Introduction to Database Systems

ACS 575: Database Systems

Instructor: Dr. Jin Soung Yoo, Professor
Department of Computer Science
Purdue University Fort Wayne

References

- R. Elmasri et al., Fundamentals of Database Systems, 7th ed. Ch1, Ch2
- W. Lemanhieu, et al., Principles of Database Management, Ch 1, Ch 2
- J. A. Hoffer et al., Modern Database Management, 13th ed. Ch1

Outline

- □ What is a database?
- □ Historical root of database systems
- □ Advantages of database approach
- □ Components of database environments
- □ Types of Database Applications
- □ Data Model
- □ Database system development life cycle
- □ Evolution of database technology
- □ Range of database systems

Data and Data Management

- Data are everywhere and come in different shapes and volumes
  - Traditional numeric and alphanumeric data in an application
  - Multimedia data such as pictures, audio, or video
  - All types of spatial or geographical data
  - *Volatile* data – high-frequency (speed) data
  - *Big Data*
- These data need to be stored and managed using appropriate data management

Data

- ***Data***
  - Facts concerning objects and events that could be recorded and stored on computer media.
  - Stored representations of meaningful objects and events
  - A set of values of qualitative or quantitative variables
  - Measured, collected, and analyzed.

Structured vs. Unstructured Data

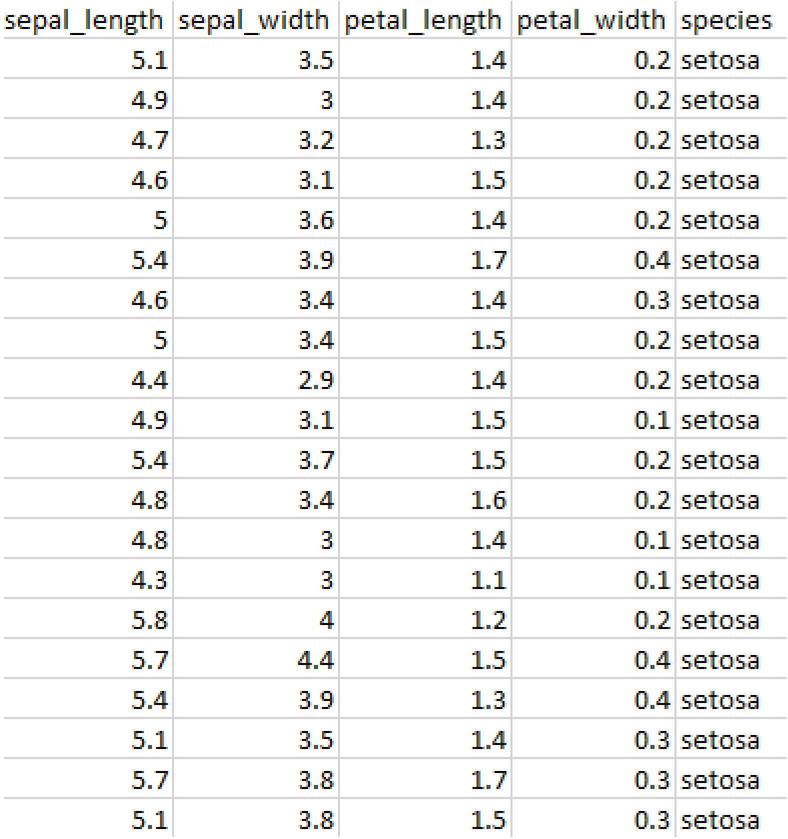
- *Structured data:* data organized in a structure
  - Individual characteristics of data items can be identified and formally specified and can be described according to a formal logical data model
  - E.g., a salesperson’s data would include facts such as name, address, and telephone number.
  - Structured data is presented with traditional data types such as numeric, character, and date data types.
- *Unstructured data:* data with no identifiable structure,
  - Not organized in a pre-defined manner.
  - E.g., documents, emails, tweets, short text messages, maps, images, sound, and video segments

Semi-Structured Data

- *Semi-structured data* is data which does have a certain structure, but the structure may be very irregular or highly volatile.
  - The data may loosely exhibit the same structure, but which do not comply entirely with a single, rigid format.
  - E.g., Individual users’ webpages on a social media platform, or resume documents in a human resources database

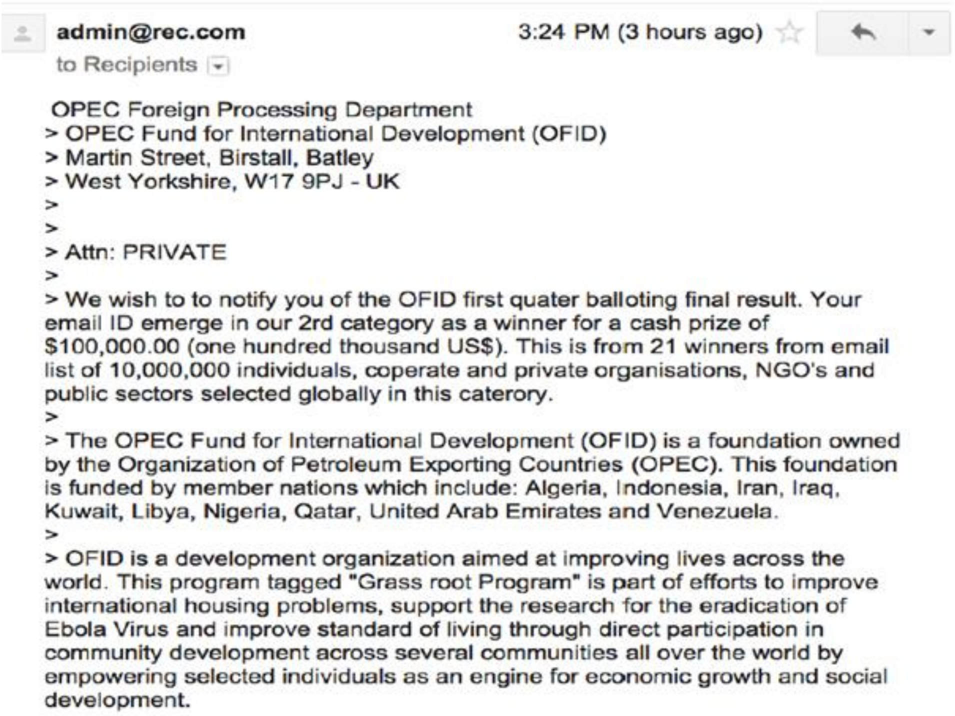
Examples

(1) Structured record data



sepal_length	sepal_width	petal_length	petal_width	species
5.1	3.5	1.4	0.2	setosa
4.9	3	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa
5.4	3.7	1.5	0.2	setosa
4.8	3.4	1.6	0.2	setosa
4.8	3	1.4	0.1	setosa
4.3	3	1.1	0.1	setosa
5.8	4	1.2	0.2	setosa
5.7	4.4	1.5	0.4	setosa
5.4	3.9	1.3	0.4	setosa
5.1	3.5	1.4	0.3	setosa
5.7	3.8	1.7	0.3	setosa
5.1	3.8	1.5	0.3	setosa

(2) Unstructured email text data



admin@rec.com 3:24 PM (3 hours ago)

to Recipients

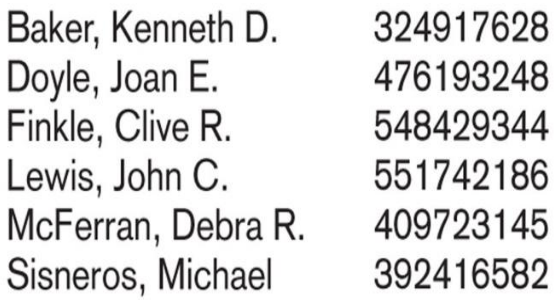
OPEC Foreign Processing Department
> OPEC Fund for International Development (OFID)
> Martin Street, Birstall, Batley
> West Yorkshire, W17 9PJ - UK
>
>
> Attn: PRIVATE
>
> We wish to to notify you of the OFID first quater balloting final result. Your email ID emerge in our 2rd category as a winner for a cash prize of $100,000.00 (one hundred thousand US$). This is from 21 winners from email list of 10,000,000 individuals, coperate and private organisations, NGO's and public sectors selected globally in this caterory.
>
> The OPEC Fund for International Development (OFID) is a foundation owned by the Organization of Petroleum Exporting Countries (OPEC). This foundation is funded by member nations which include: Algeria, Indonesia, Iran, Iraq, Kuwait, Libya, Nigeria, Qatar, United Arab Emirates and Venezuela.
>
> OFID is a development organization aimed at improving lives across the world. This program tagged "Grass root Program" is part of efforts to improve international housing problems, support the research for the eradication of Ebola Virus and improve standard of living through direct participation in community development across several communities all over the world by empowering selected individuals as an engine for economic growth and social development.

Databases

- ***Database***: an organized collection of logically related data
- A database is typically organized to model relevant aspects of reality (e.g., availability in hotels),
  in a way that supports processes requiring this information (e.g., finding a hotel with vacancies)
- Structured and unstructured data are often combined in the same database to create a true multimedia environment.
  - E.g., An automobile repair shop database may have structured data (e.g., customers and automobiles) with multimedia data (photo images of the damaged autos and scanned images of insurance claim forms).

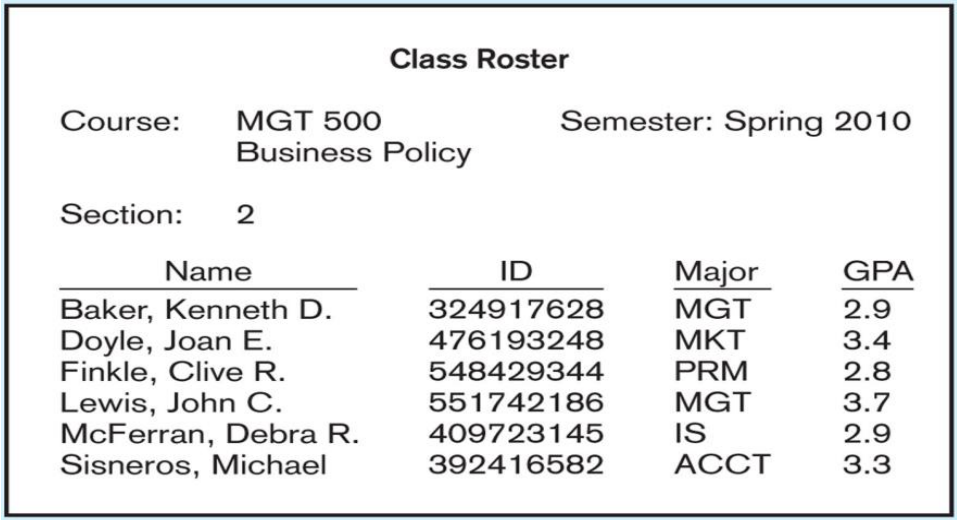
Data vs. Information

- ***Information***: data that have been processed in such a way that the knowledge of the person who uses the data is increased.
  - The terms data and information are closely related and in fact are often used interchangeably.



Baker, Kenneth D.	324917628
Doyle, Joan E.	476193248
Finkle, Clive R.	548429344
Lewis, John C.	551742186
McFerran, Debra R.	409723145
Sisneros, Michael	392416582

(a) Raw data (list of facts)
(What the entries mean?)



Class Roster

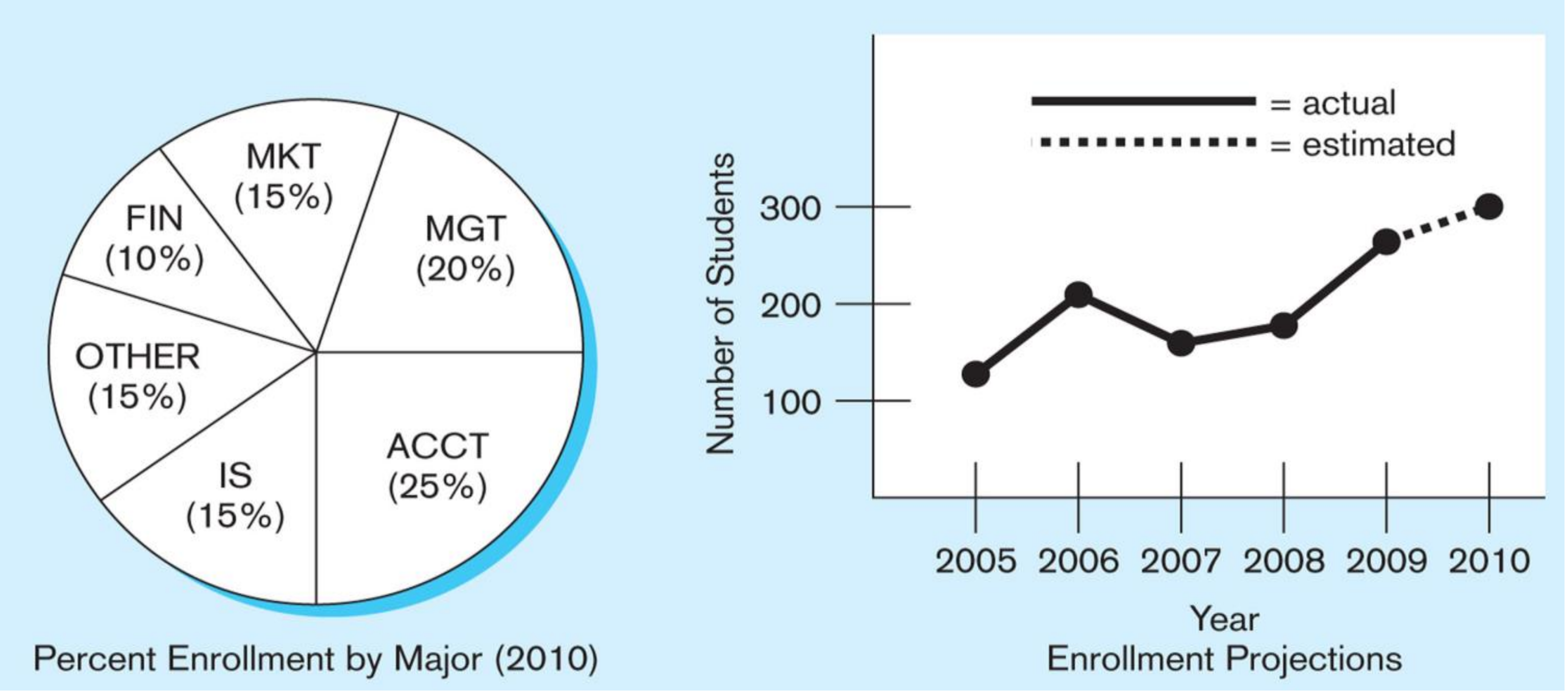
Course: MGT 500 Business Policy Semester: Spring 2010

Section: 2

Name	ID	Major	GPA
Baker, Kenneth D.	324917628	MGT	2.9
Doyle, Joan E.	476193248	MKT	3.4
Finkle, Clive R.	548429344	PRM	2.8
Lewis, John C.	551742186	MGT	3.7
McFerran, Debra R.	409723145	IS	2.9
Sisneros, Michael	392416582	ACCT	3.3

(b) **Information (data in context)**
(Context helps users understand the data, with providing some structure)

Information

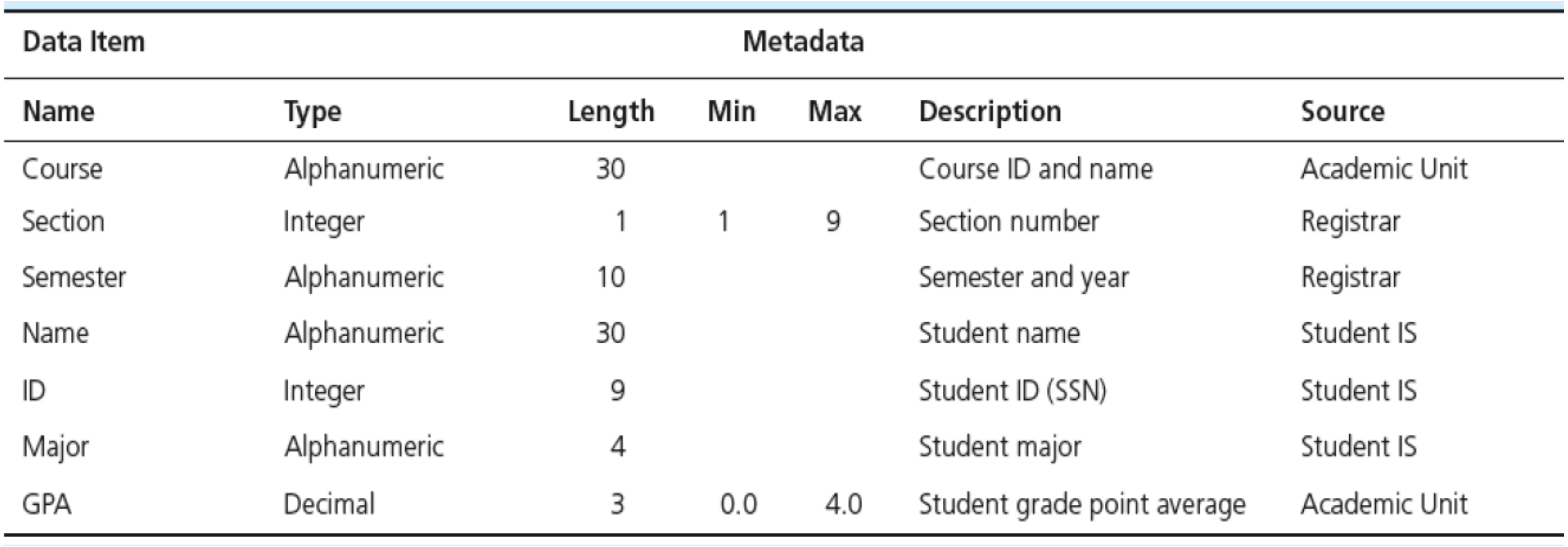


(c) **Information (Summarized data)**

- Another way to convert data to information is to summarize them, or process and present them for human interpretation.

Metadata

- ***Metadata*** is data that describe the properties or characteristics of user data
  - E.g., data item name, length (or size), data types and allowable values
  - The primary mechanism for providing contexts for data.



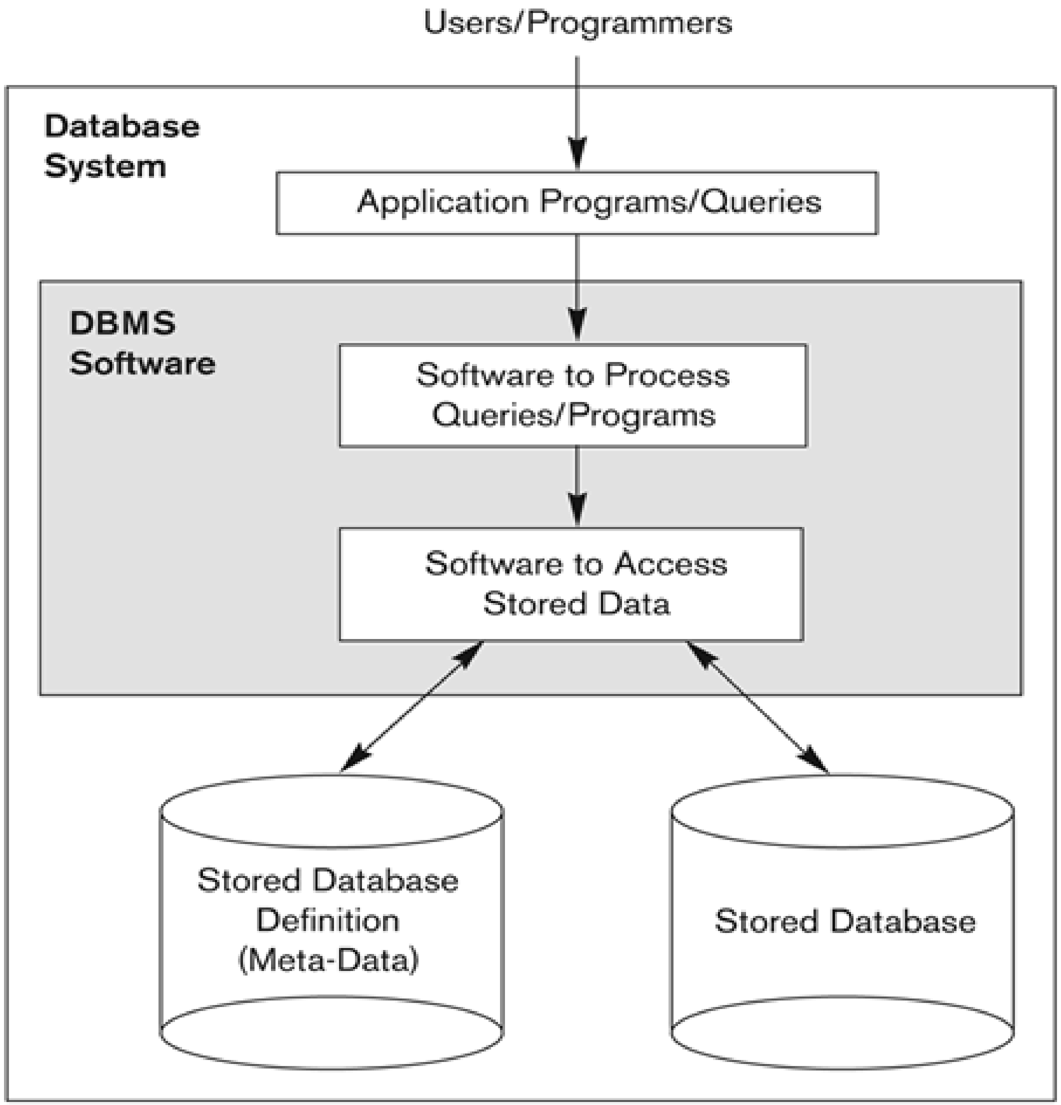
Data Item	Metadata					
Name	**Type**	**Length**	**Min**	**Max**	**Description**	**Source**
Course	Alphanumeric	30			Course ID and name	Academic Unit
Section	Integer	1	1	9	Section number	Registrar
Semester	Alphanumeric	10			Semester and year	Registrar
Name	Alphanumeric	30			Student name	Student IS
ID	Integer	9			Student ID (SSN)	Student IS
Major	Alphanumeric	4			Student major	Student IS
GPA	Decimal	3	0.0	4.0	Student grade point average	Academic Unit

DBMS vs. Database Systems

- ***Database Management System (DBMS)***
  - DBMS is a software package designed to store and manage databases.
  - DBMS is used to create and maintain user databases, and provide controlled access to the database.
  - DBMS manages data resources like an operating system manages hardware resources

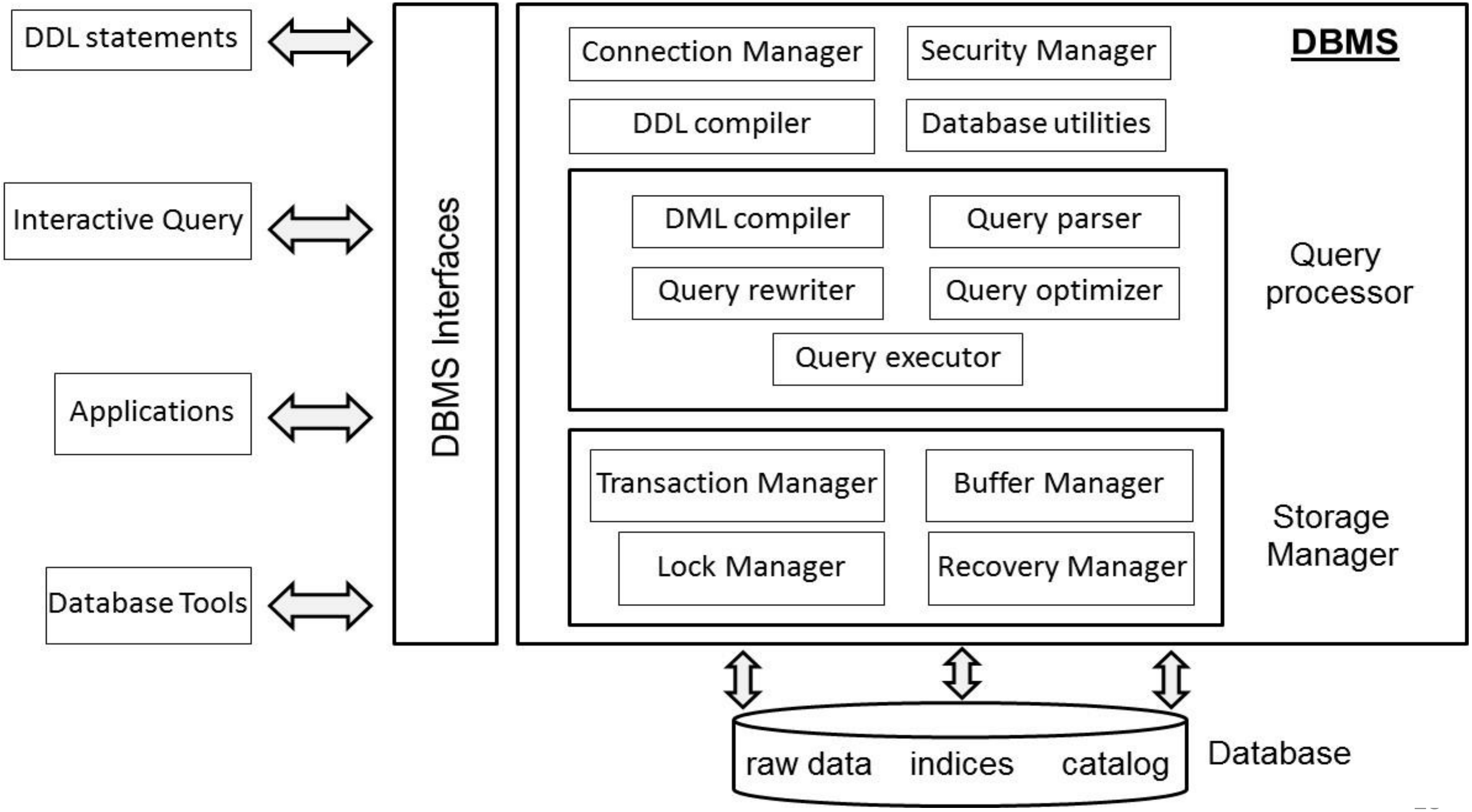
- ***Database System***
  - A system with DBMS software and the data itself. Sometimes, the applications are also included.

