

Database Management Systems

Lecture - 1

Lecture Overview

- Database Definition
- Explanation of its different parts
- Other Definitions of Database
- Data
- Types of Data
- Information
- Metadata

What is a Database ?

Database Def-1

A database is a shared collection of logically related data that is stored to meet the requirements of different users of an organization

Key Points - Def-1

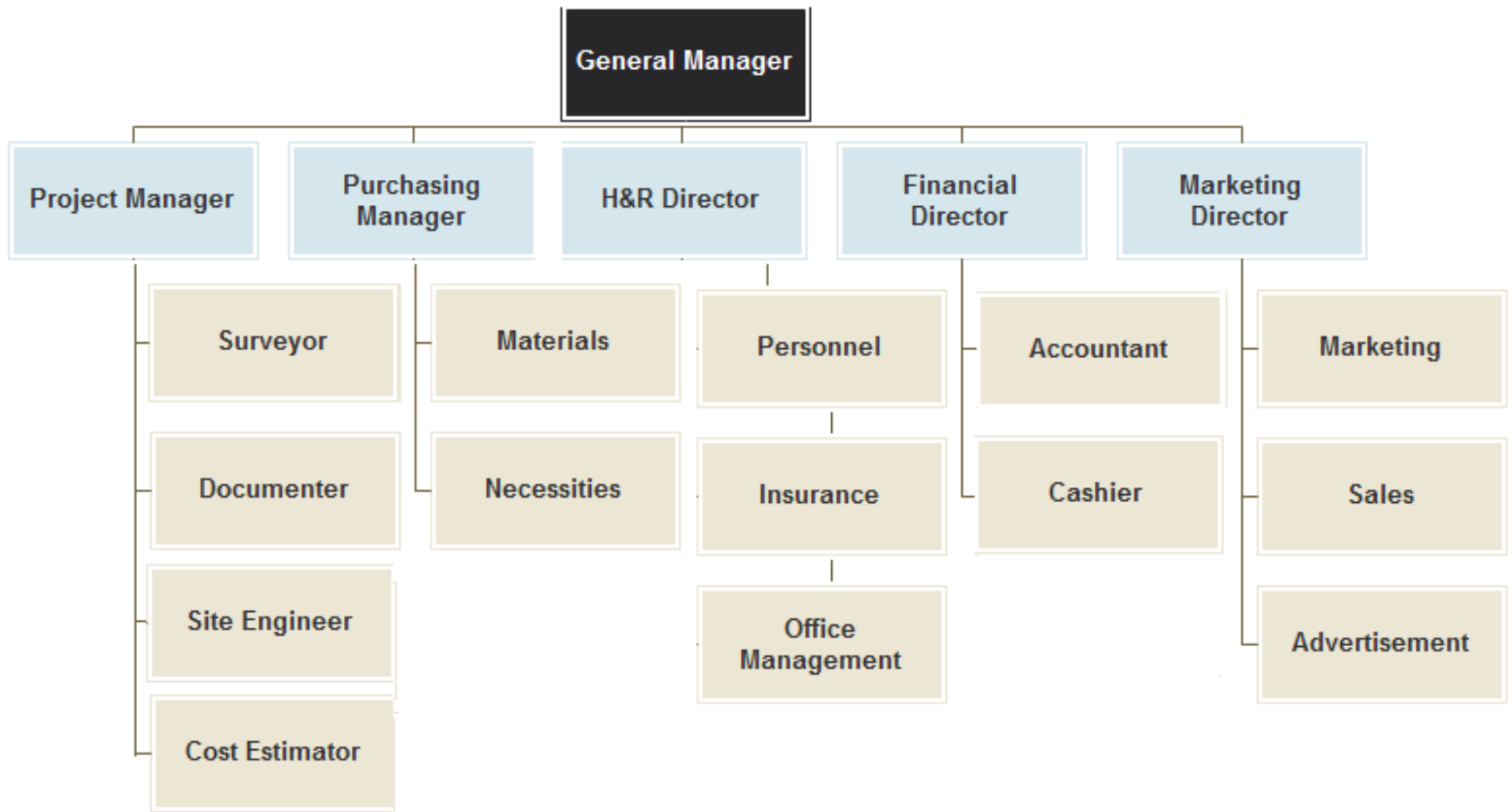
shared

logically related

different users

organization

Example: X-Construction Company



Database Def-2

A collection of organized data of an organization (e.g. part numbers, product codes, customer information, etc.) It usually refers to data organized and stored on a computer that can be searched and retrieved by a computer program.

Database Def-3

An organized collection of information in
computerized format.

Database Def-4

A collection of information organized and presented to serve a specific purpose.

Database Def-4

Example:

A telephone book is a common database.

A computerized database is a large, updated, and organized repository of information that is rapidly searched and retrieved by computer application programs.

Database Def-4

A computerized database is a large, updated, and organized repository of information that is rapidly searched and retrieved by computer application programs.

Example:

A telephone book is a common database.

Database Def-5

A Computerized representation, flow and storage of data/information of any organization

.

Database Def-6

A data structure that stores metadata, i.e. data about the structure of stored data. More generally we can say an organized collection of information.

Database Def-7

A database is a self-describing collection of integrated records

Database Def-8

A database models a particular real world system in the computer in the form of data.

Data

Data: Facts concerning things, such as people, objects, or events

Data is raw, unorganized facts that need to be processed. Data can be something simple and seemingly random and useless until it is organized.

Types of Data

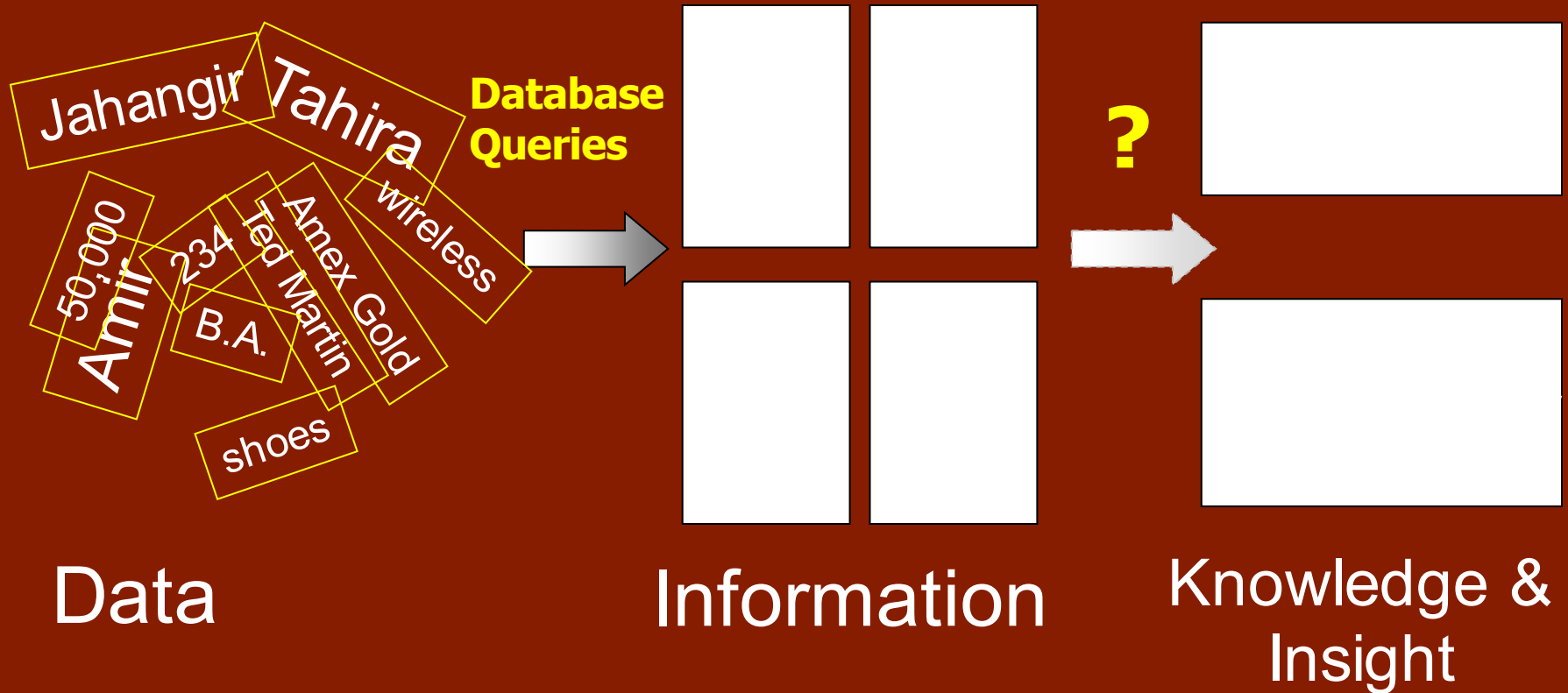
- Numeric
- Alphabetic
- Alphanumeric
- Image
- Audio
- Video
- Any Other than the Above

Information

Information: Data that have been processed and presented in a form suitable for human interpretation

When data is processed, organized, structured or presented in a given context so as to make it useful, it is called information

Data, Information & Knowledge



Metadata

Metadata is defined as data about data, it is used to describe the properties and characteristics of data.

Study Material

1-A Fundamental Study of DATABASE MANAGEMENT SYSTEMS, 3rd Edition (it- Series)

2-Fundamentals of Database Systems (7th edition)

3-Database Systems: Principles Design and Implementation
by Catherine M. Ricardo

4-Lecture Slides & Handouts

End of Lecture

Database Management Systems

Lecture -2

Lecture Overview

- File Processing System
- Disadvantages of File Processing System
- Data Management
- Database
- DBMS
- Database Environment

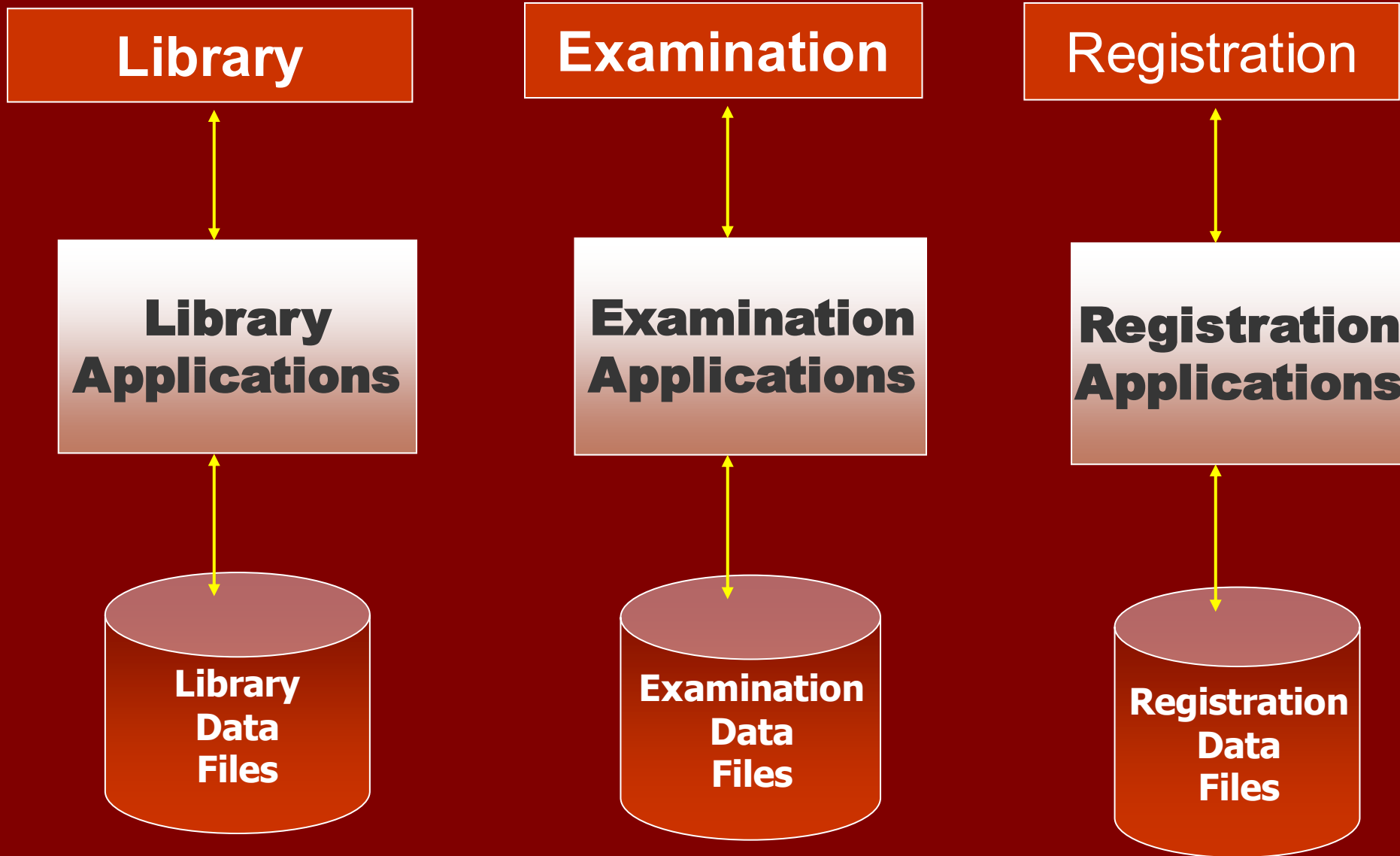
Lecture Overview

- Components of Database Environment
- Advantages of Database Approach
- Disadvantages of Database Approach
- Data as a Resource
- Data Forms / levels

File Processing Systems

A collection of programs that perform for ***data management services*** for the end-users but using individual files. It is the first computer-based approach of handling the management of ***data*** in commercial or business applications. That is why it is also called a replacement of the ***pure manual file system***.

