SLIIT
Discover Your Future

Information Systems and Data Modeling

IT1090

SLIIT
FACULTY OF COMPUTING

Contact Person

- Lecturer In Charge:

  Mrs. Manori Gamage

- Lecturers:

  Prof. Mahesha Kapurubandara
  Mrs. Muditha Thissera
  Mrs. Sanjeewi Chandrasiri
  Mrs. Pradeepa Bandara
  Mrs. Lokesha Weerasinghe

- Enrolment Key: **IT1090**

Ground Rules

- No food or drinks in the classroom.
- Mute all cell phones during the class session.
- No mini-conferences during the class session.
- Sending short bills(Chits) are not encouraged.
- Class attendance is compulsory.
- Attendance and active participation are required for the tutorial classes.
- Be punctual (perform exactly at the time appointed).

Learning Outcomes

Learning Outcome	Description
LO1	Explain the importance and impact of information systems in business organizations.
LO2	Evaluate the information systems strategies to achieve organizational goals.
LO3	Model data requirements using data models.
LO4	Apply formal methods to refine the data model.
LO5	Use SQL to store and modify the data in the database, and to write queries to satisfy real world application level problems.

Module Delivery

- Lectures
  - 2 × 1 hours of lectures per week
- Tutorials
  - 1 × 1 hour tutorial per week
- Lab Sessions
  - 2 × 1 hours of lab session per week

Assessment Criteria

Assessment components	Contribution to the final grade	Learning outcomes covered
<u>**Continuous assessments**</u>		
•Mid-term Examination	20%	LO1-LO3
•Assignment	20%	LO1-LO5
<u>**Final Examinations**</u>		
•Final Examination (individual)	60%	LO1-LO5
Total	**100%**	

Primary References

1. Laudon, K.C and Laudon, J.P., *Management Information Systems: Managing the Digital Firm*, 13th ed., Pearson Prentice Hall
2. Ramakrishnan, R. and Gehrke, J., *Database Management Systems*, 3rd ed., McGraw-Hill
3. *SQL Server 2016 Documentation*
4. Elmasri, R. and Navathe, S.B., *Fundamentals of Database Systems*, 5th ed., Addison-Wesley.
5. Silberschatz A., Korth H.F. and Sudarchan S., *Database Systems Concepts*, 3rd ed., McGrawHill , 1996
6. Connolly and Begg, *Database Systems: A Practical Approach to design, Implementation and management*, 3rd ed., Addison-Wesley

SLIIT Discover Your Future

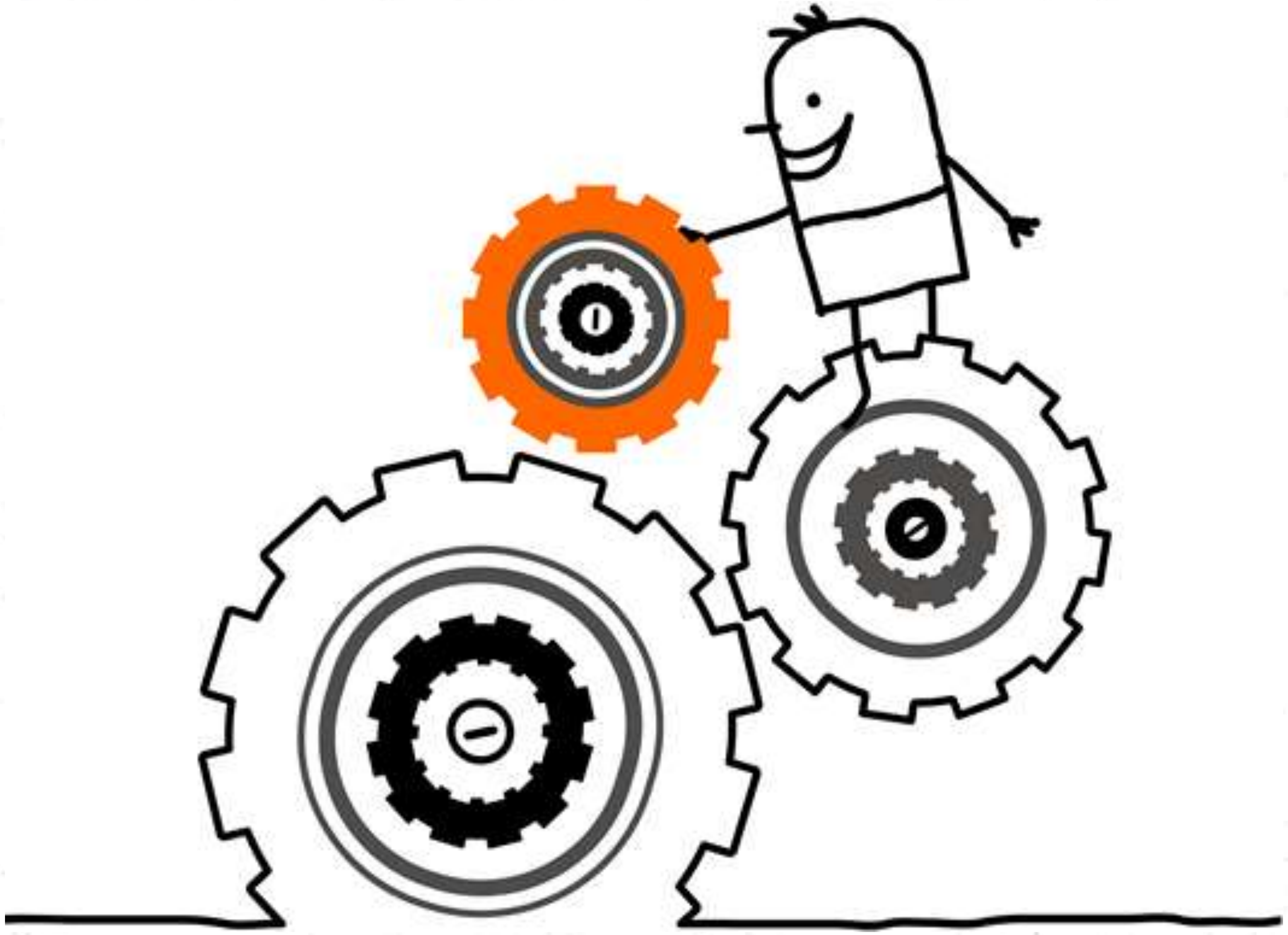
Information systems in business organizations

Lecture – 01

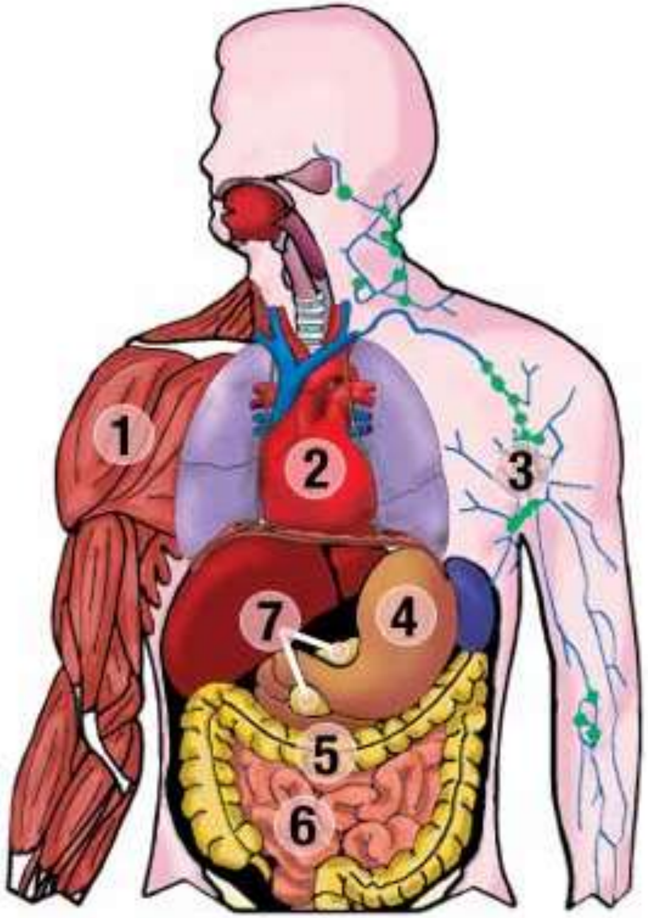
Learning Outcome

- **LO1:** Explain the importance and impact of information systems in business organizations.

What is a System..?