Simplified Architecture of Database System


Users/Programmers
Database System
Application Programs/Queries
DBMS Software
Software to Process Queries/Programs
Software to Access Stored Data
Stored Database Definition (Meta-Data)
Stored Database

Source: R. Elmasri et al., Fundamentals of Database Systems, Ch1

Architecture of a DBMS



Outline

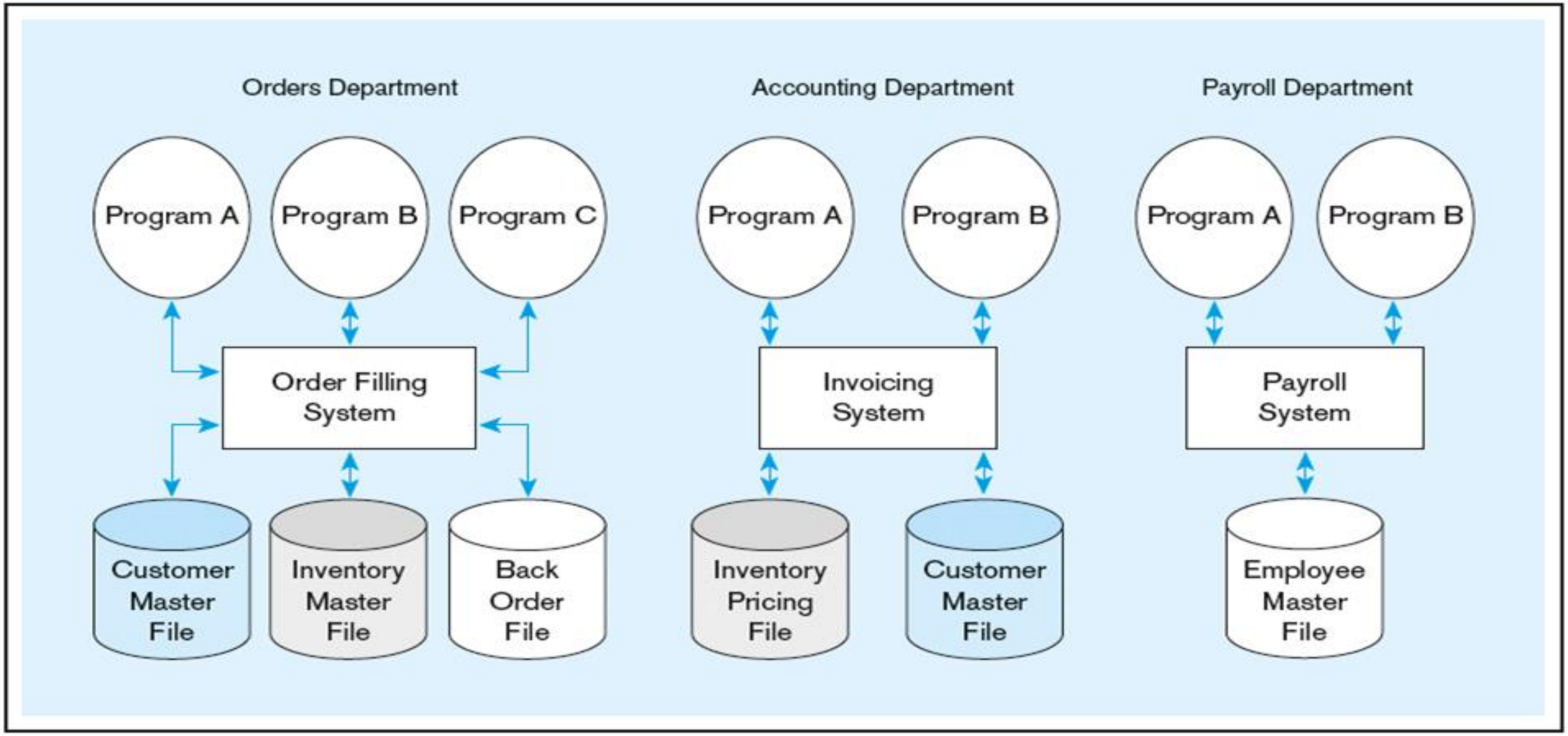
- □ What is a database?
- ☞ **Historical root of database systems**
- □ Advantages of database approach
- □ Components of database environments
- □ Types of Database Applications
- □ Data Model
- □ Database system development life cycle
- □ Evolution of database technology
- □ Range of database systems

Historical Root of Database: Traditional File Processing Systems

- In times with no databases, computer ***file processing systems*** were used to store, manipulate and retrieve large files of data for business applications.
  - A ***file*** is a collection of related records.

- Traditional file processing systems are designed for the data processing needs of individual departments rather than the overall information needs of the organization.

File-based Approach for Data Management



Major data files (such as customer file) are associated with each application (such as order filling and invoicing)

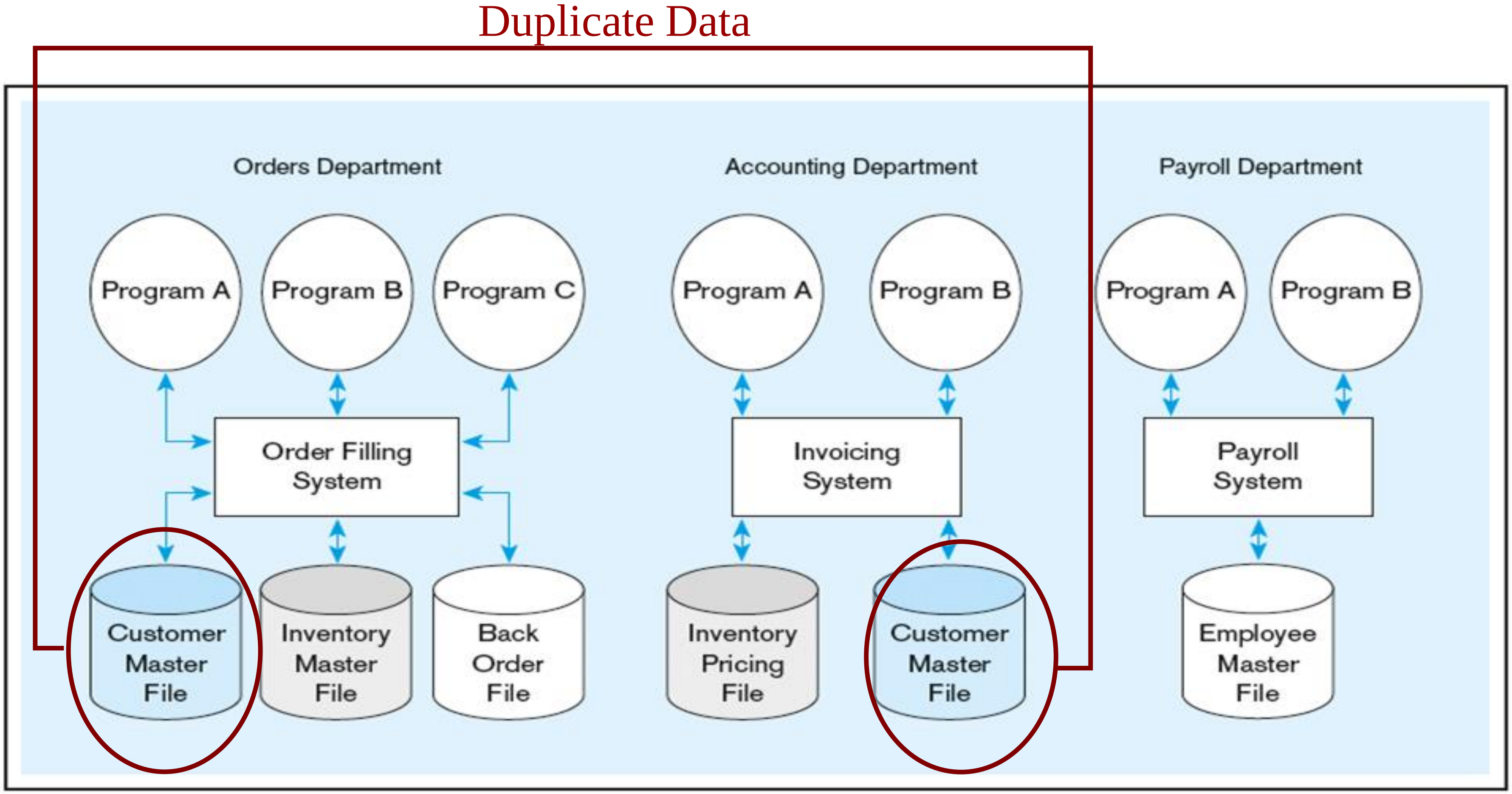
Disadvantages of Data Management in File System

- Program-Data Dependence
- Duplication of Data
- Limited Data Sharing
- Lengthy Development Times
- Excessive Program Maintenance

Program-Data Dependence

- Each application program must maintain its own data
  - Each program has its own processing routines for reading, inserting, updating, and deleting data from files
- File descriptions are stored within each application program that accesses a given file.
- Any change to a file structure requires changes to the file descriptions for all programs that access the file.
  - E.g., If the customer address field length in the records is changed from 30 to 40 characters, all the programs associated with the data file should be changed for correctly accessing other fields after the address field.

Duplication of Data



Problems with Duplication of Data

- Waste of space to have duplicate data
- Causes more maintenance headaches
- The biggest problems are
  - Data inconsistencies

    : Different and conflicting versions of same data

    e.g., the same data item may have different names in different files.
  - Data anomalies

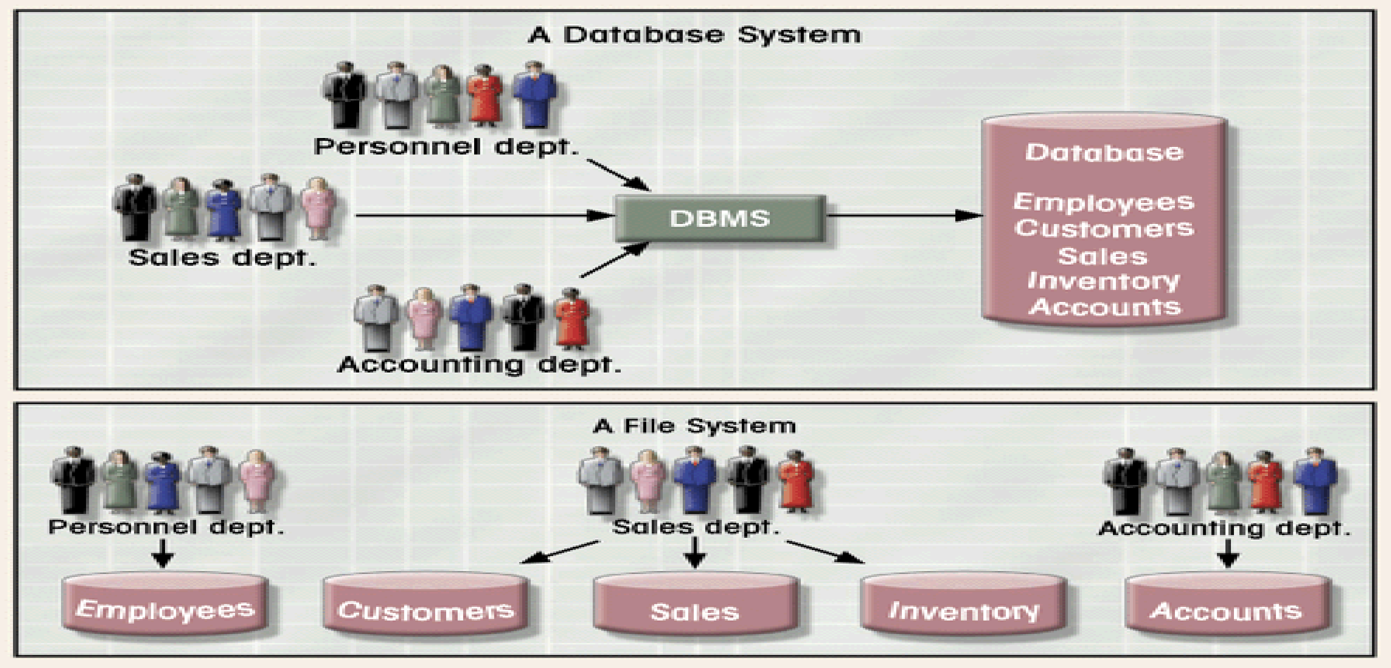
    : Abnormal status by forcing field value changes in different locations with inconsistent data

    e.g., update of agent's phone in AGENT file but not update of the same agent's phone in CUSTOMER file
  - Lack of data integrity

Other Problems

- Limited Data Sharing
  - Each application has its own private files and has little opportunity to share data outside the application
  - Lack of coordination and central control
  - Non-standard file formats
- Lengthy Development Times
  - Programmers start from scratch by design their own file formats
- Excessive Program Maintenance
  - 80% of information systems budget

Solution: The DATABASE Approach



- ***Database Approach***
  - Central repository of shared data
  - Stored in a standardized, convenient form
  - Data is managed by a controlling agent
- It requires a Database Management System (DBMS)

Advantages of the Database Approach

- Program-Data Independence
- Planned Data Redundancy
- Improved Data Consistency
- Improved Data Sharing
- Increased Productivity of Application Development
- Enforcement of Standards
- Improved Data Quality
- Improved Data Accessibility and Responsiveness
- Reduced Program Maintenance
- Improved Decision Support

Program-Data Independence

- The database approach provides independence between programs, data and data definition
- **Data independence** implies that changes in data definitions have minimal to no impact on the applications
  - **Program-data independence.**
    - Applications separated from how data is structured and stored.
    - Allow an organization’s data to change and evolve without changing the application program that process the data

☝ *One of the most important benefits of using a DBMS!*

Program-Data Independence (cont.)

- **Physical data independence** implies that neither the applications, nor the logical data model must be changed when changes are made to the physical data storage specifications
- **Logical data independence** implies that software applications are minimally affected by changes in the conceptual or logical data model (e.g., table structure)

Data Sharing and Consistency

- Improved data sharing
  - By controlling data redundancy, reduce the opportunities for data inconsistency.
  - *Syntactical rules*, *Sementical rules*, *and Data integrity rules* can be explicitly defined to enforce the correctness of the data.
  - Processing and Sharing by a set of concurrent users and application programs – yet, keeping all data valid and consistent

Concurrency Control

- DBMS has built-in facilities to support concurrent or parallel execution of database programs
- Allowing a set of *concurrent users* to retrieve from and to update the database.
  - Read/write operations can be executed at the same time by the DBMS
- DBMS must support **ACID (Atomicity, Consistency, Isolation, Durability)** properties
- A key concept is a database *transaction*.
  - A transaction is a sequence of read/write operations considered to be an atomic unit in the sense that either all operations are executed or none at all

Other Main Characteristics of Database Approach

- Planned data redundancy
  - Good database design integrates separate (and redundant) data files into a single, logical structure.
- Improved data consistency
  - By controlling data redundancy, reduce the opportunities for data inconsistency.
- Improved data quality
  - Database designer can specify integrity constraints that are enforced by the DMBS.

Others of Database Approach (Cont.)

- Improved productivity of application development
  - DBMS provides a number of high-level development tools and high-level languages
- Backup / crash recovery
  - deal with the effect of loss of data due to hardware or network errors, or bugs in system or application software
- Security and access control
  - Data access can be managed via logins
  - Privilege role-base functionality
    - Some users have read access, whilst others have write access to the dat

When Not to Use a DBMS

- Overhead costs using a DBMS
  - High initial investment in hardware, software and training
  - The generality that a DBMS provides for defining and processing data
  - Overhead for providing security, concurrency control, recovery and integrity functions.

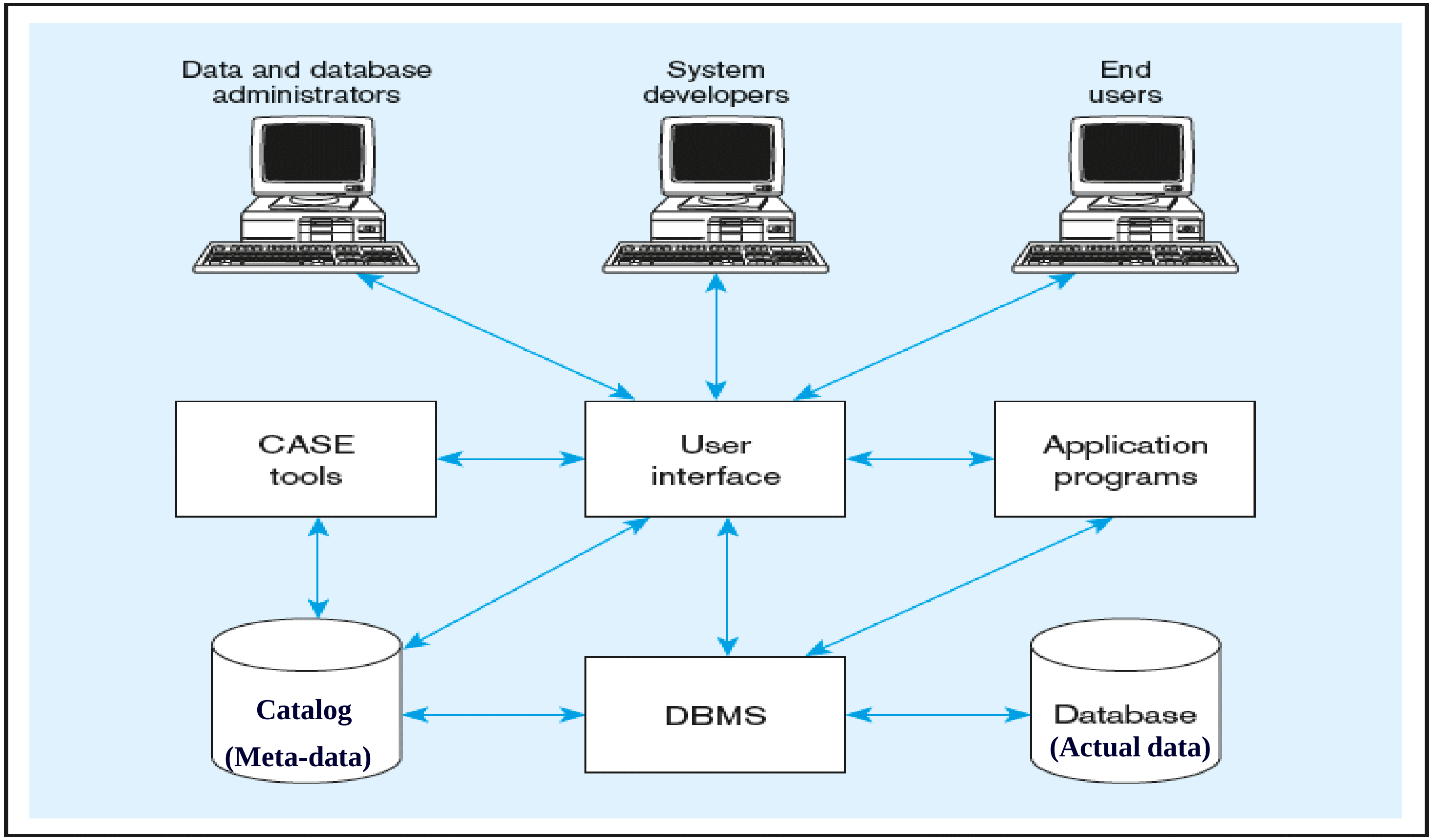
When Not to Use a DBMS (Cont.)

- Cases better to develop customized database applications than using DBMS
  - Simple, well-defined database applications that are not expected to change at all
  - Stringent, real-time requirements for some application programs that may not be met because of DBMS overhead
  - Embedded systems with limited storage capacity, where a general-purpose DBMS would not fit.
  - No multiple-user access to data

Outline

- What is a database?
- Historical root of database systems
- Advantages of database approach
- ☞ **Components of database environments**
- Types of Database Applications
- Data Model
- Database system development life cycle
- Evolution of database technology
- Range of database systems

Components of the Database Environment



The database operational environment is **an integrated system of hardware, software, and people**, designed to facilitate the storage, retrieval, and control of the information resource.

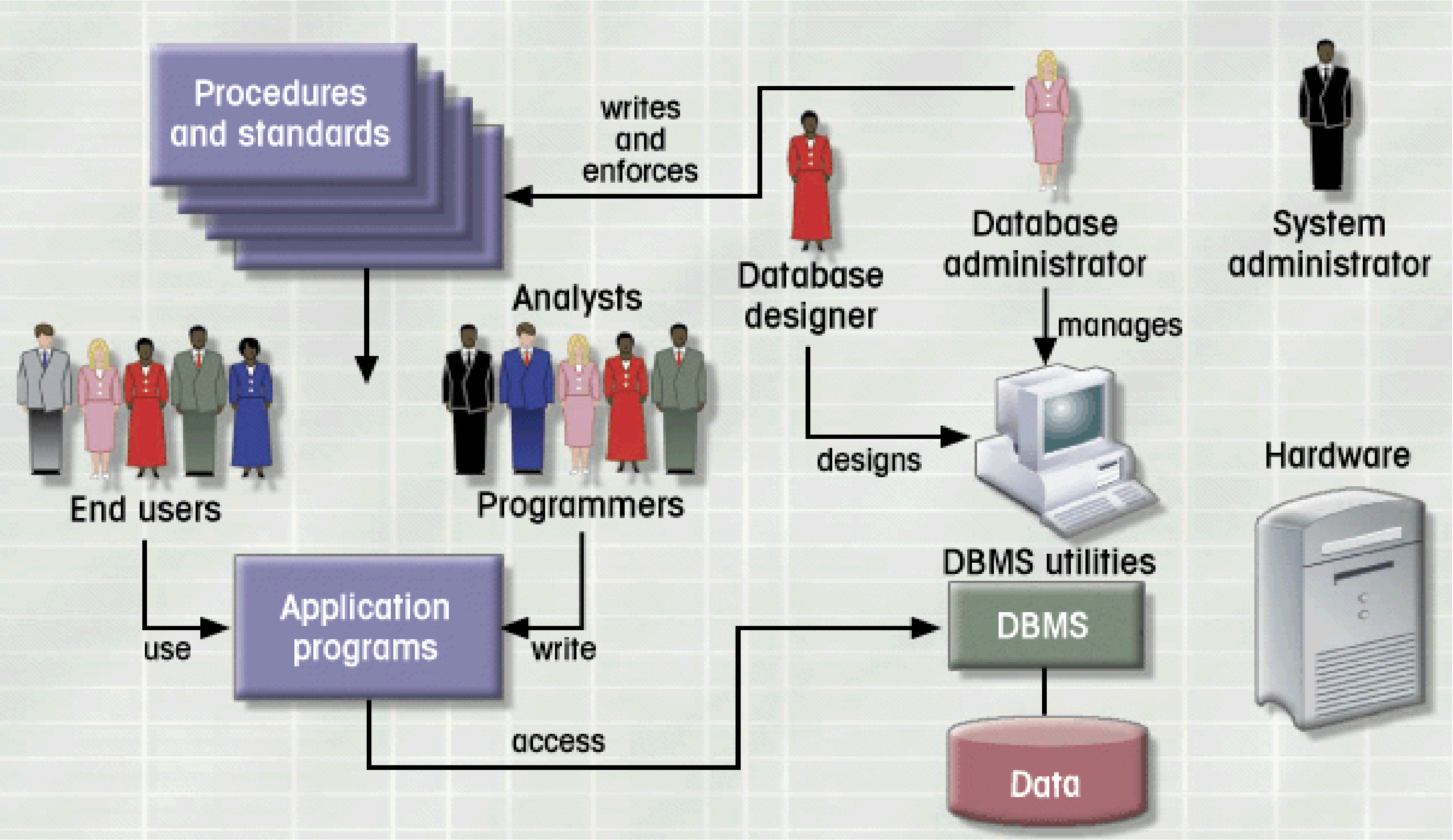
Components of the Database Environment

- Database Management System (DBMS) –software for managing the database
- Databases–DBMS stores two types of data:
  - **Raw data** (Actual Data)
  - **Metadata**: the data definition which are stored in the **catalog** of the DBMS
- Application Programs –software using the data
  - User Interface–text and graphical displays to users
    - E.g., languages, menus and other facilities
- CASE Tools –computer-aided software engineering tool
  - Used to design databases and application programs.
- Database Users

Database Users

- Users may be divided into
  - Those who actually use and control the database content, and those who design, develop and maintain database applications (called “Actors on the Scene”), and
  - Those who design and develop the DBMS software and related tools, and the computer systems operators (called “Workers Behind the Scene”).