File Processing Systems



Program and Data Interdependence

File Processing Systems

Library
Reg_Number
Name
Father Name
Books Issued
Fine

Examination
Reg_Number
Name
Address
Class
Semester
Grade

Registration
Reg_Number
Name
Father Name
Phone
Address
Class

Duplication of Data

Vulnerable to Inconsistency

File Processing Systems

It has the following shortcomings.

- Data Redundancy
- Data Isolation
- Integrity Problems (data constraints & rules)
- Program and Data Interdependence
- Atomicity Problem
- Security Problems
- Program Maintenance Problem

Data Management

Originally data was managed pure manually on papers, drafts, receipts, vouchers, registers, sheets, tables, etc.

Files, Folders, Drawers, Cabinets Shelves and even division of Rooms was used to maintain data hierarchy and management.

e.g. Library Books Data Organization and Management via Shelves, Ranks, Floors, Catalogs and Registers.

Data Management

Next computers were used for record keeping and data management

When Computers involved for record keeping and data management the two approaches were introduced;

- Initially ***File Processing Approach*** was used.
- Next ***Computer Database Approach*** was introduced.

Data Management

Note:

computer database approach is simply called ***database approach*** and it provided a more natural form of data management in terms of rows and columns.

It stores the data electronically in computer more likely in the form as it is presented in ***real world*** on papers, receipts, vouchers, registers or sheets.

Data Management

Thus in the perspective of this subject we conclude the three main methods of data management as follows;

- **Pure Manual Approach**
- **File Processing Approach**
- **Database Approach**

Database

A database is an organized collection of related data, that is stored electronically in computer in an efficient and compact manner. (Ref. to Definitions)

The term database was introduced in 1960's when it was possible to store data in disks.

In database data is organized into Rows and Columns

Database

Rows *are called* Records

Columns *are called* Fields

Examples:

- Phone Directory
- Library
- Accounts
- College Records
- Business Records
- Govt. Records
- etc

Database

Two early data models for database were;

- Flat Data Model

- Network Data Model

- Hierarchical Data Model

Recent are;

- Relational

- Object-Relational

- Semantic

- NoSQL

Database Management System

A database management system (**DBMS**) is mega system software for creating and managing databases.

The DBMS software is a *main component* of **Database Approach** of data management.

In Database Approach the DBMS software provides all necessary services to manage the real world data efficiently.

Database Management System

The DBMS provides facilities to create tables, rows / records and columns / fields.

The DBMS provides users and programmers / designers with a systematic way to create, use, store, retrieve, update and manage the data of real world.

Note: *one should know the difference between a complete database environment and the DBMS software*

Database Management System

The DBMS provides facilities to create tables, rows/records and columns/fields.

The DBMS provides **users and programmers / designers** with a systematic way to create, use, store, retrieve, update and manage the data of real world.

Database Management System

Some top ranking Commercial DBMS software tools;

- **Oracle RDBMS:**
- **IBM DB2:**
- **Microsoft SQL Server:**
- **MS Access**
- **Teradata:**
- **SQLite:**
- **MySQL:**

Database Management System

Some top ranking Commercial DBMS software tools;

- **phpMyAdmin:**
- **MongoDB:**
- **Hadoop HDFS:**
- **SAP Sybase ASE:**
- **SQL Developer:**
- **ADABAS:**
- **FileMaker:**

Database Environment

A database environment is a collective system of components that provides a complete environment of efficient usage of data, its management.

It provides facilities to efficiently develop database applications, efficient usage of the database, ensures its maintenance and security and also supports its authorization, to maintain user policies regulations.

Database Environment

It comprise and regulates the group of data, management, and use of data which consist of software, hardware, people, techniques of handling **database** and the **data** to be stored and retrieved from it.

Components of Database Environment

- **Commercial DBMS software:**
- **Central Repository:** (provided by DBMS software)
- **Created Database** (for some real world):
- **Developed Application Programs:**
- **Efficient User Interfaces:**
- **Database Designers & System Analysts:**
- **Application Programmers:**
- **End Users** (after creation, used in organization)
- **Database Administrators**

Advantages of Database Approach

Redundancy Control:

Data Consistency (if address changed no need to change it at multiple locations.)

Consistency / Integrity Constraints (data insertion updating and deletion (integrity) rules can be maintained easily)

Data Atomicity

Data Security

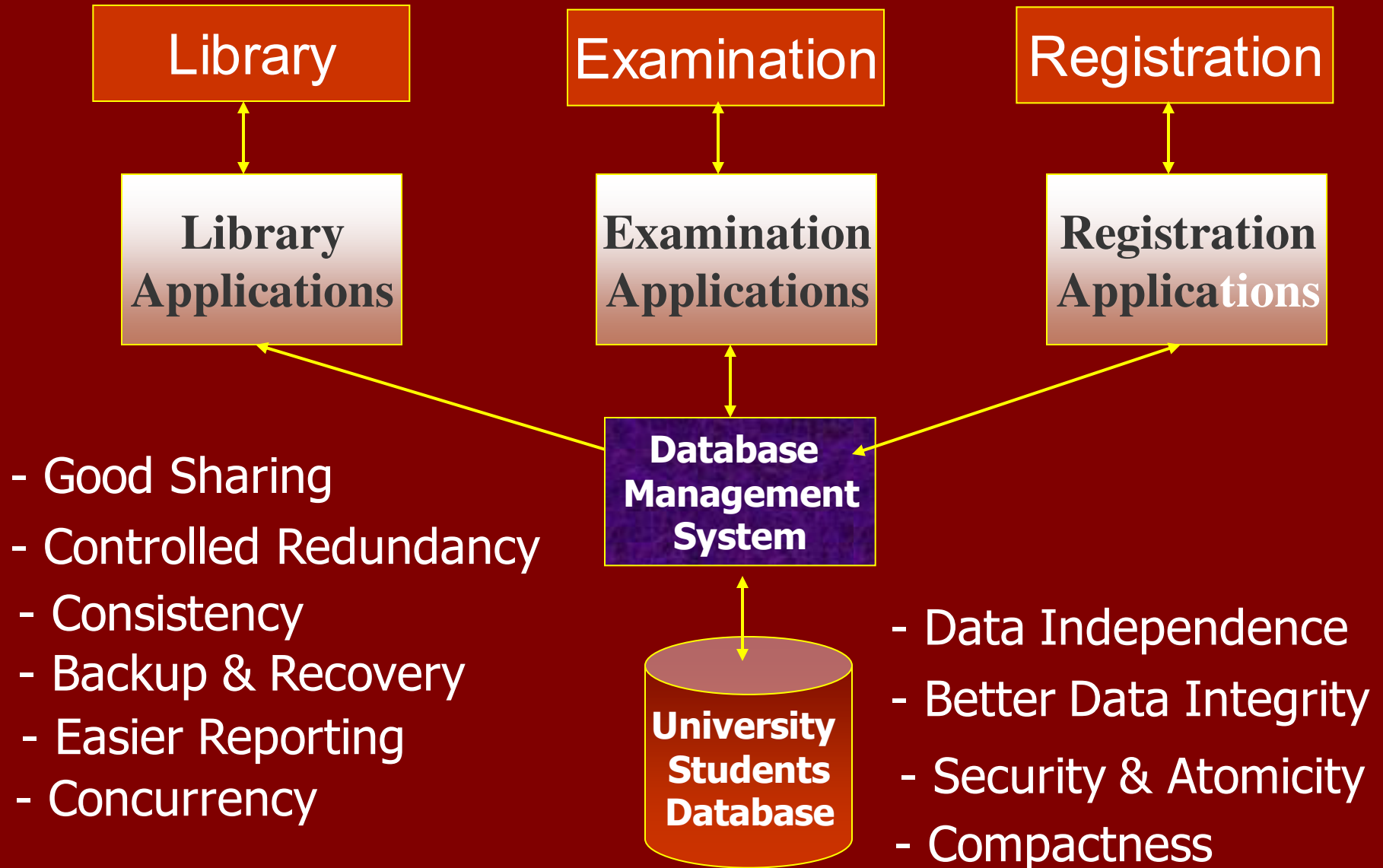
Reduced Development Time

Compactness (storage compactness)

Advantages of Database Approach

- Easier Reporting
- Data Sharing
- Increased Concurrency
- Improved Backup & Recovery
- Data Independence (from application programs)

Advantages of Database Approach



Disadvantages of Database Approach

- **DBMS Cost**
- **Hardware Cost**
- **Designing Cost**
- **Programming Cost**
- **Conversion Cost**
- **Failure Fear**
- **DMBS Performance & Complexity Issues**

Data As Resource

Organization:

An organized group of people with a particular purpose. Any business (doesn't matter whether it is a profit or non profit) organization which have (its business) objectives and also have some resources to achieve its objectives.

Data As Resource

Resource:

Any asset that is of value to an *organization* and that incurs cost. Data is the major or super resource of any organization. All the other resources are described and maintained in the form of data.

Data As Resource

Management:

All the Resources of an organization are required to be fully organized and managed. They must be recorded/stored correctly and are maintained updated.

Database approach has introduced many other parameters of management like sharing, consistency, integrity(correctness), etc.

Levels / Forms of Data

- Real-world Actual data presented within the organization.
- Metadata, structure about the data and database defined to store data in DBMS
- Data Occurrence or insertion of data, by the users of the organization in the created Database.

End of Lecture

Database Management Systems

Lecture -3

Lecture Overview

- Application Programs
- More advantages of database approach
- Disadvantages
- Data, its levels
- Database users
- Some common terms

Application Programs

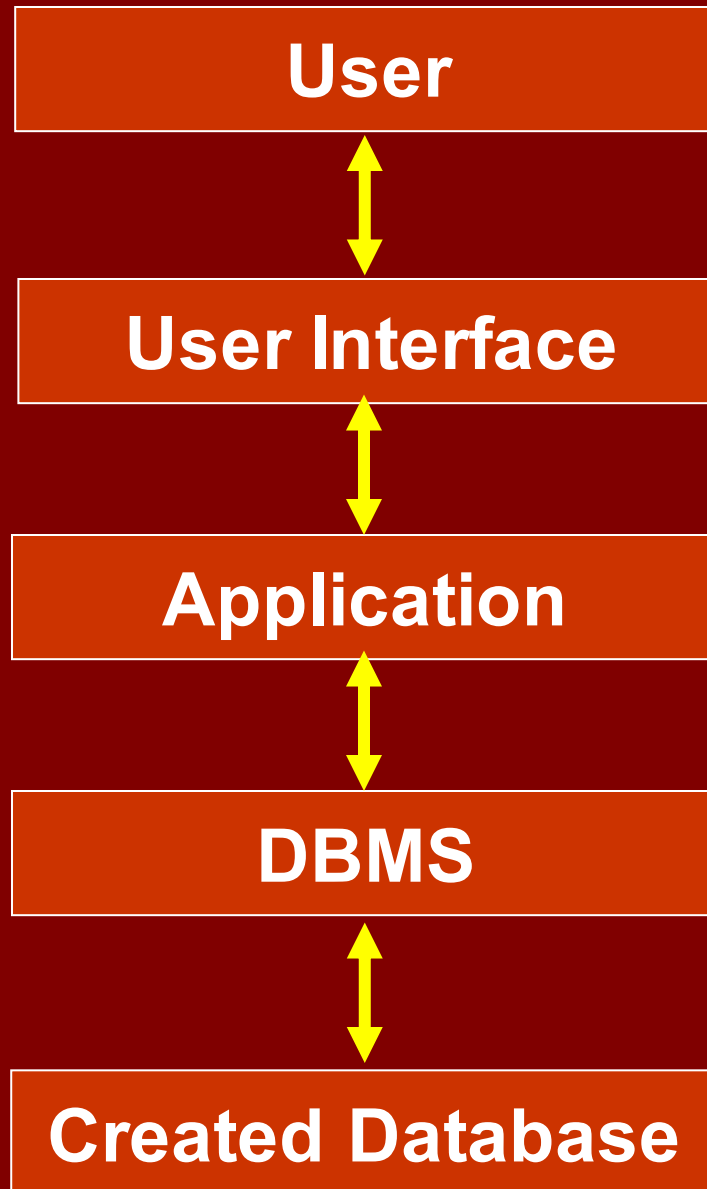
The application program provides the user-interface to send requests to database management system and to receive processed results from database management system. The database management system processed the requests and returns the results to the application program and also controls and manages the database.