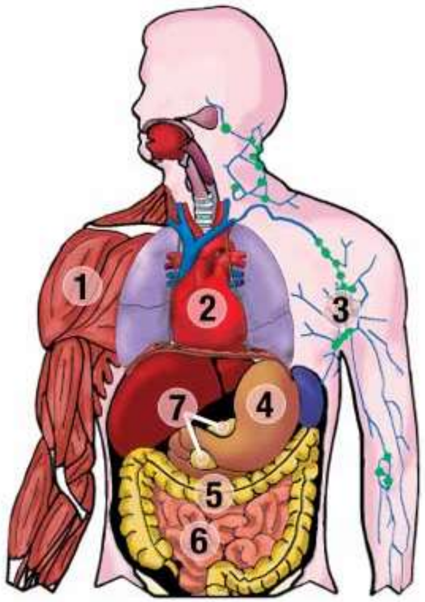


Example

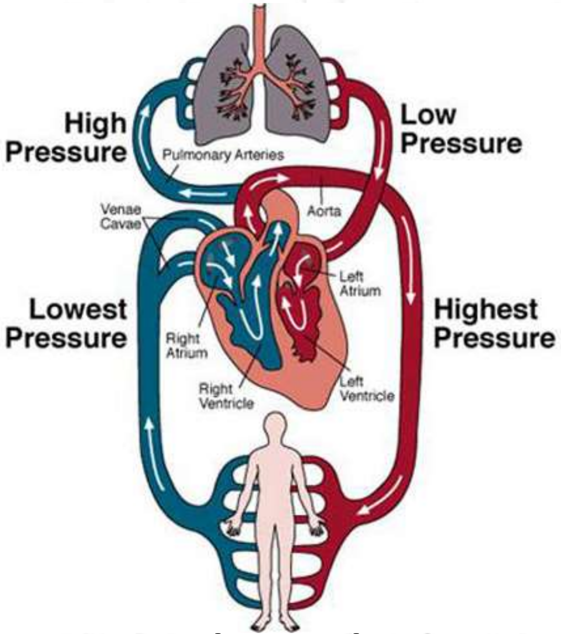


A System

- An array of components that work together to achieve a goal or goals



components

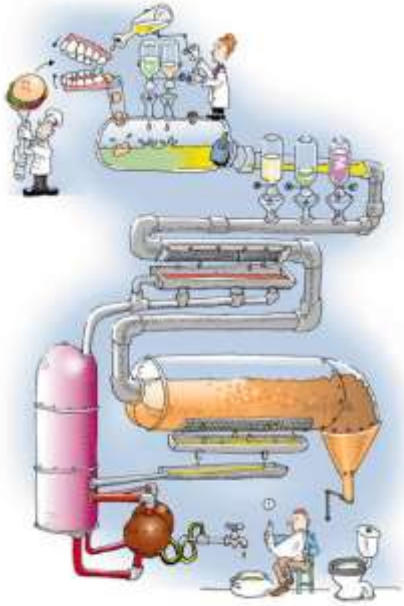


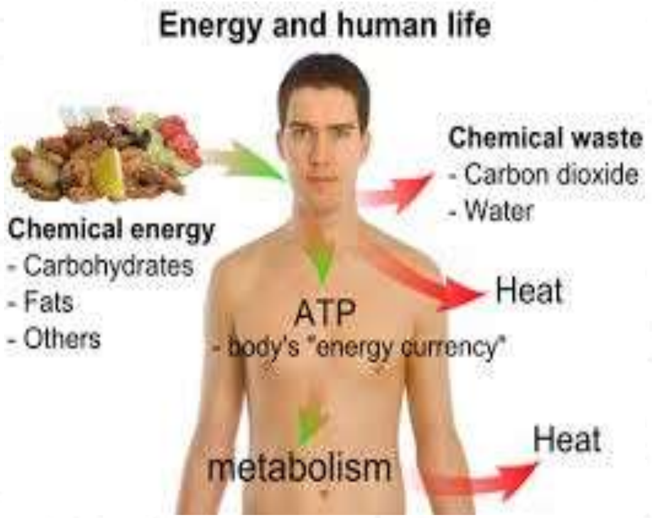
work together

achieve a goal

A System

- Accepts input
- Processes input
- Produces output

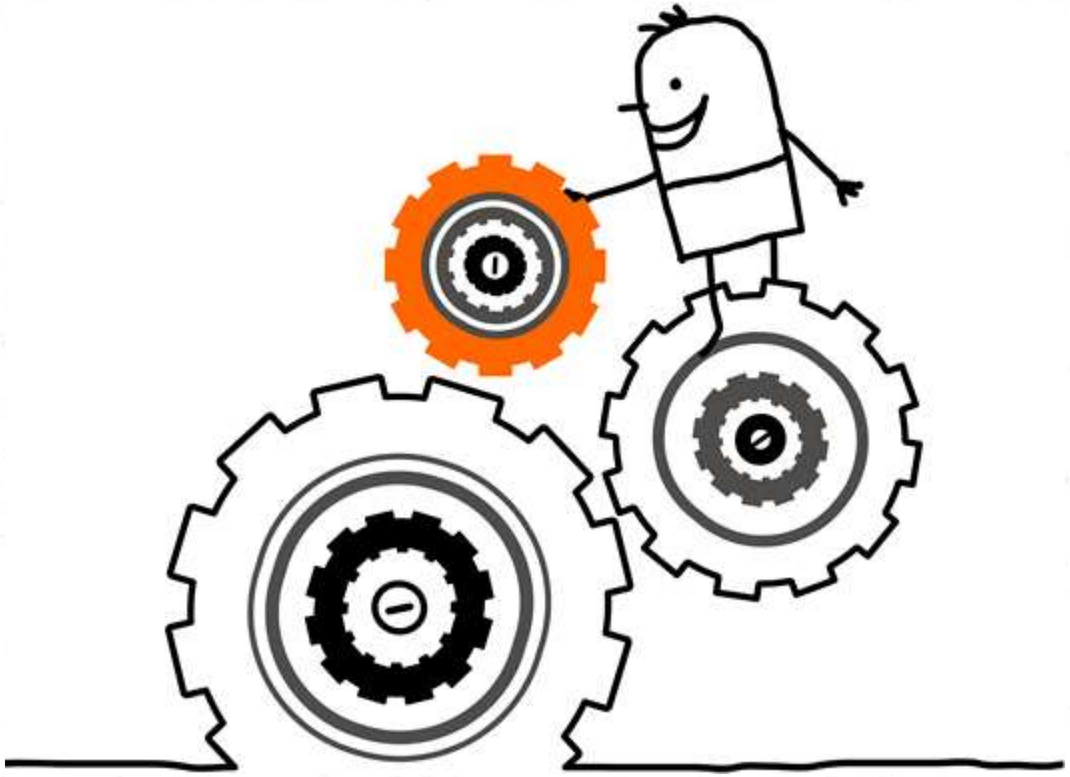




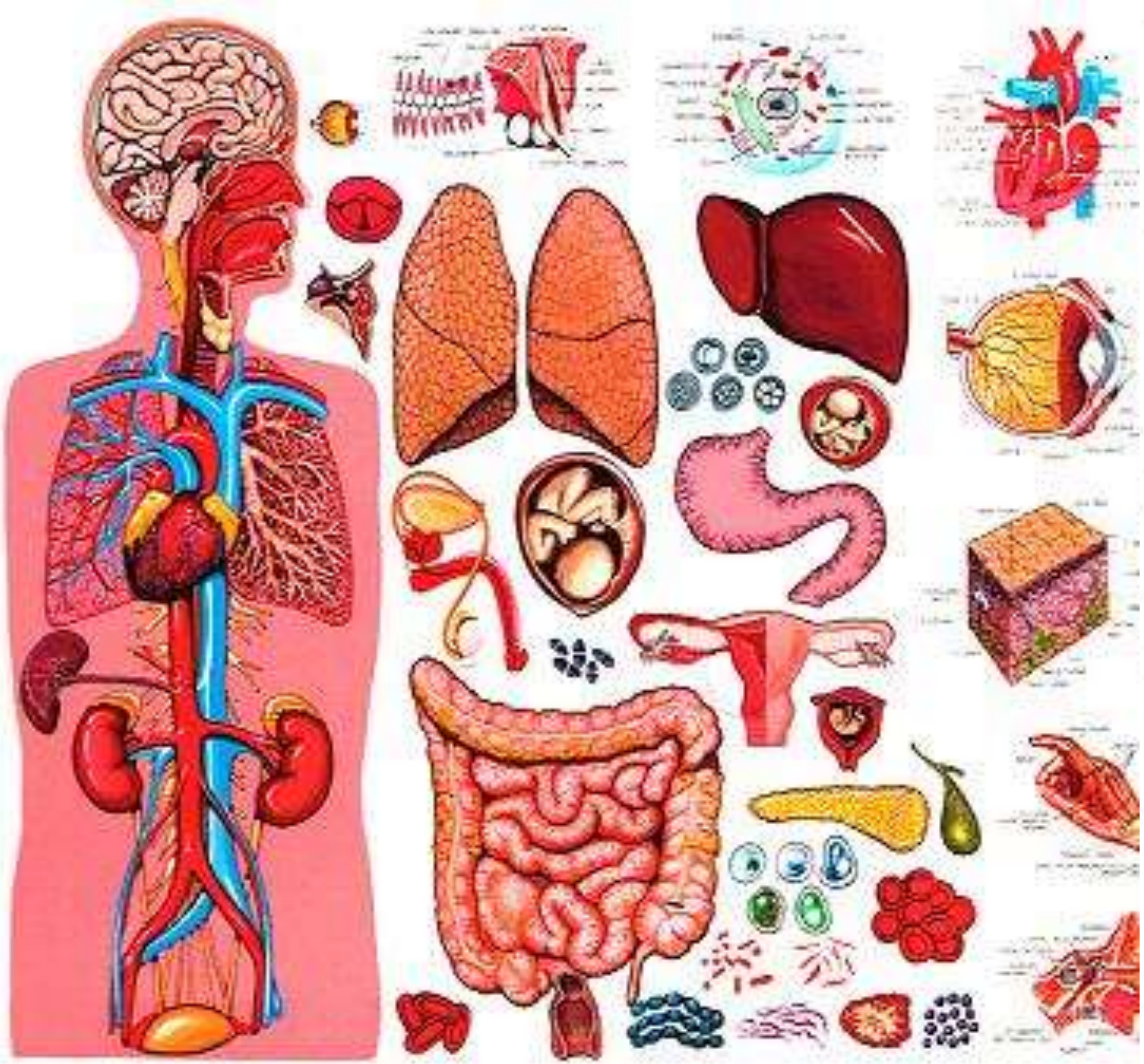
A System

- May have multiple goals
  - E.g: Goals of the human body
    - To Be Alive (Ultimate Goal)
    - To Work
    - To Survive

- Contains Subsystems

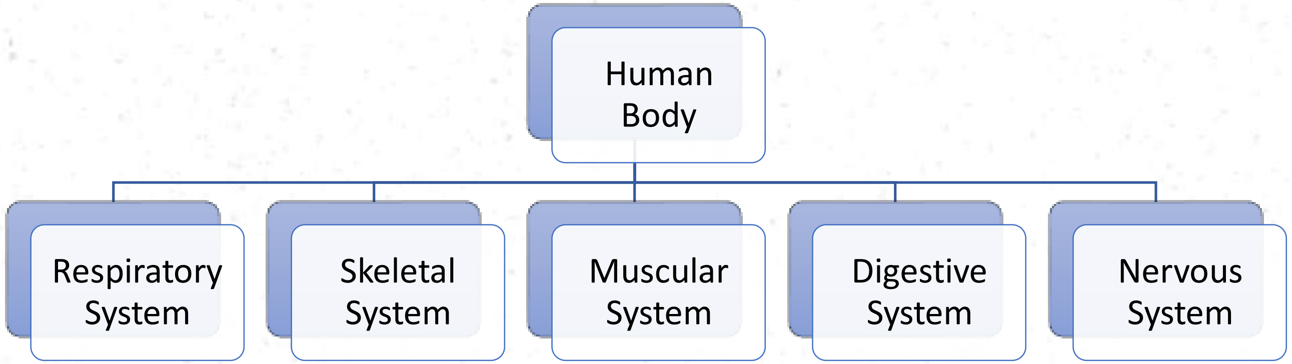


What is a Subsystem..?

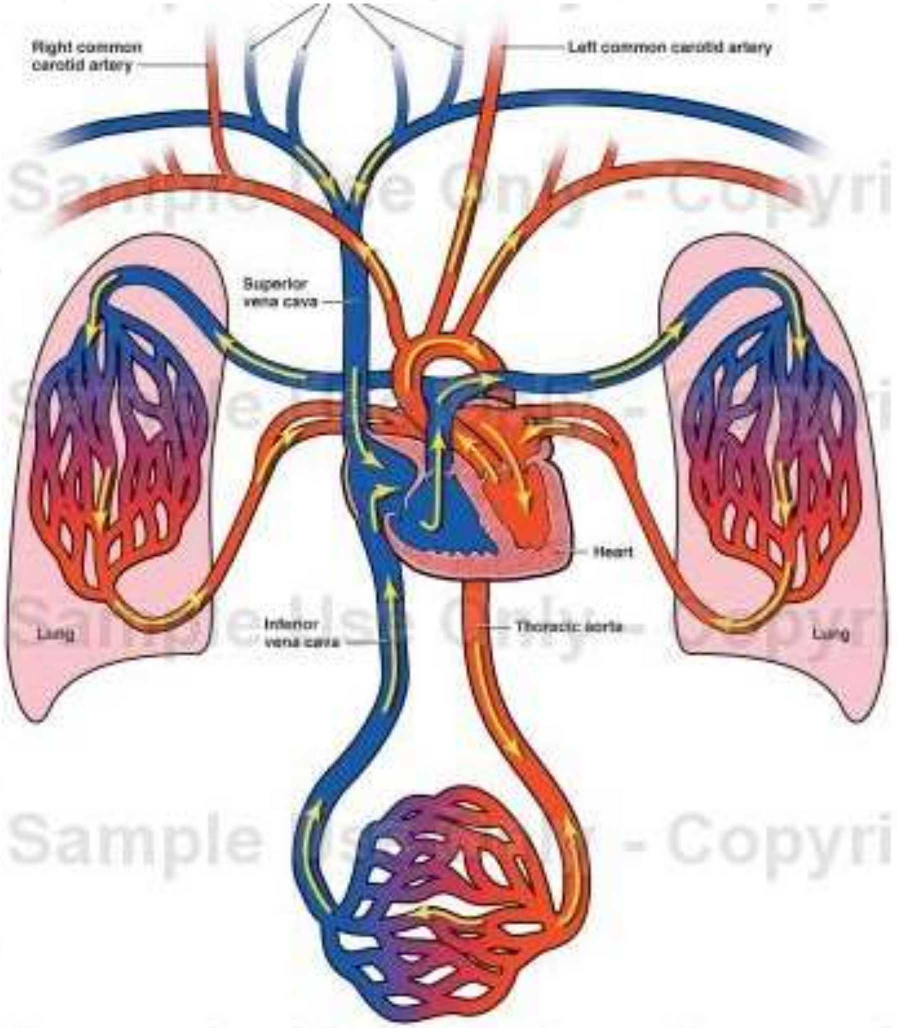


A Subsystem

- is a component of a system, it can also be considered a system in its own right.



- have sub-goals that meet the main goal
- transfer output to other subsystems

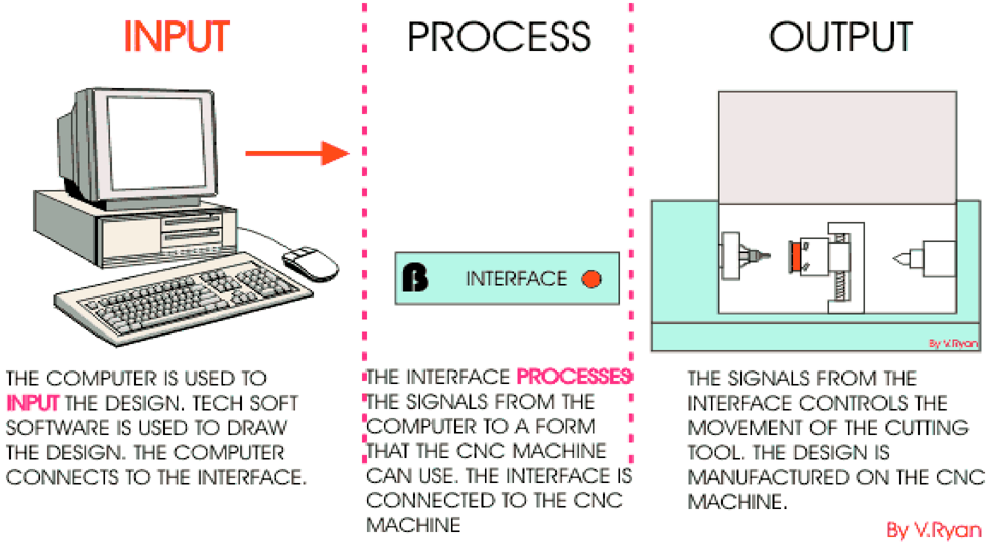


Subsystems Transfer Output to Other Subsystems

What is an Information System..?

An Information System

- is a collection of components that work together to provide information to help in the operations and management of an organization.



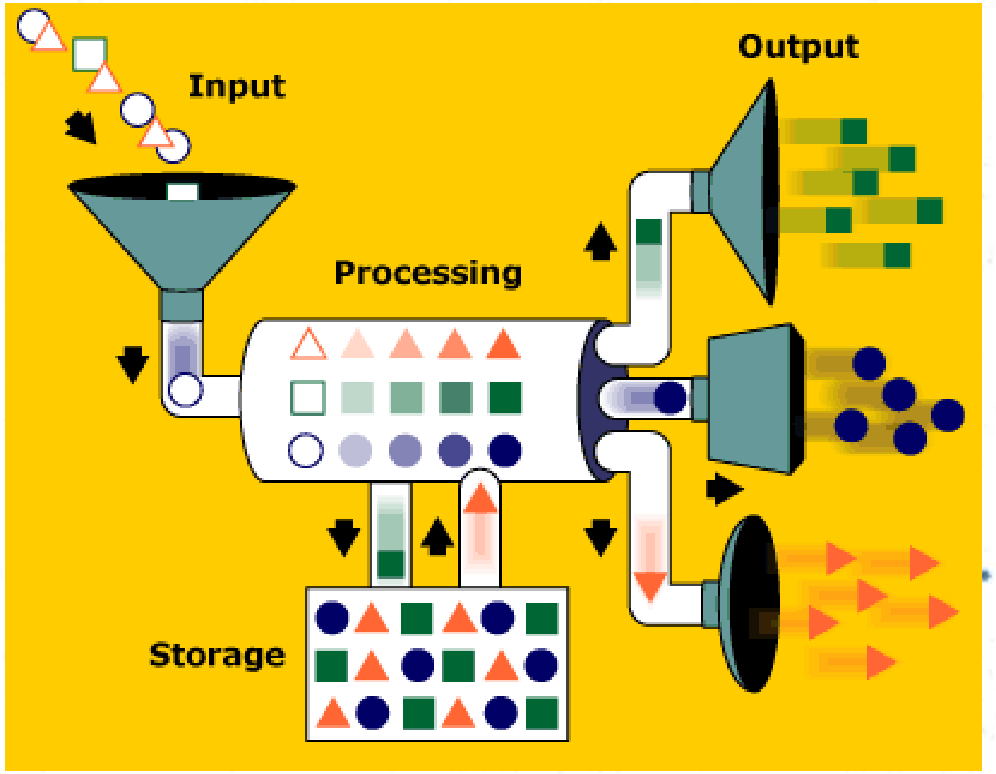
Information Systems

- An Information System collects, processes, stores, analyzes, and disseminates information for a specific purpose

- Information System is an *arrangement of people, data, processes, information presentation, and IT* that interact to support and improve day-to-day operations in an organization (As well as support the problem solving and decision making needs of various people)

- An Information System is not necessarily be computerized

Components of an Information System



Components of an Information System

- Hardware - Equipment such as computers
- Software - Instructions for the equipment
- Stored Data - Facts stored in the system
- Personnel - People who operate the system
- Procedures - for the user to follow

Examples of Information Systems

- Library Information Systems
- Banking systems
- Inventory control systems
- Billing systems
- Hospital management systems
- Hotel reservation systems

What are the information systems you know..?

E.g.: – SLIIT Library Information System

E.g.: – SLIIT Library Information System

- Let’s look at the library information system at SLIIT which provides the facility to borrow books for studies
- *Borrowing involves; students and staff registering to the system, selecting the desired material from the library and library assistant recoding it against the studentID*

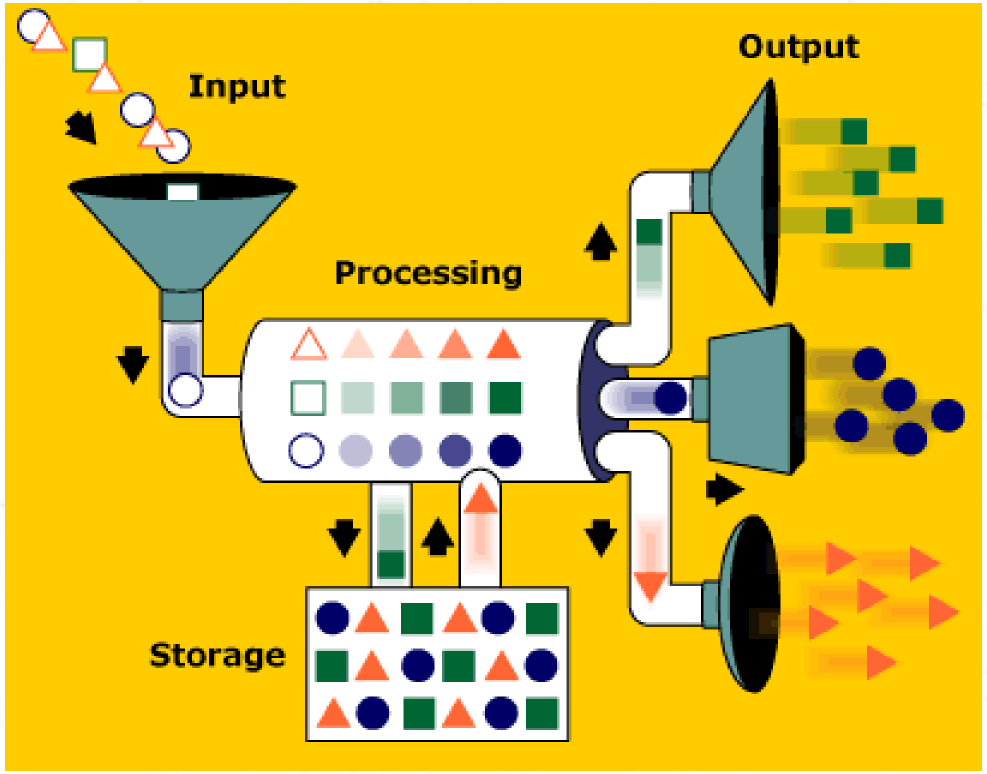
E.g.: – SLIIT Library Information System

- What information does this system provide?
  - Periodically the system reports on the books available, books overdue, membership
  - Calculates fines for overdue books
  - Check for valid membership