Database Users ("Actors on the Scene")



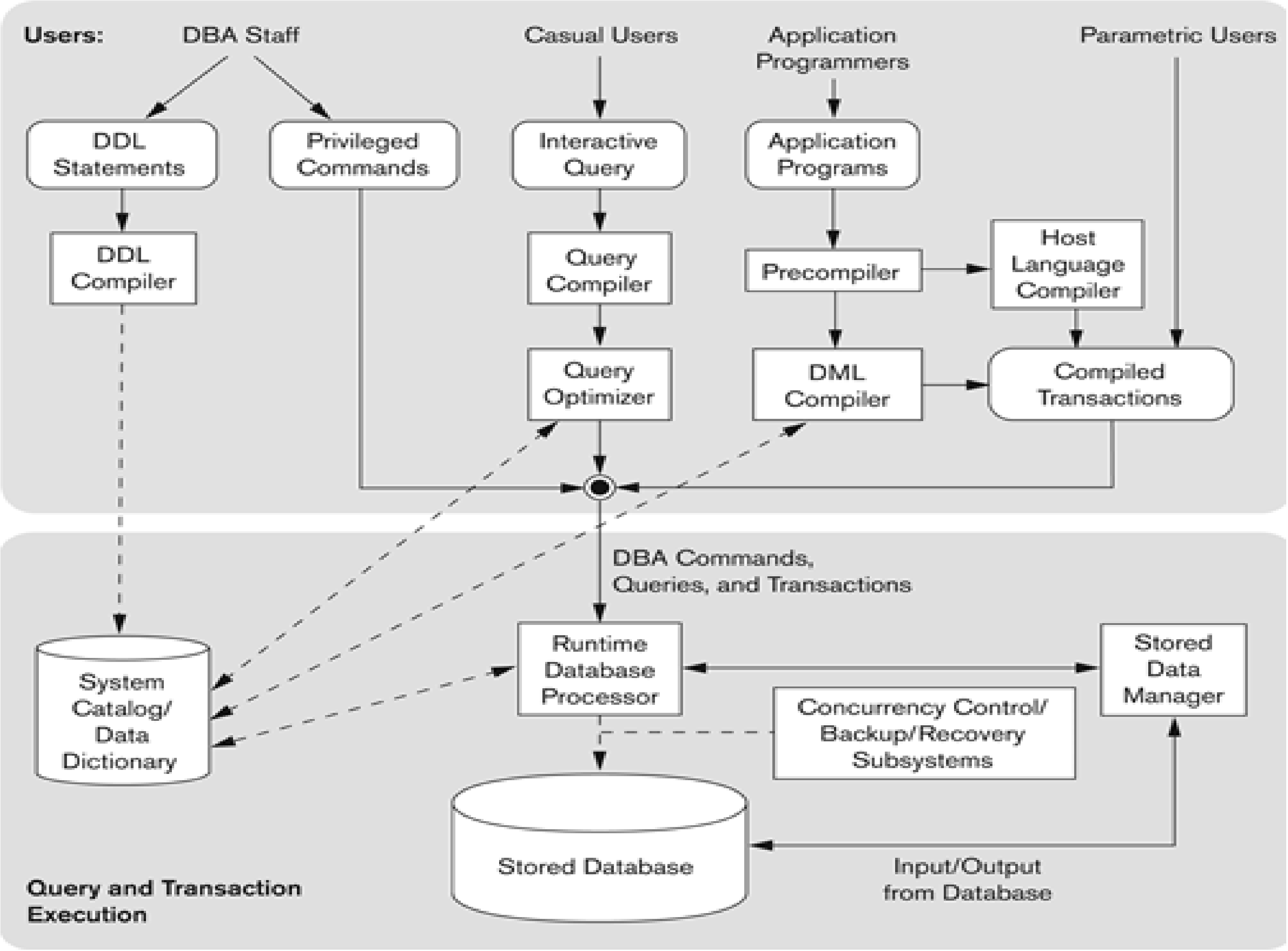
Database Users

- Database administrators (short form DBA):
  - A person responsible for
    the installation, configuration, upgrade, administration, monitoring and maintenance of databases in an organization.
  - Responsible for authorizing access to the database, for coordinating and monitoring its use, acquiring software and hardware resources, controlling its use and monitoring efficiency of operations.

Database Users (Conti)

- System developers
  - Database designers
    - Responsible to define the data content, structure and constraints, and functions or transactions against the database.
    - They must communicate with the end-users and understand their needs.
  - Programmers
    - Responsible to implement database applications
- End users
  - use the data for queries, reports, and some of them update the database content.

Database Systems with Different Database Users



Outline

□ What is a database?

□ Historical root of database systems

□ Advantages of database approach

□ Components of database environments

☞ **Types of Database Applications**

□ Data Model

□ Database system development life cycle

□ Evolution of database technology

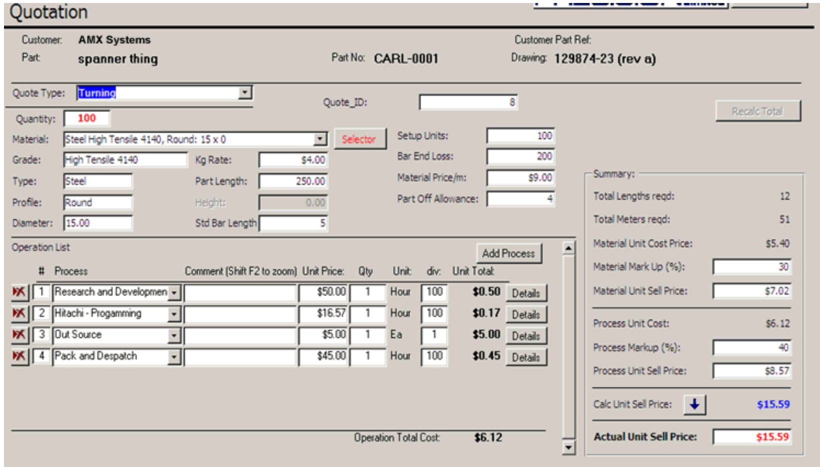
□ Range of database systems

Applications of Database Technology

- Traditional applications for traditional numeric and alphanumeric data (e.g., inventory application)
- Multimedia applications (e.g., YouTube, Spotify)
- Biometric applications (e.g., fingerprints, retina scans)
- Wearable applications (e.g., Fitbit, Apple Watch)
- Geographical Information Systems (GIS) applications (e.g., Google Maps)
- Finance applications with volatile data (e.g., investment banks)
- Sensor applications (e.g., environment monitoring, automatic shutdown system)
- Big Data applications (e.g., a retailer’s point-of-sale database system)

Traditional Applications

- A **database application** is a computer program whose primary purpose is entering and retrieving information from a computerized database.
- **Traditional Applications:**
  - A part of an information system within any organization
  - Banking transaction, purchases from supermarket, purchases using credit card, booking a holiday at travel agents, using local library, etc.
    - Numeric and character databases



Quotation

Customer: AMX Systems
Part: spanner thing
Part No: CARL-0001
Customer Part Ref:
Drawing: 129874-23 (rev a)

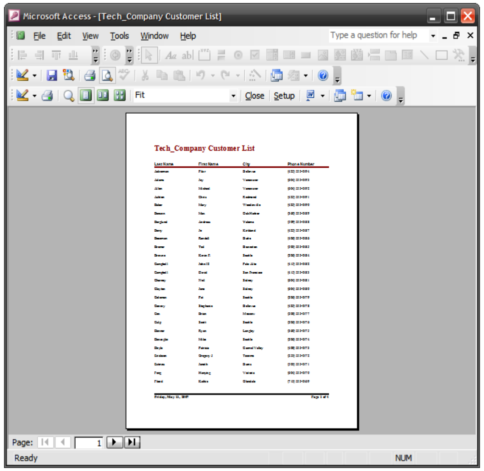
Quote Type: Turning
Quote_ID: 8
Quantity: 100
Material: Steel High Tensile 4140, Round: 15 x 0
Grade: High Tensile 4140
Type: Steel
Profile: Round
Diameter: 15.00
Kg Rate: $4.00
Part Length: 250.00
Height: 0.00
Std Bar Length 5
Setup Units: 100
Bar End Loss: 200
Material Price/m: $9.00
Part Off Allowance: 4

Operation List

#	Process	Comment (Shift F2 to zoom)	Unit Price:	Qty	Unit:	div:	Unit Total:
1	Research and Developmen		$50.00	1	Hour	100	$0.50
2	Hitachi - Progamming		$16.57	1	Hour	100	$0.17
3	Out Source		$5.00	1	Ea	1	$5.00
4	Pack and Despatch		$45.00	1	Hour	100	$0.45

Operation Total Cost: $6.12

Summary:
Total Lengths reqd: 12
Total Meters reqd: 51
Material Unit Cost Price: $5.40
Material Mark Up (%): 30
Material Unit Sell Price: $7.02
Process Unit Cost: $6.12
Process Markup (%): 40
Process Unit Sell Price: $8.57
Calc Unit Sell Price: $15.59
Actual Unit Sell Price: $15.59



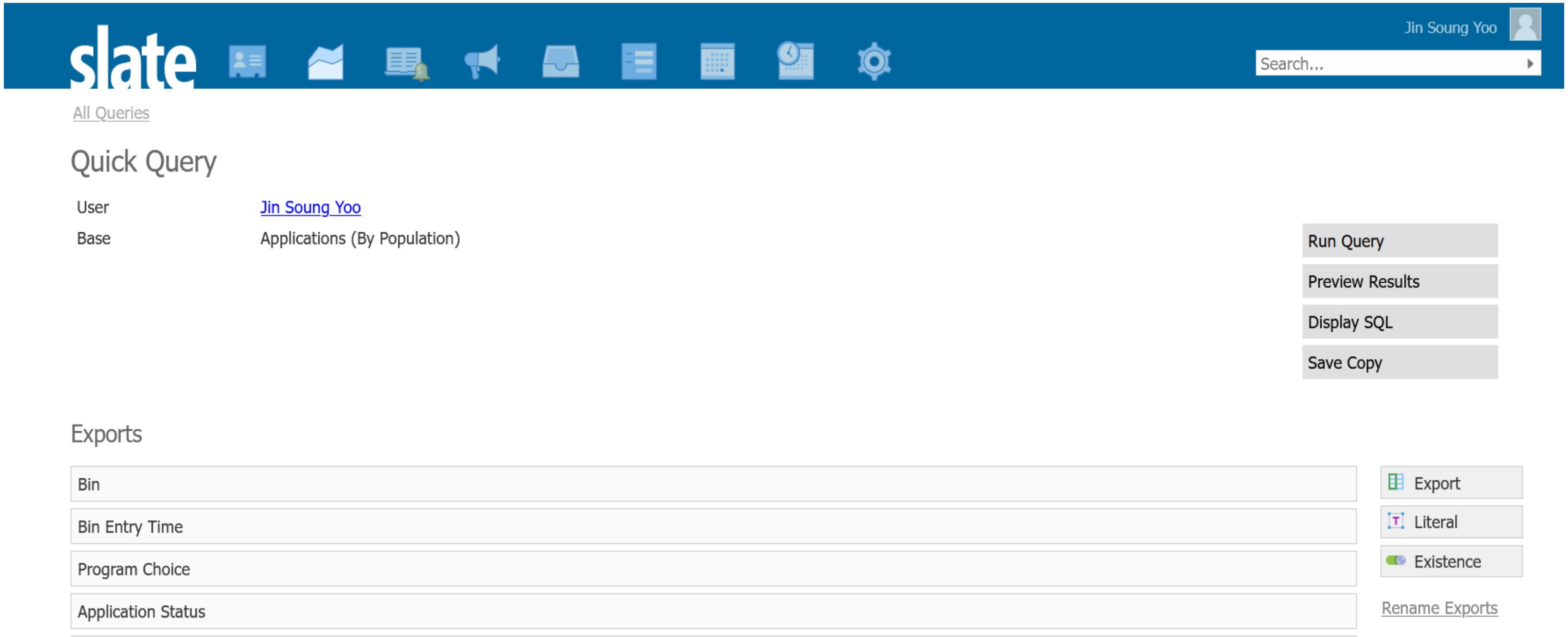
Microsoft Access - [Tech_Company Customer List]

Tech_Company Customer List

Last Name	First Name	City	Phone Number

Web-based Applications

- Web-based applications for transaction processing



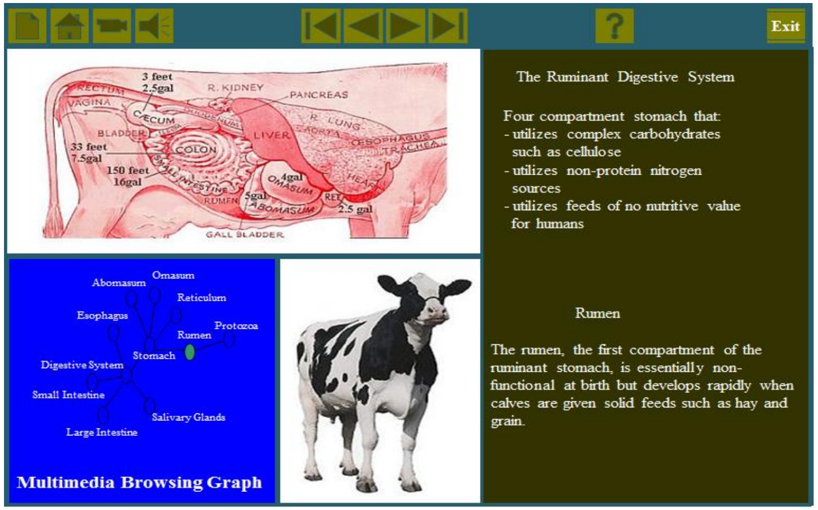
Quick Query

User	Jin Soung Yoo
Base	Applications (By Population)

Exports

- Bin
- Bin Entry Time
- Program Choice
- Application Status

Multimedia Databases

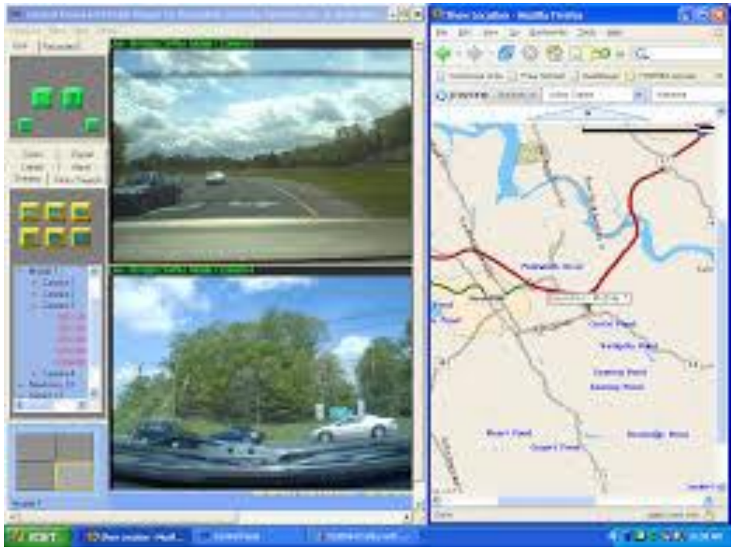


The Ruminant Digestive System

Four compartment stomach that:
- utilizes complex carbohydrates such as cellulose
- utilizes non-protein nitrogen sources
- utilizes feeds of no nutritive value for humans

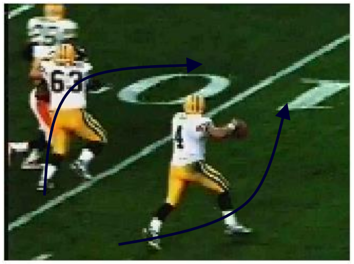
Rumen

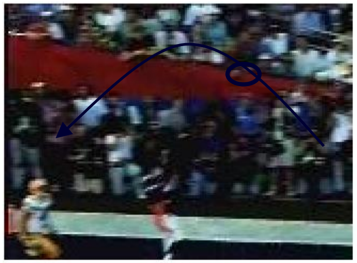
The rumen, the first compartment of the ruminant stomach, is essentially non-functional at birth but develops rapidly when calves are given solid feeds such as hay and grain.



Vehicle Surveillance System

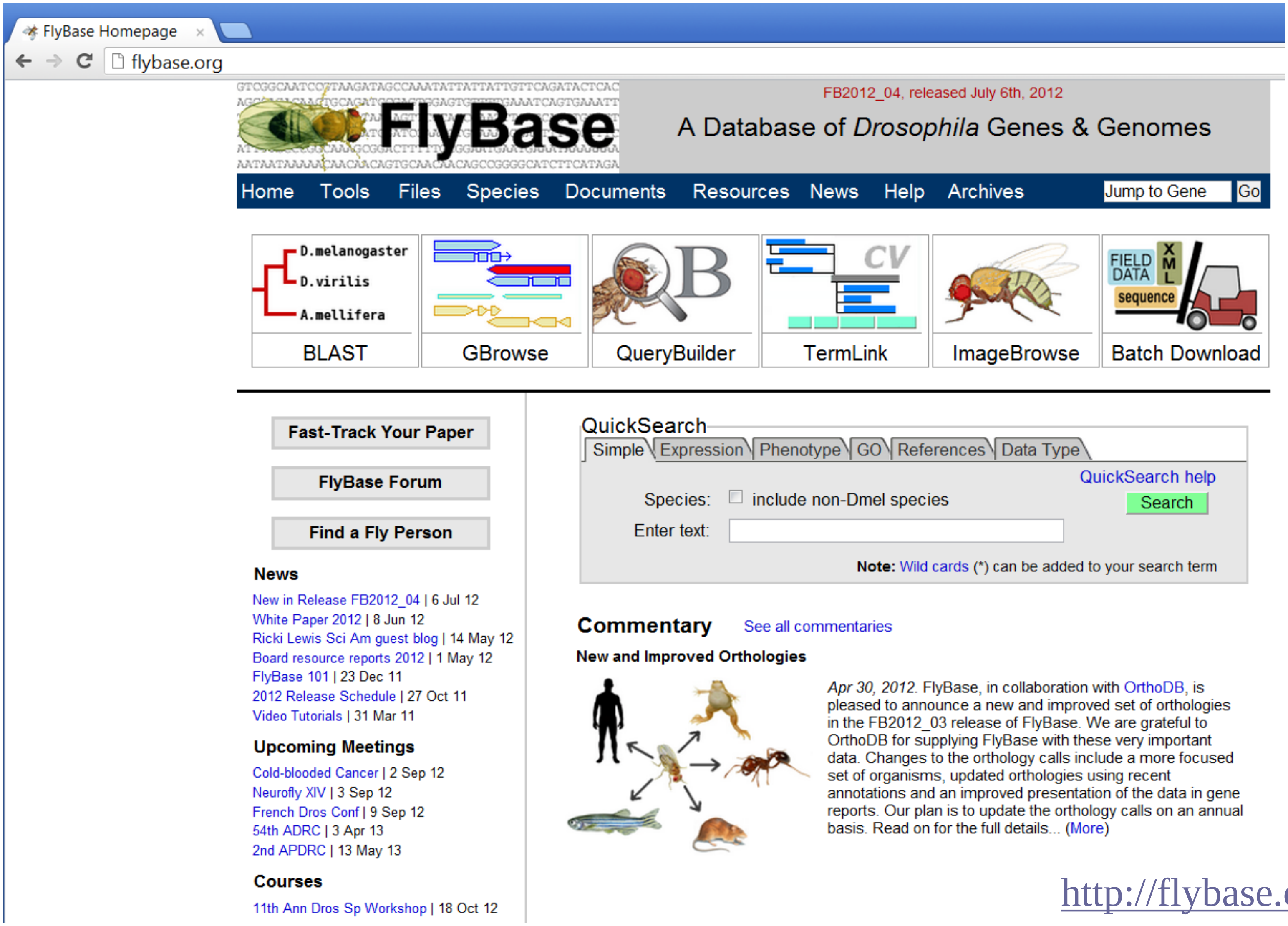
A **multimedia database (MMDB)** is a collection of related multimedia. The multimedia database includes one or more primary media data types such as images, graphic objects, animation(image) sequences, audio and video.





* Source: Multimedia Database Lab, Purdue University

Application: Database for Life Science



FB2012_04, released July 6th, 2012

A Database of *Drosophila* Genes & Genomes

News

New in Release FB2012_04 | 6 Jul 12
White Paper 2012 | 8 Jun 12
Ricki Lewis Sci Am guest blog | 14 May 12
Board resource reports 2012 | 1 May 12
FlyBase 101 | 23 Dec 11
2012 Release Schedule | 27 Oct 11
Video Tutorials | 31 Mar 11

Upcoming Meetings

Cold-blooded Cancer | 2 Sep 12
Neurofly XIV | 3 Sep 12
French Dros Conf | 9 Sep 12
54th ADRC | 3 Apr 13
2nd APDRC | 13 May 13

Courses

11th Ann Dros Sp Workshop | 18 Oct 12

QuickSearch

QuickSearch help

Species: □ include non-Dmel species

Enter text:

Note: Wild cards (*) can be added to your search term

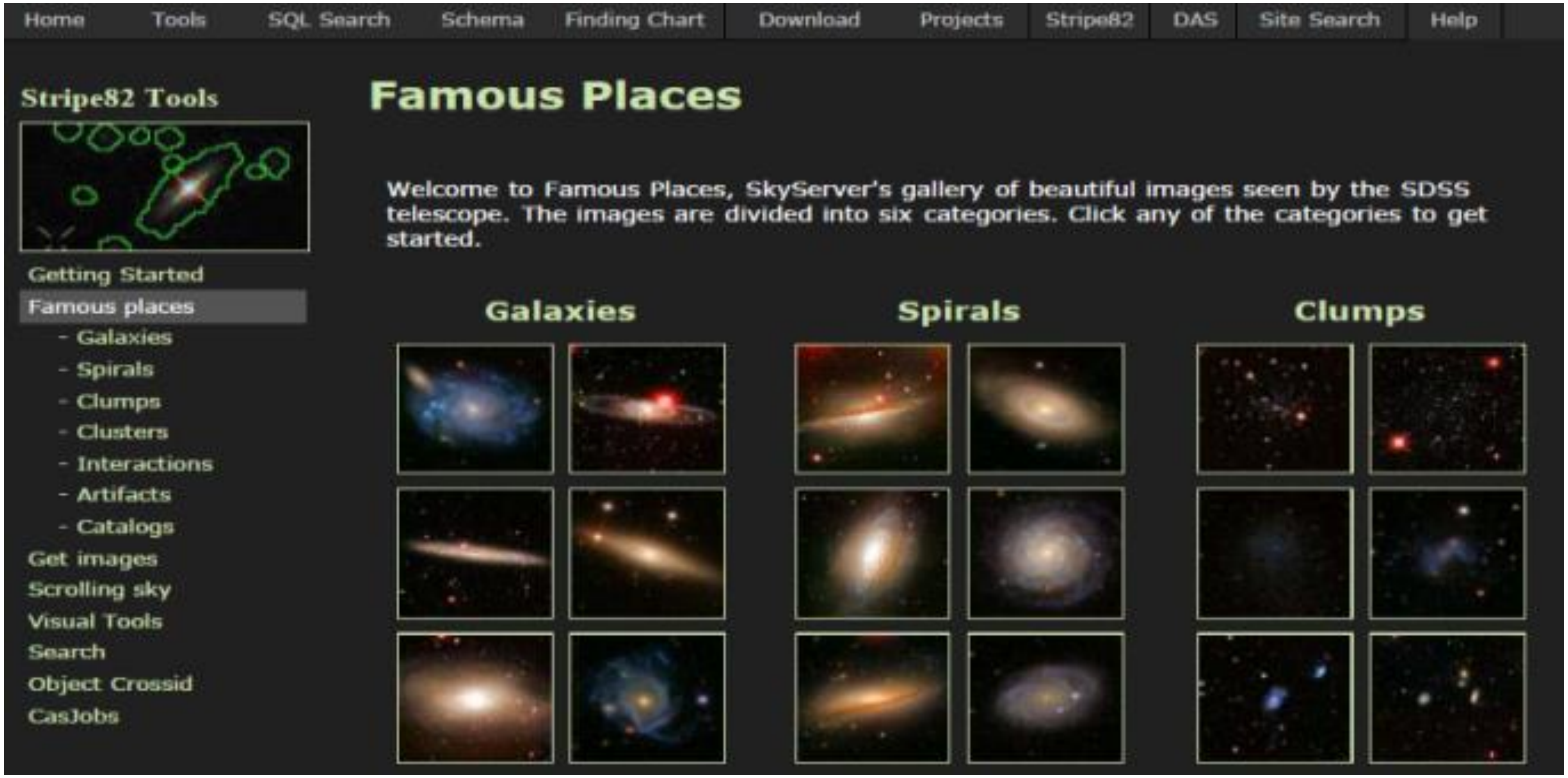
Commentary See all commentaries

New and Improved Orthologies

Apr 30, 2012. FlyBase, in collaboration with OrthoDB, is pleased to announce a new and improved set of orthologies in the FB2012_03 release of FlyBase. We are grateful to OrthoDB for supplying FlyBase with these very important data. Changes to the orthology calls include a more focused set of organisms, updated orthologies using recent annotations and an improved presentation of the data in gene reports. Our plan is to update the orthology calls on an annual basis. Read on for the full details... (More)

http://flybase.org/

Application: Sloan Digital Sky Survey



Famous Places

Welcome to Famous Places, SkyServer's gallery of beautiful images seen by the SDSS telescope. The images are divided into six categories. Click any of the categories to get started.