Application Programs

A Database application is developed by using both, application program and DBMS. Application programs provides user interface for commutating with DBMS

Application Program is called Front-End
Database is known as Back-end

Application Programs



Range of Database Applications

- **Personal Computer Database**
- **Workgroup Database**
- **Department Database**
- **Enterprise Database**

Types of Users

- **Application Programmers**
- **End Users**
 - **Naïve Users**
 - **Sophisticated Users**
- **Database Administrator**

History of Database Systems

- **1960's**
 - Network Model
 - Hierarchical Model
 - Low level details were required to use them
- **1970-72**
 - Relational Model by E.F Codd
- **1974-77 (two main models developed)**
 - Ingres.
 - Led to Ingres Corp, Sybase. MS SQL Server, etc
 - System R.
 - Led to IBM SQL/DS& DS2, Oracle, etc

History of Database Systems

- **1976**
 - ER Model by P.Chen
- **1980's**
 - Commercialization of Relational Data model
- **Mid 1980's**
 - SQL became standard, RIM, RBASE 5000, PARADOX, OS/2 Database Manager, Dbase III, Visual FoxPro, Wetcom SQL

History of Database Systems

- **Early 1990's**
 - Most work was on Front-end
 - **PowerBuilder, Oracle Developer, VB...etc**
 - **OODBMS**
- **Mid 1990's**
 - Concept of Web DB
- **Late 1990's**
 - ASP, JSP, FrontPage, JavaServlets, JDBC, Coldfusion, Dreamweaver, Apache, OLTP,
 - OLAP, etc

History of Database Systems

- **Early 21 Century**
 - Concept of DB on PDA's, Data warehousing,
 - IBM, Microsoft and Oracle have captured the market
- **Future Trends**
 - Concept of Big Data, Terabyte Systems, ,
Huge Science Database:
 - **genome project, geological, national security, space app, data mining, data marts, XML with Java, Mobile Databases, Distributed Database, etc ...**

End of Lecture

Database Management Systems

Lecture 4

Database Architecture

Three Level Architecture

Three Level Architecture

A basis for understanding DBMS functionalities, as it is divided into three levels.

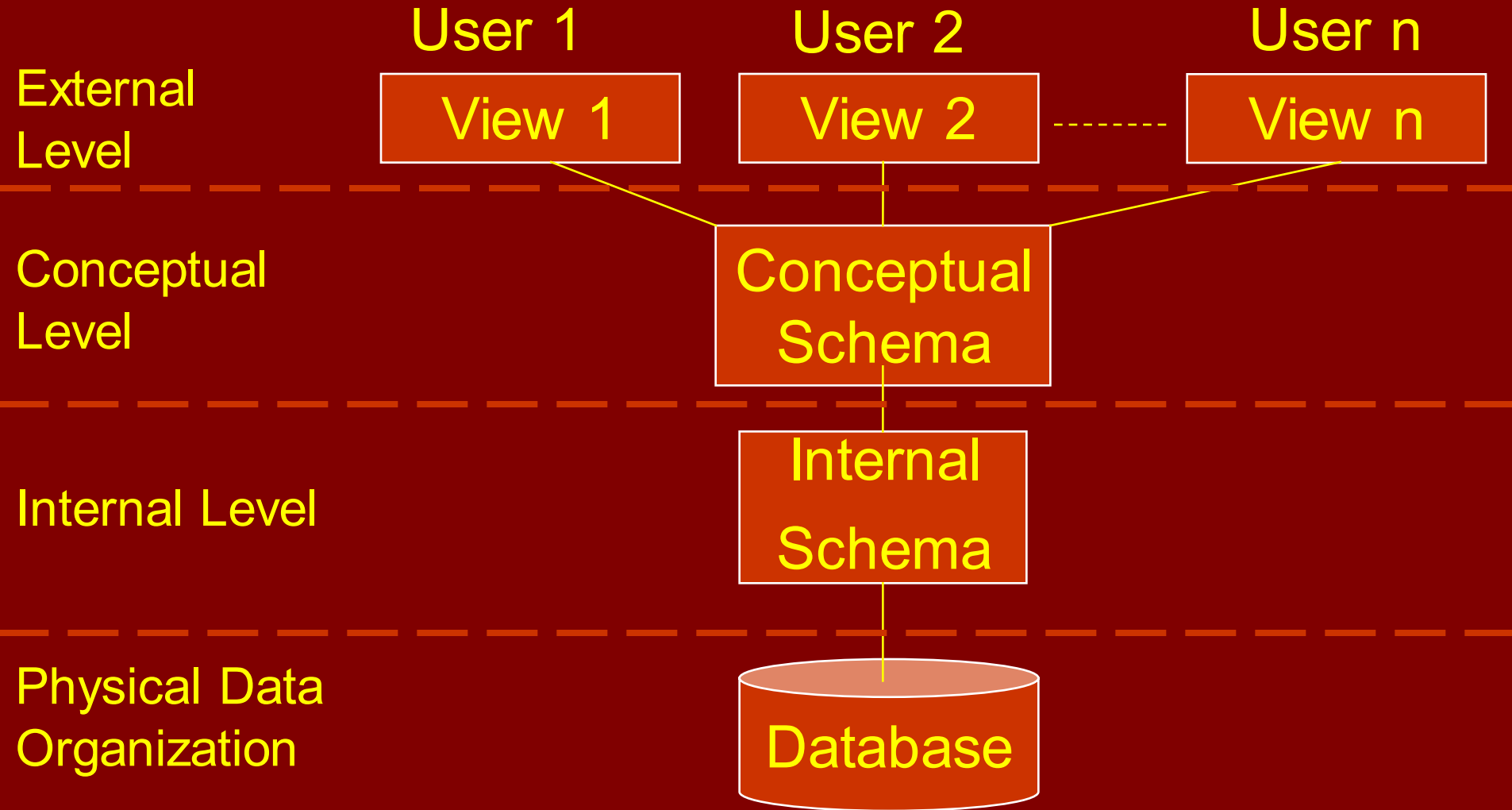
Depicted by three schemas or three models

Refers to permanent structure or *intention* or **Schema** of database. Where incoming user data is called **Instance** or **Extension** to the Structure.

Objective

- Separate users view from the Logical and physical representation
- Why?
 - Different views of same data
 - Consolidated representation
 - Data Independence

The Three-Level Architecture



External View

Each user has a view of the database limited to the appropriate portion of the user's *perspective of reality*.

External View

Users may have *different* views of the same data e.g. date, time etc.

External View

Virtual/calculated data:

that is not actually stored in the database but is created when needed e.g. age, Averages, or other statistical data etc.

External View

DBMS uses external views to create user interfaces (**Data entry Forms**) for different users which is both the *facility and barrier*

External View

User's external view is created after considering data access, reports, and the transactions needs.

External View

External schema evolves as user needs are modified over time

Employee Data

First Name: Rana
Last Name: Aslam
Date of Birth:
12 Sep, 1985

Aisha



Workers

Name: R. Aslam
Age: 33y
Dept: Sales




Saleem

External Layer

Lower Layers

External View

 **Programs**

Program S#

(New)

Program Code

Program Title

Save

External View



Subjects

S#

(New)

Subject
Code

Subject
Title

Save

External View



Teacher

Serial No

1

Teacher
Name

Mr. Dost Muhammad Khan

Teacher Emp
Code

111

Qualification

PHD

Joining Date

01-10-1995

Save

External View

New Class Form

Class ID

Program

Semster

Session

Subject

Teacher

Save

Level 2

Logical or Conceptual View

A complete description of the information content of the database according to the real world or of organization

Conceptual Schema

The entire information structure of the database, as seen by the DBA.

The community view of data

Conceptual Schema

All entities, attributes and their relationships are tried to be represented here as they exist in real world actually.

Conceptual Schema

Contains Tables, Fields, record types representing entities, data item types with their attributes, relationships and constraints on data.

Conceptual Schema

Contains Semantic information about the data meaning, security and integrity rules.

Conceptual Schema

Relatively constant:

designed with the present as well as future
needs of an organization

Employee Data

First Name: Rana
Last Name: Aslam
Date of Birth:
12 Sep, 1985

Aisha



Saleem

Workers

Name: R. Aslam
Age: 33y
Dept: Sales

External Layer

Logical Record Interface

Conceptual Layer

Name

DoB

Grade

DepId

Rana Aslam

12/09/85

17

D001

Marya Wasti

29/02/86

18

D005

Conceptual Schema

Program Table

[illegible]

Conceptual Schema

Subject Table

[illegible]

Conceptual Schema

Teacher Table

Serial No	Teacher Name	Emp Code	Qualification
1	Mr. Dost Muhammad Khan	111	PHD
2	Miss Afsah Imtiaz	112	Mphill
3	Mr. Zahid Akhtar	95	PHD
4	Mr. Ali Nawaz	192	Mphill
5	Mr. Zeshan Jhandir	101	PHD
6	Mr. Waheed Anwar	135	Mphill
*(New)			

Conceptual Schema

New Class Table

ID	Program	Semester	Session	Subject	Teacher
1	BSCS	4rth	1721	CSIT-01401	Database Systems
2	BSCS	4th	1721	CSIT-01402	Software Engineering - I
3	BSCS	4th	1721	CSIT-01403	Linear Algebra
4	BSCS	4th	1721	CSIT-01404	Digital Logic & Design
5	BSCS	4th	1721	CSIT-01405	Data Communication
6	BSCS	4th	1721	CSIT-01406	Web Programming
7	BSCS	2nd	1822	CSIT-01201	Programming Fundamentals
8	BSCS	3rd	1822	CSIT-01301	Object Oriented Programming
* #####					

End of Lecture

Database Management Systems

Lecture 6

Functions of DBMS

Some important functions of DBMS are as follows:

1. Data Processing
2. User-Accessible Catalog
3. Transaction Support
4. Concurrency Support
5. Recovery Support
6. Data Communication Support
7. Integrity Services
8. Authorization Services

Database Development Process

The development of complete database application is lengthy and complicated process. Different strategies can be used to develop database applications which are as follows:

1. Top-Down Development
2. Bottom-up Development

System Development Life Cycle

System development life cycle is a conventional way to develop an information system. It consists of many steps and involves different persons.

The steps of SDLC are as follows:

1. Preliminary Investigation / Planning
2. Requirement Analysis
3. System Design
4. Software Development

System Development Life Cycle

- 5. System Testing
- 6. System Implementation
- 7. System Maintenance

System Development Life Cycle



Staged Database Design Approach

Another way to design an information system is known as staged as data database design approach.