E.g.: – SLIIT Library Information System

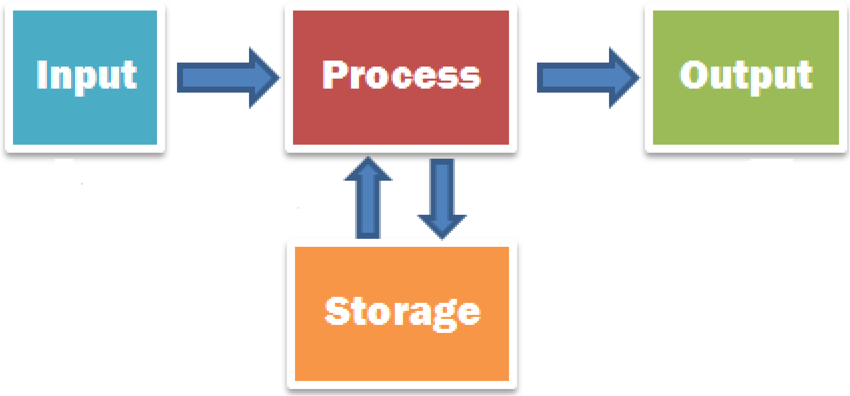
- How does this information helps?
  - This helps the Librarian to decide whether *to understand the needs of the membership (add new books)*, track the borrowings and over dues, easy communication with membership

Functions of an Information System



Functions of an Information System

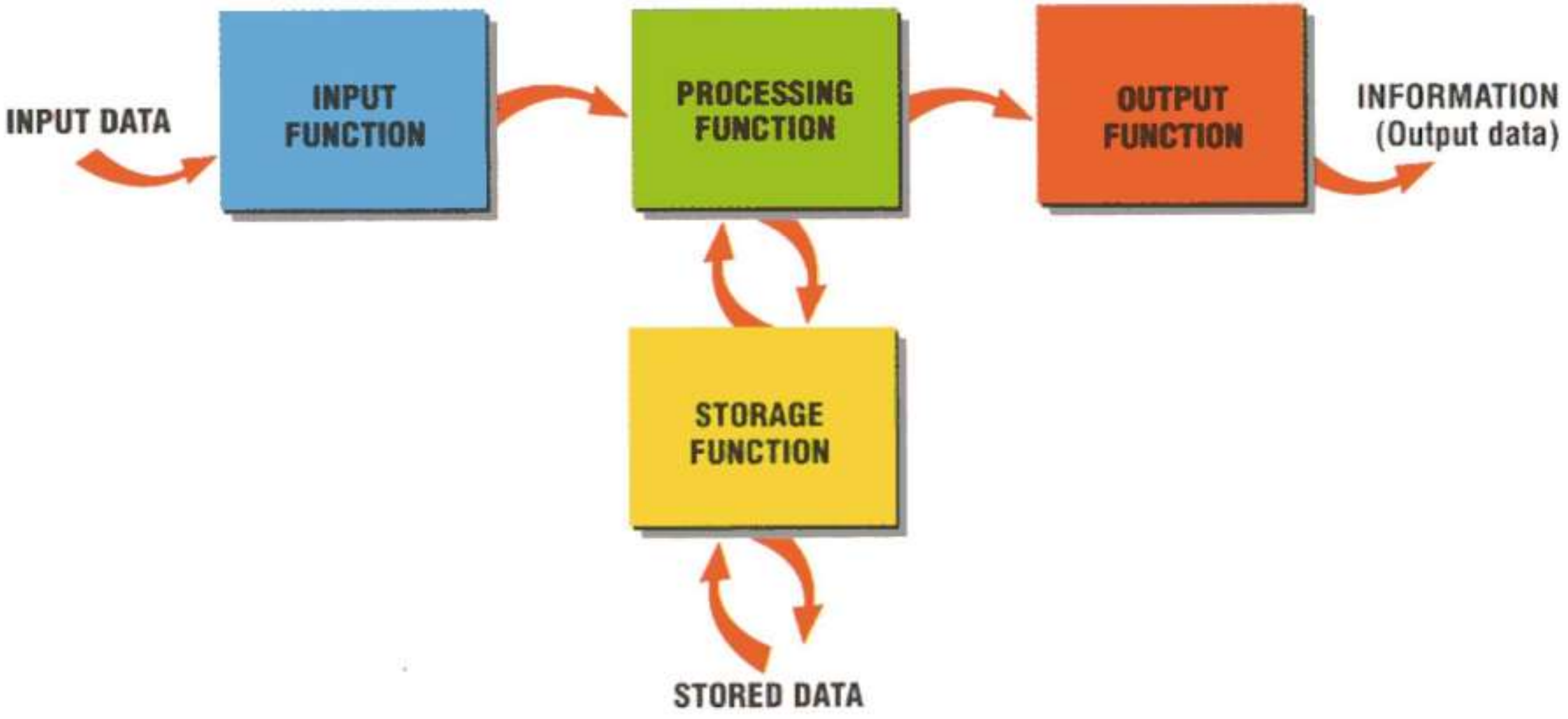
- An Information System accepts facts or data from outside the system, stores and processes the facts, and produces the results for use outside the system



Functions of an Information System

- Input Function accepts the input data from outside the system.
- Storage Function retains input data and retrieves stored data.
- Processing Function calculates and in other ways manipulates the input and stored data.
- Output Function produces results of processing for use outside the system.

Functions of an Information System



Characteristics of Useful Information

- **Not all information is useful**
- Useful information should be:
  - Relevant
  - Complete
  - Accurate
  - Current
  - Obtained economically (in business)

Information Systems Users

Information System Users

- An information system user is a non-computer oriented person who gains some benefit from using an information system in his or her personal or work life
- Sometimes called as an end-user, to distinguish from computer professionals

Information System Users

When you go to a fast food restaurant and buy a hamburger, the counter person is the user of the retail food system

When you go to a retail goods store and buy some essential items, the cashier is the user of the point of sales system

Types of Information Systems Users

- **Direct Users** - Users may use an information system **directly** by pressing keys on a computer keyboard or by operating a piece of equipment that sends input data to a computer
  - Eg.: A person using spreadsheet software to do financial projections
- **In-direct Users** - A user may use an information system **indirectly** by having someone else enter the data and receive the output which is then given to the user
  - Eg.: A sales manager who receives periodic, printed sales reports

Any information system you know..??

Are Systems Unique..?

- Some systems are unique to a particular industry;

Eg.: Health care industry

- Patient-scheduling systems
- Nursing allocation systems
- Pharmaceutical systems

Are Systems general...?

- Some systems are general enough for the use in a variety of industries;
  - Inventory
  - Payroll
  - Billing
  - Invoice Processing
  - Human Resource Management
  - Personal Productivity software

The Categorization of Information Systems

- So, we need a way to categorize Information Systems…there are many ways;
  - By <u>number of people</u> whose work is affected by the system
  - By organizational structure
  - By major functions
  - By support provided

Types of Information Systems

- Personal Information Systems
- Workgroup Information Systems
- Organizational Information Systems
- Inter-organizational Information systems
- Global Information systems

Benefits of Information Systems

- Better information
- Improved service
- Increased productivity
- Competitive Advantage

How organizations use IS Today

- Information systems transform business
  - Increase in wireless technology use, Websites
  - Increased business use of Web technologies
  - Cloud computing, mobile digital platform allow more distributed work, decision-making, and collaboration
- Globalization opportunities
  - Internet has drastically reduced costs of operating on global scale
  - Presents both challenges and opportunities

How organizations use IS Today

- In the emerging, fully digital firm
  - Significant business relationships are digitally enabled and mediated
  - Core business processes are accomplished through digital networks
  - Key corporate assets are managed digitally
- Digital firms offer greater flexibility in organization and management
  - Time shifting, space shifting

How organizations use IS Today

Growing interdependence between ability to use information technology and ability to implement corporate strategies and achieve corporate goals.

How organizations use IS Today

- Business firms invest heavily in information systems to achieve six strategic business objectives:
  1) Operational excellence
  2) New products, services, and business models
  3) Customer and supplier intimacy
  4) Improved decision making
  5) Competitive advantage
  6) Survival

1) Operational excellence

- Improvement of efficiency to attain higher profitability
- Information systems, technology an important tool in achieving greater efficiency and productivity

1) Operational excellence

- E.g.:
  - PickMe’s vehicle reservation system connecting driver and traveler based on location based services (GPS).

(1). Select pick-up location via GPS... or simply type the address

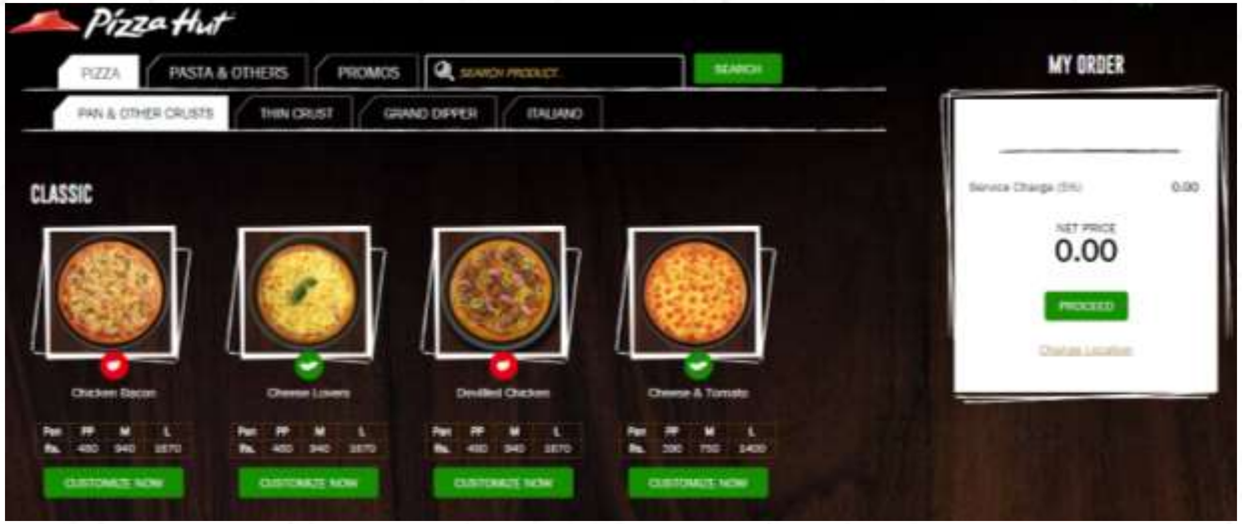
(2). Select taxi type and make your booking

(3). View driver and taxi details

(4). At the end of the trip make the payment as indicated. You can also rate the driver and submit your comments.

1) Operational excellence

- E.g.:
  - Pizzahut's ordering system providing an interactive menu for customers to place an order.



1) Operational excellence

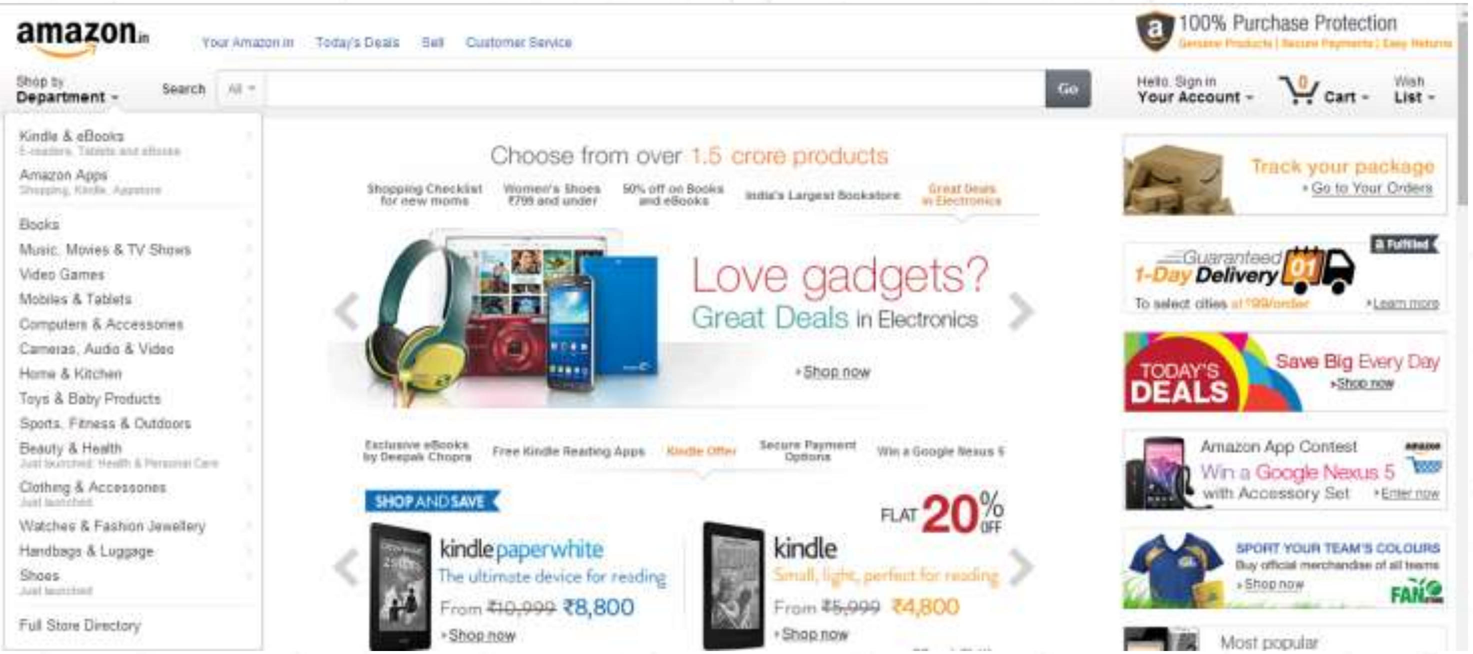
- E.g.:
  - Cargill's Foodcity's point of sales (POS) system allows the counter person to calculate payments.

2) New products, services, and business models

- Business model: describes how company produces, delivers, and sells product or service to create wealth
- Information systems and technology a major enabling tool for new products, services, business models

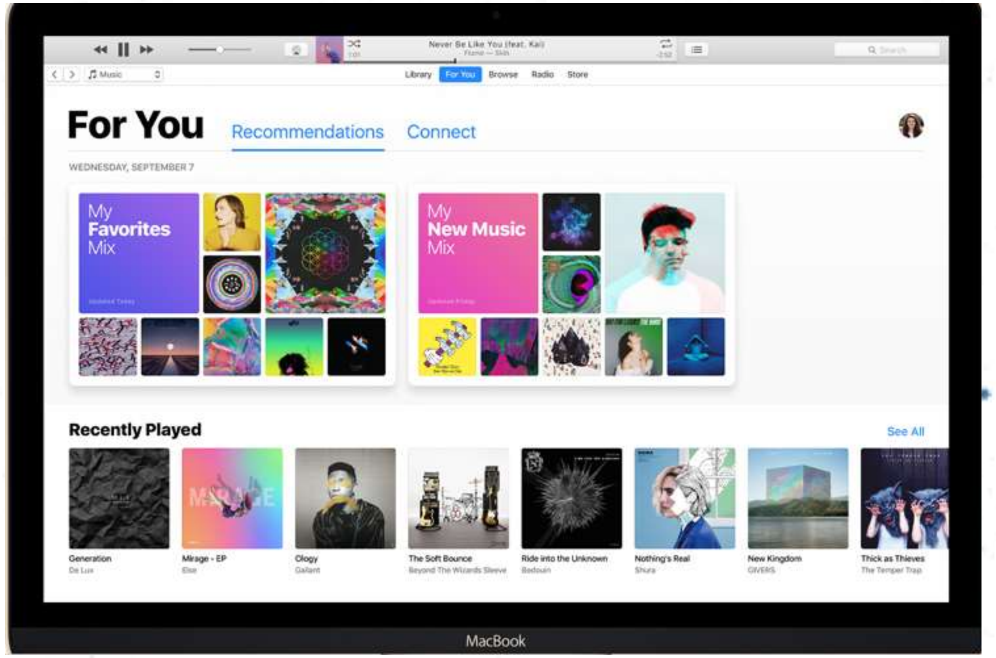
2) New products, services, and business models