Go to SkyServier : http://skyserver.sdss.org/dr12/en/home.aspx

* Source: Sloan Digital Sky Survey

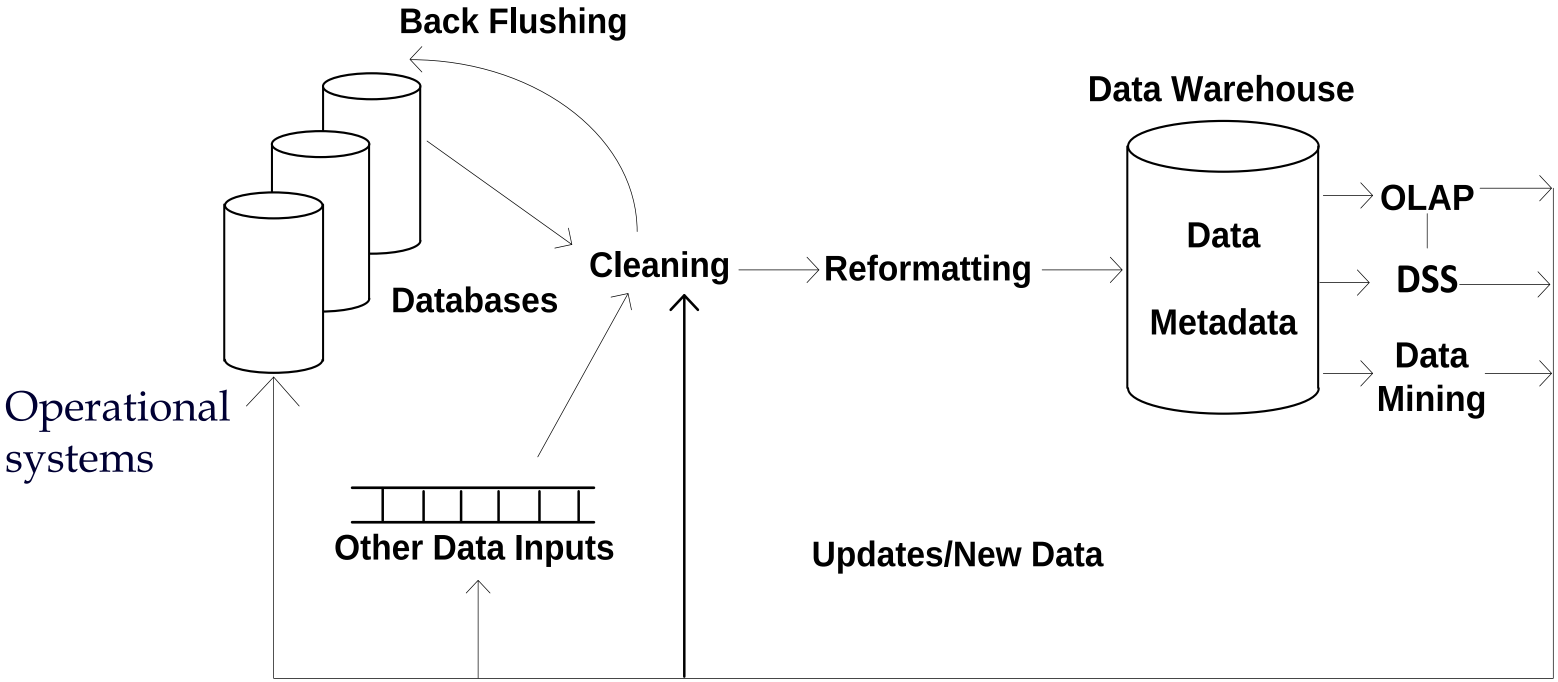
Database schema:

SQL query search: http://skyserver.sdss.org/dr12/en/tools/search/sql.aspx

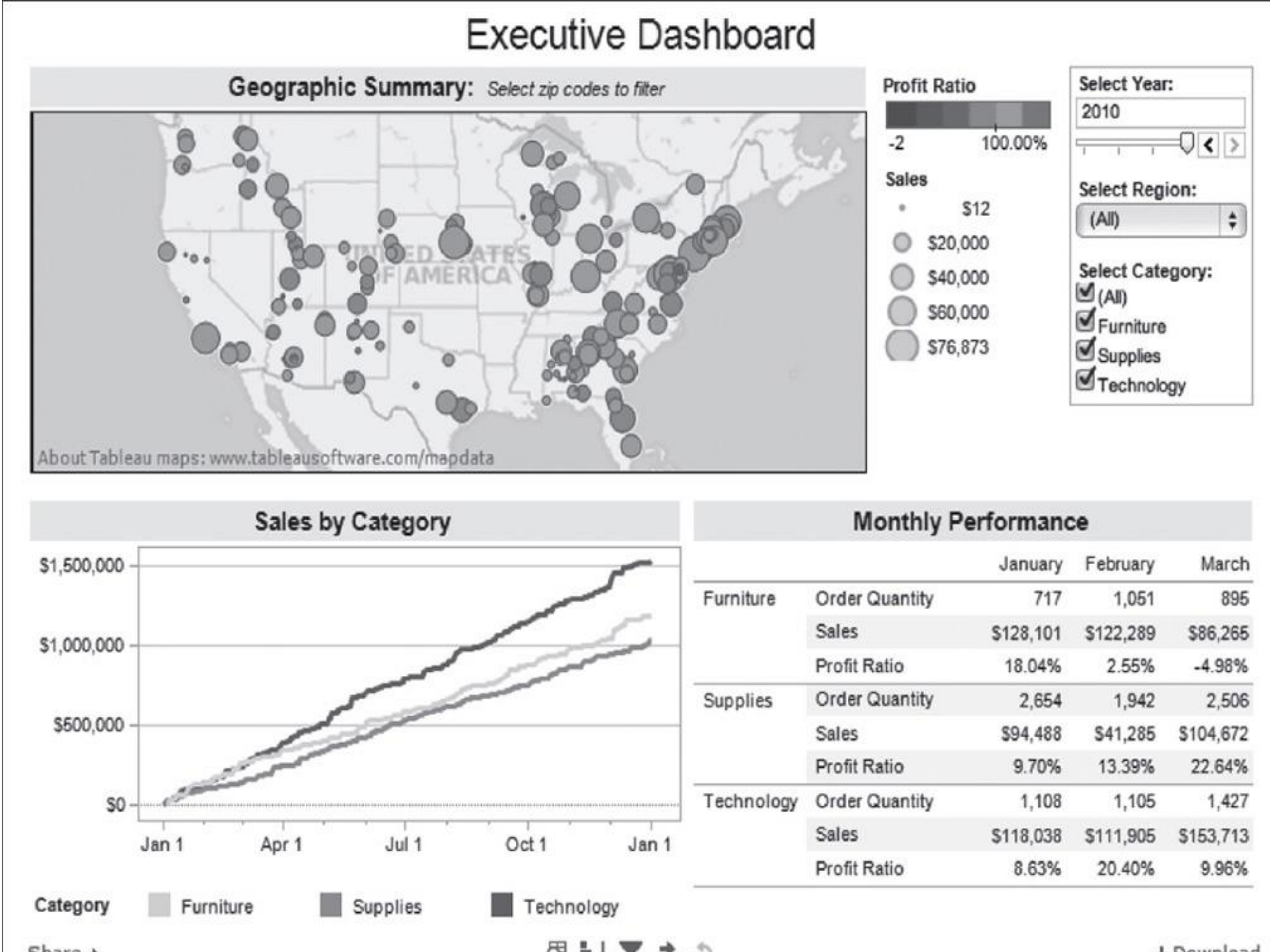
Data Warehouse

- Data Warehouse is a system used for reporting and data analysis.
  - Integrate data from one or more disparate sources, and create a central repository of data, a data warehouse.
  - The data is often extracted and uploaded from the operational systems (such as marketing, sales, etc.)
  - Data warehouses store current and historical data, and are used for senior management reporting to help business decision support.

Conceptual Structure of Data Warehousing



- Data are extracted from the various data source systems.
- The data from the various source systems are transformed and integrated
- And then the integrated data is loaded into the data warehouse.
- Data warehouse data is used for OLAP, DSS, etc.
- The results may be fed back to operational system.



Executive Dashboard

Geographic Summary: *Select zip codes to filter*

Select Year:
2010

Select Region:
(All)

Select Category:
☑ (All)
☑ Furniture
☑ Supplies
☑ Technology

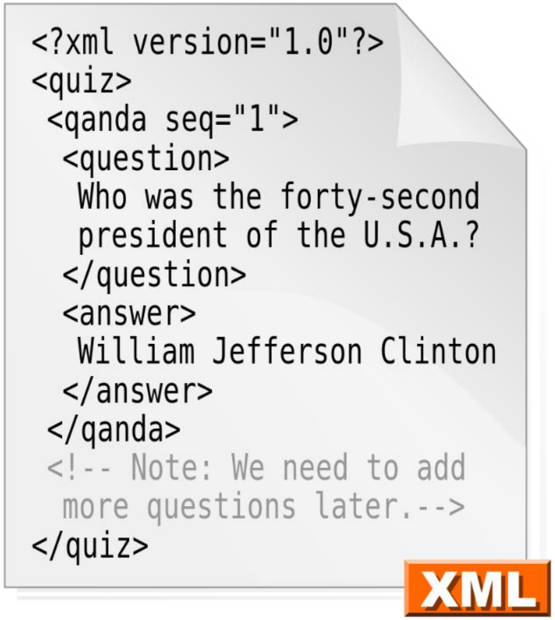
Sales by Category

Monthly Performance

		January	February	March
Furniture	Order Quantity	717	1,051	895
	Sales	$128,101	$122,289	$86,265
	Profit Ratio	18.04%	2.55%	-4.98%
Supplies	Order Quantity	2,654	1,942	2,506
	Sales	$94,488	$41,285	$104,672
	Profit Ratio	9.70%	13.39%	22.64%
Technology	Order Quantity	1,108	1,105	1,427
	Sales	$118,038	$111,905	$153,713
	Profit Ratio	8.63%	20.40%	9.96%

We as Database and XML Database

- Most internet users gain access to the web using search engines
- The data is a best “*semi-structured*”.
  - A form of structured data that does not obey the formal structure of data models associated with relational databases or other forms of data tables.
  - But contains tags or other markers to separate semantic elements and enforce hierarchies of records and fields within the data.



```xml
<?xml version="1.0"?>
<quiz>
 <qanda seq="1">
  <question>
   Who was the forty-second
   president of the U.S.A.?
  </question>
  <answer>
   William Jefferson Clinton
  </answer>
 </qanda>
 <!-- Note: We need to add
  more questions later.-->
</quiz>
```

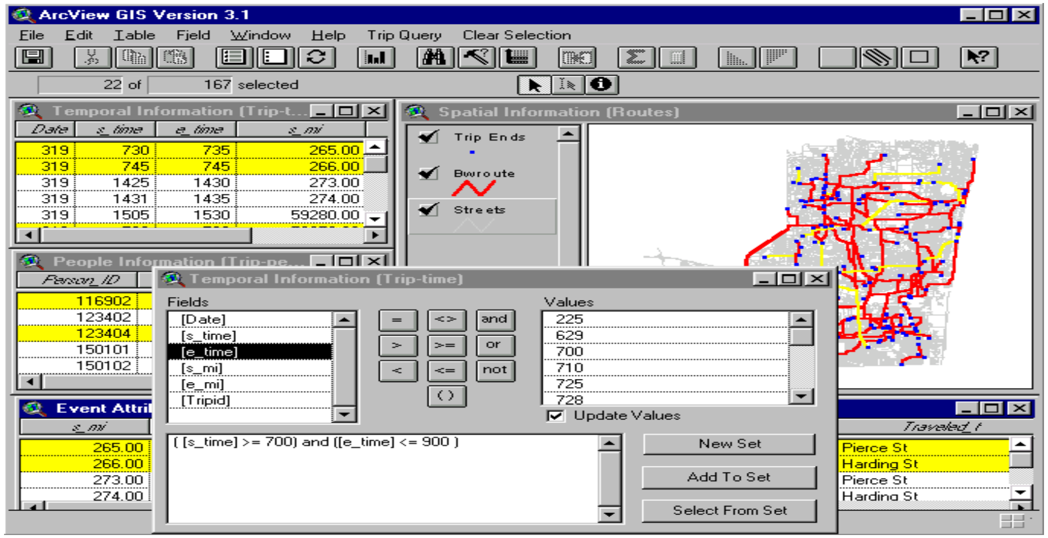
- XML database and application
  - http://students.mimuw.edu.pl/~pd291528/zbd/materialy/Xml%20Databases.pdf
  - https://www.marklogic.com/blog/importance-xml-database/

Data Stream Management

- □ Data is generated **continuously, in real time**, and in fixed order
- □ Data sources:
  - Sensor network data – weather monitoring, road traffic monitoring, motion detection
  - Internet (Web) data – financial trading, news/sports tickers
  - Network traffic (IP packet headers) – bandwidth usage, routing decisions, security
  - Transaction log data - telecom, point-of-sale purchases
- □ Collect and process the data on-line and use for applications such as:
  - Environment monitoring
  - Location monitoring
  - Correlations across stock prices
  - Denial-of-service attack detection
  - Up-to-date answers generated continuously or periodically
- □ Data Stream Management Systems (DSMS) : Aurora, IBM Streams, Hortonworks' Data Flow, NIAGARA Query Engines, STREAM, etc.

Geospatial Databases

- A **spatial database**, or **geodatabase** is a general-purpose database that has been enhanced to include objects defined in a geometric space, along with tools for querying and analyzing such data.
  - Simple geometric objects: points, lines and polygons
  - Complex geometric objects: 3D objects, topological coverages, linear networks, triangulated irregular networks.



Date	s_time	e_time	s_mi
319	730	735	265.00
319	745	745	266.00
319	1425	1430	273.00
319	1431	1435	274.00
319	1505	1530	59280.00

Person_ID
116902
123402
123404
150101
150102

s_mi
265.00
266.00
273.00
274.00

Traveled_
Pierce St
Harding St
Pierce St
Harding St

Fields: [Date], [s_time], [e_time], [s_mi], [e_mi], [Tripid]

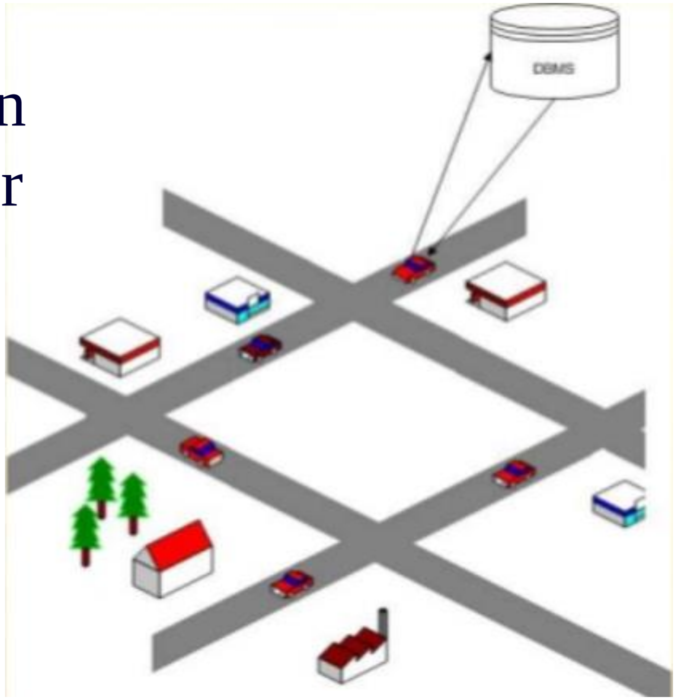
Values: 225, 629, 700, 710, 725, 728

☑ Update Values

```
( [s_time] >= 700) and ([e_time] <= 900 )
```

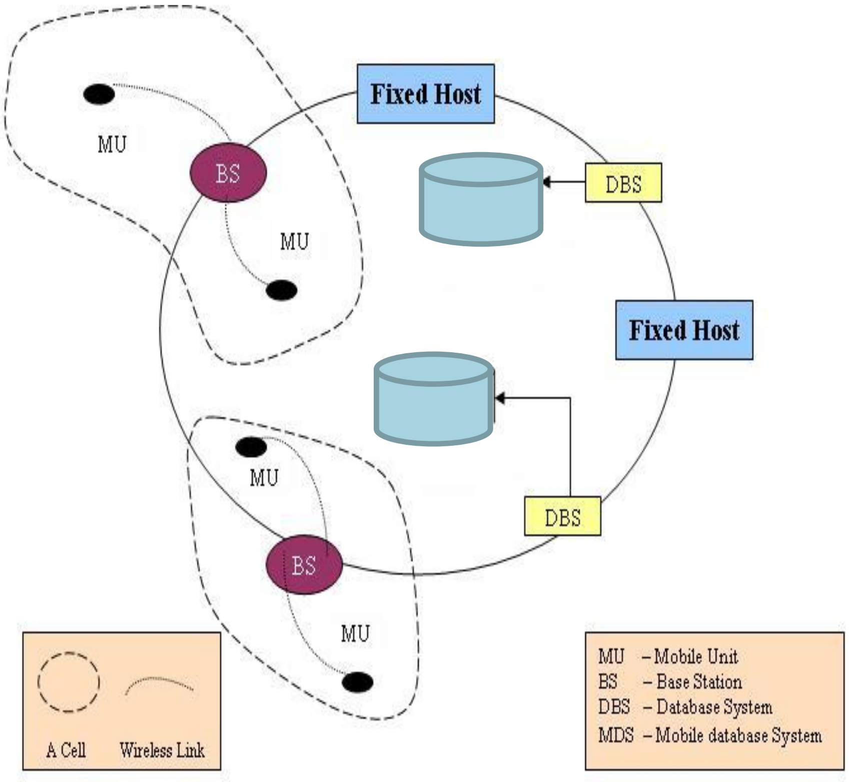
Moving Object Databases

- Database which can represent the ***moving object*** and allow to query such objects and movements.
- Moving objects could be cars, trucks, aircraft, ships, polar bears, or users with mobile phone, position-aware devices such as personal digital assistants
- There are also moving entities with an extent, called as ***moving regions*** - for example, hurricanes, forest fires, oil splits, armies, epidemic diseases, and so on.
- Moving objects are time-dependent geometries. Supported by ***Spatio-temporal databases.***



Mobile Databases

- A mobile database is either a stationary database that can be connected to by a mobile computing devices (such as smart phones) over a mobile network or a database which is actually carried by the mobile device.
- The database could include a list of contacts, price information, distance travelled, or any other information such as your contacts and calendar



DBMS Interfaces

- Keyword-based Database Search
- Menu-based Interface for Web Clients or Browsing
- Apps for Mobile Devices
  - present mobile users with access to their data
- Forms-based Interfaces
  - Users can fill out all of the form entries to insert new data
- Graphical User Interfaces
  - A GUI display a schema to the user
- Natural Language Interfaces
  - accept requests written in English and attempt to understand them.
- Speech Input and Output
- Interface for Parametric Users (e.g., bank tellers which perform a small set of operations repeatedly)
- Interface for the DBA

Outline

- □ What is a database?
- □ Historical root of database systems
- □ Advantages of database approach
- □ Components of database environments
- □ Types of Database Applications
- ☞ **Data Model**
- □ Database system development life cycle
- □ Evolution of database technology
- □ Range of database systems

Database Model vs. Instances

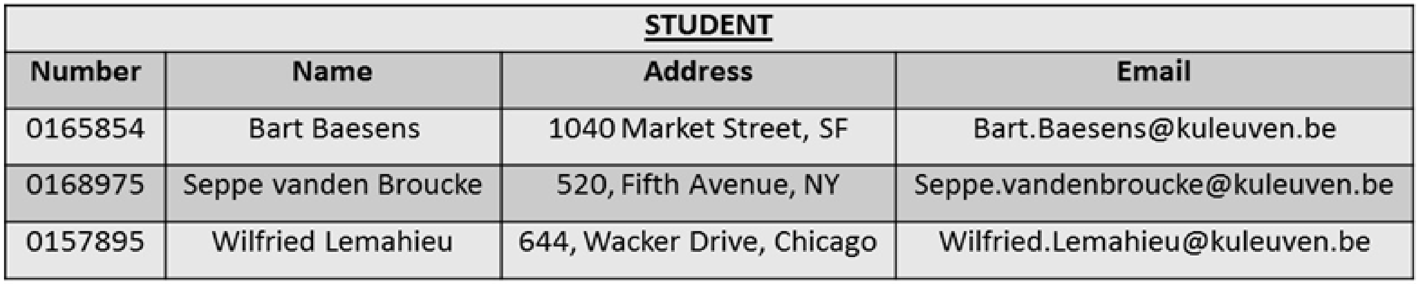
- **Database model** or **database schema** provides the description of the database data at different levels of detail and specifies the various data items, their characteristics and relationships, constraints, storage details, etc.
  - The database model is specified during database design and not expected to change too frequently
  - It is stored in the catalog of the DBMS.
- **Database state** represents the data in the database at a particular moment (also called the *current set of instances*)
  - The database state is typically changes on an ongoing basis depending on data manipulation such as adding, updating or removing data

Examples

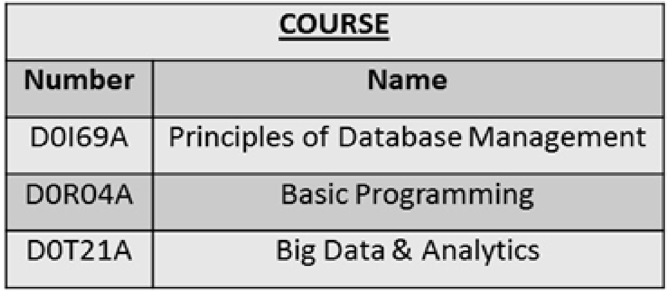
- □ Database model

  Student (number, name, address, email)
  Course (number, name)
  Building (number, address)

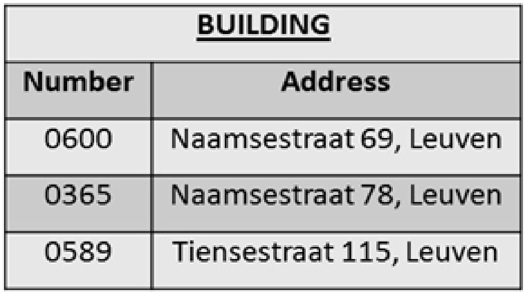
- □ Database state



STUDENT			
Number	Name	Address	Email
0165854	Bart Baesens	1040 Market Street, SF	Bart.Baesens@kuleuven.be
0168975	Seppe vanden Broucke	520, Fifth Avenue, NY	Seppe.vandenbroucke@kuleuven.be
0157895	Wilfried Lemahieu	644, Wacker Drive, Chicago	Wilfried.Lemahieu@kuleuven.be



COURSE	
Number	Name
D0I69A	Principles of Database Management
D0R04A	Basic Programming
D0T21A	Big Data & Analytics



BUILDING	
Number	Address
0600	Naamsestraat 69, Leuven
0365	Naamsestraat 78, Leuven
0589	Tiensestraat 115, Leuven

Data Model

- A **database model** is comprised of different data models, each describing the data from different perspectives
- A good data model is the start of every successful database application.
- A good data model provides a clear and unambiguous description of the data items, their relationships and various data constraints from a particular perspective.