Steps in this approach as follows:

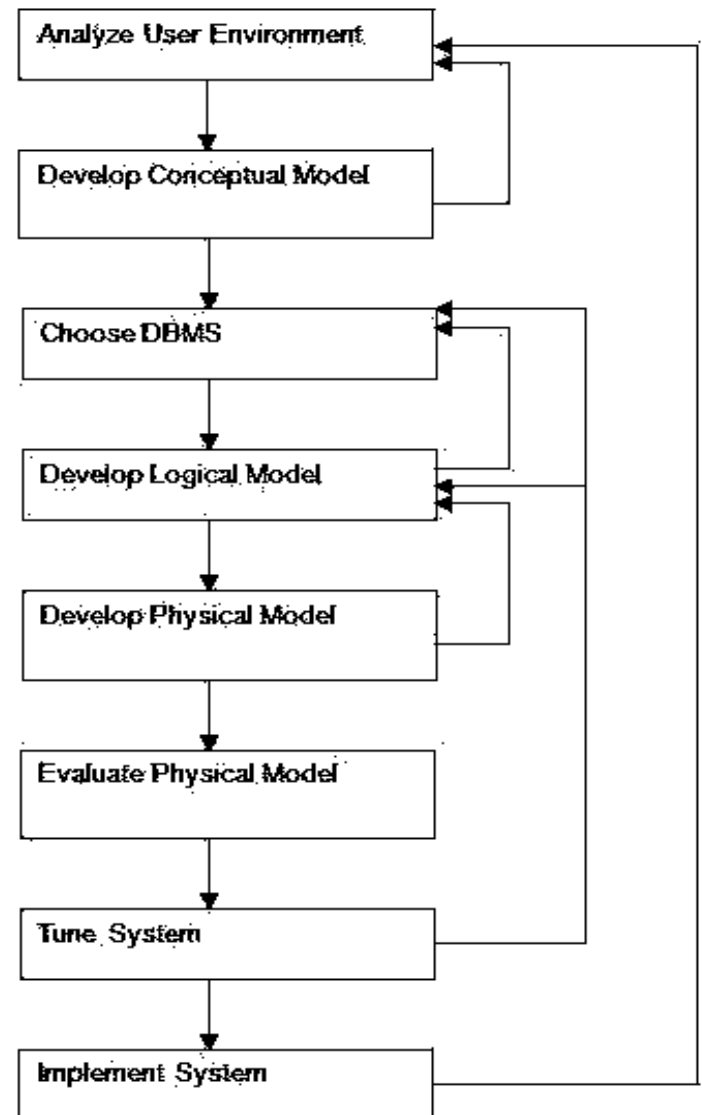
1. Analyze User Environment
2. Develop Logical/Conceptual Data Model
3. Choose a DBMS
4. Map Logical Model to DBMS
5. Develop Physical Model

Staged Database Design Approach

6. Evaluate Physical Model
7. Perform Tuning
8. Implement Physical Model and Maintenance

Staged Database Design Approach

- Analyze user environment
 - Develop conceptual data model
 - Choose a DBMS
 - Develop logical model, by mapping conceptual model to DBMS
 - Develop physical model
 - Evaluate physical model
 - Perform tuning, if indicated
 - Implement physical model
- See **Figure 2.3** – note loops



Design Tools

- Design tools used to describe the design process in a standard way. A standard tool is important because it provides standard notation for designing specific systems.
- If there is no standard tool, everyone may use different design notations that can be difficult to understand for others.

Data Flow Diagram

- The most commonly used tool to design database system is data flow diagram (DFD).
- A data flow diagram shows the flow of data through an organization
- It is used to design systems graphically.
- DFD is very simple and it hides complexities of the system.

Limitations of DFDs

DFDs have the following limitations:

1. They do not provide us a way of expressing decision points.
2. DFDs are focused on flow of information only.

Symbols in DFD

Symbols in DFDs

DFDs uses the following symbols:

1. Data Flow:

The purpose of the dataflow in a DFD is to express the flow of information from one entity to another entity in the system.

Some Rules for Data Flows

- Data in motion, moving from one place to another in the system
 - From external entity (source) to system
 - From system to external entity (sink)
 - From internal symbol to internal symbol, but always either start or end at a process

Data Flow Symbol



Symbols in DFDs

2. Data Store:

Data store is a repository for the storage of the data. When in a system the data is to be permanently stored somewhere for future reference or use the data store is used for this purpose.

Some Rules for Data Stores

- Internal to the system
- Data at rest
- Include in system if the system processes *transform* the data
 - Store, Add, Delete, Update
- Every data store on DFD should correspond to an entity on an ERD
- Data stores can come in many forms:
 - Hanging file folders
 - Computer-based files
 - Notebooks

D1

Data Stores

Symbols in DFDs

3. Process:

Processes are expressed with ovals or rounded rectangles. Processes are used to express the transformation of incoming dataflow into outgoing dataflow.

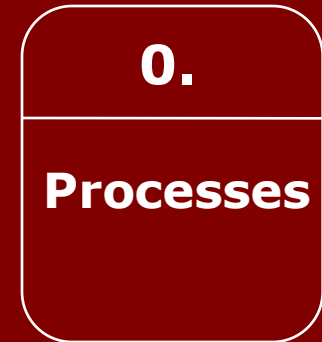
Some Rules for Processes

- Always internal to system
- Law of conservation of data:

#1: Data stays at rest unless moved by a process.

#2: Processes cannot consume or create data

- Must have at least 1 input data flow (to avoid miracles)
- Must have at least 1 output data flow (to avoid black holes)
- Should have sufficient inputs to create outputs (to avoid gray holes)



Processes

- Logical process models omit any processes that do nothing more than move or route data, thus leaving the data unchanged. Valid processes include those that:
 - Perform computations (e.g., calculate grade point average)
 - Make decisions (determine availability of ordered products)
 - Sort, filter or otherwise summarize data (identify overdue invoices)

Processes

- Logical process models omit any processes that do nothing more than move or route data, thus leaving the data unchanged. Valid processes include those that:
 - Organize data into useful information (e.g., generate a report or answer a question)
 - Trigger other processes (e.g., turn on the furnace or instruct a robot)
 - Use stored data (create, read, update or delete a record)

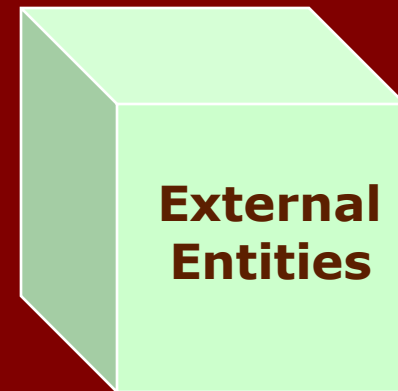
Symbols in DFDs

4. External Entities:

These are the entities interacting with the system in any of two different ways. They may be either receiving the data from the system, or may be producing the data for the system to consume.

Some Rules for External Entities

- External people, systems and data stores
- Reside outside the system, but interact with system
- Either a) receive info from system, b) trigger system into motion, or c) provide new information to system
- e.g. Customers, managers
- Not clerks or other staff who simply move data



Symbols in DFDs

6. Separator:

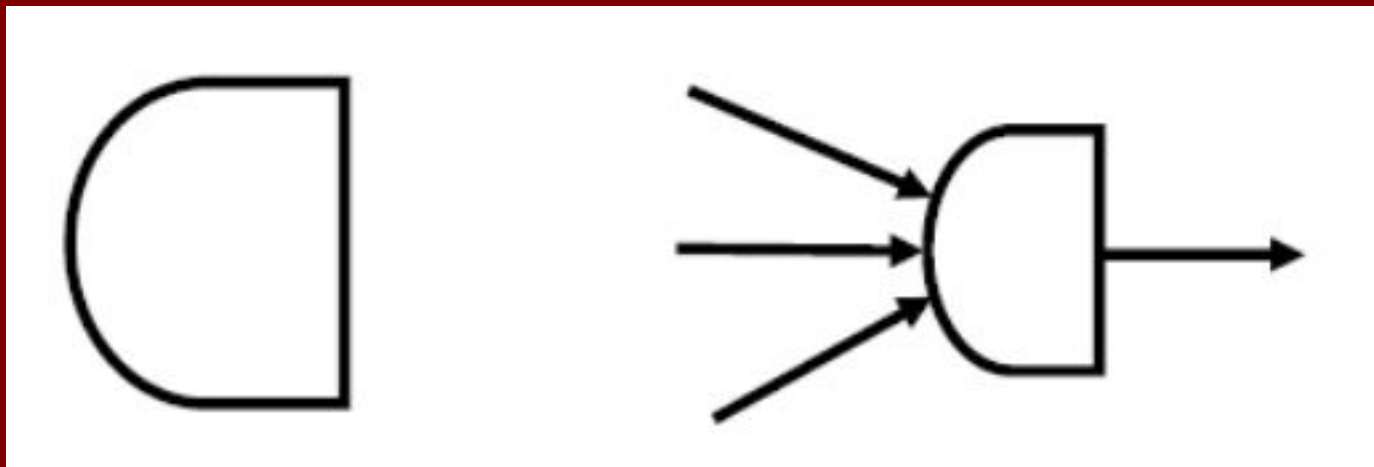
The dataflow symbol which is used for separating data from a single source to multiple sinks is known as a separator.