- E.g.:
- Online business model of Amazon.com



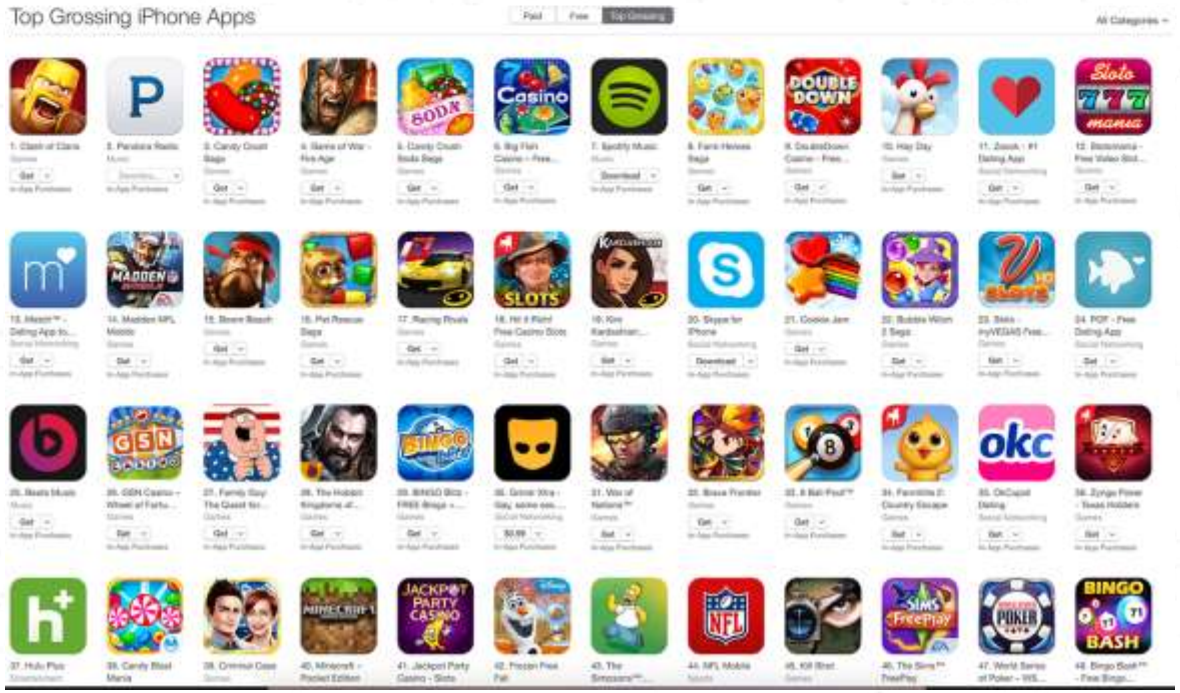
2) New products, services, and business models

- E.g.:
- Apple's iTunes



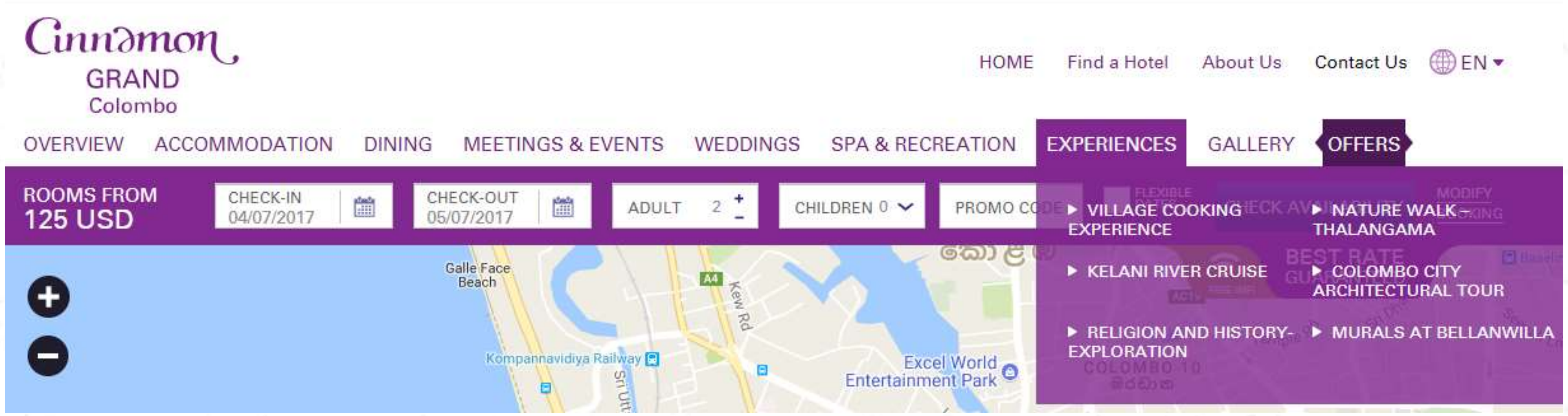
2) New products, services, and business models

- E.g.:
- Google's apps store for mobile applications



3) Customer and supplier intimacy

- Serving customers well leads to customers returning, which raises revenues and profits
- E.g.:
    - High-end hotels that use computers to track customer preferences and use to monitor and customize environment



3) Customer and supplier intimacy

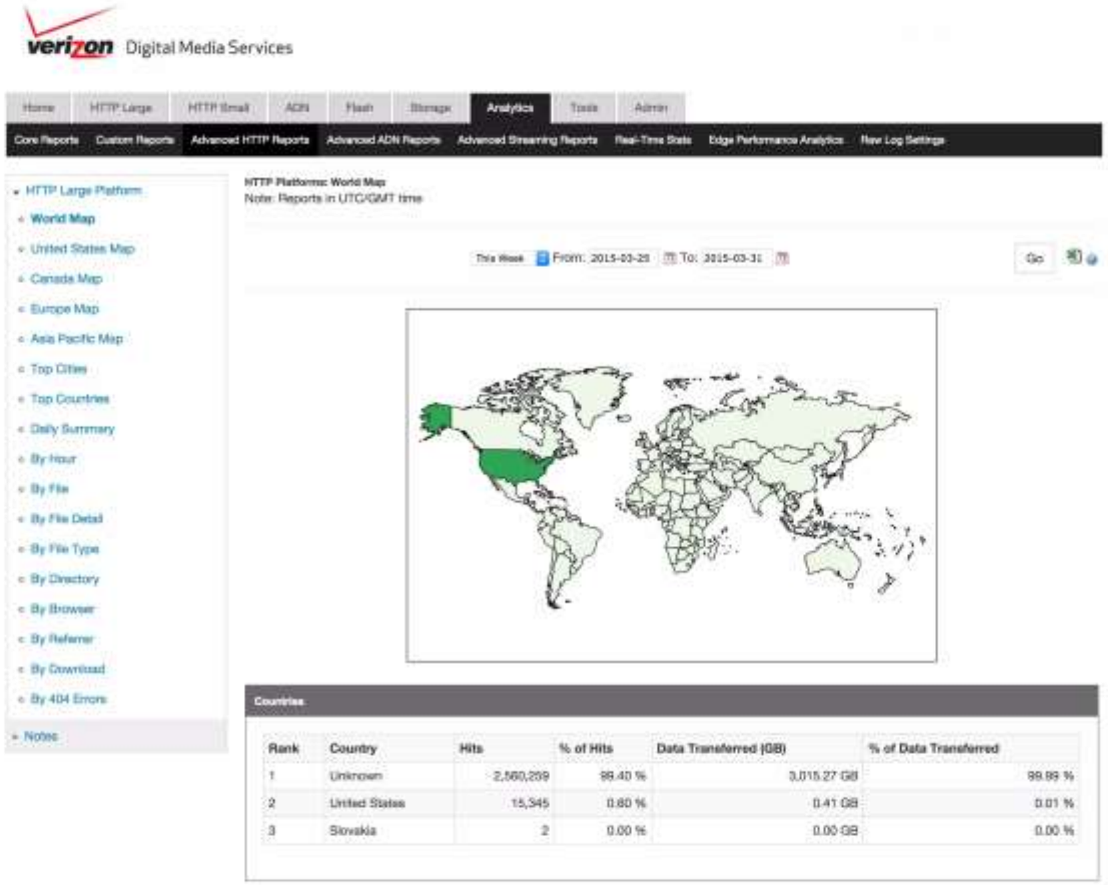
- Intimacy with suppliers allows them to provide vital inputs, which lowers costs
- E.g.:
  - Brandix's information system which links supplier to contract manufacturer

4) Improved decision making

- Without accurate information:
  - Managers must use forecasts, best guesses, luck
  - Leads to:
    - Overproduction, underproduction of goods and services
    - Misallocation of resources
    - Poor response times
  - Poor outcomes raise costs, lose customers

4) Improved decision making

- E.g.:
  - Verizon’s Web-based digital dashboard to provide managers with real-time data on customer complaints, network performance, line outages



5) Competitive advantage

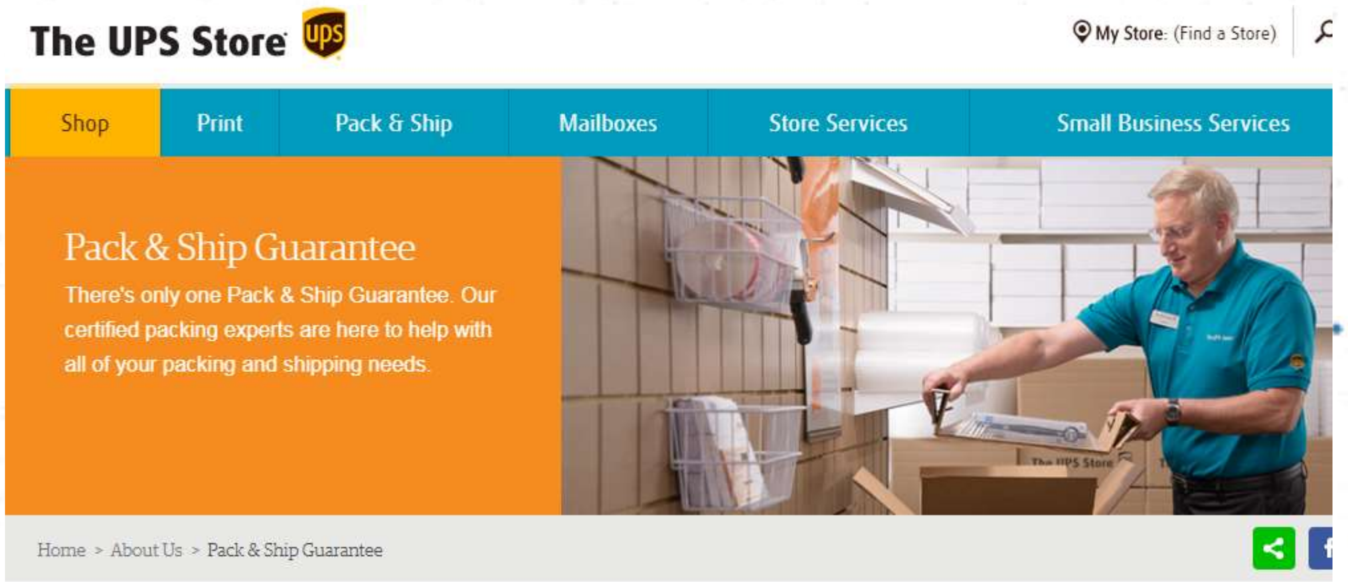
- Delivering better performance
- Charging less for superior products
- Responding to customers and suppliers in real time

5) Competitive advantage

- E.g.:
- Online banking for customers

5) Competitive advantage

- E.g.:
- UPS online product tracking system for shipping and delivery



6) Survival

- Some industries are heavily dependent on technology
- Hence all businesses in the industry need to maintain the basic technologies used in the industry

6) Survival

- E.g.:
- ATM systems for banks

End of Lecture - 01

Questions..?

Reference

- K. C. Laudon and J.P. Laudon, "Management Information Systems: Managing the digital Firm", Chapter 1, INFORMATION SYSTEMS IN BUSINESS TODAY, 13th Ed, 2014

Next Lecture

- Process Mapping

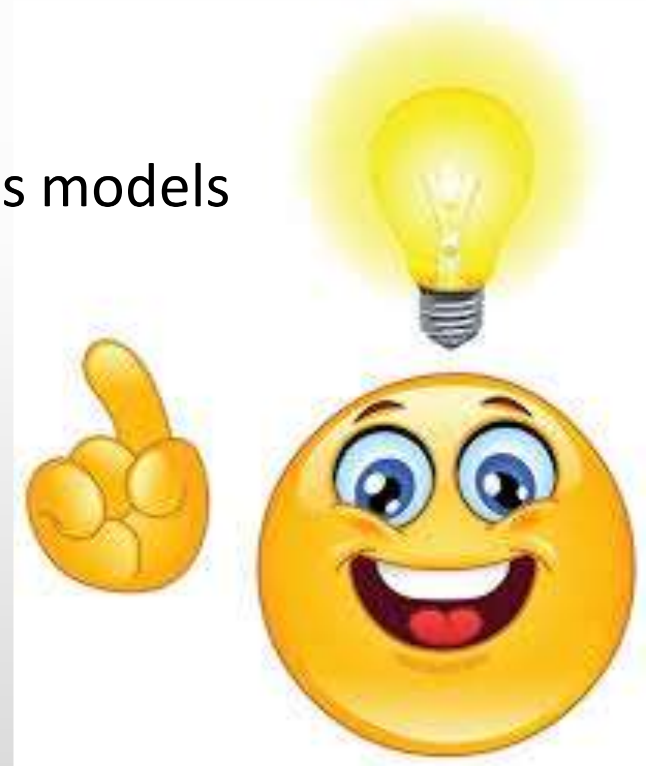
SLIIT
FACULTY OF COMPUTING

Business Processes Mapping

Lecture – 02

Lecture 02 - Recap

- How organizations use IS Today – Strategic business objectives
  - Operational excellence
  - New products, services, and business models
  - Customer and supplier intimacy
  - Improved decision making
  - Competitive advantage
  - Survival



Learning Outcomes

- **LO2:** Evaluate the information systems strategies to achieve organizational goals

Session Outcome

After completing this session you will be able to;

1. Develop complete consistent process maps
2. Effectively analyze flow charts to identify improvement opportunities.

What is a Business Process?

- A business process is a collection of interrelated activities that takes one or more kind of inputs and creates an output that is of value to the customer.

Lets walk to McDonald's. Your favorite fast food restaurant.

What kind of business processes you can find there?

Example: Preparing a Burger Meal

- Activities

  Take order, prepare meal, deliver meal

- Inputs

  Burger bun, meat, tomatoes, potatoes, labor, etc.

- Output

  Hamburger

Why Process Mapping?

Because…

These questions require a good, documented and communicated understanding of business processes!

- How do the processes look like?
- How do we know that we manage our processes well?
- How can we train staff in our processes?
- How can we ensure required process compliance?
- How can we design and use IT to support our processes?
- How can we organize and conduct process improvement?

Process mapping?

Break down the process in to activities/steps and visualize it pictorially to highlight the flow of performing them.

Process mapping - Characteristics

- A visual representation of a flow of a process for a product or service.
- Within specified boundaries.
- Uses symbols and arrows to display inputs, outputs, tasks performed and task sequences.
- Helps to analyze the process.
- Also referred to as Flow charts or Flow Diagrams.
- First step in Process improvement effort.

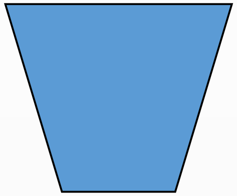
Questions to understand the process flow

- What is happening?
- When is it happening?
- Who is doing it?
- Where is it happening?
- How long does it take?
- How is it being done?
- Why is it being done?

Developing a Process Map

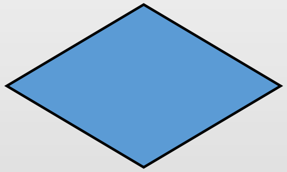
Process Mapping - Notations

Start or End point

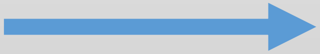


A Manual operation depicted as a sub process

Totally automated activity depicted as a sub-process

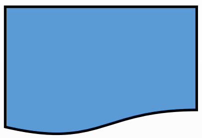


Decision

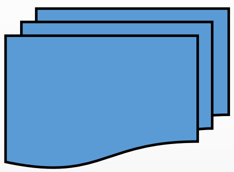


Process flow direction

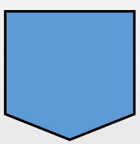
Process Mapping – Notations Cont.