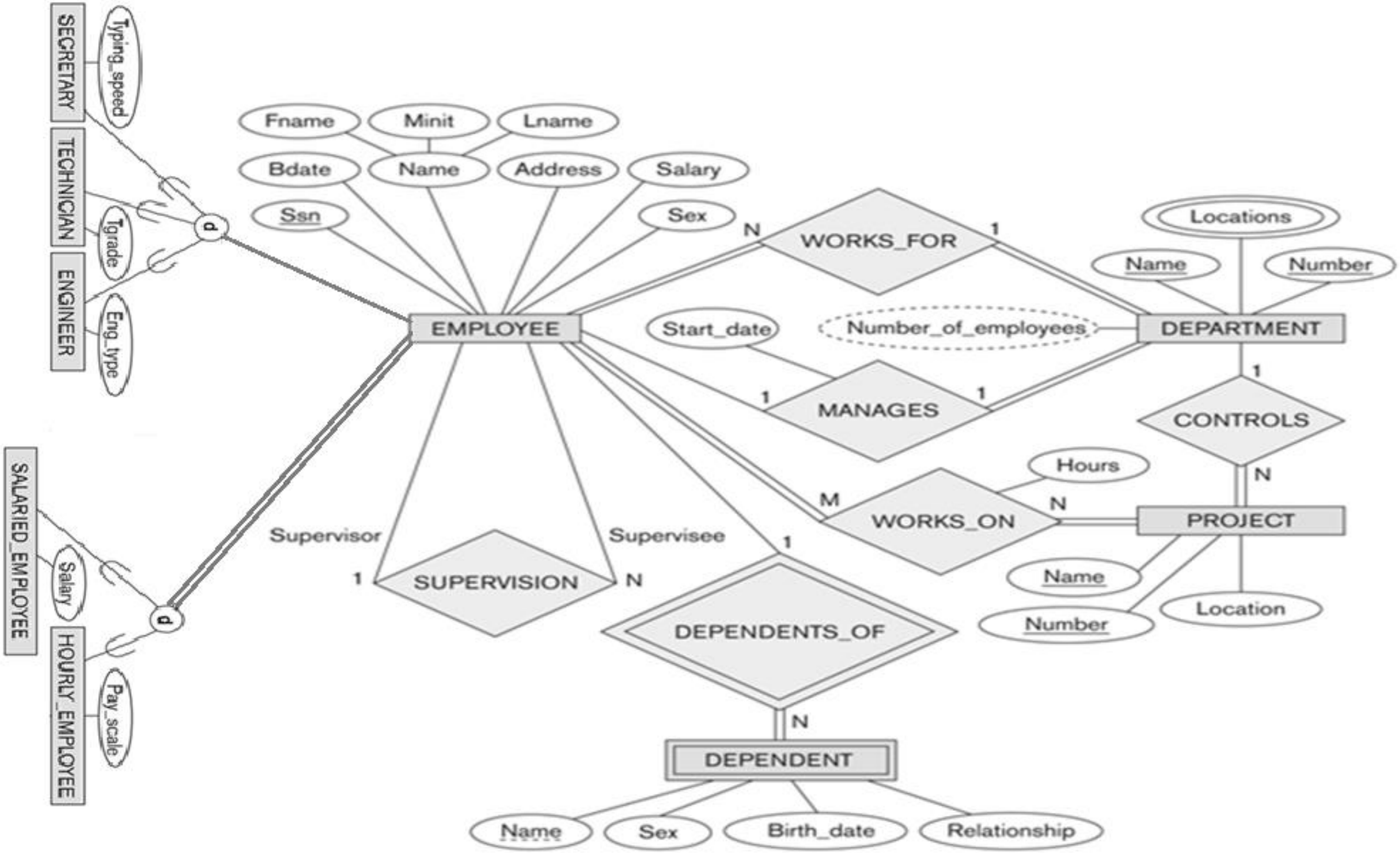
Types of Data Models

- Four types of data models
  - External Data Model
  - Conceptual data model
  - Logical data model
  - Internal data model (Physical data model)
- Three types of data models
  - External data model
  - Conceptual data model
  - Internal data model

Conceptual Data Model

- A **conceptual data model** provides a high-level description of the data items with their characteristics and relationships
- It is a communication instrument between information architect and business user
- It should be **implementation-independent**, user-friendly, and close to how the business user perceives the data
- It will usually be represented using an Enhanced-Entity Relationship (EER) model, or an object-oriented model

Example of EER Model



Internal Data Model

- Internal Data Model

  : View of data that involved data management technology.

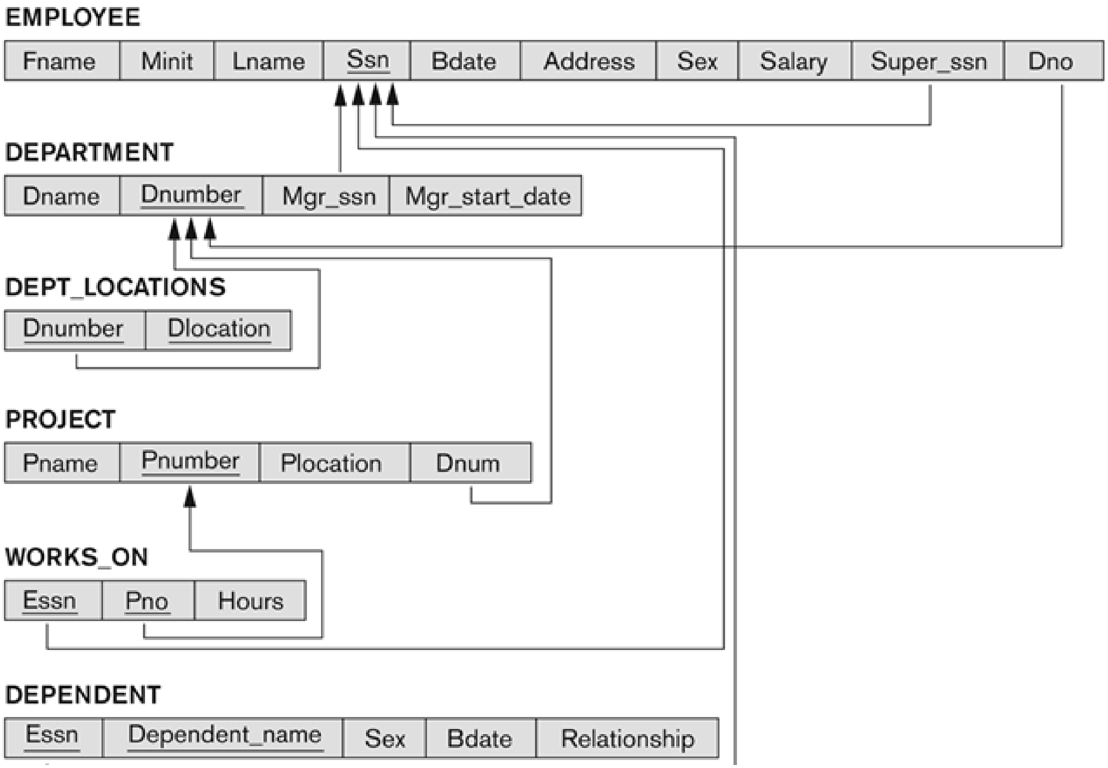
  - **Logical Data Model**
  - **Physical Data Model** (also called Internal Data Model)

Logical Data Model

- A **logical data model** is a translation or mapping of the conceptual data model towards a specific implementation environment
  - The logical data items may still be understood by business users, but are not too far removed from the physical data organization
- Depending on the development environment, it can be a hierarchical, relational, object-oriented, extended relational, XML or NoSQL model
- The logical data model can be mapped to an internal data model.

Logical Data Model (Relational DB Schema)

If relational database technology will be used, the conceptual data model will be transformed and represented using elements of the relational model such as tables, columns, rows, primary keys, foreign keys, and other constraints



EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date

DEPT_LOCATIONS

Dnumber	Dlocation

PROJECT

Pname	Pnumber	Plocation	Dnum

WORKS_ON

Essn	Pno	Hours

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship

Often, include data type, length and other domain constraints, e.g., DEPENDENT

Essn : Number(10)	Dependent_name : Varchar2(30)	Sex : Char(1)	Bdate: Date	Relationship : Varcahr2(20)

Physical Data Model (Internal Data Model)

- A **physical data model (**also called internal data model**)** represents the data's physical storage details.
- It clearly describes which data are stored where, in what format, which indexes are provided to speed up retrieval, etc.
- It is highly DBMS-specific.

External Data Model

- An **external data model** contains various subsets of the data items in the logical model, also called ***views***, tailored towards the needs of specific applications or groups of users

Three Layer Architecture for Database Models

- The **three-layer database architecture** is an essential element of every database application and describes how the different underlying data models are related.
  - **External Layer**
  - **Conceptual/Logical Layer**
  - **Internal Layer**
- Ideally, changes in one layer should have no to minim impact on the others.
- The three-layer architecture has advantages in efficiency, maintenance, performance, security, etc.

Three Layer Architecture

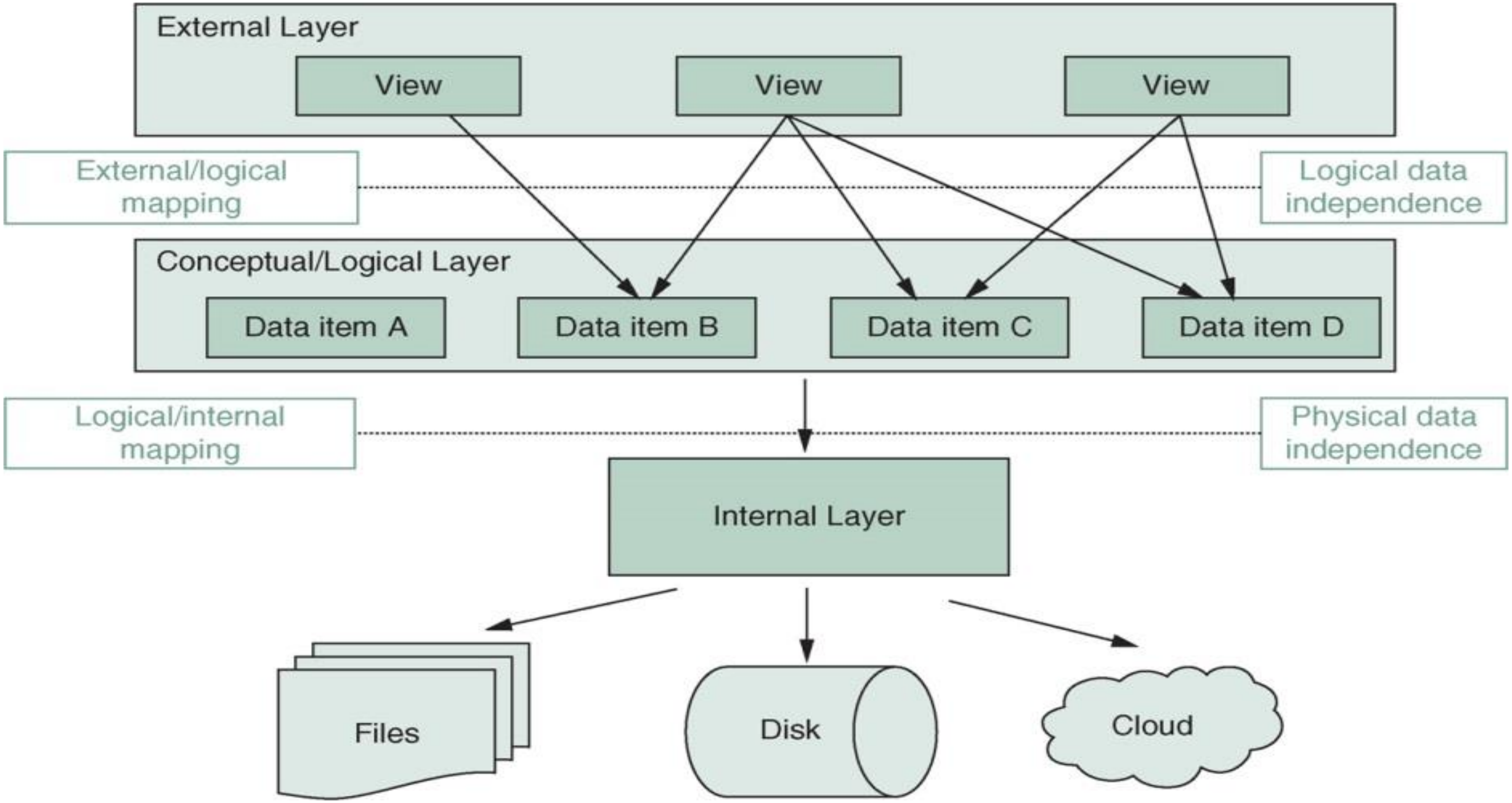


Figure. The Three-layer database architecture

Conceptual/Logical Layer

- The **conceptual/logical layer** has the conceptual and logical data models which (both) focus on the data items, their characteristics, and relationships without too much about the actual physical DBMS implementation.
- The **conceptual data model** should be a user-friendly, implementation-independent, and transparent data model, constructed in close collaboration between the information architect and business user(s).
- The conceptual data model will be refined to a **logical data model b**ased on the implementation environment

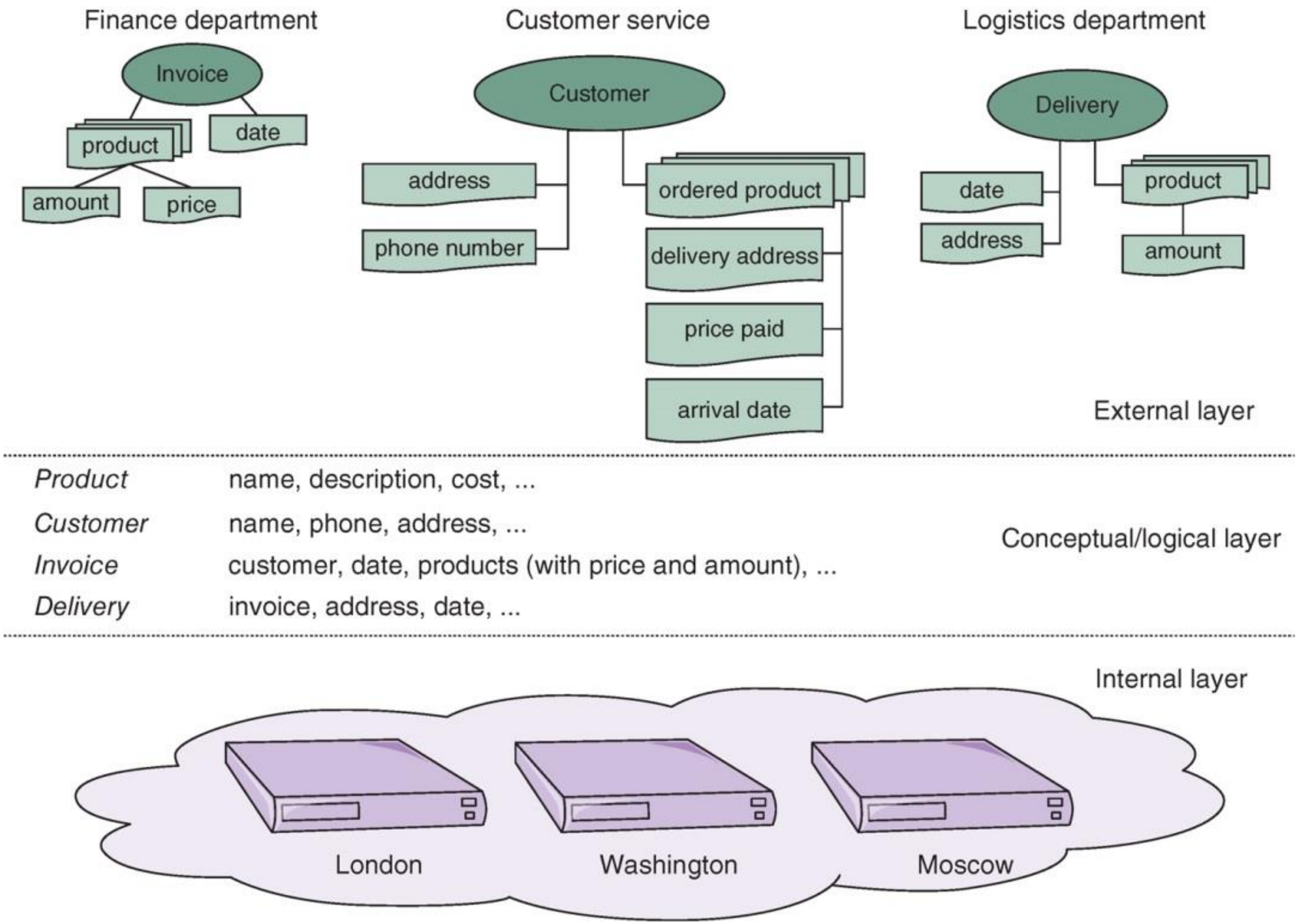
External Layer

- An **external data model** includes views offering a window on a carefully selected part of the logical data model.
- A **view** describes the part of the database that a particular application or user group is interested in, hiding the rest of the database.
  - The views will be tailored to the data needs of an application of (group of) user(s).
  - E.g.,
    - A view offering only student information to a student registration application
    - A view offering only building information to a capacity planning application
  - A view can serve one of more applications.
  - A view is used to control data access and enforce security.

Internal Layer

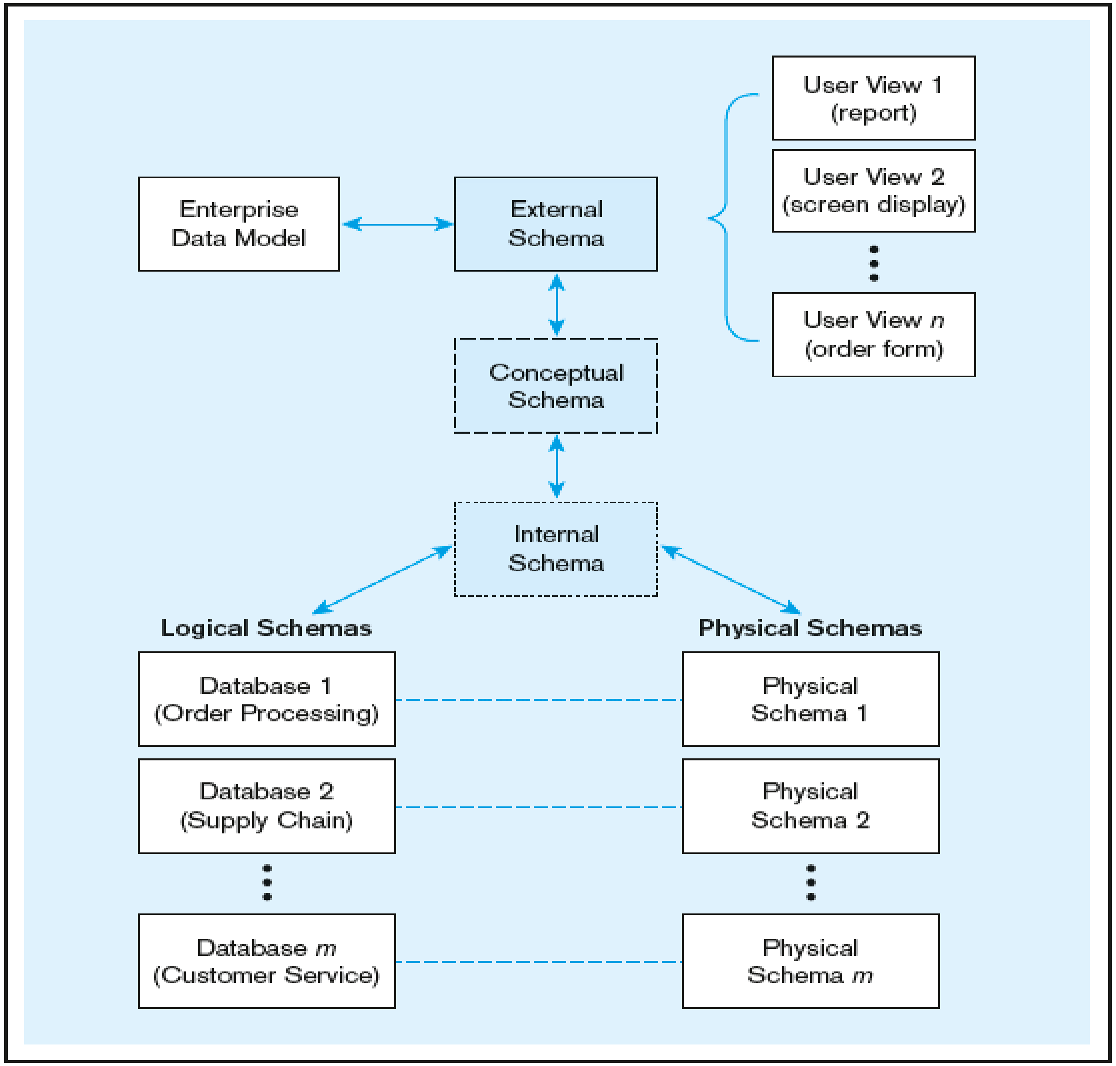
- The internal layer includes the **internal data model**.
- An internal data model specifies how the data are stored or organized physically.

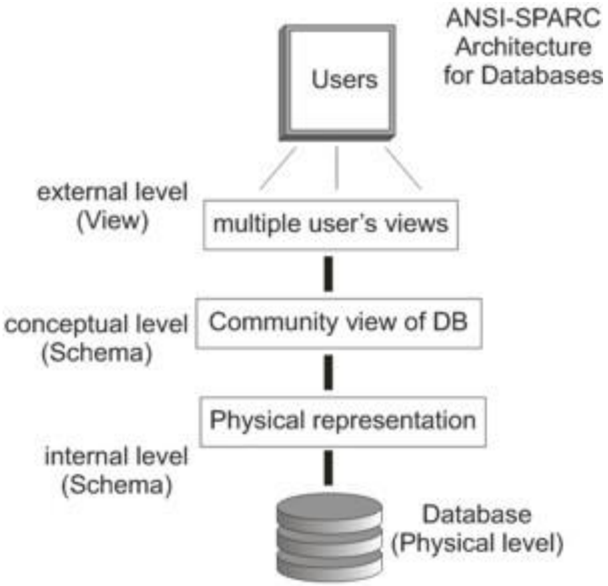
Example of Three Layer Architecture for a Procurement Business Process



Product	name, description, cost, ...
Customer	name, phone, address, ...
Invoice	customer, date, products (with price and amount), ...
Delivery	invoice, address, date, ...

Three Layer Architecture (Different Perspective)



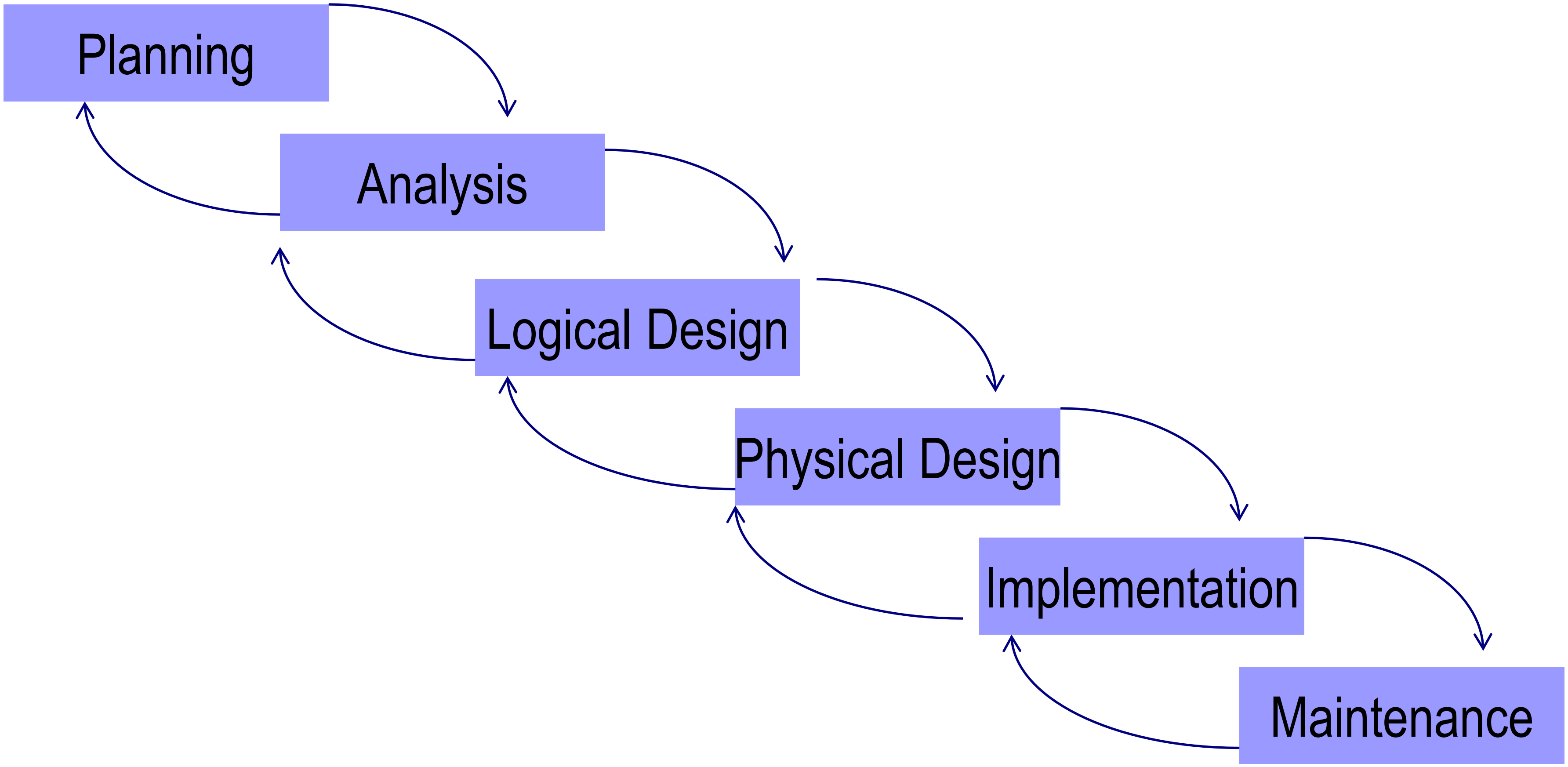


ANSI-SPARCH (American National Standards Institutes, Standards Planning And Requirements Committee) architecture is an abstract design standard for a database management systems).