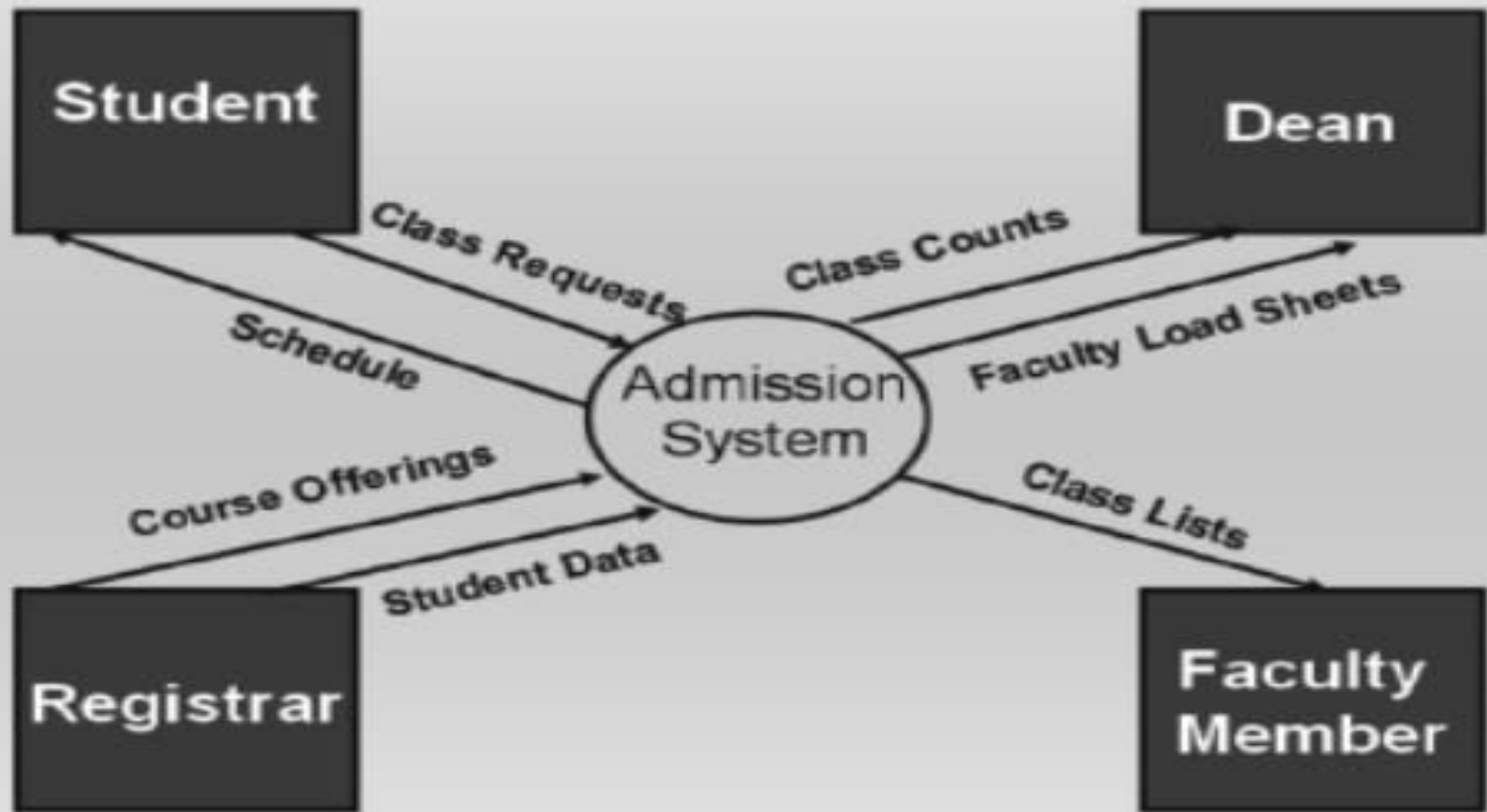
Symbols in DFDs

5. Collector:

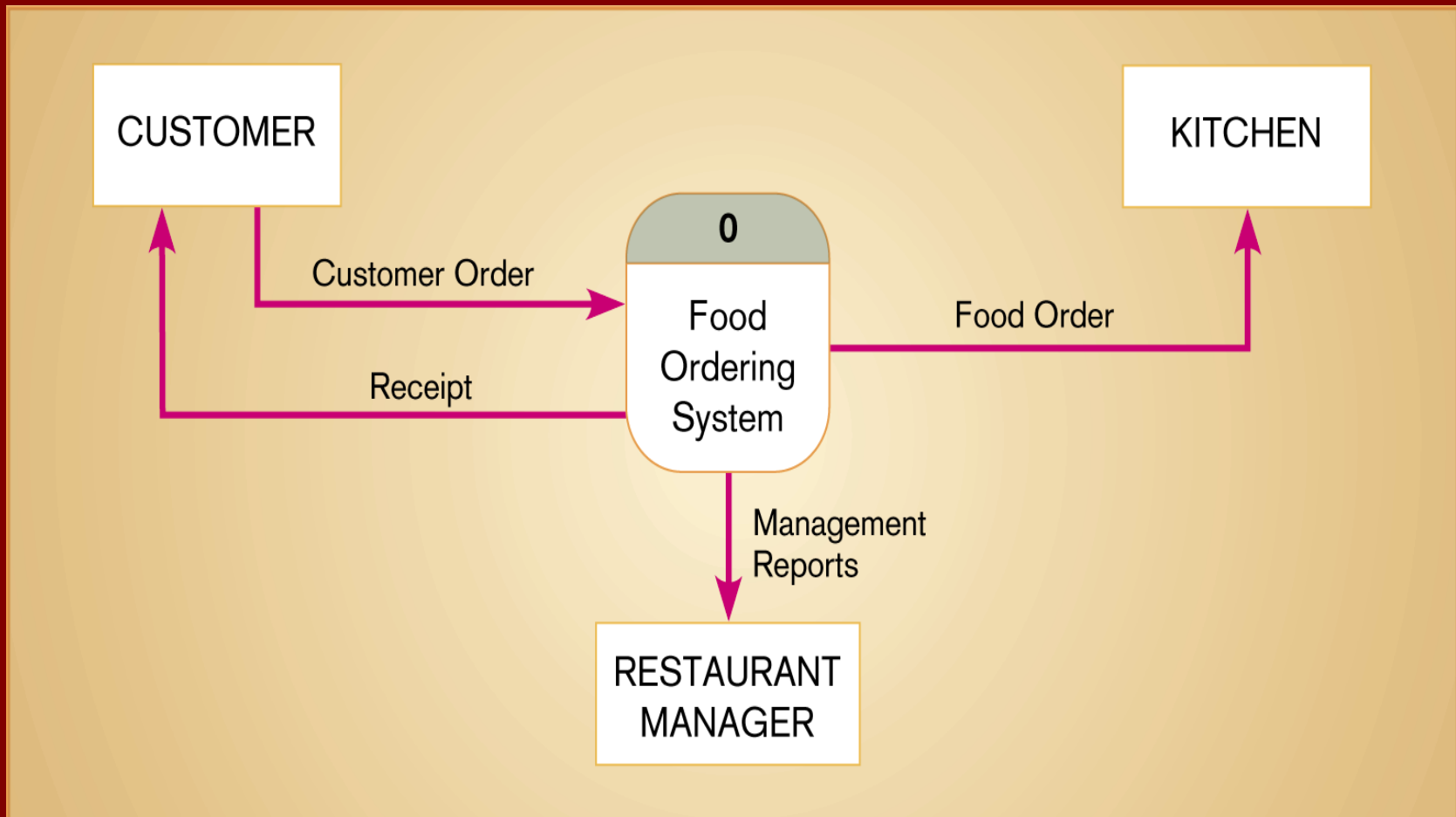
This DFD shape is used to express several dataflow connections terminating at a single location. Collector is used to show the convergence of data to a single point.



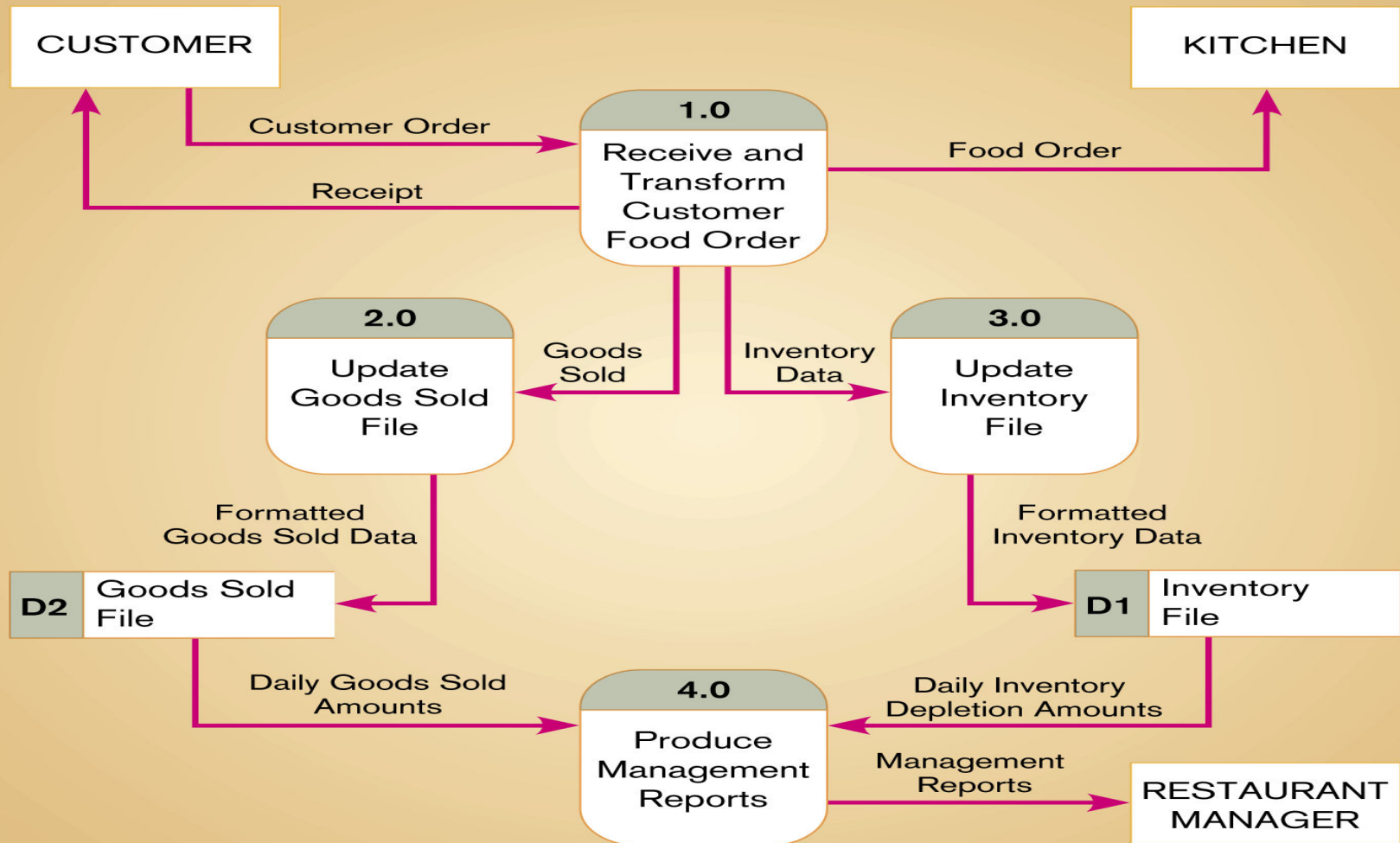
Context diagram of Food ordering system – Level 0



Context diagram of Food ordering system – Level 0



Context diagram of Food ordering system – Level 1



End of Lecture

Database Management Systems

Lecture 7

Database Administrator (DBA)

- DBA is an important person in the development of any information system. He is responsible for design, operation and management of database.
- He must be technically competent, good manager and skilled diplomat.

Functions of DBA

1. Preliminary Database Planning
2. Identifying User Requirements
3. Developing & Maintaining Data Dictionary
4. Designing Logical Model
5. Choosing a DBMS
6. Developing Physical Model
7. Creating & Loading Database
8. Developing User Views

Functions of DBA

- 9. Writing & Maintaining Documentation
- 10. Developing & Enforcing Data Standard
- 11. Developing Operating Procedures
- 12. Training the Users
- 13. Helping Database Users
- 14. Defining Backup and Recovery Procedures
- 15. Monitoring Performance
- 16. Tuning & Reorganizing

Data Administrator

- The need of data administrator arises in a very large organization where many databases may exist.
- Data administrator is responsible for the whole information resource.
- He develops the requirements for database, develops logical design and other non-technical functions.

Data Dictionary

- Data dictionary is a repository of information that describes the logical structures of the database.
- It contains record types, data item types and data aggregates etc.
- Data dictionary contains metadata.

Uses of Data Dictionary

- Data dictionary contains Information about data, i.e. data about data.
- Data dictionary is used for communication with different users of the database.
- Keeps record of change in Database Structure, like creation, deletion or updating of elements.

Uses of Data Dictionary

- Determining the impact of change by analyzing its relationships with other items.
- It Keeps Record of all Access Control Information and authorized accesses.
- Its stored Information is also used later for system and organization's audit.

Types of Data Dictionary

Different types data dictionaries are as follows:

1. Integrated Data Dictionary
2. Freestanding Data Dictionary

Types of Data Dictionary

Integrated Data Dictionary: it is part of DBMS and it performs many functions throughout the life of the database not only during design phase but also during running

.

- 1. Active Data Dictionary:** which is checked for every access to the DBMS.
- 2. Passive Data Dictionary:** which is checked periodically.

Types of Data Dictionary

Freestanding Data Dictionary: it is not part of DBMS and is available by some third commercial product and mostly it is used till to design phase before the selection of a commercial DBMS.

Logical Database Design

- The logical database design contains the **definition of the data** to be stored in database and definition of the whole database **structure**.
- It also contains the **rules and information** about the structure and **type** of data.

Logical Database Design

- All entities, their attributes and their relationships are described in logical model.
- It defines the complete description of data to be stored in the database

Logical Database Design Process

- Represent Entities
- Represent Relationships
- Merge the Relations
- Normalize the Relations

Physical Database Design

- Physical design is the **last stage** of database design process.
- The major objective of physical database design is to implement the database as a set of **records, files, indexes** and other data structures.

Major Inputs to Physical Design

Three major **inputs** to Physical database design are as follows:

1. Logical Database Structure
2. User Processing Requirements
3. Characteristics of DBMS

Components of Physical Design

Different components of physical database design are as follows:

1. Data Volume and Usage Analysis
2. Data Distribution Strategy
3. File Organization
4. Indexes
5. Integrity Constraints

Components of Physical Design

Different components of physical database design are as follows:

1. Data Volume and Usage Analysis
2. Data Distribution Strategy

Components of Physical Design

1. Data Volume and Usage Analysis

How much data will be inserted? Its day-by-day analysis.

Analysis of Its pattern, storage devices and costs are estimated.

It also helps to plan file organization selection, indexing and data distribution strategies.

Components of Physical Design

2. Data Distribution Strategies:

i. Centralized

All data is placed at a single location or site

ii. Partitioned

Different sites are placed on different partitions

iii. Replicated

Full copy is replicated at some location

iv. Hybrid

Distributed at different sites

File Organization

File organization is a technique for physically arranging the records on secondary devices.

This technique organizes data carefully to support **fast access** to desired subsets of records.

File Organization

Thus **File organization** is a method of arranging the records in a **file** when the **file** is stored on disk. A **relation** is typically stored as a **file** of records.

File Organization

The system designer must recognize several constraints for selecting a file organization, like:

- **Physical characteristics of secondary storage devices**
- **Available operating systems and file management software**
- **User requirements for storing and accessing data**

File Organization

- Available operating systems and file management software
- User requirements for storing and accessing data

File Organization Methods

The files are organized on storage media in the following methods:

- **Sequential Files**
- **Direct or Random Files**
- **Indexed Sequential Files**

Indexes

- An index is a **table** that is used to determine the **location of rows** in a table.
- Indexes are used to **speed up** the sorting and searching process.

Indexes

- The **performance** of database is improved with these indexes.
- The index may be created on **primary key**, **secondary key** and **foreign key** etc.

Integrity Constraints

- **Data Integrity** is maintained with the help of integrity constraints.
- These constraints are the **rules** that are designed to keep data **consistent** and **correct**.

Integrity Constraints

- They act like a **check** on the **incoming data**.
- It is very important that a database maintains the quality of the data stored in it.
- **DBMS** provides several mechanisms to **enforce integrity** of the data.

