network, relational model establishes relationships between data structures by means of foreign keys and not by physical pointers

Relational

- *Relationships* - consider relationship between CUSTOMER and ORDER
- when look for all orders for particular customer, you search through foreign key field of ORDER and find those order occurrences with identification of that customer in foreign key field

Object Relational

- demand for information continues to grow, organizations need database systems that allow representation of complex data types, user-defined sophisticated functions, and user-defined operators for data access.
- present viable solutions for handling complex data types; combines ability of object technology to handle advanced types of relationships with features of data integrity, reliability, and recovery found in relational

DATABASE LANGUAGES

- Data definition language (DDL)
- Storage definition language (SDL)
- View definition language (VDL)
- Data manipulation language (DML)
- Programming language, with embedded SQL
- Transactions

- in practice, there are not separate languages, instead they simply form parts of single database language and comprehensive integrated language is used such as widely used Structured Query Language (SQL)
- SQL represents combination of these, as well as statements for constraints specification

Data Definition Language (DDL)

- used to specify database conceptual schema
- supports definition of database objects (or data element)
- describe and name entities, attributes and relationships
  - together with associated integrity and security constraints
- theoretically, different DDLs are defined for each schema in three-level schema-architecture (for conceptual, internal and external schemas

Data Storage Definition Language (DSDL)

- used to specify internal schema in database
- mapping between conceptual schema (as specified by DDL) and internal schema (as specified by DSDL) may be specified in either one of these languages
- storage structure and access methods used by database system is specified
- define implementation details of database schemas, which are usually hidden from user

View Definition Language (VDL)

- used to specify user's views (external schema) and their mappings to conceptual schema
- in most of DBMSs, DDL is used to specify both conceptual and external schemas
- there are two views of data:
  - one is *logical view* of data - form that programmer perceives to be in
  - other is *physical view* - reflects way that data is actually stored on disk (or other storage devices)

Data Manipulation Language (DML)

- is mechanism that provides set of operations to support basic data manipulation operations on data held in database
- used to retrieve data stored in database, express database queries and updates - communicate with DBMS
- part of DML that provides data retrieval is called *query language*

https://ro.wikipedia.org/wiki/ Daniela_Crudu Educație Universitatea Spiru Haret (2009-2012)

- Create - add (or insert) records to database
- Read - retrieve data and/or records from database
- Update - modify data and/or record in database
- Delete - delete records from database

Transaction managements

- all work that logically represents single unit is called *transaction*
- sequence of database operations that represents logical unit of work is grouped together as single transaction and access database and transforms it from one state to another

Transaction managements

- when DBMS does ‘*commit*’, changes made by transaction are made permanent
- if changes are not be made permanent, transaction can be ‘*rollback*’ and database will remain in its original state

Transaction managements

- when updates are performed on database, we need some way to guarantee that set of updates will succeed all at once or not at all
- transaction (BEGIN TRANSCATION … END TRANSCATION) ensures that all work completes or none of it affects database
- necessary in order to keep database in consistent state
- for example, transaction might involve transferring money from bank saving to checking account

Transaction

- has four properties, called *ACID*:
  - Atomicity
  - Consistency
  - Isolation
  - Durability

Transaction

- *Atomicity* means that either all work of transaction or none of it is applied
- other operations can only access any rows involved transactional access either before transaction occurs or after transaction is complete, but never while transaction is partially complete
- *Consistency* means that transaction’s work will represent correct (or consistent) transformation of database’s state
- *Isolation* requires that transaction not to be influenced by changes made by other concurrently executing transactions
- *Durability* means that work associated with successfully completed transaction is applied to database and is guaranteed to survive system or media failures

Database Need

- amount of information available exploding
- value of data as organizational asset is widely recognized
  - ability to manage this vast amount of data
  - quickly find information relevant
- need for increasingly powerful and flexible data management systems

Database Importance

- there is not real application without a kind of database
- great number of DBMS packages can be found on software market, for all types of computers and processing technologies
- DBMS can be found at the top 3 of most needed, requested, sold and used products

Mathematics

- always strive for elegance and orthogonality - dislike exceptions
- system is said to be designed in an *orthogonal* way if its set of components that together make up whole system capability are *non-overlapping* and *mutually independent*
  - each capability should be implemented by only one component
  - one component should only implement one capability of system

Mathematics

- well-separated and independent components ensure that there are no side effects: using or even changing one component does not cause side effects in another area of system
- formal disciplines, most relevant ones in application of mathematics to field of databases
- Set theory, Relation theory

Mathematics

- *Everything should be made as simple as possible, but not simpler.*
    - Albert Einstein (1879–1955)

Relational model

- Professionals in any discipline need to know the foundations of their field
- Database Professional - need to know theory of relations
- is not product-specific;
  - rather, it is concerned with principles

Principles, not Products

- principle
  - source, root, origin; which is fundamental; essential nature; theoretical basis
- principles endure
- by contrast, products and technologies, change all the time