Outline

- What is a database?
- Historical root of database systems
- Advantages of database approach
- Components of database environments
- Types of Database Applications
- Data Model
- ☞ **Database system development life cycle**
- Evolution of database technology
- Range of database systems

Database System Development Life Cycle



Planning

Analysis

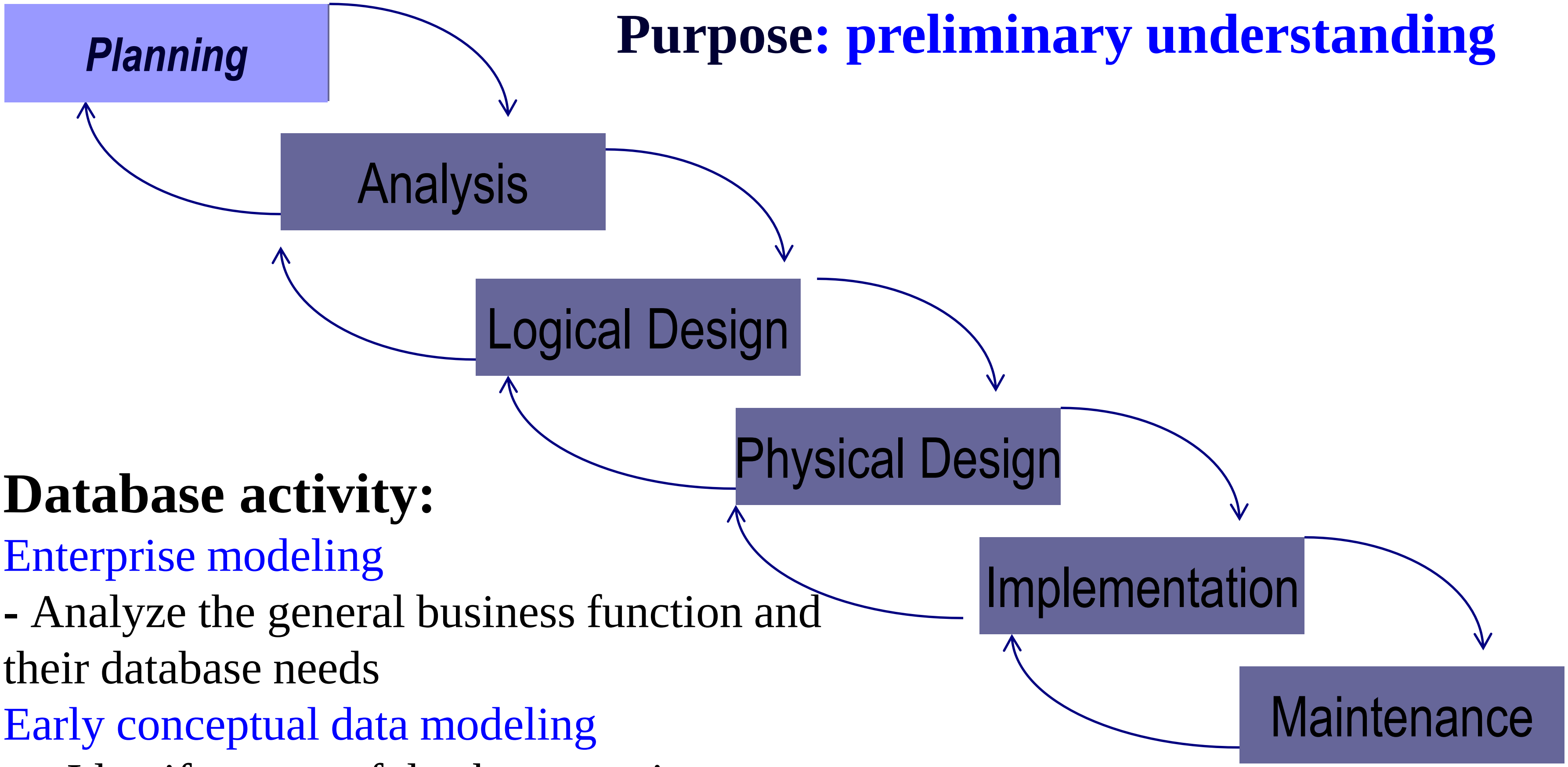
Logical Design

Physical Design

Implementation

Maintenance

Database System Development Life Cycle



Purpose: preliminary understanding

Database activity:

Enterprise modeling

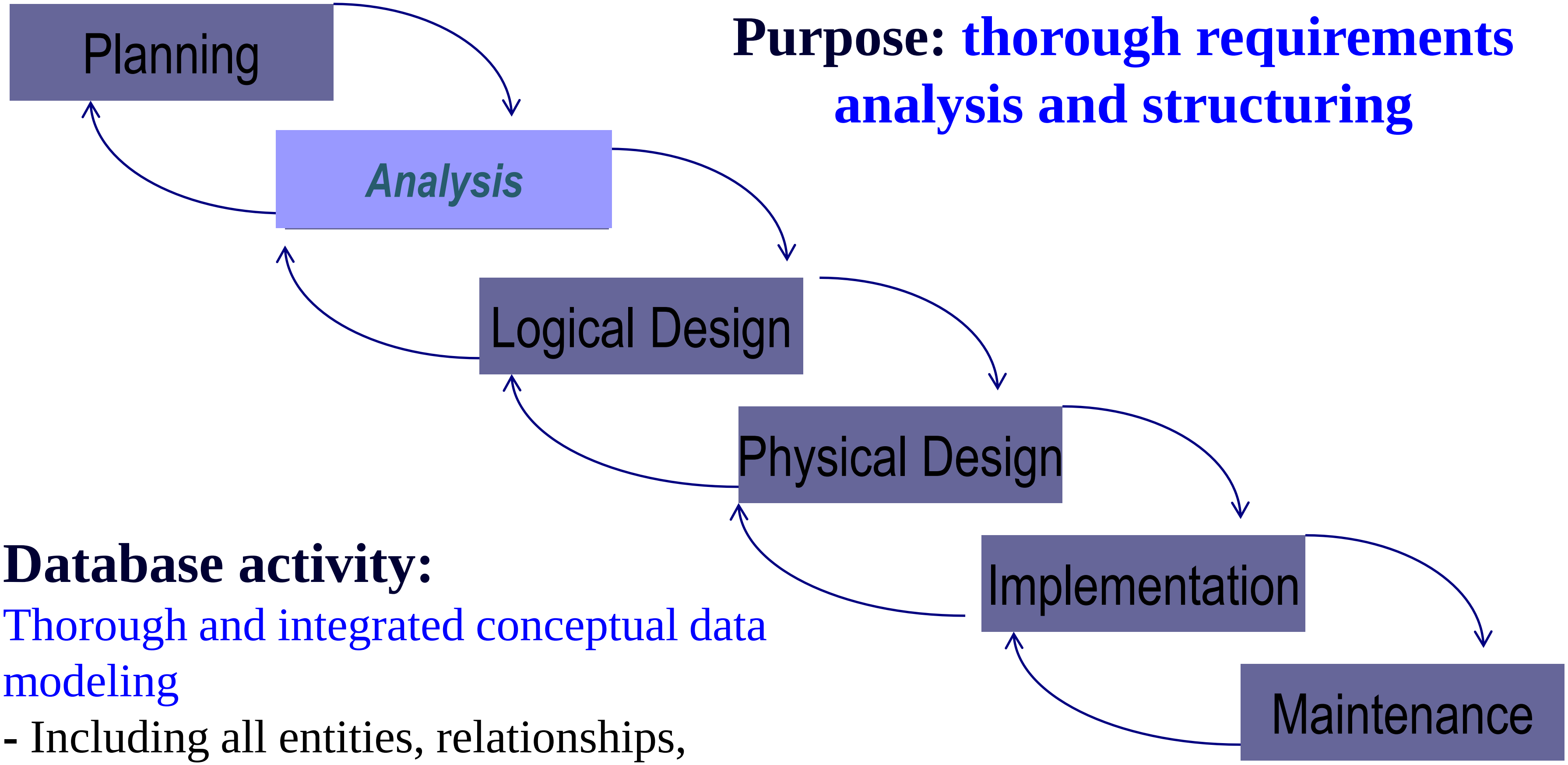
- Analyze the general business function and their database needs

Early conceptual data modeling

- Identify scope of database requirements

Output: concise written requirement documents

Database System Development Life Cycle



Purpose: thorough requirements analysis and structuring

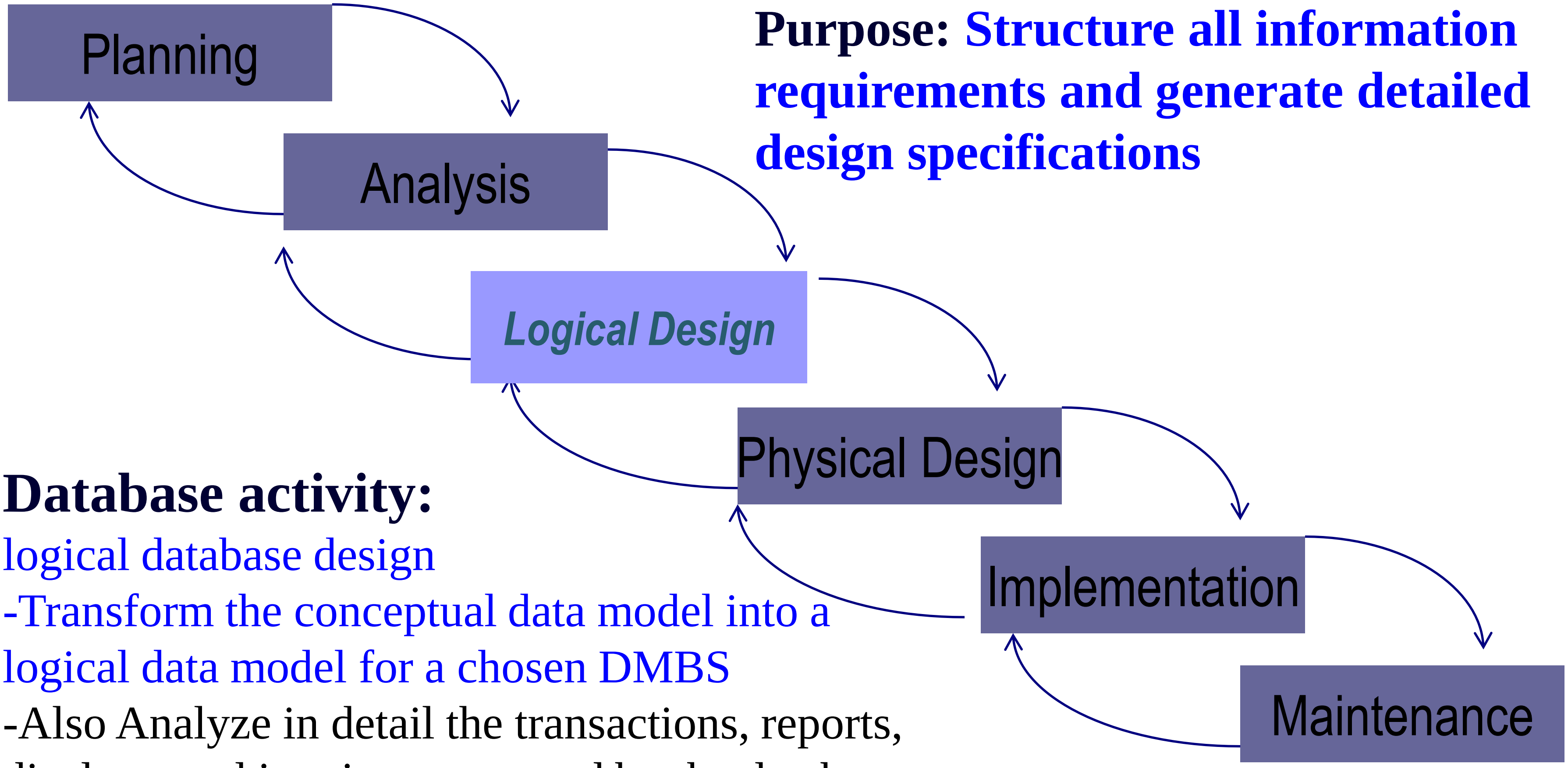
Database activity:

Thorough and integrated conceptual data modeling

- Including all entities, relationships, attributes and business rules.

Output: Conceptual database schema (e.g., ER-diagram)

Database System Development Life Cycle



Purpose: **Structure all information requirements and generate detailed design specifications**

Database activity:

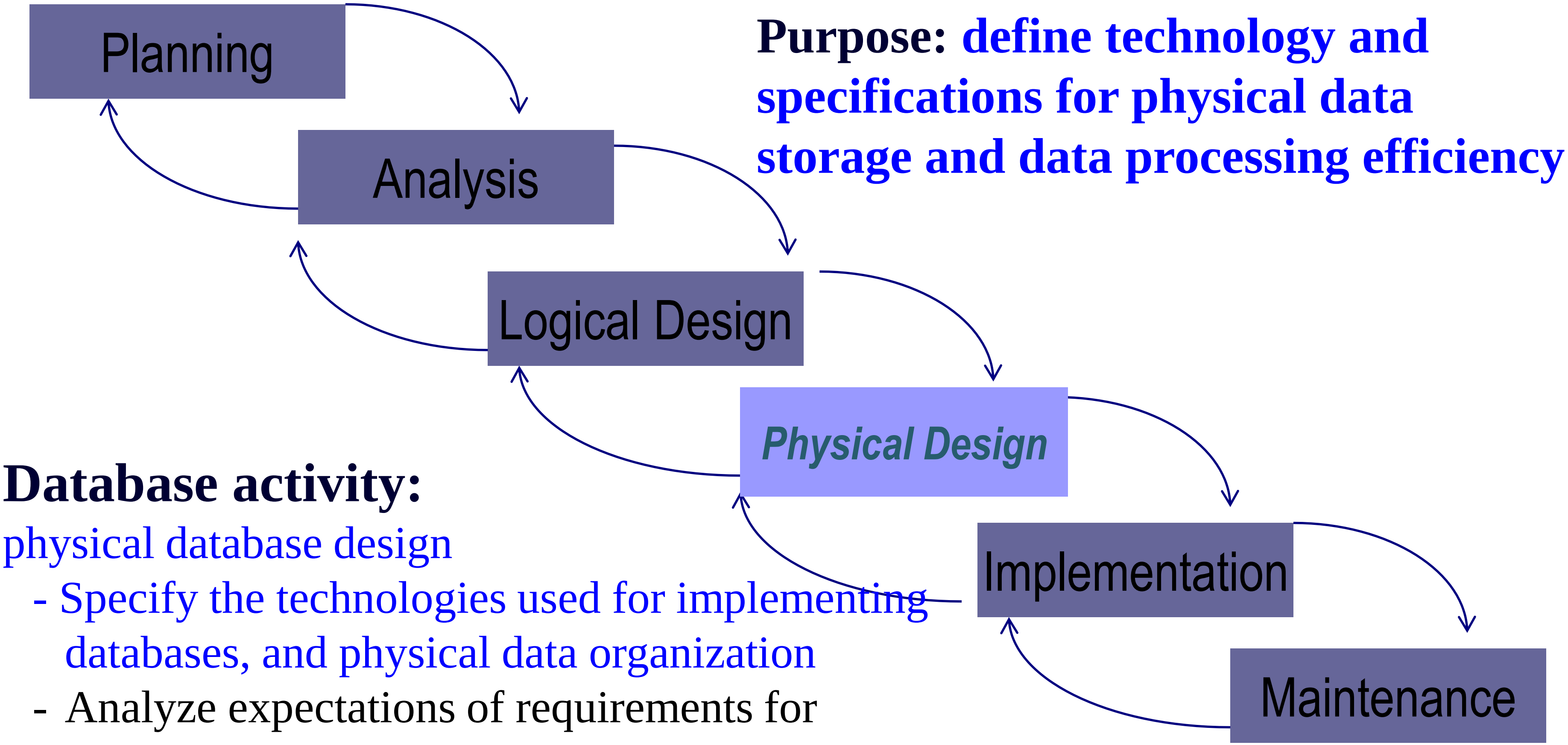
logical database design

-Transform the conceptual data model into a logical data model for a chosen DMBS

-Also Analyze in detail the transactions, reports, displays, and inquires supported by the database.

Output: Logical database schema

Database System Development Life Cycle



Purpose: define technology and specifications for physical data storage and data processing efficiency

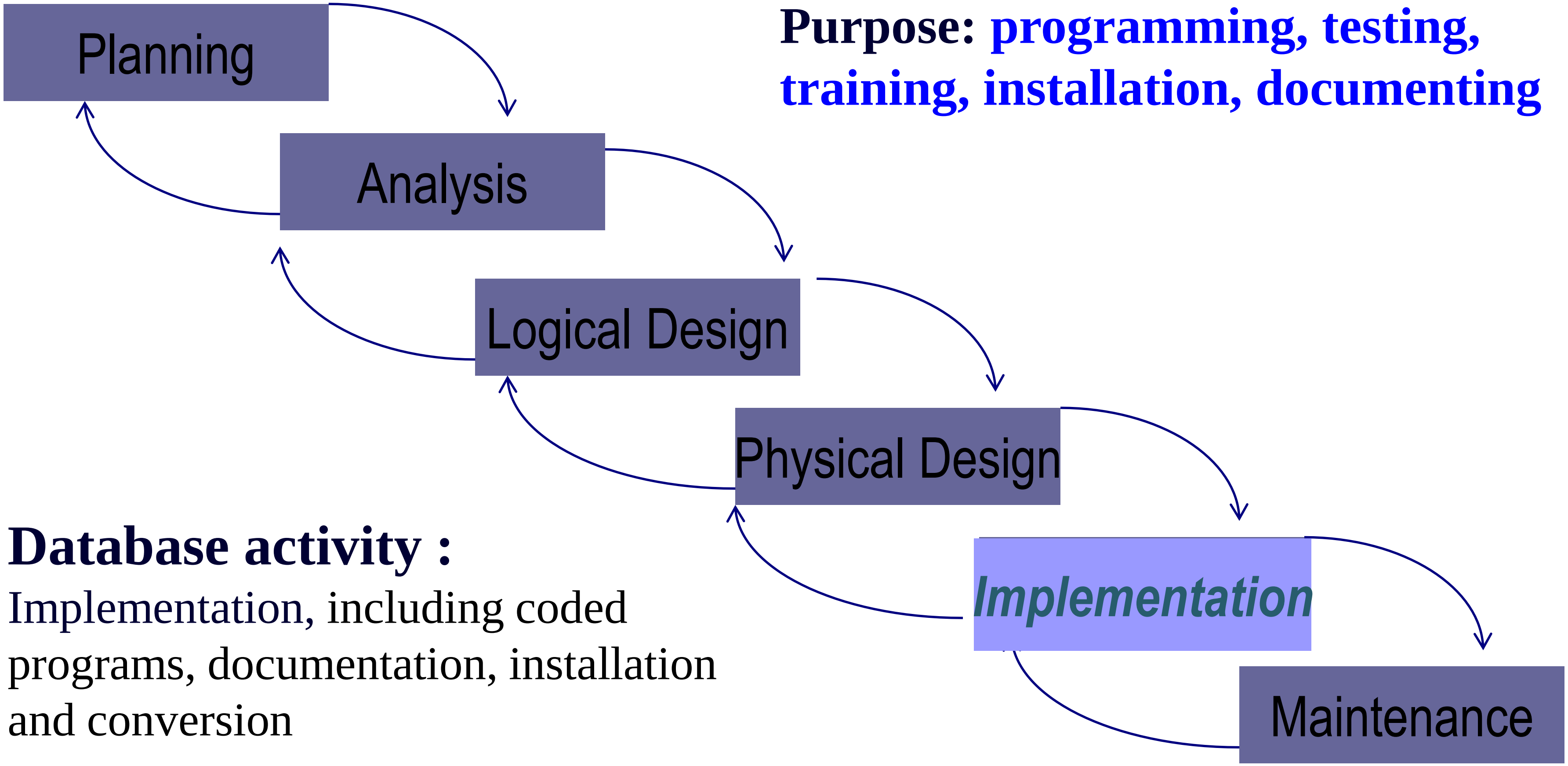
Database activity:

physical database design

- Specify the technologies used for implementing databases, and physical data organization
- Analyze expectations of requirements for response time and data security, backup, recovery, etc.

Output: A physical schema (physical database specificiation)

Database System Development Life Cycle

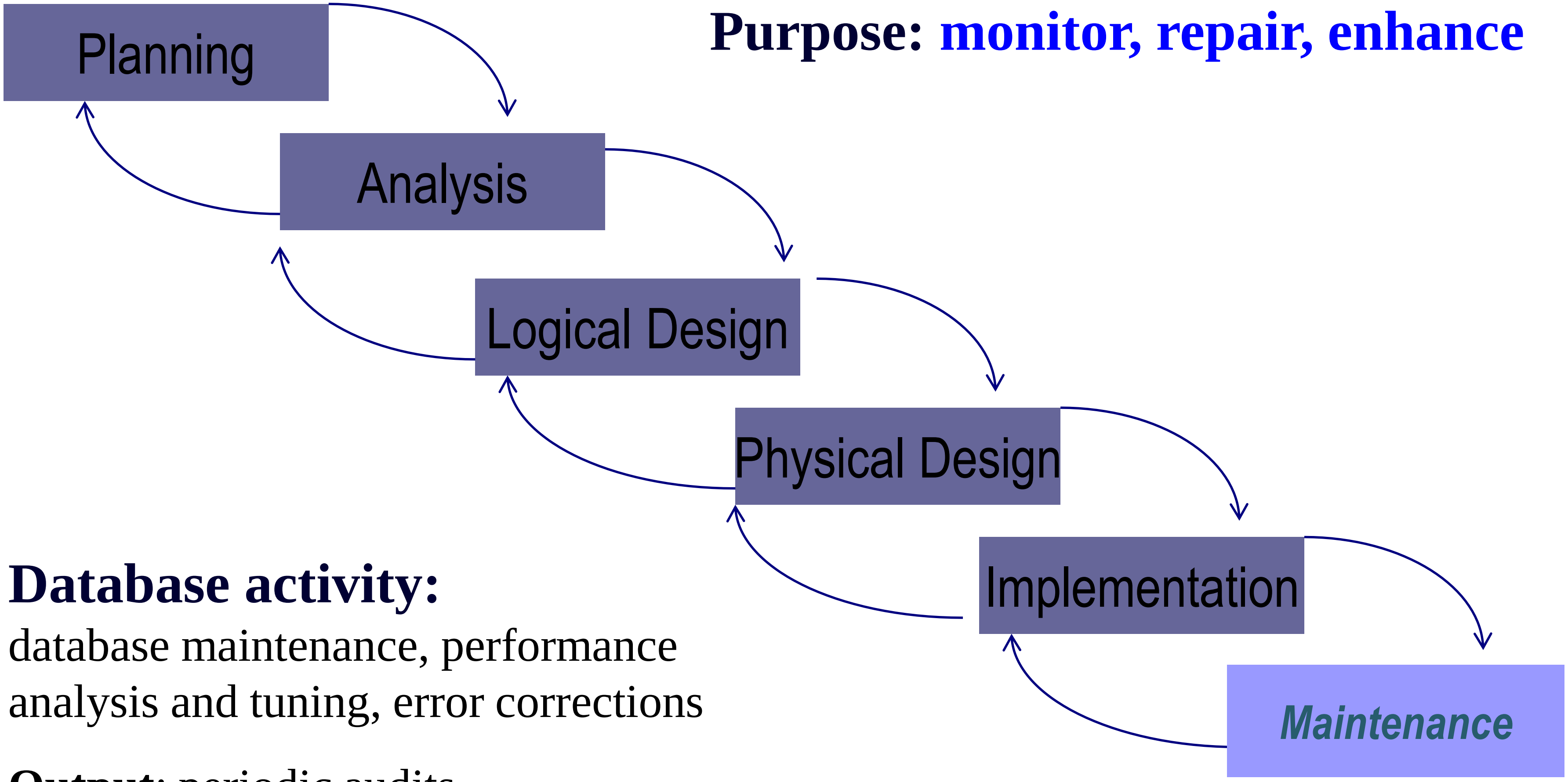


Purpose: programming, testing, training, installation, documenting

Database activity :
Implementation, including coded programs, documentation, installation and conversion