End of Lecture

Database Management Systems

Lecture 8

Data Models

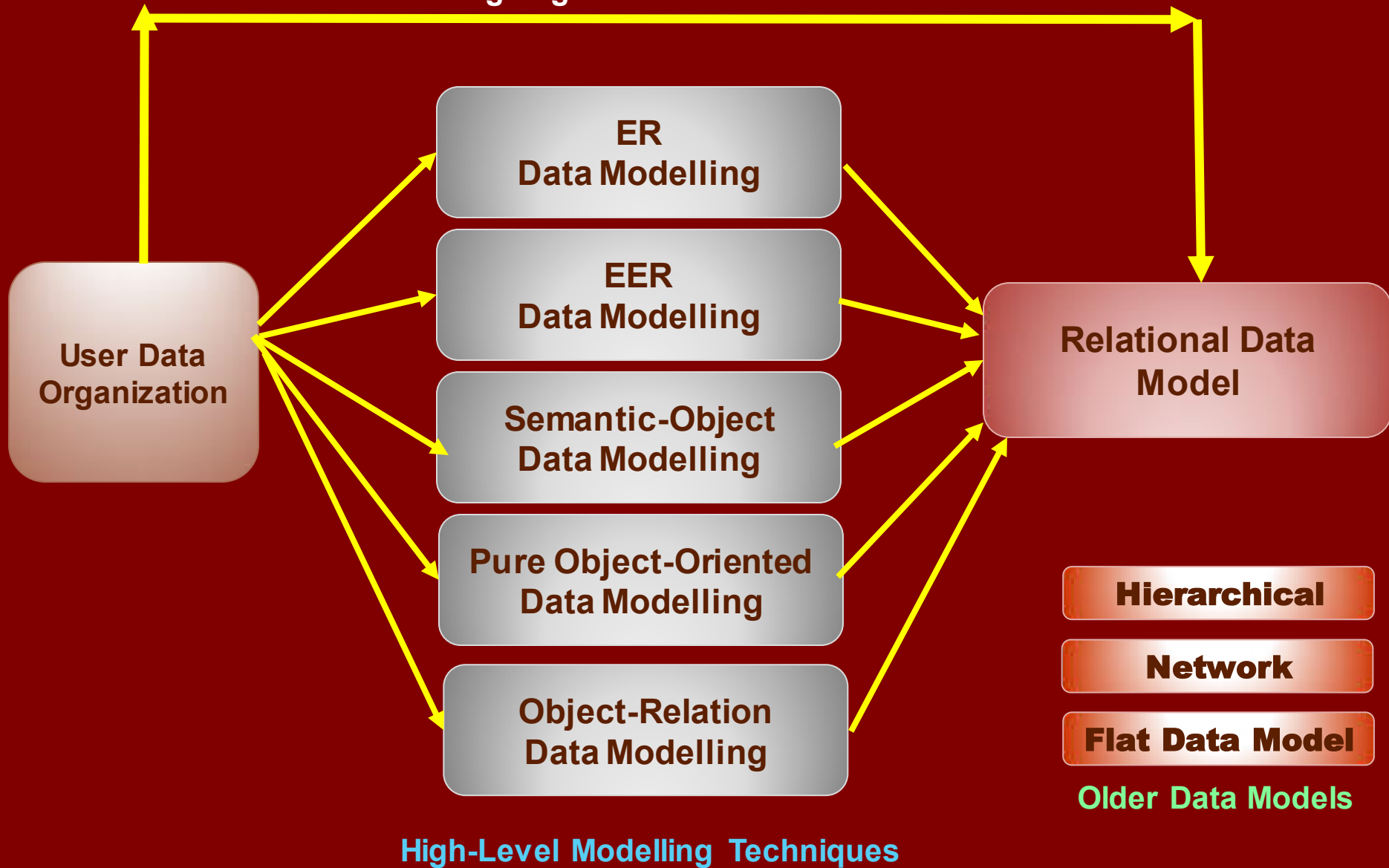
Data Models

Data model is a technique to present the a data management model of the enterprise's data, such that the model can make possible the efficient sharing of data between different stockholders in a organized way.

Initially it may be a paper representation that is then converted to computerized form (in database form).

Data Models

Direct designing in Relational is not suitable



ERM and ERD

Entity-Relationship Data Model (ERM)

is a detailed, logical representation of the data for an organization or for a business area.

Entity-Relationship Diagram (ERD)

is a graphical / diagramtic representation of a Entity-Relationship Model.

Entity Relationship Model

Entity Relationship Model

- Entity-Relationship model is a logical representation of data in an organization.
- It describes the entire system as a collection of **entities** (with their **attributes**) related to one another (i.e. **relationships**).

Entity Relationship Model

- It is used to describe the elements of a system and their relationships.
- It was introduced by Peter Chen in 1976.

Advantages of E-R Model

Some important advantages of E-R model are as follows:

1. Conceptual Simplicity
2. Visual Representation
3. Effective Communication
4. Integrated with Relational Database Model

Elements of E-R Model

Three major elements of an E-R model are as follows:

- 1. Entities**
- 2. Attributes**
- 3. Relationships**

Entities

Entity

What is an Entity?

An entity has its own identity that distinguishes it from other entities.

– Examples:

- **Person:** PROFESSOR, STUDENT
- **Place:** STORE, UNIVERSITY, CITY
- **Object:** MACHINE, BUILDING
- **Event:** SALE, REGISTRATION
- **Concept:** ACCOUNT, COURSE, EXAM

Entity

Naming Guidelines

Entity Name should be;

- A Singular Noun and in Capital letters.
- Descriptive and specific to the organization.
- Concise.

Entity

Defining Guidelines

Entity definition should;

- Include a *statement* of what the unique characteristics are for each instance of the entity.
- Make clear what *entity instances* are *included and not included*.
- Include a description of when an instance of the entity type is *created and deleted*.

Entity

Defining Guidelines

An Entity definition should;

- Specify when an instance *might change* into an instance of another entity type, e.g. **DOB to Age**
- Specify *what history / log record is to be kept about entity instances.*

Entity

Entity Type or just Entity

Entity type is a term that is used for an entity, it means that what type of entity you are defining.

Thus entity type means simply an Entity.

e.g.

STUDENT,
CUSTOMER

CLASS,

TEACHER,

Entity

Entity Instance

is a member of entity class or the single occurrence or value of an entity type.

e.g. All the individual students entered in a student table.



A diagram illustrating the concept of an entity instance. It features a blue rectangular box with the word "STUDENT" in yellow capital letters. A yellow arrow points downwards from the bottom center of this box to a red rectangular box containing a list of names in green text.

STUDENT

Ali, Mueed, Faiza, Uzair, Sajid, Khizra, Nabeel

Entity

Entity Set

The collection/set all entity instances/values of a particular entity type/entity is called entity set,

Example:

All of the students in a university of entity type STUDENT.

Attributes

Attributes

Characteristics of an **entity** are called its attributes

STUDENT

Student_ID,
Student_Address,
Student_Phone,
Student_Class,
.... etc.

Attributes

Naming Guidelines

An attribute name;

- Should be a *noun* and *capitalize the first letter of each word*. (Example: Student_ID.)
- Should be *unique*.
- Should follow a *standard format*.

Example: Student_GPA, not GPA_of_Student.

Attributes

Naming Guidelines

Similar attributes of different entity types should use similar conventions but unique names.

Example:

Faculty_Residence_City_Name

Student_Residence_City_Name

Attributes

Defining Guidelines

An attribute definition should;

- State **what** the attribute is ? and **why** it is important ?
- Make clear what is and isn't included in the **attribute's value**.
- Define any **aliases** (same attribute with different **names**)
- Indicate if this attribute is **really required** or not.
- Indicate any **relationships** with other attributes.

Attributes

Attribute Domain

The set of all possible values / instances that an attribute can take,

e.g. entity **PAKISTAN-CITY** can take any valid city of Pakistan as a value.

Relationships

Relationships

A relationship, in the context of a databases, is a **natural situation / logical connection** that exists between two entities.

The entities that participate in a relationship are called **participants**.

A relationships may be between different entities as well as to the entity itself.

Relationships

A relationship, is established on the basis of real or natural logical connection between the participant entities.

e.g.

STUDENT	and	TEACHER
STUDENT	and	STUDENT
SUPPLIER	and	SHIPPER

Relationships

A relationship may be **Total** (if all of the entities in the entity set participate in the relationship)

or

A relationship may be **Partial** (if some subset of the entity set participate into the relationship).

Entity Relationship Diagram

E-R Diagram

E-R Diagram is a graphical representation of E-R model using a set of standard symbols.

ER-diagram represents the previous three elements of ER-Model in more detail using different types of symbols / diagrams.

Each symbol/diagram is actually a description of an element, conveying or adding some particular meanings

Entities

**ERD Symbols Representing
Different Types of Entities**

ERD Symbol of An Entity

Entities should always be placed in a plane rectangle.



Types of Entities

Different types of entities are represented in E-R diagram as follows;

- 1. Strong Entities**
- 2. Weak Entities**
- 3. Associative Entities**

ERD Symbols For Types of Entity

Strong Entity

The Strong Entity is the one whose existence **does not depend** on the existence of any other entity in a schema.

A **strong entity** always has the **primary key** in its **set of attributes**.

ERD Symbols For Types of Entity

Strong Entity

It is denoted by a **single rectangle**, the same symbol is used for the entity itself.



Strong Entity

ERD Symbols For Types of Entity

Weak Entity

The existence of a **weak entity** depends on the existence of an other **entity** in the entity set of a database schema.

A weak entity does not have a **primary key** rather it must have a foreign key.

The **foreign key** will be the **primary key** of some other (**strong and related**) entity.

ERD Symbols For Types of Entity

Weak Entity

In a relational database, a **weak entity** cannot be uniquely identified by its **attributes** alone; therefore, it uses a **foreign key** in conjunction to some **primary key** of another related **entity**



ERD Symbols For Types of Entity

Associative Entity

This is an **additional entity** which come into existence where **many-to-many relationships** are build.

It is denoted by a cross rectangle within another rectangle



ERD Symbols For Types of Entity

Associative Entity

Relational database requires the implementation of a **separate base relation** (or base table) to resolve many-to-many relationships.

This **base relation** represents this entity and is called, informally, an **associative table**.



Attributes

**ERD Symbols Representing
Different Types of Attributes**

Types of Attributes

Different types of **Attributes** are represented in RED as follows;

1. **Simple Attribute**
2. **Composite Attribute**
3. **Single Valued Attribute**
4. **Multi-valued Attribute**
5. **Stored Attribute**
6. **Derived Attribute**
7. **Identifier Attribute**

End of Lecture

Database Management Systems

Lecture 9

Slides are not Prepared

3.4 – Degree of Relationships

3.5 – Subtype & Supertype

Relationships

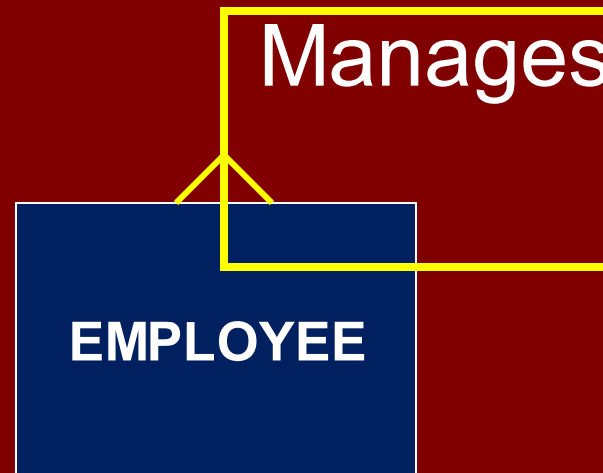
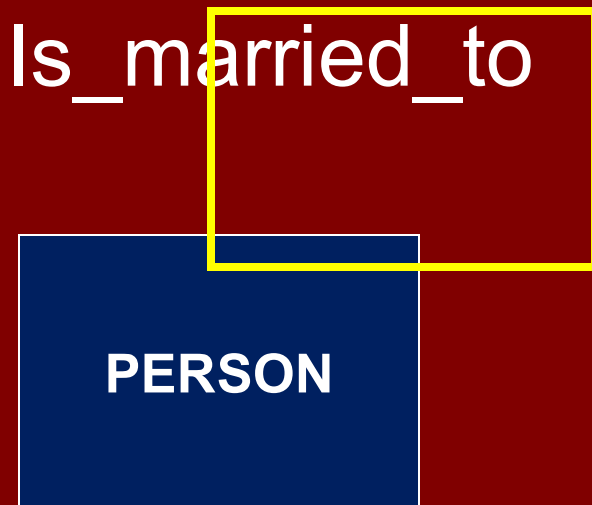
Degree of Relationships

A relationship may consist of many entities. The number of entities in a relationship is called degree of relationship. The types of relationships with respect to degree are as follows:

- **Unary Relationship:** only one entity is involved
- **Binary Relationship:** Two entities are involved
- **Ternary Relationship:** Three entities are involved

Unary Relationship

- Relationship between the instances of one entity type.