Document

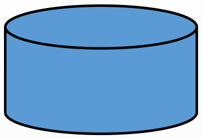


Multidocument

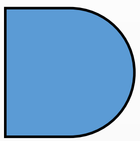


Off-page connector

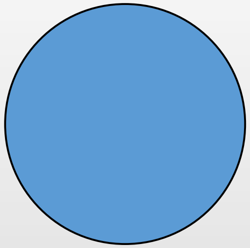
Phase Separator



Database

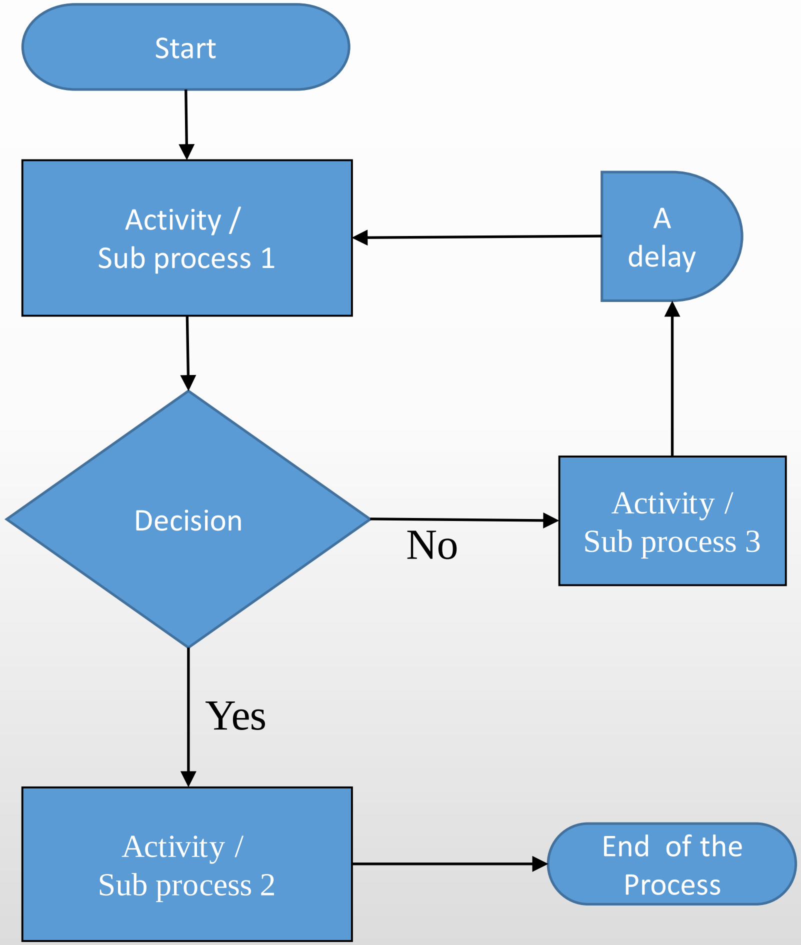


Delay



Another Process

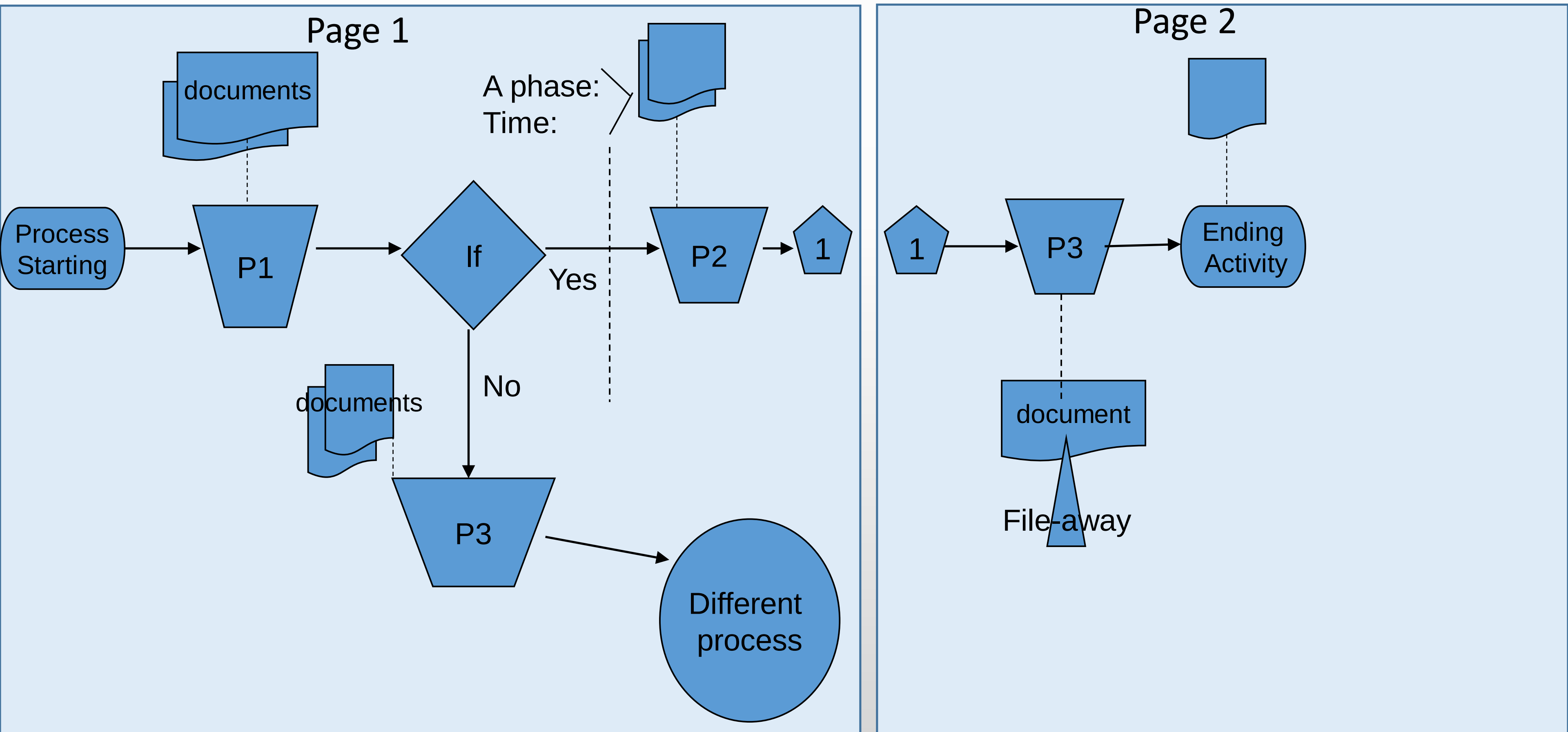
Example:



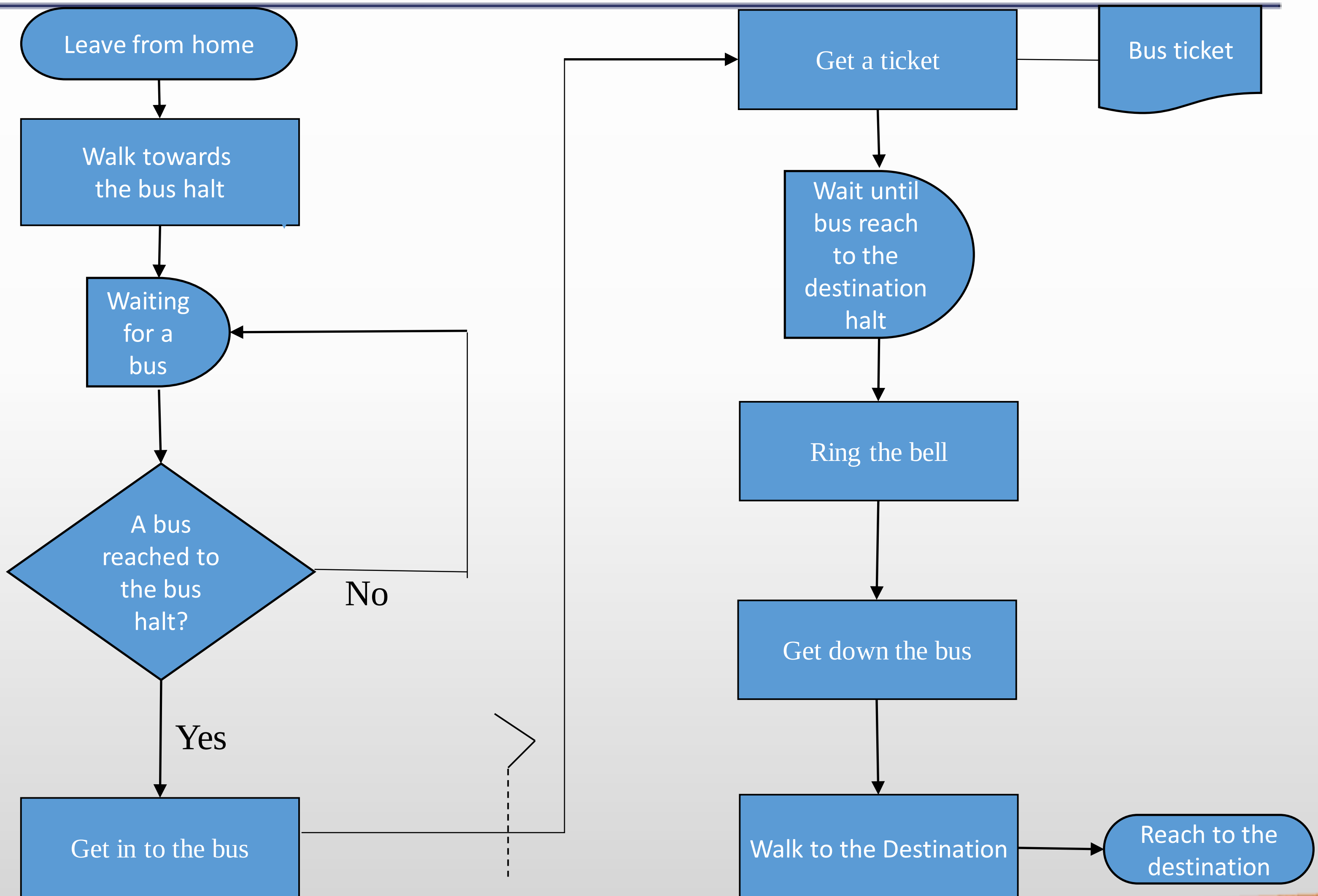
Note:

- use terminator/start symbol to show start and end of the process
- Use off page reference to link to another process map page
- Use delay symbol to show when delays occurs: waiting on queue's, waiting for an event, etc.
- Always try to identify timing for each activity + total phase time

Possible process map drawing



Example: Travelling process

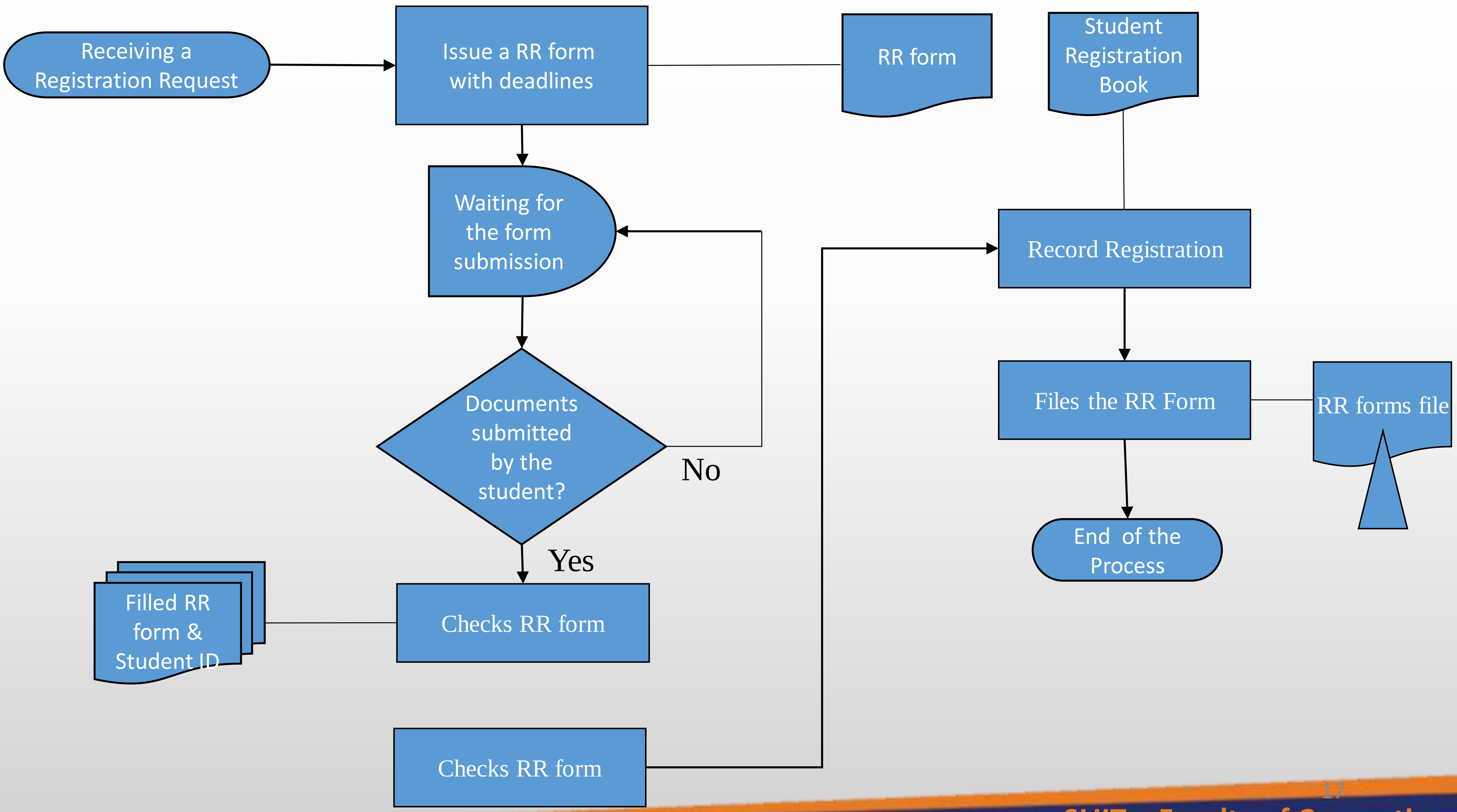


Class Activity – Draw a Process Map for the following process

Register for hostel accommodation

- Go to hostel and Request a 'Registration Request Form' from hostel manager.
- Hostel manager gives a RR form to student, and inform of last date and time to submit the form to hostel manager.
- Student Fill in the registration request form.
- Hand-over registration form and Student ID to hostel manager.
- Hostel manager checks the RR form.
- Hostel manager record the student registration in student registration book.
- Hostel manger files the form in current registration request forms file.