Output: operational programs, documentation, training materials

Database System Development Life Cycle



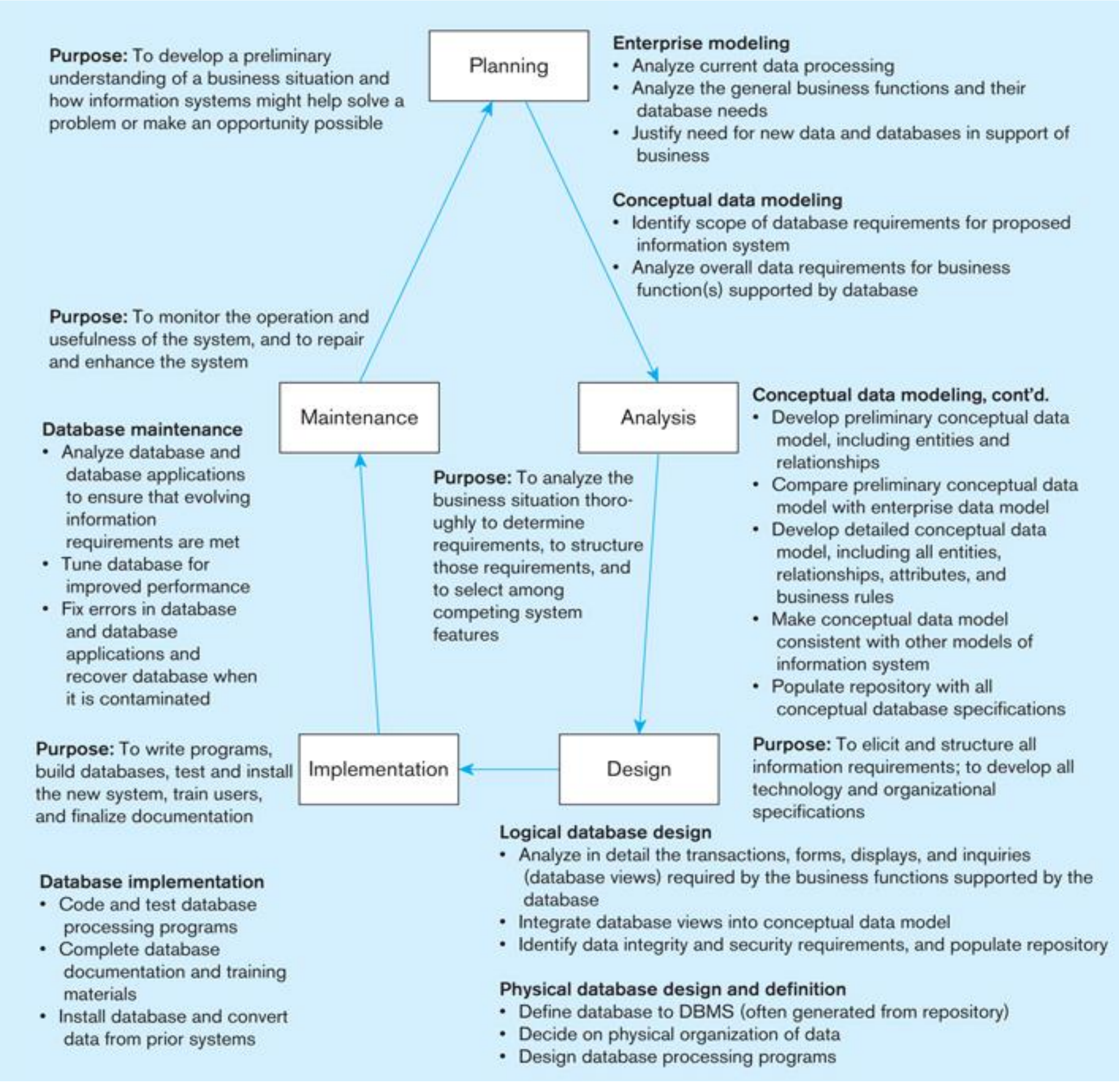
Purpose: monitor, repair, enhance

Database activity:
database maintenance, performance analysis and tuning, error corrections

Output: periodic audits

Summary

Database development activities during the system development life cycle(SDLC)



Planning

Purpose: To develop a preliminary understanding of a business situation and how information systems might help solve a problem or make an opportunity possible

Enterprise modeling
- Analyze current data processing
- Analyze the general business functions and their database needs
- Justify need for new data and databases in support of business

Conceptual data modeling
- Identify scope of database requirements for proposed information system
- Analyze overall data requirements for business function(s) supported by database

Analysis

Purpose: To analyze the business situation thoroughly to determine requirements, to structure those requirements, and to select among competing system features

Conceptual data modeling, cont'd.
- Develop preliminary conceptual data model, including entities and relationships
- Compare preliminary conceptual data model with enterprise data model
- Develop detailed conceptual data model, including all entities, relationships, attributes, and business rules
- Make conceptual data model consistent with other models of information system
- Populate repository with all conceptual database specifications

Design

Purpose: To elicit and structure all information requirements; to develop all technology and organizational specifications

Logical database design
- Analyze in detail the transactions, forms, displays, and inquiries (database views) required by the business functions supported by the database
- Integrate database views into conceptual data model
- Identify data integrity and security requirements, and populate repository

Physical database design and definition
- Define database to DBMS (often generated from repository)
- Decide on physical organization of data
- Design database processing programs

Implementation

Purpose: To write programs, build databases, test and install the new system, train users, and finalize documentation

Database implementation
- Code and test database processing programs
- Complete database documentation and training materials
- Install database and convert data from prior systems

Maintenance

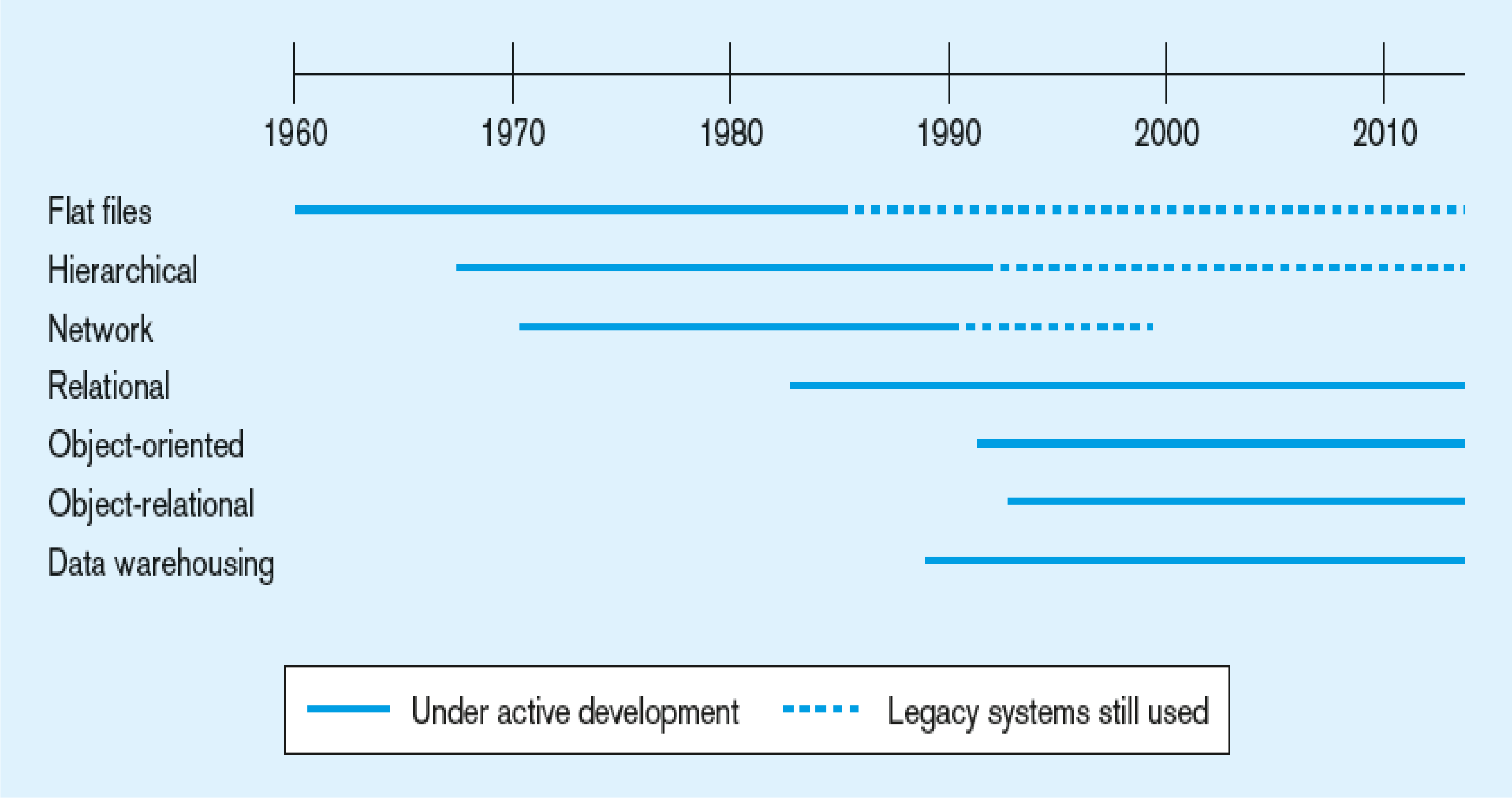
Purpose: To monitor the operation and usefulness of the system, and to repair and enhance the system

Database maintenance
- Analyze database and database applications to ensure that evolving information requirements are met
- Tune database for improved performance
- Fix errors in database and database applications and recover database when it is contaminated

Outline

- □ What is a database?
- □ Historical root of database systems
- □ Advantages of database approach
- □ Components of database environments
- □ Types of Database Applications
- □ Database system development life cycle
- ☞ **Evolution of database technology**
- □ Range of database systems

Evolution of Database Technology



Evolution of Database Technology

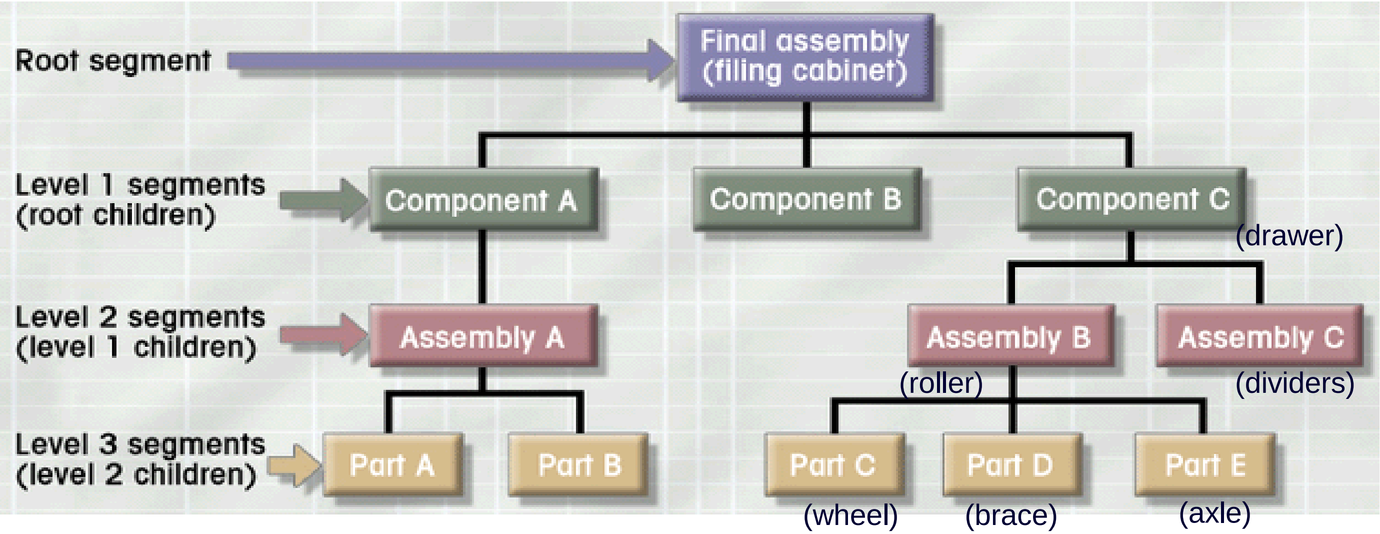
Time	Database Model	Products	Description
1960s-1970s	**File system**	VMS/VSAM	Managed records, not relationship
1970s	**Hierarchical**	IMS, ADABS, IDS-II	Early database systems Navigation access
	Network		
Mid-1970s to present	**Relational**	DB2, Oracle, MS SQL-Server	Conceptual simplicity ER modeling support
Mid-1980s to present	**Object-oriented**	Versant VFS/FastObjects Objectivity/DB	Support complex data Extended relational products support objects, and data warehousing and web database applications
	Object-relational	DB2 UDB Oracle 12c	

Evolution of Database Technology

Time	Database Model	Products
Present	**XML (semi-structured data), Complex data such as text, multimedia, spatial, temporal (time series) , spatio-temporal, etc**	dbXML, Tamino, DB2 UDB, Oracle 12c
	NoSQL	MongoDB, Apache CouchDB, Apache Cassandra, Apache HBase, Membase, Redis, HyperTables, Tokyo Cabinet, Scalaris, Riak
	NewSQL	Akiban, Clustrix, dbShards, GenieDB, H-Store, NuoDB, ScaleArc, ScaleBase, ScaleDB, SchoonerSQL, VMWareSQLFire, VoltDB, Xeround

Hierarchical DMBS

- Adopt a **tree** like data model
  - A segment is equivalent with a record in file system
  - Hierarchical structure supports 1:M relationship
    - Each parent segment can have many children segment. Each child segment has only one parent segment
- Products: IMS (IBM), Register (in Microsoft Windows)

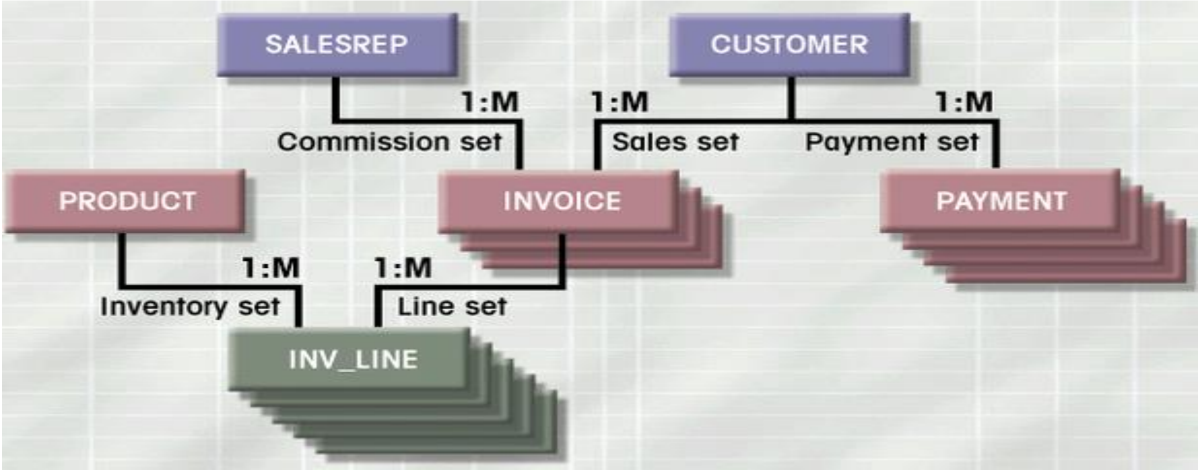


Hierarchical DBMS

- **Advantages**
  - Over the file system model
    - – can represent relationships
  - Conceptual simplicity – tree data structure
- **Disadvantages**
  - Many common data relationships are not 1:M form.
  - No query processor (logical and internal data model intertwined)
  - Complex to implement
  - Difficult to manage and lack of standards
  - No standards for how to implement the model

Network DBMS

- Adopt a **network** data model
  - Allow each record to have multiple parent records and multiple child records with supporting 1:M, M:1, and N:M
    - E.g., INVOICE record is owned by both SALESREP and CUSTOMER
  - Is composed of sets (= relationships). Each set has the owner record (parent) and the member record (child)



Network DBMS

- **Advantages**
  - More flexible than a tree-like data model
  - Handles more relationship types
- **Disadvantages**
  - Complexity to model
  - No query processor (logical and internal data model intertwined)

- Products: CA-IDMS (Computer Associates), UDS from Siemens Nixdorf, DMS 1100 from Unisys, and Image from HP

Limitations of Early Database Systems - Hierarchical and Network DBMSs

- Intermixing of conceptual relationships with the physical storage and placement of records on disk.
- Did not provide sufficient *data abstraction* and *program-data independence* capabilities
  - Data abstraction refers to providing only essential information about the data to the outside world, hiding the background details or implementation.

Limitations of Early Database Systems - Hierarchical and Network DBMSs (Cont.)

- Laborious to reorganize the database when changes where made to the application’s requirements
  - They may provide efficient access for the original queries and transaction that the database was designed to handle, but not provide enough flexibility to access records efficiently for new queries and transactions.
- No database query language - Provide only general programming language interfaces
  - Time consuming and expensive to implement new queries and transactions

Relational DBMS

- Relational databases were originally proposed to
  - separate the physical storage of data from its conceptual representation
    - hiding complexity of internal implementation
  - provide a mathematical foundation (i.e., relational model) for data representation and querying
    - Relational model is from set theory in Mathematics

Example: Relational Database

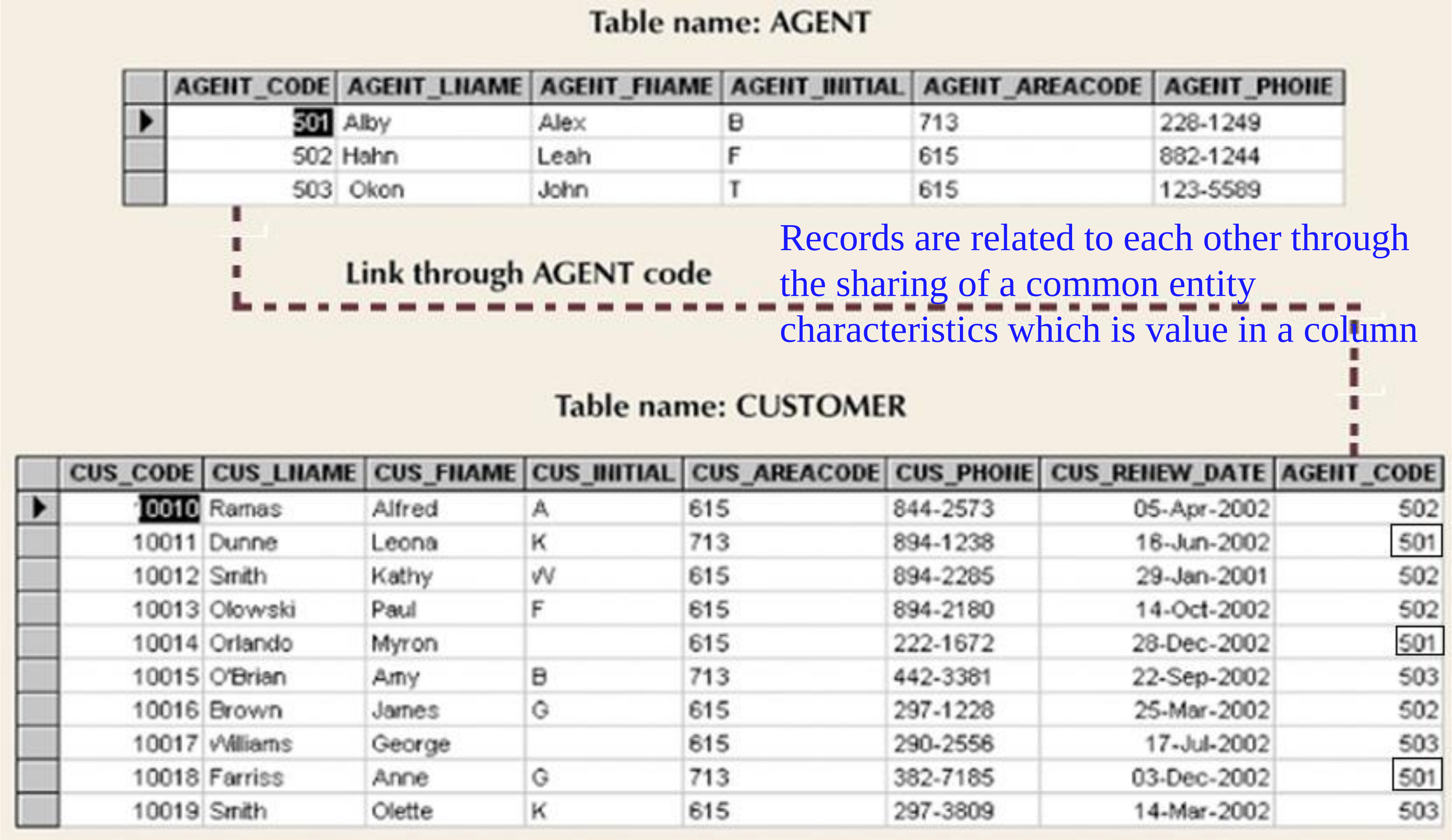


Table name: AGENT

AGENT_CODE	AGENT_LNAME	AGENT_FNAME	AGENT_INITIAL	AGENT_AREACODE	AGENT_PHONE
501	Alby	Alex	B	713	228-1249
502	Hahn	Leah	F	615	882-1244
503	Okon	John	T	615	123-5589

Records are related to each other through the sharing of a common entity characteristics which is value in a column

Table name: CUSTOMER

CUS_CODE	CUS_LNAME	CUS_FNAME	CUS_INITIAL	CUS_AREACODE	CUS_PHONE	CUS_RENEW_DATE	AGENT_CODE
10010	Ramas	Alfred	A	615	844-2573	05-Apr-2002	502
10011	Dunne	Leona	K	713	894-1238	16-Jun-2002	501
10012	Smith	Kathy	W	615	894-2285	29-Jan-2001	502
10013	Olowski	Paul	F	615	894-2180	14-Oct-2002	502
10014	Orlando	Myron		615	222-1672	28-Dec-2002	501
10015	O'Brian	Amy	B	713	442-3381	22-Sep-2002	503
10016	Brown	James	G	615	297-1228	25-Mar-2002	502
10017	Williams	George		615	290-2556	17-Jul-2002	503
10018	Farriss	Anne	G	713	382-7185	03-Dec-2002	501
10019	Smith	Olette	K	615	297-3809	14-Mar-2002	503

Relational DBMS

- Introduced by E.F. Codd (of IBM) in 1970
- RDBMS (Relational DBMS) products emerged in the early 1980s.- The performance was quite slow
- The system was heavily researched and experimented within IBM Research and several universities to improve
  - storage, indexing techniques, better query processing and optimization
- Most commercial database systems such as Oracle, IBM's DB2, Informix, MS SQL Server, PostgreSQL, MySQL (open source, Oracle) and many others

Relational DBMS

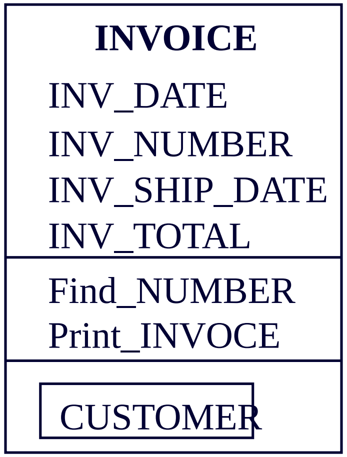
- *Data abstraction* and *program-data independence* were much improved than earlier systems.
- Introduce high-level query languages (SQL)
  - making it much faster to write new queries than programming language - Ad hoc query capability
- Well tuned for OLTP (online transaction processing) systems
- Dominant database technology for traditional database systems

Object-Oriented DMBS (OODBMS)

- Motivation
  - Need to capture complex data processing in CAD and other applications
  - The emergence of object-oriented programming language in the 1980s
- OODBMS combines
  - **OO programming concepts** such as
    - encapsulation, polymorphism and inheritance
  - **Database management concepts** such as
    - ACID (Atomicity, Consistency, Isolation and Duration) for system integrity
    - Features that are common in the RDBMS world such as transactions, the ability to handle large amounts of data, indexes, deadlock detection, backup and restoration features and data recovery mechanisms also exist

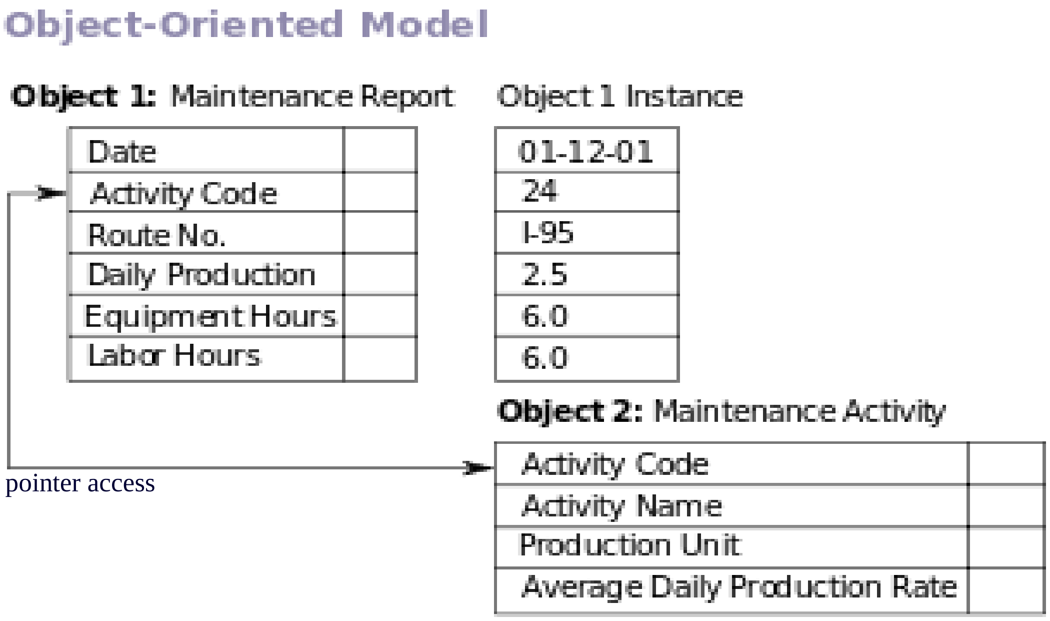
Object-Oriented DMBS (OODBMS)

- Information (data) is represented in the form of ***object*** as used in object-oriented programming.
- **O**bject-**O**riented **data model**
  - More closely represents the real world
  - An **class** is an abstraction of a real-world entity.
    - **Attributes** describe the properties of an object, e.g., name
    - **Method** represents a real-word action, e.g., finding a selected name, changing name, printing address
  - Both data entity instance and its behavior are contained in a single structure known as object class.
    - Both data and relationships are contained in an object.