Binary Relationship

Binary Relationships:

one-to-one

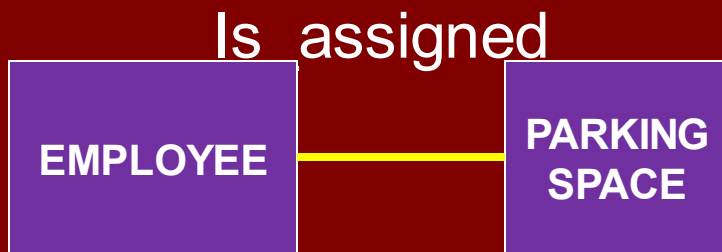
one-to-many

many-to-one

many-to-many

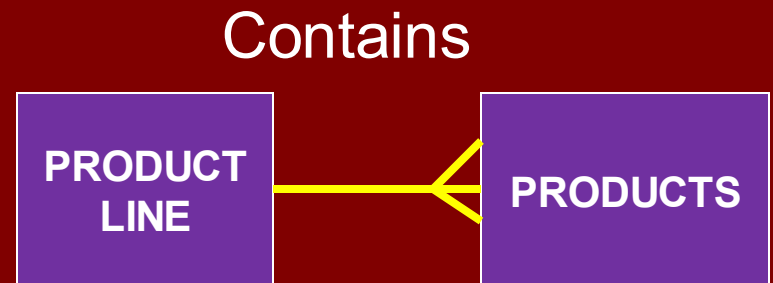
Binary Relationship

Relationship between the instances of two entity type.



One-to-One

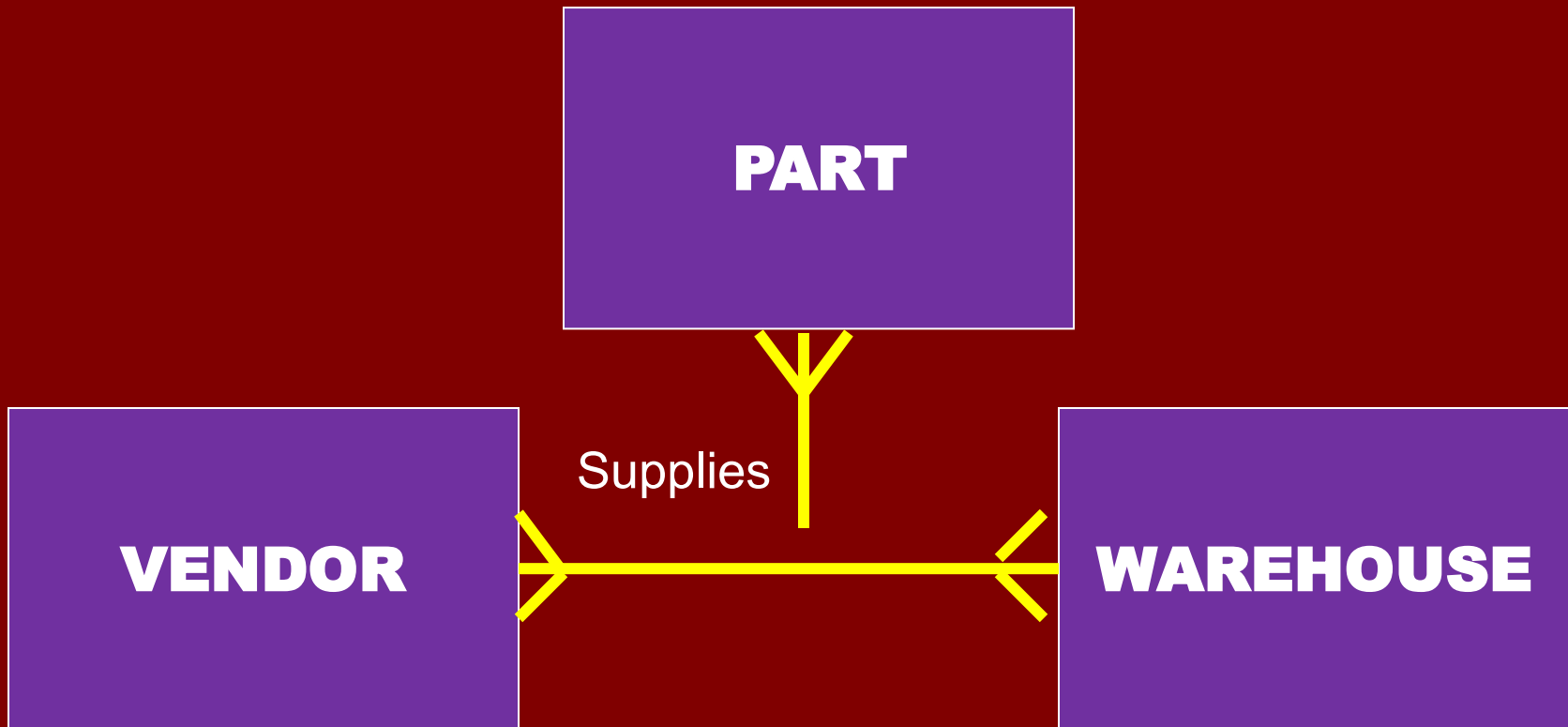
Can also have many to many!



One-to-Many

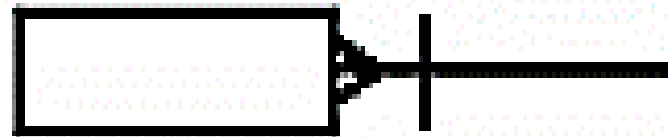
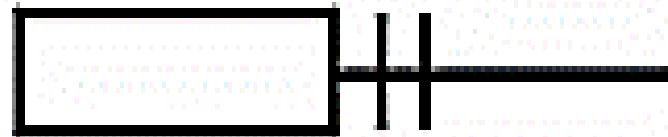
Ternary Relationship

A simultaneous relationship among instances of three entity types.



Relationship Cardinalities

- Optional one
- Mandatory one
- Optional many
- Mandatory many



Relationship Cardinalities

One-to-many (Optional):

STUDENT---BOOK (in library)

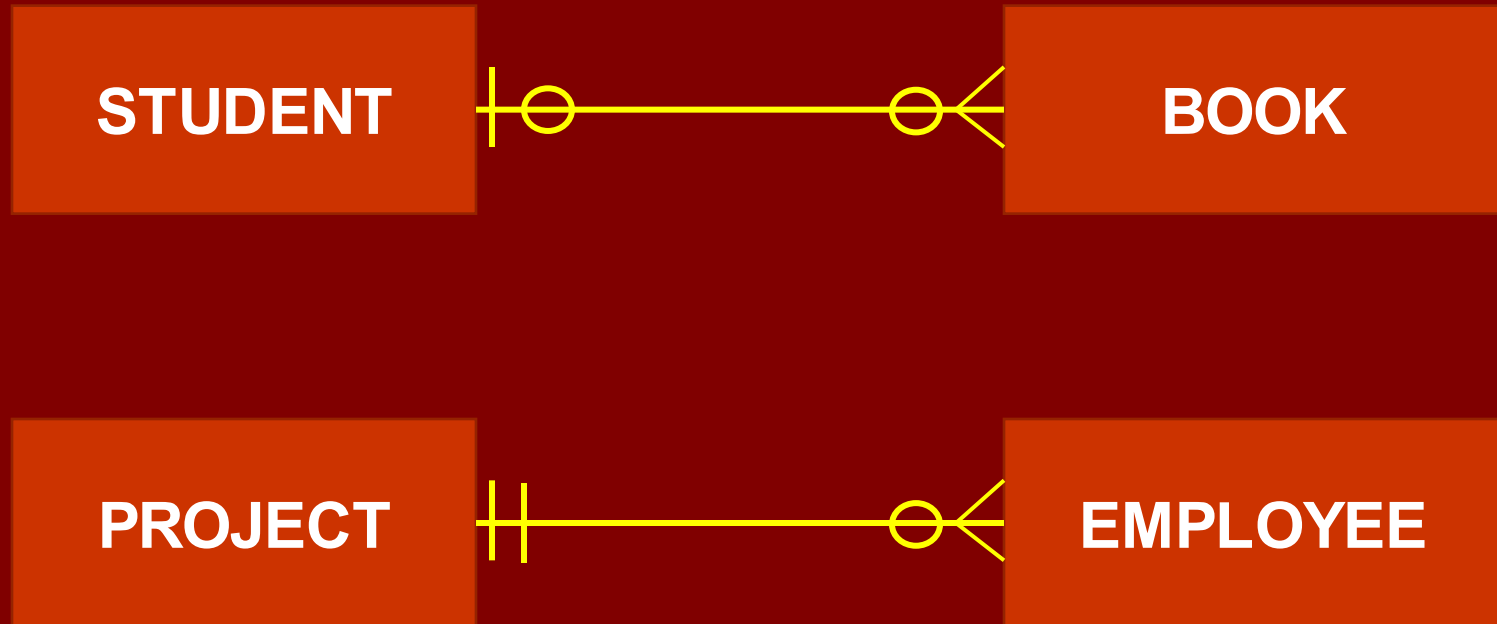
both are optional for each other

One-to-many (Mandatory):

PROJECT---EMPLOYEE

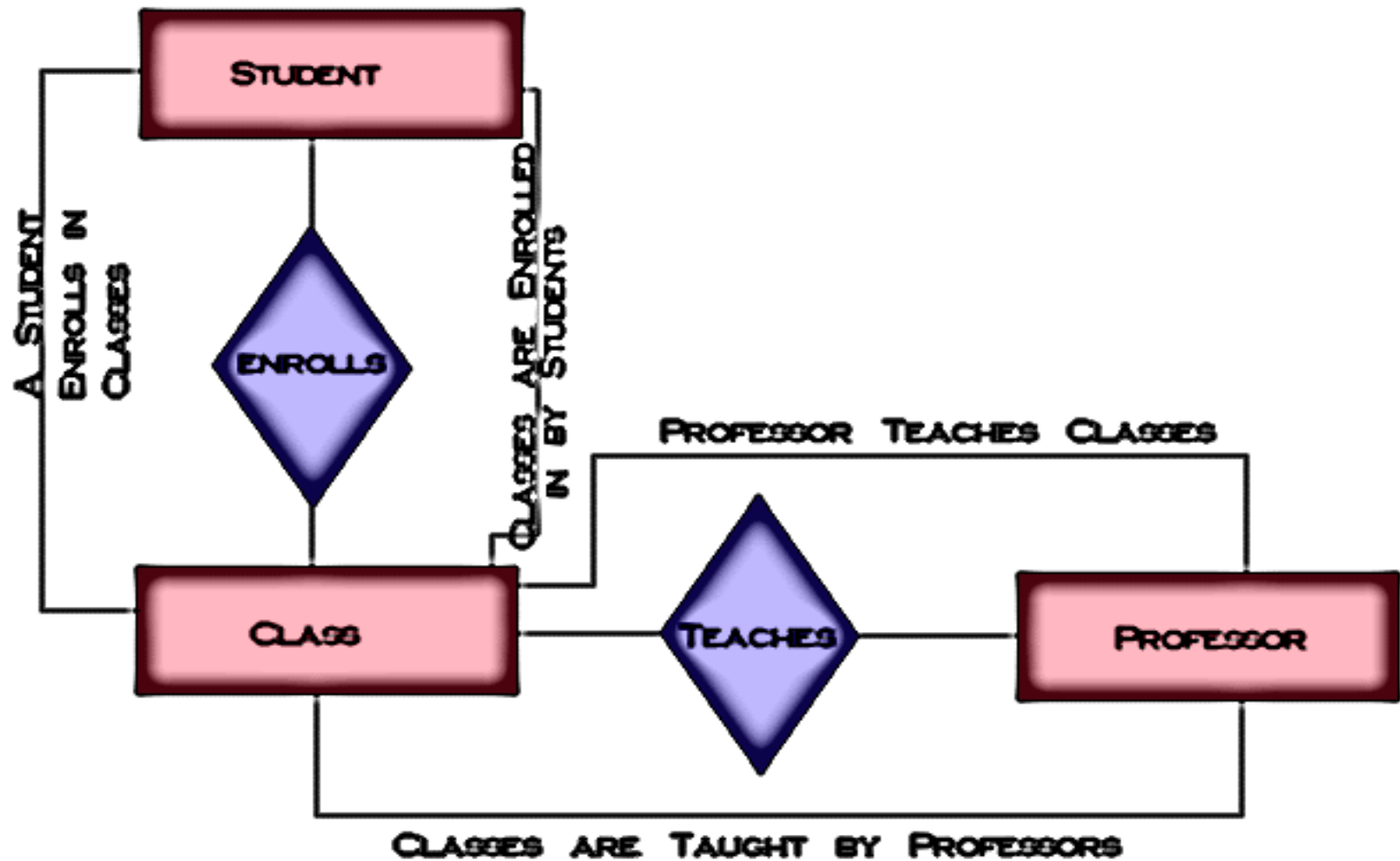
project must has some members but for employee it is optional