- E. F. Codd, at the time researcher at IBM; late in 1968 that Codd, a mathematician, first realized that discipline of mathematics could be used to inject some solid principles and rigor into field of database

original definition of relational model

- E. F. Codd
  - *Derivability, Redundancy, and Consistency of Relations Stored in Large Data Banks*, IBM Research Report RJ599, 1969.
  - *A Relational Model of Data for Large Shared Data Banks*, CACM 13, No. 6, 1970.
- C. J. Date
  - *An Introduction to Database Systems*, 8th Ed. Boston, Mass.: Addison-Wesley, 2004.
  - first edition of book was published in 1975

Database Development Process

Database Design Paradigm Shift

- structure of database is determined during **database design**
- to produce system that will satisfy organization's information needs requires different approach from that of file-based systems, where work was driven by application needs of individuals
- for database approach to succeed, organization now has to think of data first and application second – this change in approach is referred to as *paradigm shift*

Database Design Paradigm Shift

- system to be acceptable to end-users, database design activity is crucial
- poorly designed database will generate errors that may lead to bad decisions, which may have serious repercussions for organization
- well-designed database produces system that provides correct information for decision making process to succeed in an efficient way

- before database can be built, must understand how database will achieve its goals; verify that database will solve business problems
- design is necessary
- well-designed database is based on good **understanding of business process**
- first step in development of database is to identify business requirements and goals

- What are the business requirements for future database?
- What information needs to be stored in database?
- How can the data be stored?
- How can the data be presented to end users?

- next step is to **understand relationships**
- database designers translate data model to database tables with integrity constraints: specify tables and columns to organize business data into a well-defined **database structure**

- designers can use data model to verify if future database will meet database requirements
- usually, several modifications are needed
- might take some time to make sure that database design and business partners are all satisfied

- next step **implement database** with DBMS package; choose proper DBMS
- database designers or database administrators (DBAs) **create database** tables and populate them with business data
- test database; further modify database to meet users’ requirements

- once database is ready database application developers **create database application** objects to assist users in accessing database
- database should enforce necessary **security** measures

- for large applications database must be partitioned into multiple parts, and **partition database** distributed to multiple computers
- to support business decision making, database should also provide **data analysis tools**, such as OLAP, data mart or data warehouse, and data mining
- analyzer can store history data for fast information search and identify trend and patterns hidden in daily business operation data

Type of DataBases

Type of Databases

- ***How Databases Are Used***
- Production Database
- Decision Support System
- Mass Deployment

Production Database

- Support business functions
- Online transaction processing
- Usage includes CRUD activities (Create, Read, Update, Delete)
- Features include concurrency, security,
- transaction processing, security; OnLine Transaction Processing (OLTP) systems

Decision Support System

- Used for analysis, querying, and reporting
- Generally read-only
- Features include query tools and custom applications
- Data warehouse and OnLine Analytical Processing (OLAP) systems

Mass Deployment

- Intended for single user environments
- Workstation / mobile versions of database products
- Ease of use important
- Features include report and application generation capabilities

Type of Databases

- ***Where Data Are Used***
- Centralized database
- Distributed database
  - Homogeneous databases
  - Heterogeneous databases

DataBase Market

DataBase Market

- the five leading commercial relational database vendors by revenue
- Oracle (48.8%)
- IBM (20.2%)
- Microsoft (17.0%)
- SAP including Sybase (4.6%)
- Teradata (3.7%)

DataBase Market

- the three leading open source implementations are MySQL, PostgreSQL, and SQLite
- MariaDB is a prominent fork of MySQL prompted by Oracle's acquisition of MySQL AB

DataBase Market

- the percentage of database sites using any given technology were (site may deploy multiple)
- Oracle Database - 70%
- Microsoft SQL Server - 68%
- MySQL (Oracle Corporation) - 50%
- IBM DB2 - 39% IBM Informix - 18%
- SAP Sybase Adaptive Server Ent. - 15%
- SAP Sybase IQ - 14%
- Teradata - 11%

Review Questions

Review Questions

- list five government sectors that use database systems
- discuss each of the following terms:
  - data; database
  - database management system
  - database application program
  - data independence
  - security; integrity; views
- describe role of DBMS database approach

Review Questions

- discuss why knowledge of DBMS is important for DBA
- describe main characteristics of database approach and contrast it with file-based
- describe five components of DBMS environment and discuss how they relate to each other
- discuss the three generations of DBMS

Review Questions

- discuss roles of following personnel in database environment:
  - data administrator
  - database administrator
  - logical database designer
  - physical database designer
  - application developer
  - end-users
- why are views important aspect of database

Review Questions

- Why is business process information important?
- Describe functions of database in business process
- What is metadata?
- Explain functions of ODBC and ADO
- Name commonly used database components
- Describe tables and their structures

- as always it is not possible to close whiteout **thank you for your kindly attention!**
- *If you get half as much pleasure - the guilty variety, to be sure - from reading this slides as I get from writing it, we're all doing pretty well.*