Answer:


Receiving a Registration Request
Issue a RR form with deadlines
RR form
Student Registration Book
Waiting for the form submission
Documents submitted by the student?
No
Yes
Record Registration
Files the RR Form
RR forms file
End of the Process
Filled RR form & Student ID
Checks RR form
Checks RR form

Analyzing a Process Map

[Purpose of analyzing is to improve the process]

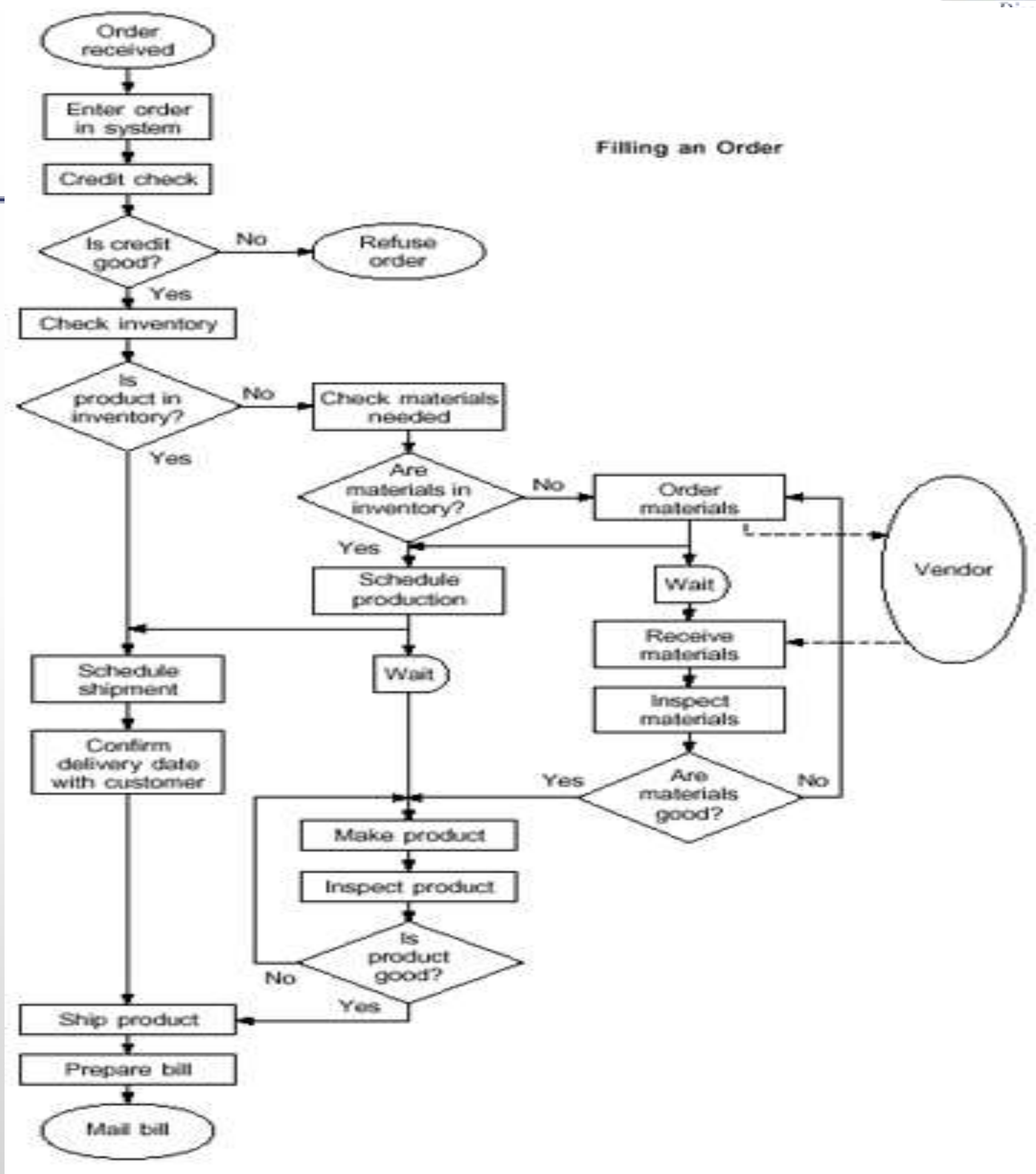
Questions to improve the process:

- What problems do you experience with the process?
- Where can the process be improved?
- Describe the ways in which other people do this process differently?
- Does the process always work this way, are there exception?
- What happens when things get really busy, do people do things differently?
- What happens when people go on leave, is it a problem to get everything done?
- What are all the documents that can be used in this process?
- What reports does this process produce and how are they used?
- At what stages are the documents handed over to others?

SLIIT
Discover Your Future

Class Activity :

Analyze this process map and prepare a list of questions to ask in order to improve the process.



Benefits

- Gives everyone a clear understanding of the process
- Identify different steps in the process with the right order.
- Easy to learn and understand.
- Helps to identify non-value-added operations.
- Good communication language and supports team/group work.
- Keeps everyone on the same page.

Next Lecture

- Modern Information Systems in Business

End of Lecture 02

SLIIT
Discover Your Future

Answer for the class activity (Filling the order)

- How does the process decide whether the credit is good or bad? Is there any threshold credit limit which distinguish between good and bad. Does the process follow that customer credit validation rule for all the customers in a similar manner or is does it vary from one customer to another?
- When refusing the order, how does the process communicate that with the customer?
- When the customer has breached the credit limit, what are the strategies in the process to encourage customer to pay back his credit. Is it included in the scope of this process?
- Does the company have an automated inventory management system?
- When scheduling shipment how does the process interact with the shipping company?
- In addition to the customer, who are the other parties that receive the bill? (such as accounting dept.)
- Are product inventory and material inventory both handled by the same inventory management system or by two systems separately?
- What are the sub processes of shipment scheduling and production scheduling?
- How does the order material process communicate with the external vendor?
- How long (average time) does it take to make the product? What if the customer does not agree to wait if that production time is too long?
- What are the remedies that we can try out to reduce the delays in production scheduling and order material etc. ?

Modern Information Systems

Lecture – 03

Lecture 03

- What is a business process.?
- How business processes work
- What are its components

Learning Outcomes

- **LO2:** Evaluate the information systems strategies to achieve organizational goals

Session Objectives

At the end of this session, you will be able to;

- Explain the evolution of Computing world.
- Compare and contrast various modern Information Systems including Enterprise Systems.
- Know what are real world examples for various modern IS.
- Identify advanced technologies used in Modern IS.

Evolution of Computing World

Early computers: Computing in 1945

- Extremely difficult to use
- Large & expensive
- Used by specialists for scientific calculations etc.

Harvard Mark I

ASCC: IBM Automatic Sequence Controlled Calculator

55 feet long, 8 feet high, 5 tons

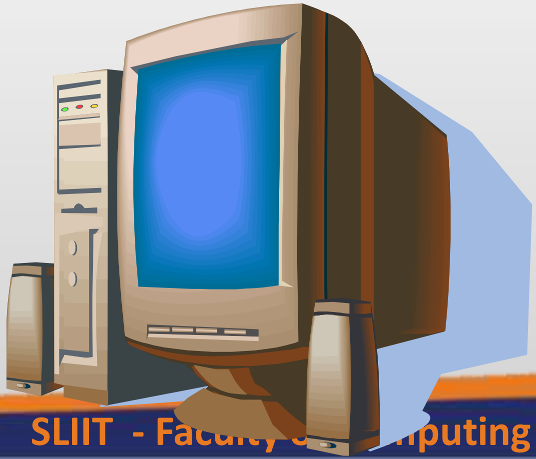
Computing Paradigms Cont.

Late 1950s - 1970s, Mainframe computers and dumb terminals

- Batch processing
- Time sharing
- No graphical interfaces

In 1980s

- Invention of a microprocessor
- Development of PC's - major landmark
- Start using computers by people, customers and end users of Business organizations



SLIIT
Discover Your Future

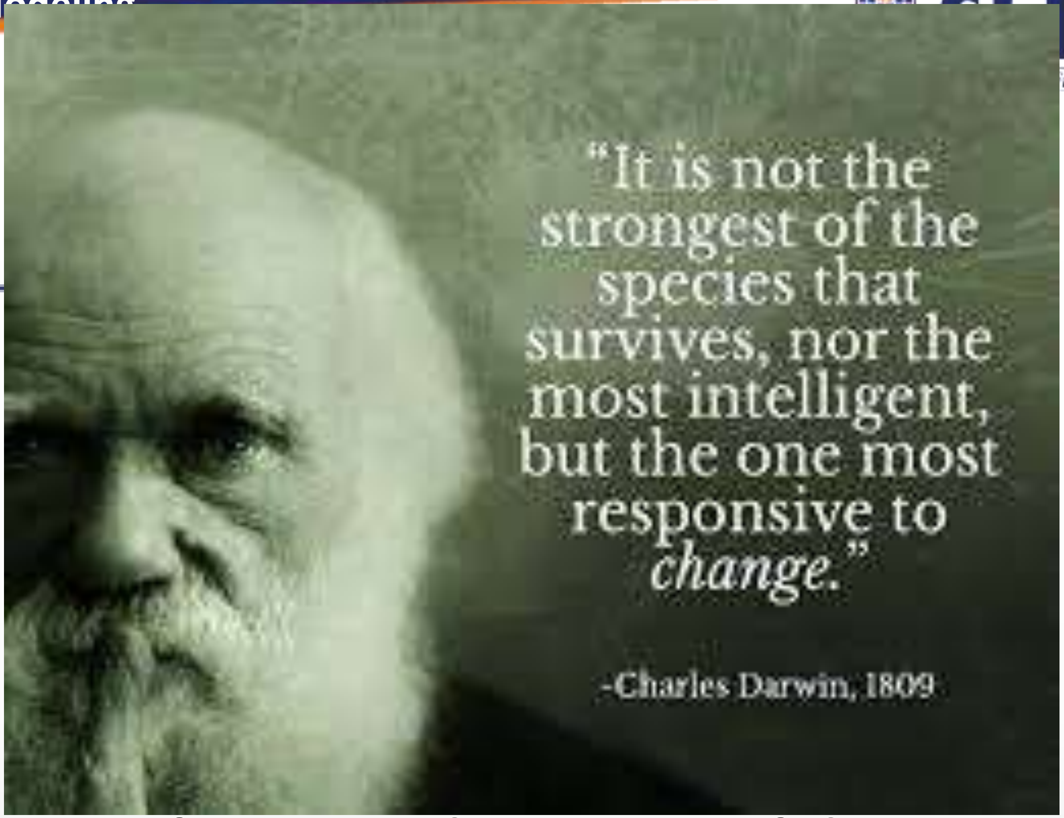
Computing Paradigms Cont.

Today (This era)

What made this change?

- Advancement of technology
  - Rapid advances in speed and capacity
- Pervasiveness of Internet
  - E-enabled world
- Wireless, portable devices
  - Any where, Any time work capability

Technology is configured into *Information systems* that help manage *information* to improve *organizational performance.*



Key Idea:

Get ready to adopt to the modern world with modern Information Systems!!!!

Modern Information Systems

- Office Automation Systems
- Process Control Systems
- E-Commerce Systems
- Decision Support Systems
- Enterprise Systems
- Cloud based Systems
- IoT (Internet of Things) Systems
- BI(Business Intelligence) Systems

Office Automation Systems (OAS)

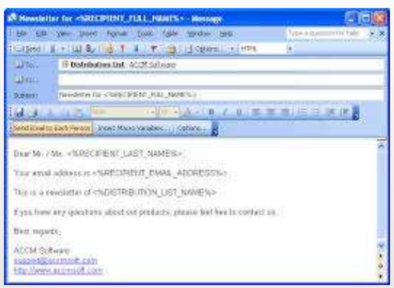
Used to manage the administrative functions in an office environment and are often critical to service-based industries.

OASs intended to increase the productivity of office workers.

Ex:

Groupware

Assists teams of people working together through facilities such as email and teleconferencing within or between companies



Document imaging process (DIP)

Converts documents into digital format using a scanner which can then be stored, retrieved and manipulated across a computer network.

Office automation systems Cont.

Workflow management systems (WFMS)

Automate a business process by providing a structured framework to support the process as follows:-

- Assign tasks to people.
- Allow collaboration between people sharing tasks.
- Retrieve information needed to complete a task e.g. customer details.
- Provide an overview of the status of each task.
- Used in conjunction with DIP to provide automated routing of documents.

Process control systems

These systems support and control manufacturing processes.

Ex:

CAD/CAM (Computer Aided Design/Computer Aided Manufacturing)–

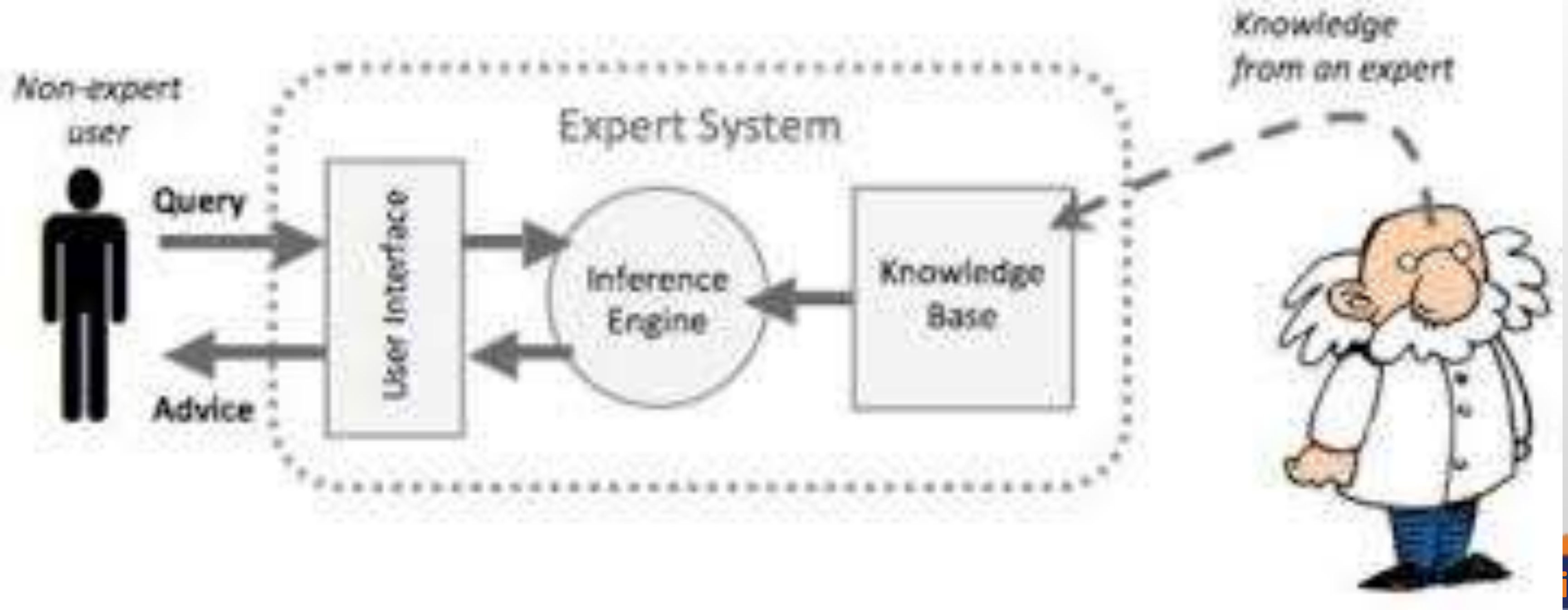
provides a graphics program which allows the design (CAD) and automated manufacture (CAM) of components.

Expert systems

Based on Artificial Intelligence technologies.

Computer system that emulates the decision-making ability of a human expert.

Represent the knowledge and decision-making skills of specialists.