Relationship Cardinalities



End of Lecture

University ERD





Database Management Systems

Lecture 9

3.4 – Degree of Relationships

3.5 – Subtype & Supertype

Relationships

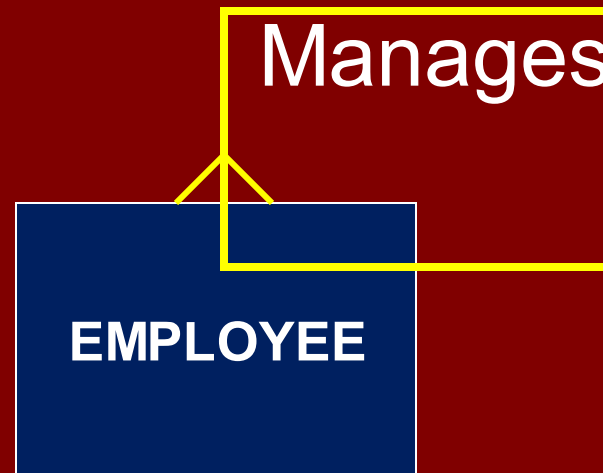
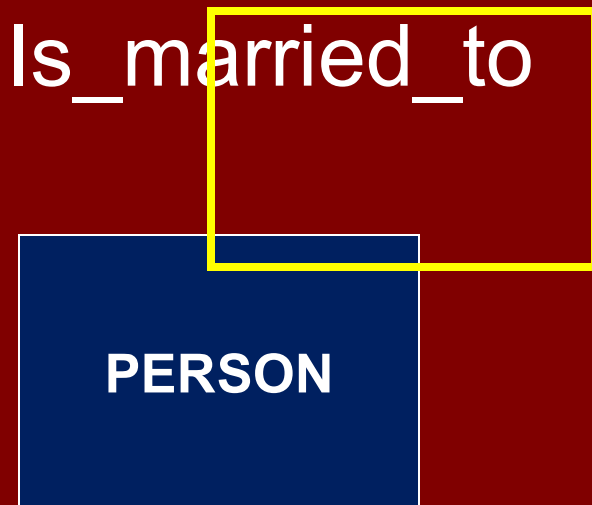
Degree of Relationships

A relationship may consist of many entities. The number of entities in a relationship is called degree of relationship. The types of relationships with respect to degree are as follows:

- **Unary Relationship:** only one entity is involved
- **Binary Relationship:** Two entities are involved
- **Ternary Relationship:** Three entities are involved

Unary Relationship

- Relationship between the instances of one entity type.



Binary Relationship

Binary Relationships:

one-to-one

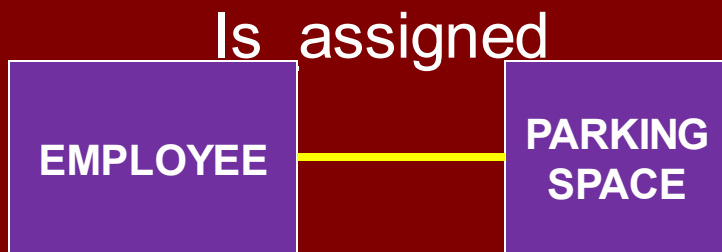
one-to-many

many-to-one

many-to-many

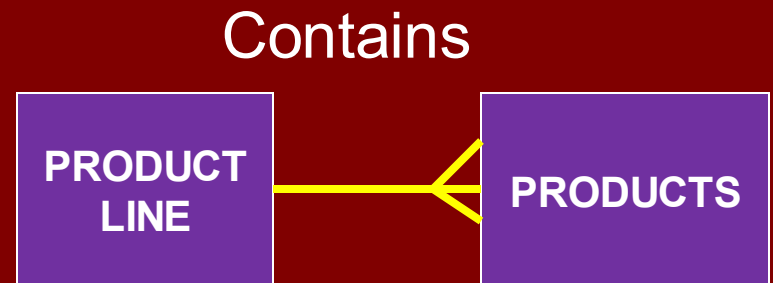
Binary Relationship

Relationship between the instances of two entity type.



One-to-One

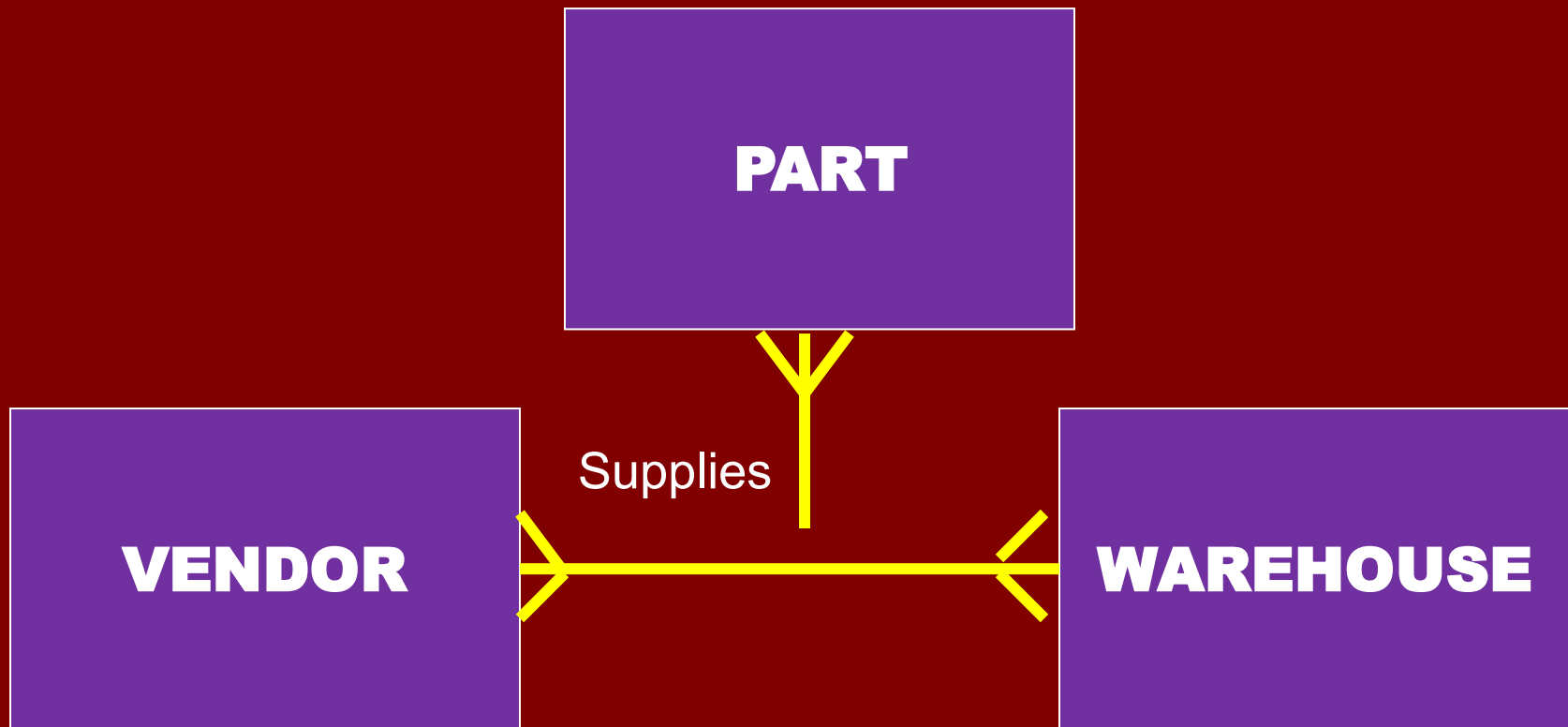
Can also have many to many!



One-to-Many

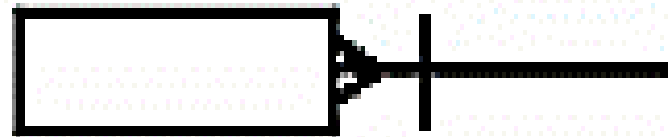
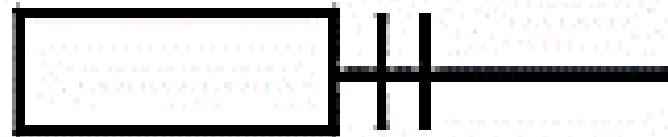
Ternary Relationship

A simultaneous relationship among instances of three entity types.



Relationship Cardinalities

- Optional one
- Mandatory one
- Optional many
- Mandatory many



Relationship Cardinalities

One-to-many (Optional):

STUDENT---BOOK (in library)

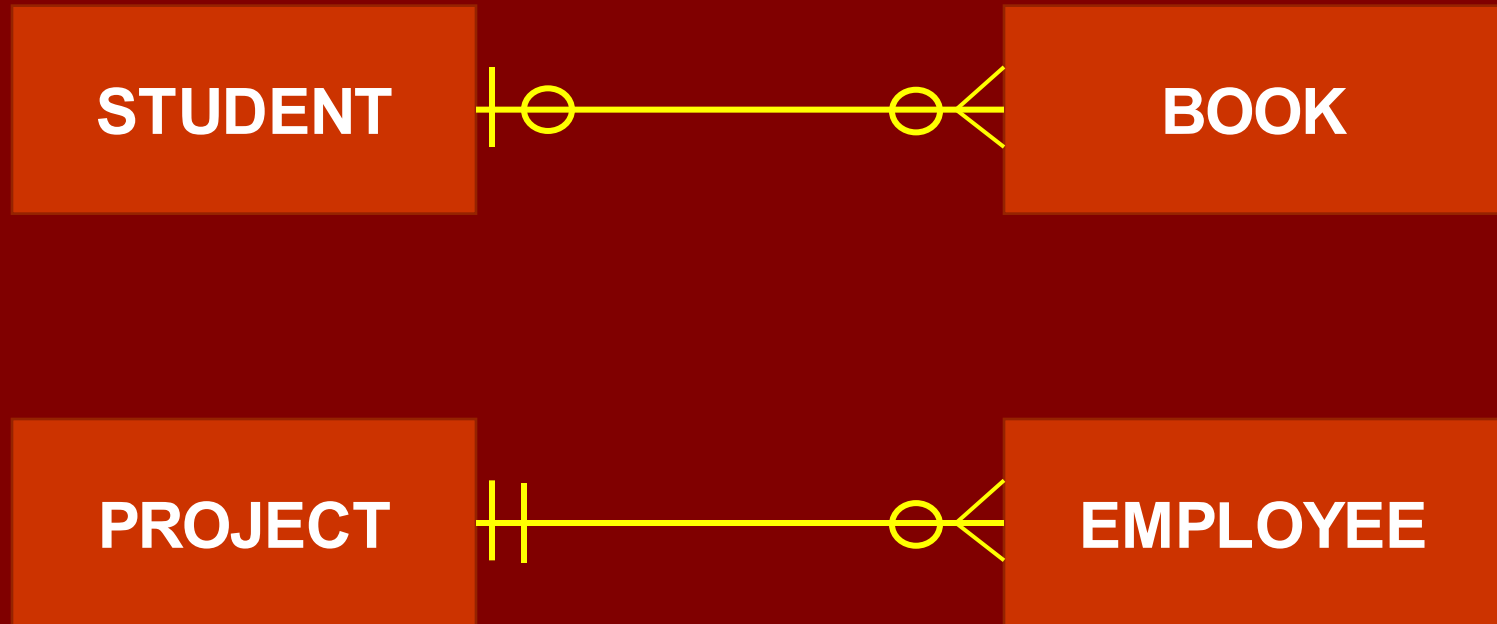
both are optional for each other

One-to-many (Mandatory):

PROJECT---EMPLOYEE

project must has some members but for employee it is optional

Relationship Cardinalities



End of Lecture

University ERD