Object-Oriented DMBS (OODBMS)

- □ The programming language and the database schema use the same type definitions
- □ Access to data can be faster because joins are often not needed as RDBMS



Object-Oriented Model

Object 1: Maintenance Report

Date	
Activity Code	
Route No.	
Daily Production	
Equipment Hours	
Labor Hours	

Object 1 Instance

01-12-01
24
I-95
2.5
6.0
6.0

Object 2: Maintenance Activity

Activity Code	
Activity Name	
Production Unit	
Average Daily Production Rate	

pointer access

Data Integration Glossary, U.S. Department of Transportation, August 2001.

Object-Oriented DMBS (OODBMS)

- **Disadvantage** of OODBMS
  - Complexity
  - Increased (maintenance) costs
  - Supporters of relational approach believe simplicity and purity of relational model are lost
  - Some believe OODBMS will be a minority of applications
  - Object-oriented DBMS’s are failed because they did not offer the efficiencies of well-entrenched relational DBMS’s.

Object-Relational DMBS (ORDBMS)

- Object relational database management system (ORDBMS) simply puts an object oriented front end on a relational database (RDBMS)
  - Object-relational data model includes the OO model’s best features within an simpler relational data model.
    - User-defined data types
    - User-defined methods
    - Inheritance, etc.
    - SQL3
  - Also called to **extended relational data model**
- Most commercial DBMS vendors, including IBM, Informix, Microsoft and Oracle are adding ORDM functionality.

XML Databases

- **XML DBMSs**
  - use the XML data model to store data
    - XML is a very popular standard to exchange data between various applications.
  - An XML specification essentially represents data in hierarchical way.
    - E.g., the address tag is further subdivided into street, number, zip code, city and country tags.

```
<employee>
    < firstname>Bart </firstname>
    < lastname>Fox</lastname>
    <address>
        <street> Sycamore Hills </street>
        <number> 69 </number>
        <zipcode> 46714 </zipcode>
        <city> Fort Wayne </city>
        <state> Indiana </state>
     </address>
     < gender>Male</gender>
</employee>
```

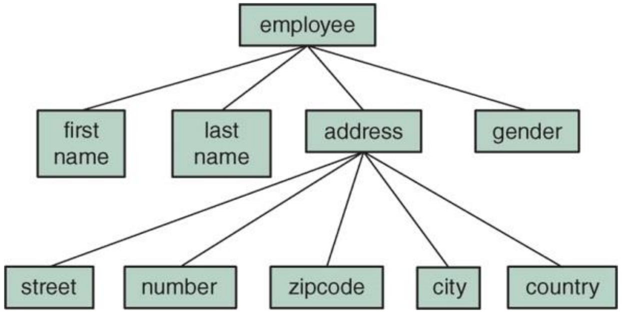


Fig. Tree-based XML representation

XML Databases (cont.)

- Types of XML DBMS
  - **Native XML DBMSs** (e.g., BaseX, eXist) map the tree structure of an XML document to a physical storage structure
  - **XML-enabled DBMSs** (e.g., Oracle, IBM DB2) are existing DBMSs that are extended with facilities to store XML data in an integrated and transparent way.
  - Both types of DBMSs provide facilties to query XML data

Cloud DBMS and Federated DBMS

- **Cloud** DBMS
  - DBMS and database are hosted by a third-party cloud provider
  - E.g., Apache Cassandra project and Google's BigTable
- **Federated** DBMS
  - provides a uniform interface to multiple underlying data sources
  - hides the underlying storage details to facilitate data access

In-Memory DBMS

- **In-memory** DBMS
  - stores all data in internal memory instead of slower external storage (e.g., disk)
  - often used for real-time purposes
  - E.g., Hana (SAP)

Emergence of Big Data Storage Systems

- **Big Data** from modern data sources such as social media (e.g., Twitter, Facebook), large e-commerce companies, Web search indexes, cloud storage/backup
- Characteristics of Big Data:

  3Vs (Volume, Variety, Velocity)
- Relational DBMSs might not compatible with these data. We need more flexible or even schema-less data structure.
- New types of database systems are needed
  - NoSQL (Not Only SQL)
  - NewSQL

NoSQL (non-only SQL) DBMSs

- NoSQL databases
  - are targeted at storing big and unstructured data
  - focus on scalability and the ability to cope with irregular or highly volatile data structures (which are some shortcomings of relational DBMS)
  - can be classified into
    - key-value stores,
    - Tuple or document stores,
    - column-oriented databases, and
    - graph databases
  - E.g., Apache Hadoop, MongoDB, Neo4j

Outline

- What is a database?
- Historical root of database systems
- Advantages of database approach
- Components of database environments
- Types of Database Applications
- Database system development life cycle
- Evolution of database technology
- ☞ **Range of database systems**

Categorization based on Degree of Simultaneous Access

- **Single user** vs **multi user systems**
  - In a **single-user system**, only one user at a time is allowed to work with the DBMS. It is not desirable in a networked environment
  - **Multi-user systems** allow multiple users to simultaneously interact with the database in a distributed environment

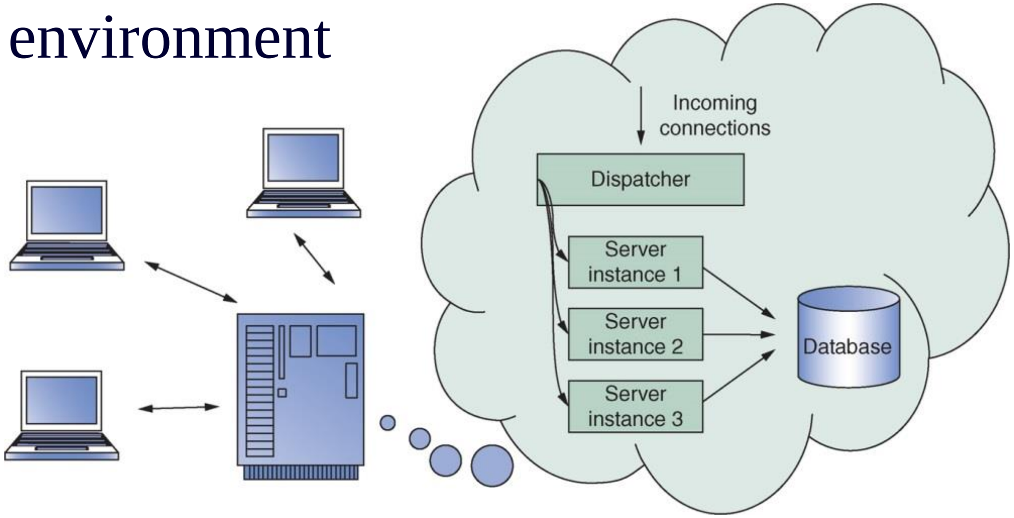
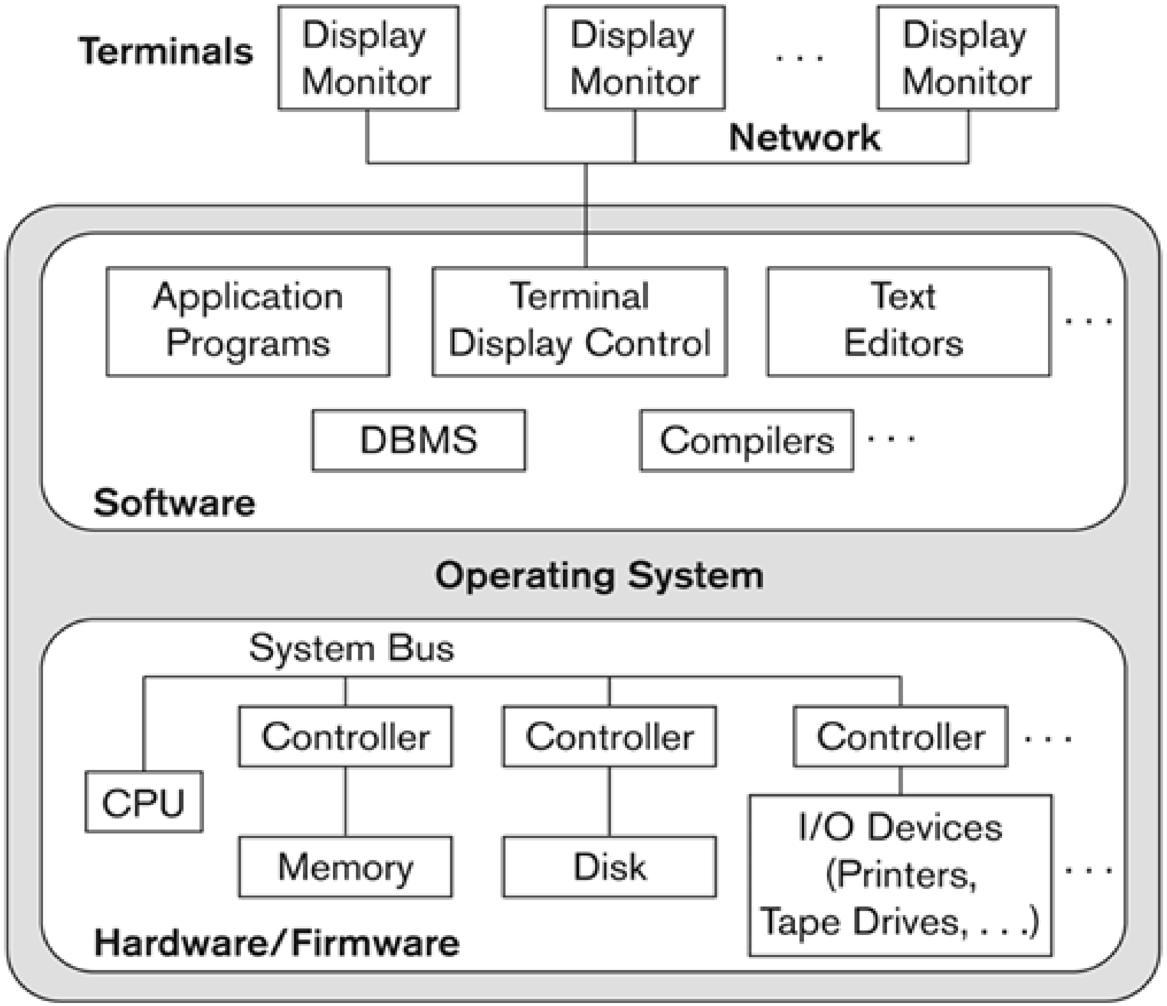


Fig. three clients are being served by three server instances or threads

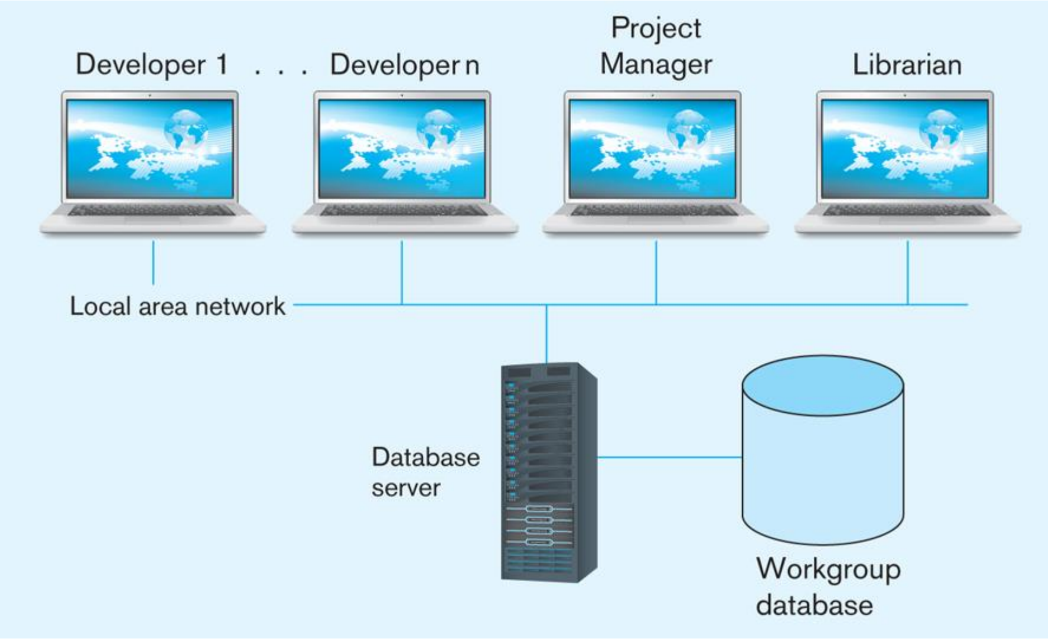
Categorization based on Architecture

- **Centralized** DBMS architecture
  - data and applications are maintained on a centralized server
- **Client server** DBMS architecture
  - active clients request services from passive servers
  - fat server versus fat client variant
- ***n*-tier** DBMS architecture
  - client with GUI functionality, application server with applications, database server with DBMS and database, and web server for web based access

Centralized DBMS



Two-tier Database System Architecture



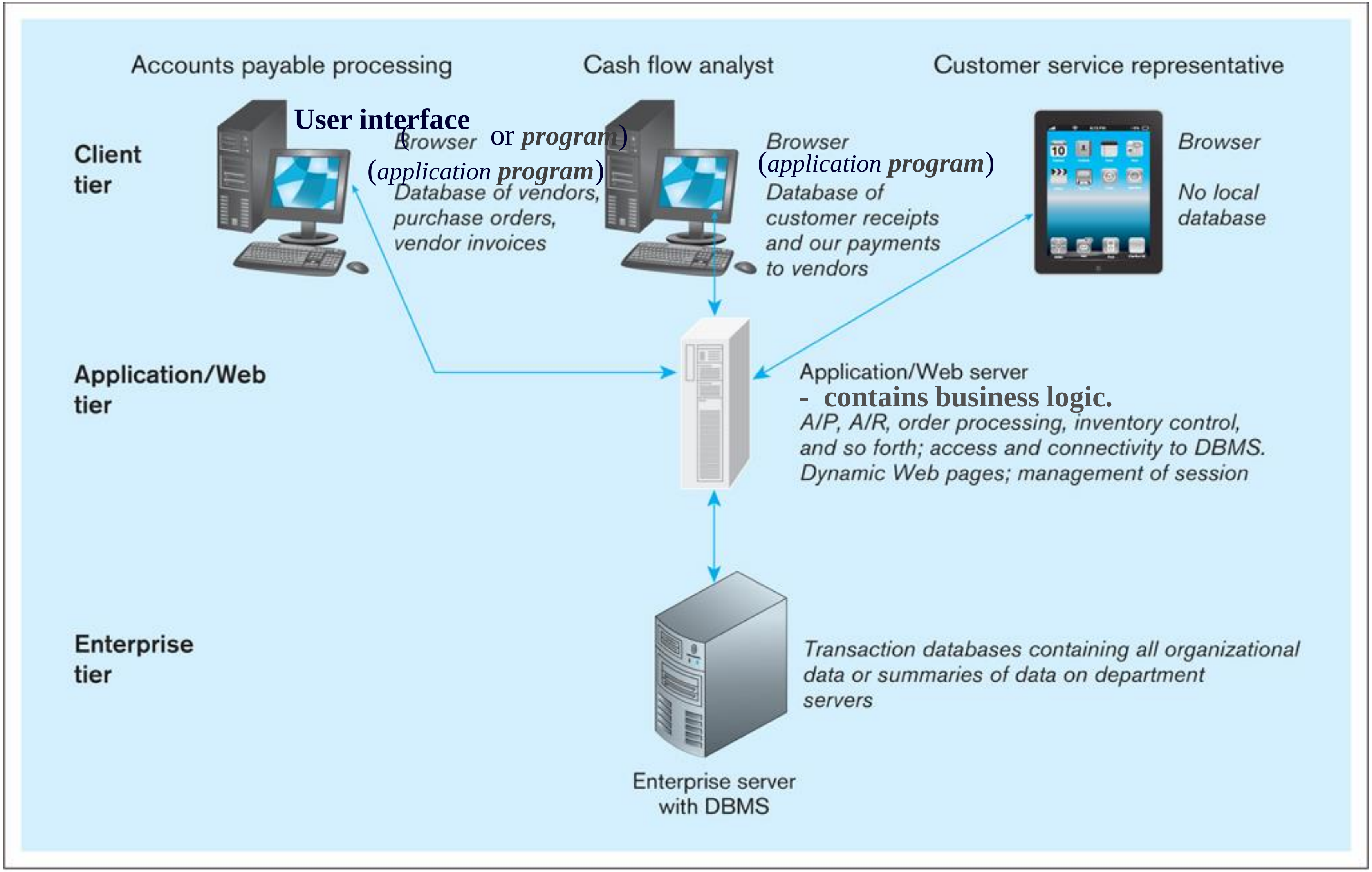
Client

DBMS Server

- For a small team of people (< 25 or <100 persons) who collaborate on the same project or on a group of similar projects.
- Each member has a computer, and **the computers are linked by network.**

  Each computer has a copy of database application program.
- The database itself and the DBMS are stored on a central device (“database server).

Three-tiered Client/Server Database Architecture


Accounts payable processing
Cash flow analyst
Customer service representative
Client tier
Browser
Database of vendors, purchase orders, vendor invoices
Browser
Database of customer receipts and our payments to vendors
Browser
No local database
Application/Web tier
Application/Web server
A/P, A/R, order processing, inventory control, and so forth; access and connectivity to DBMS. Dynamic Web pages; management of session
Enterprise tier
Transaction databases containing all organizational data or summaries of data on department servers
Enterprise server with DBMS

User interface

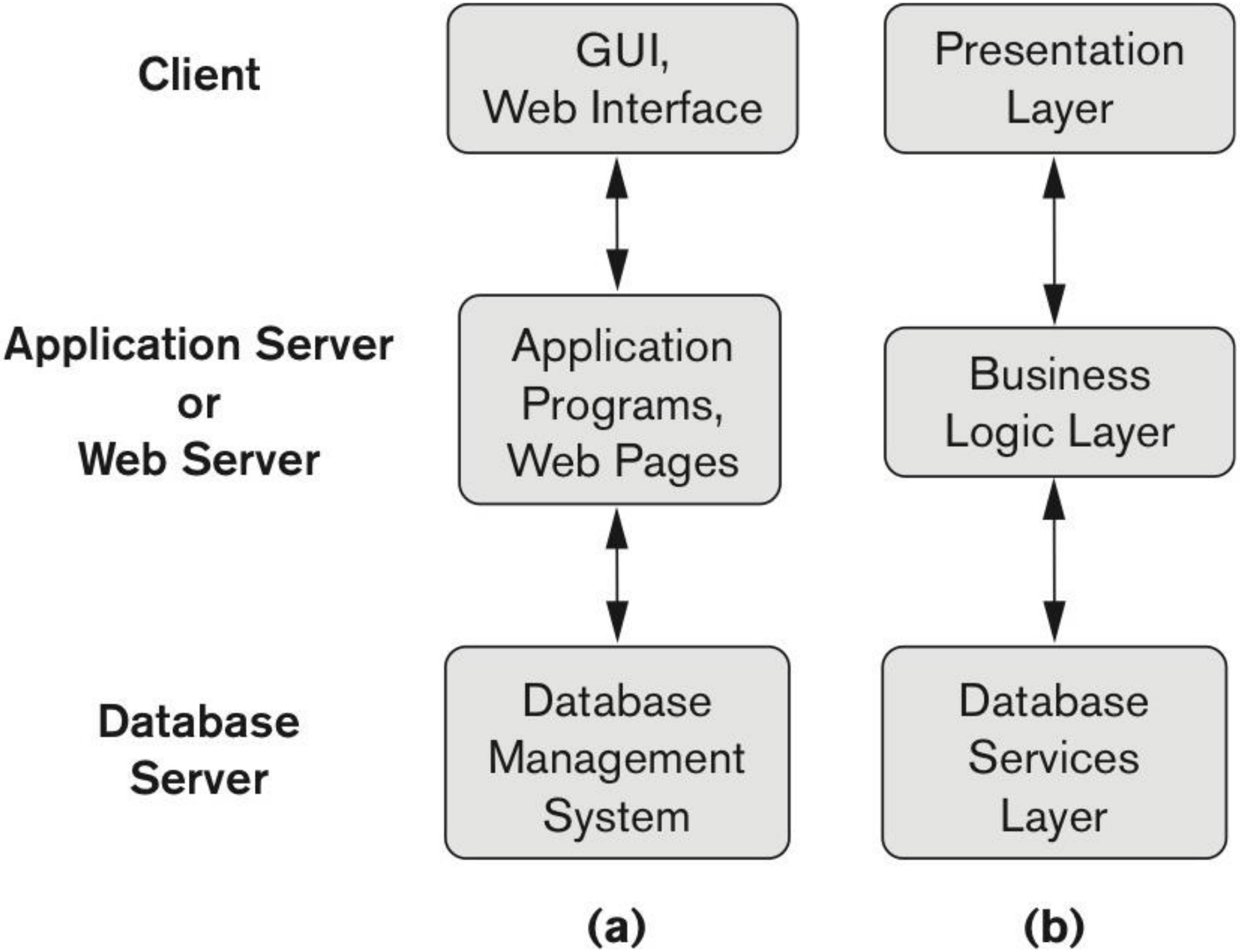
(or ***program***)

(*application* ***program***)

(*application* ***program***)

- **contains business logic.**

Logical Tree-tier Client/Server Architecture



Client

Application Server
or
Web Server

Database
Server

GUI,
Web Interface

Application
Programs,
Web Pages

Database
Management
System

(a)

Presentation
Layer

Business
Logic Layer

Database
Services
Layer

(b)

Categorization based on Usage

- **On-line transaction processing (OLTP)**
  - focus on managing operational or transactional data
  - database server must be able to process lots of simple transactions per unit of time
  - DBMS must have good support for processing a high volume of short, simple queries
- **On-line analytical processing (OLAP)**
  - focus on using operational data for tactical or strategical decision making
  - limited number of users formulates complex queries
  - DBMS should support efficient processing of complex queries which often come in smaller volumes

Categorization based on Usage

- **Big Data** & Analytics
  - All arounds these days
  - New database technology is needed, e.g., NoSQL
  - **NoSQL databases** (CouchDB, MongoDB, Cassandra, Neo4j)
    - focus on more flexible, or even schema-less, database structures
      - Good for storing unstructured information such as emails, text documents, Twitter tweets, Facebook posts.
    - scale more easily in terms of storage capacity

Categorization based on Usage

- **Multimedia DBMSs**
  - provide storage of multimedia data such as text, images, audio, video, 3D games, etc.
  - should also provide content-based query facilities (e.g., find images of polar bear
  - may require specific hardware support due to very resource-intensive transactions
  - Multimedia data are usually stored as a binary large object (BLOB)

Categorization based on Usage

- **Spatial DBMSs**
  - support storage and querying of spatial data (both 2D and 3D)
  - support spatial operations such as calculating distances or relationships between objects (e.g., contain, intersect, detached)
  - is a key building block of geographical information systems (GIS)
- **Sensor DBMSs**
  - manages sensor data such as biometric data from wearables, or telematics data which continuously record driving behavior.

Categorization based on Usage

- **Mobile DBMSs**
  - run on smartphones, tablets or other mobile devices.
  - should always be online, have a small footprint and be able to deal with limited processing power, storage and battery life
  - E.g., Oracle Lite, Sybase SQL Anywhere, Microsoft SQL Servier Compact, SQLite, and IBM DB2 Every place
- **Open-source DBMSs**
  - code of open source DBMSs is publicly available and can be extended by anyone
  - See www.sourceforge.net
  - E.g., MySQL (Oracle), PostgresSQL, Twig OODBMS(Google), Perst OODBMS (McObject)

Why Study DBMSs ??

- Shift from *computation* to *information*
- Data increasing in diversity and volume.
- DBMS is used to maintain, query large datasets.
- Benefits include recovery from system crashes, concurrent access, quick application development, data integrity and security.
- DBAs and database tuners hold responsible jobs and are well-paid!

- Further, data analytics, and data scientists ☺
  (although you need more data analytics knowledge in data mining, machine learning and statistics…)