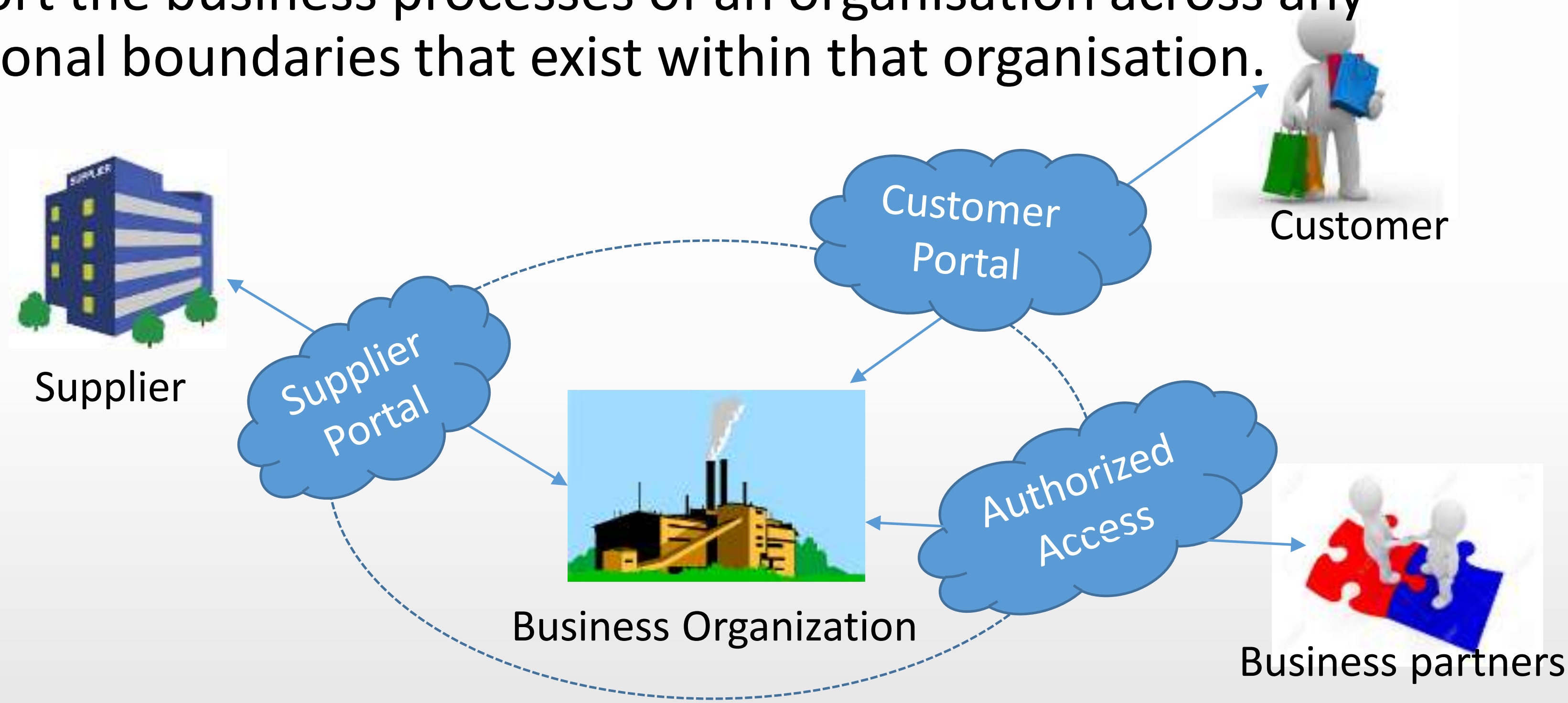
Expert systems Cont.

MYCIN: medical diagnosis

The first well known medical expert system to help doctors, who are not expert in antibiotic drugs. It prescribes such drugs for blood infections with the dosage adjusted for patient's body weight.

Enterprise Systems

Support the business processes of an organisation across any functional boundaries that exist within that organisation.



Use internet technology to connect the business with suppliers, customers and partners.

Main elements of Enterprise systems are ERP, CRM, SCM and SRM.

Enterprise Systems Cont.

Enterprise applications from SAP

Watch video on

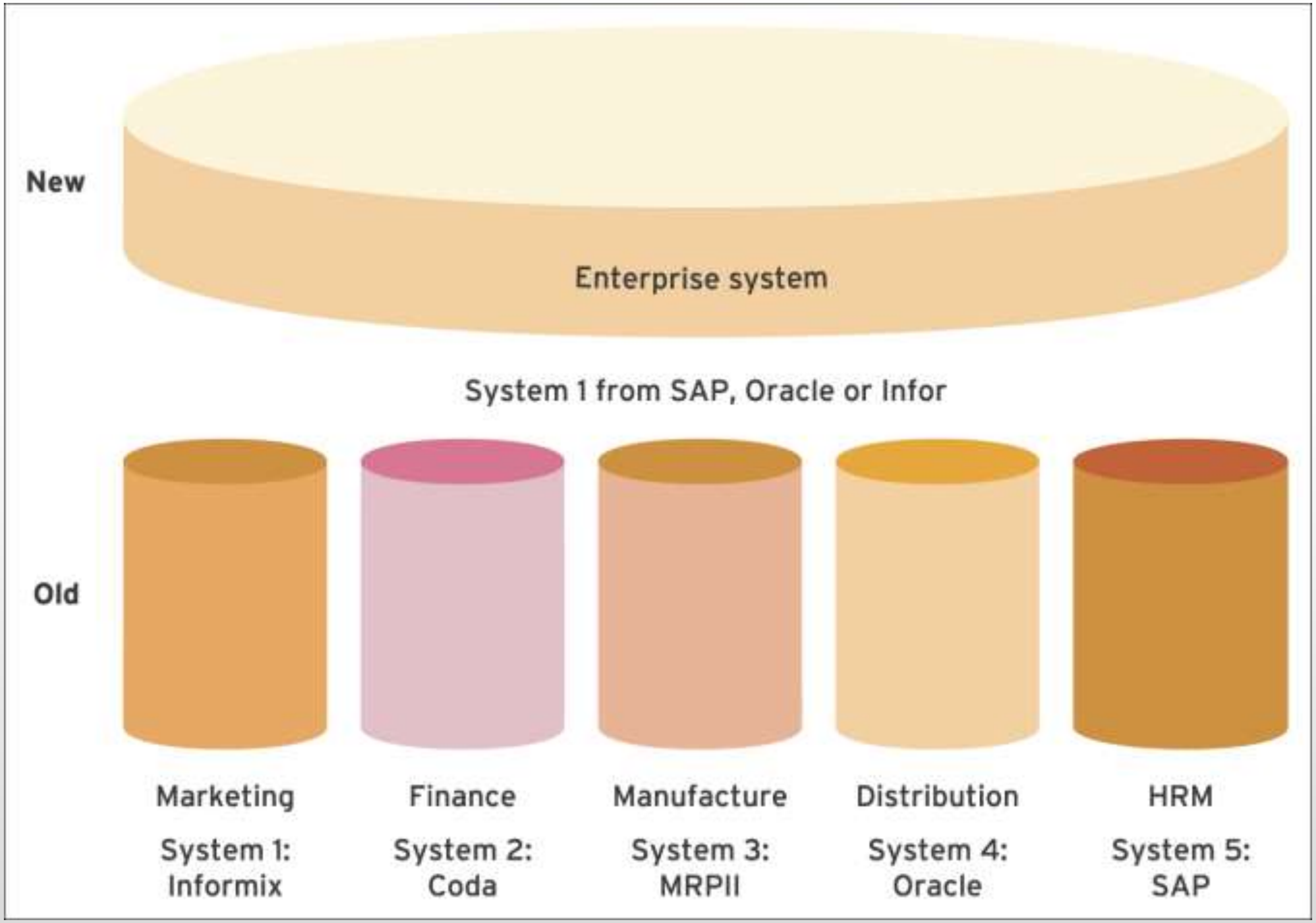
[SAP BUSINESS SUITE 7 RETAIL DEMO]

Enterprise Systems Cont.

Enterprise Resource Planning(ERP) Systems:

- ERP provides a single integrated solution from a single supplier for major business functions such as marketing, finance etc.
- Advantage of ERP is it removes ‘information islands’ (separate applications and data in different parts of the company).
- Disadvantages of ERP include high cost, major change required for implementation and need to change working methods to software.

Enterprise Systems Cont.



ERP system in comparison to separate functional applications

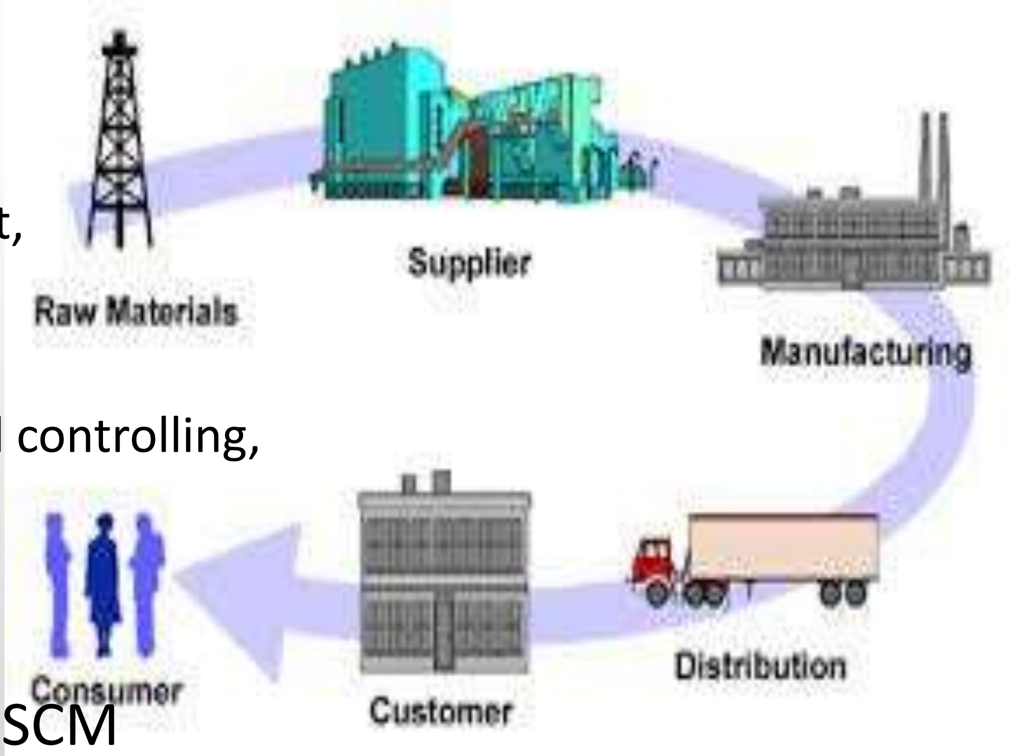
Enterprise Systems Cont.

Supply Chain Management (SCM) Systems:

A supply chain consists of the series of activities that moves materials from suppliers through the organisation to customers.

Activities involves

- Warehouse Management,
- Logistics management,
- Transportation Management,
- Demand Forecasting,
- sales forecasting,
- Manufacturing planning and controlling,
- Marketing,
- Quality Planning,
- Financial Planning, …..



Ex: QuickBase, SAP SCM, Infor SCM

Enterprise Systems Cont.

Customer Relationship Management (CRM) Systems:

CRM systems analyze customer interactions and data throughout the customer lifecycle, with the goal of improving business relationships with customers, assisting in customer retention and driving sales growth.

Designed to integrate the range of information systems that contain information regarding the customer.

Also, It collects and compile customer data through company's website, telephone, live chat, direct mail, marketing materials and social media.

CRM provides customers' personal information, purchase history, buying preferences and concerns. That enables giving individual attention to the customer.

Helps to know your Customer well and treats individually.

Enterprise Systems Cont.

Supplier Relationship Management (SRM) Systems

These systems refer to all the activities involved with obtaining items from a supplier, including procurement, transportation and warehousing.

Aim to streamline and make more effective the processes between an enterprise and its suppliers

Procurement is an important aspect of SRM as the cost of materials can represent a substantial amount of the total cost of a product or service.

Choosing a supplier is another important aspect of SRM.

SRM is a component of SCM information flow.

Lead to lower production costs and a higher quality,

but lower priced end product.

Popular Vendors:

12 Technologies, Manugistics, PeopleSoft, SAP

E-Commerce Systems

What is E-Commerce?

E-commerce is usually associated with buying and selling over the Internet

or

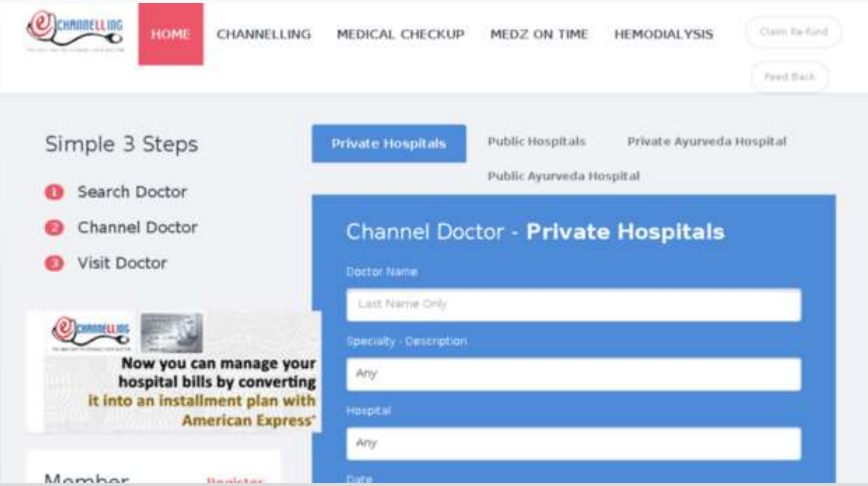
Conducting any transaction involving the transfer of ownership or rights to use goods or services through a computer-mediated network

Thomas L. Mesenbourg, *Measuring Electronic Business: Definitions, Underlying Concepts, and Measurement Plans*

E-Commerce Systems Cont.

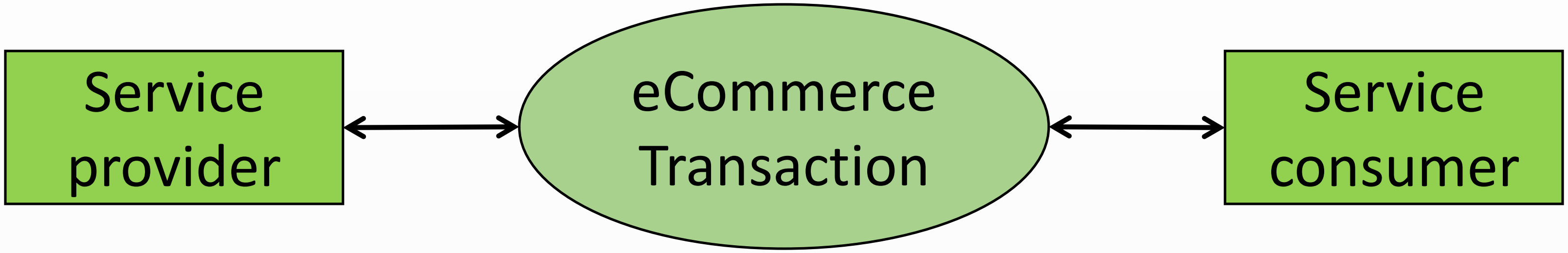
Service provider ⟷ eCommerce Transaction ⟷ Service consumer

eCHANNELLING
THE EASY WAY TO CHANNEL YOUR DOCTOR

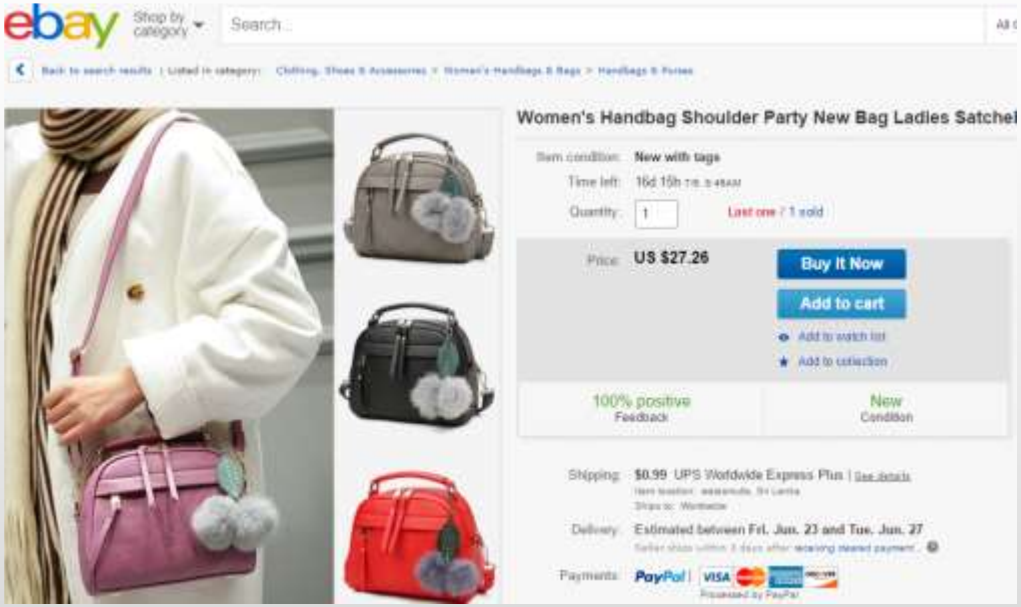


SLIIT Discover Your Future

E-Commerce Systems Cont.



ebay



Cloud based Systems

Cloud computing –

Provides availability of entire computing infrastructure (HW/SW) over the Internet.

Users purchase computing capacity **on-demand (as need arises or decreases).**

Free of Capital Expenditure, Staff and maintenance cost

Work from anywhere.

Cloud based backup and recovery solutions.

Any Information System can be converted in to a cloud based system.

SLIIT
Discover Your Future

Cloud based Systems Cont.

Cloud Service Providers:

Saleforce.com
Google Cloud Platform (GCP)
Oracle Cloud,
Amazon Cloud Drive
Microsoft Azure

Cloud based Systems Cont.

Popular Cloud based systems:

Systems that store, access data and programs (Infrastructure) over the cloud/internet.

Ex:

- **Oracle NetSuite** - a complete cloud-based ERP solution on Oracle Cloud.
- **Office 365 –** provides group of software and related services to subscribers such as email, OneDrive (Cloud storage) on MS Azure

IoT (Internet of Things) Systems

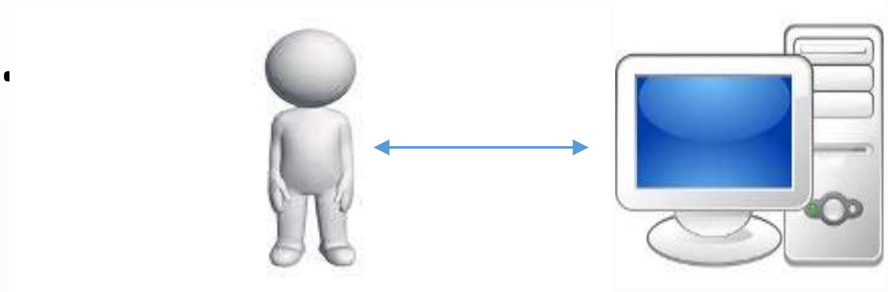
IOT Concept:

The Internet of Things (IoT) is a network of objects each embedded with sensors(things) which are connected to the Internet.

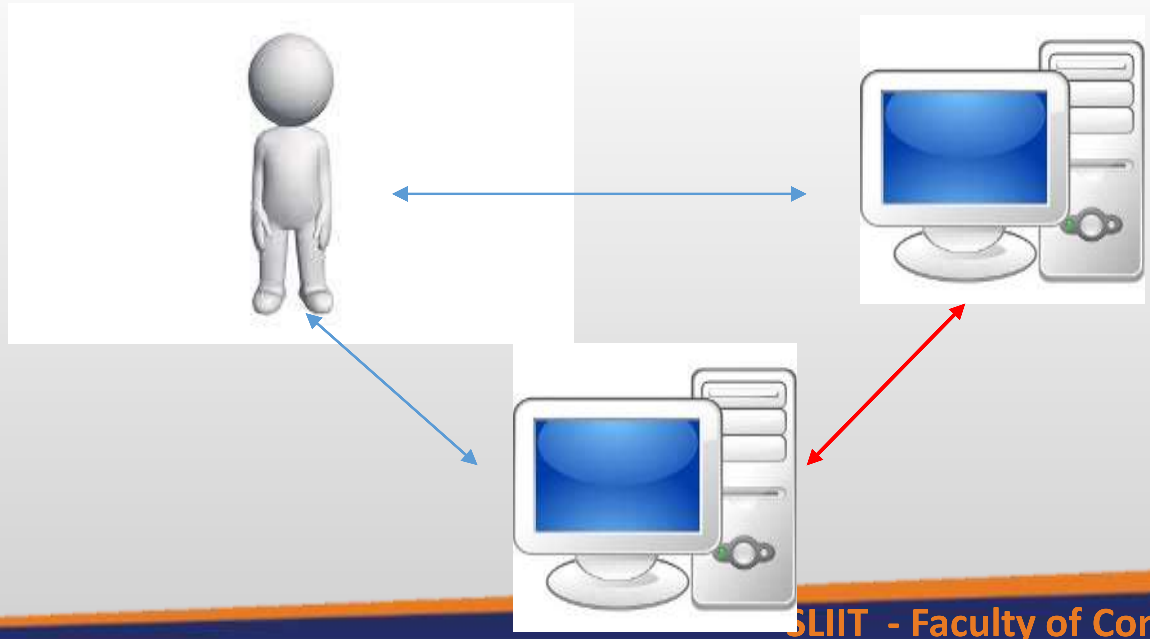
This concept is connecting any device with an on and off switch to the Internet. This includes everything from mobiles, air conditioners, headphones, wearable devices and almost anything.

IoT (Internet of Things) Systems Cont.

Without IoT...



With IoT...



IoT (Internet of Things) Systems Cont.

IoT makes anything "**SMART**" enhancing every aspect of life with the power of data collection (through sensors), AI algorithms and networks.

Ex:

Sensors connected to your refrigerator and cabinets detect when milk and your favorite cereal run low, then automatically place an order with your preferred grocer.

Sensors such as heart rate monitors, blood pressure monitors, implant in your body, track you and when it
the risk level analyze and diagnose the critical
tically send details to your doctor and the
to the hospital.

IoT systems have applications across industries through their unique flexibility and ability to be suitable in any environment.

Ex:

Transportation and logistics , Healthcare, Agriculture,
Smart environment (home, office, plant), Personal and social,
Energy and power

SLIIT
Discover Your Future

BI(Business Intelligence) Systems

Raw Data

Process

Information

meaningful and support decision making

Ex: Reports with organized information

Facts

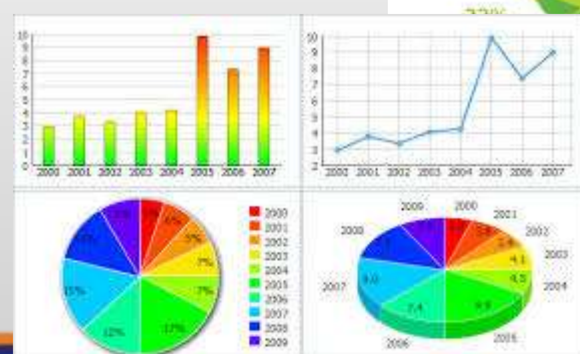
Ex: Data in an Application form

Business Analytics

Intelligence/ Knowledge

more meaningful and predictable information to better decision making (specially for strategic management for strategic planning)

Ex: Prediction on profitability of a product with existing and future market condition.



BI Systems

BI(Business Intelligence) Systems Cont.

- BI systems analyze/mine organizations raw data and Information to reveal insights of the business and build intelligence to help a business make faster and more accurate decisions.
- BI systems uses techniques such as data mining, statistics, online analytical processing, querying and reporting.
- Integrate data/information from across the enterprise. (Big Data).
- Illustrates business intelligence in the areas of customer profiling, customer support, market research, market segmentation, product profitability, statistical analysis, inventory and distribution analysis, . . .

SLIIT
Discover Your Future

BI(Business Intelligence) Systems Cont.

Existing systems:

Yellowfin

https://www.yellowfinbi.com/

Oracle Business Intelligence Enterprise Edition OBIEE

http://www.oracle.com/technetwork/middleware/bi-enterprise-edition/overview/index.html?ssSourceSiteId=opn

SAP Business Intelligence

http://www.sap.com/pc/analytics/business-intelligence.html

SISENSE

https://www.sisense.com/

Technologies behind Modern Information Systems

- Internet (WWW)
- Mobile Technology
- Sensor networks
- Global Positioning Systems (GPS)
- Cloud Computing and Virtual environments
- Object tracking technologies Ex. RFID(Radio Frequency IDentification) , NFC (Near Field Communication)
- Wireless communication
- Artificial Intelligence (AI)
- Distributed storage – Ex. Hadoop
- Payment gateways (Credit card payment)
- Web Services

Data Management within Information Systems Context

Data is one of the important aspects of any kind of Information System.

Modeling and managing data is a fundamental requirement for a successful Information System.

Reference

- K. C. Laudon and J.P. Laudon, "Management Information Systems: Managing the digital Firm", INFORMATION SYSTEMS IN BUSINESS TODAY, 13th Ed, 2014

Learning Outcomes

- **LO2:** Evaluate the information systems strategies to achieve organizational goals

Next Lecture

- Data Modeling
- (Introduction to Database and DBMS)

End of Lecture 03

Introduction to Database Modelling

Lecture - 04

Introduction to Database

To better understand what drives the design of databases, first need to understand the difference between *data* and *information*.

- What is Data?
- What is Information?
- What is Database (DB)?
- What is Database Management System (DBMS)?

File based Approach

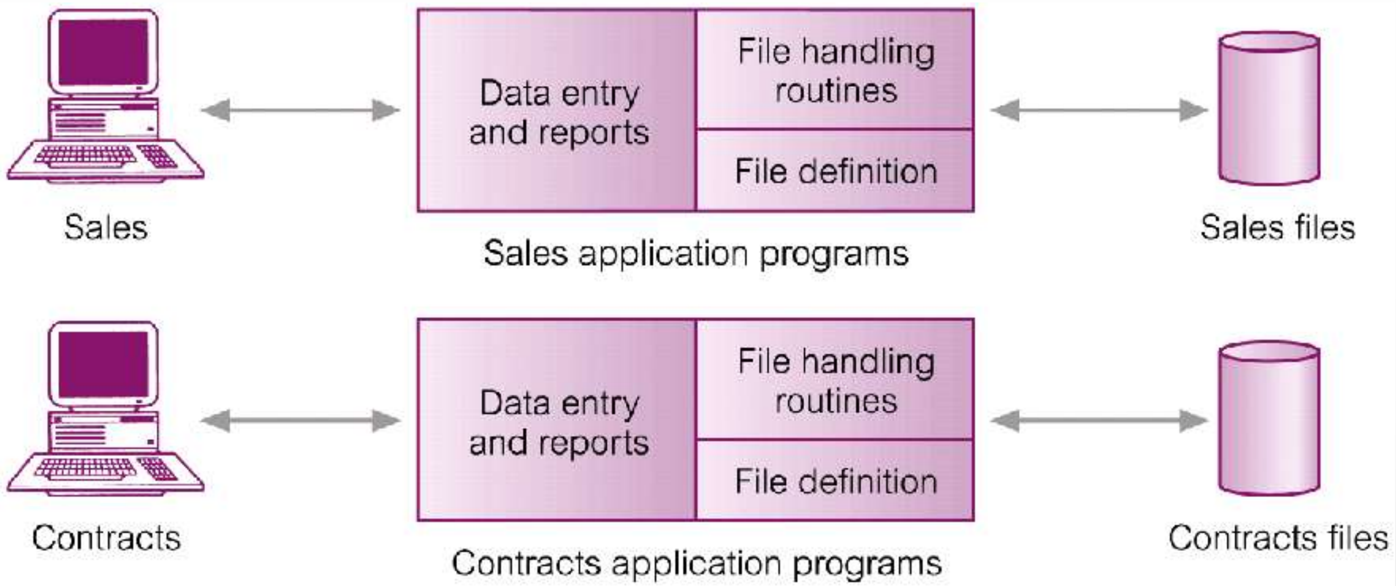
This is one of the *obsolete approach* which used to data management. Such a system would typically consist of a set of application programs (separate computer files) that perform various tasks. Each program would define and manage its own data.

Basic File Terminologies

- Data
- Field
- Record
- File

Cont.

File based Approach



Sales Files

PropertyForRent (propertyNo, street, city, postcode, type, rooms, rent, ownerNo)

PrivateOwner (ownerNo, fName, lName, address, telNo)

Client (clientNo, fName, lName, address, telNo, prefType, maxRent)

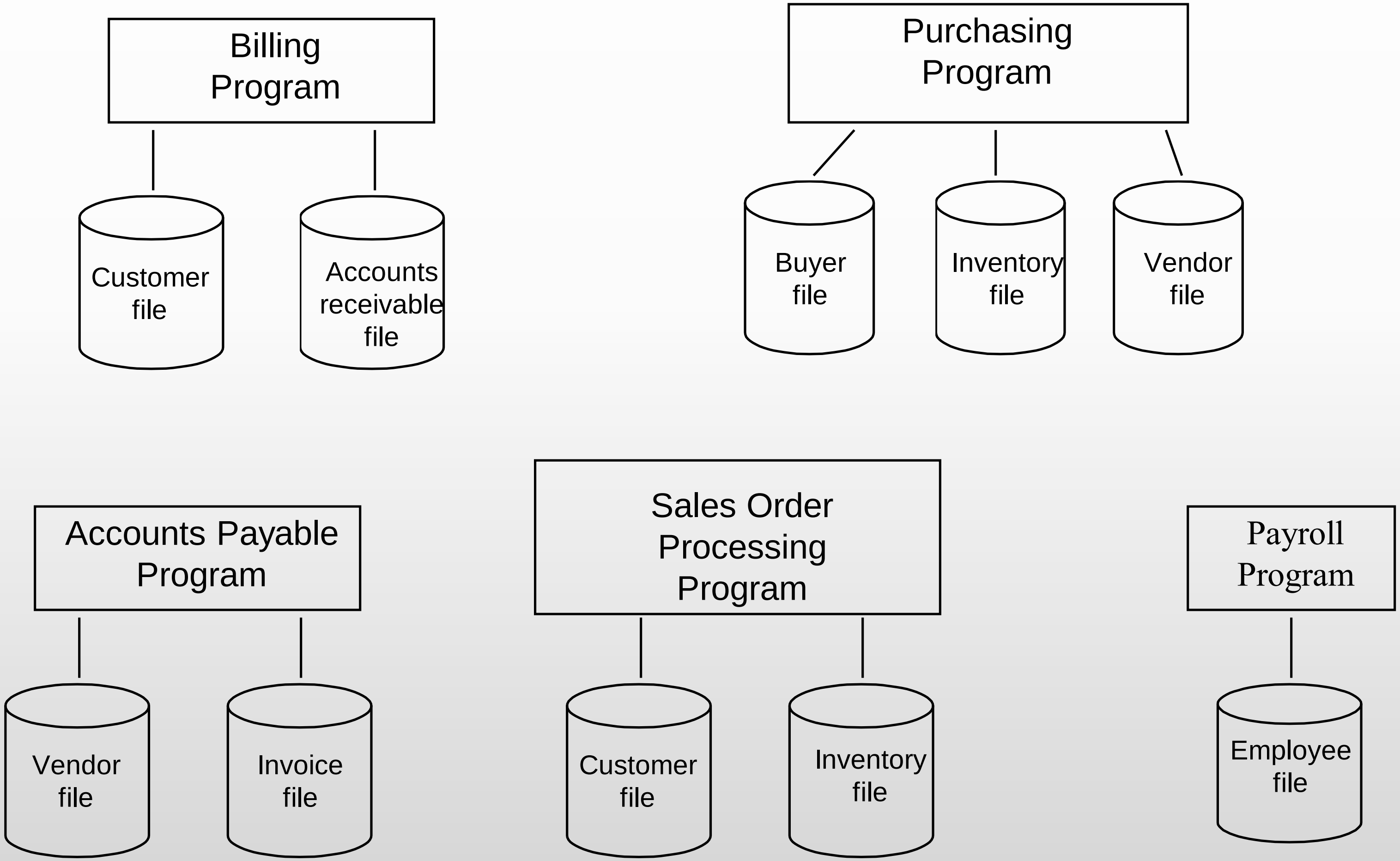
Contracts Files

Lease (leaseNo, propertyNo, clientNo, rent, paymentMethod, deposit, paid, rentStart, rentFinish, duration)

PropertyForRent (propertyNo, street, city, postcode, rent)

Client (clientNo, fName, lName, address, telNo)

File based Approach



Database Approach

Limitations of Conventional File-based Approach:

- Uncontrolled redundancy (data redundancy)
- Data inconsistence
- Inflexibility
- Limited data sharing
- Poor enforcement of standards
- Extensive program maintenance

Directed Reading *Section 1.1 and 1.2* in Elmasri and Navathe.

Database Approach

- Arose because:

  Definition of data was embedded in application programs, rather than being stored separately and independently

  No control over access and manipulation of data beyond that imposed by application programs

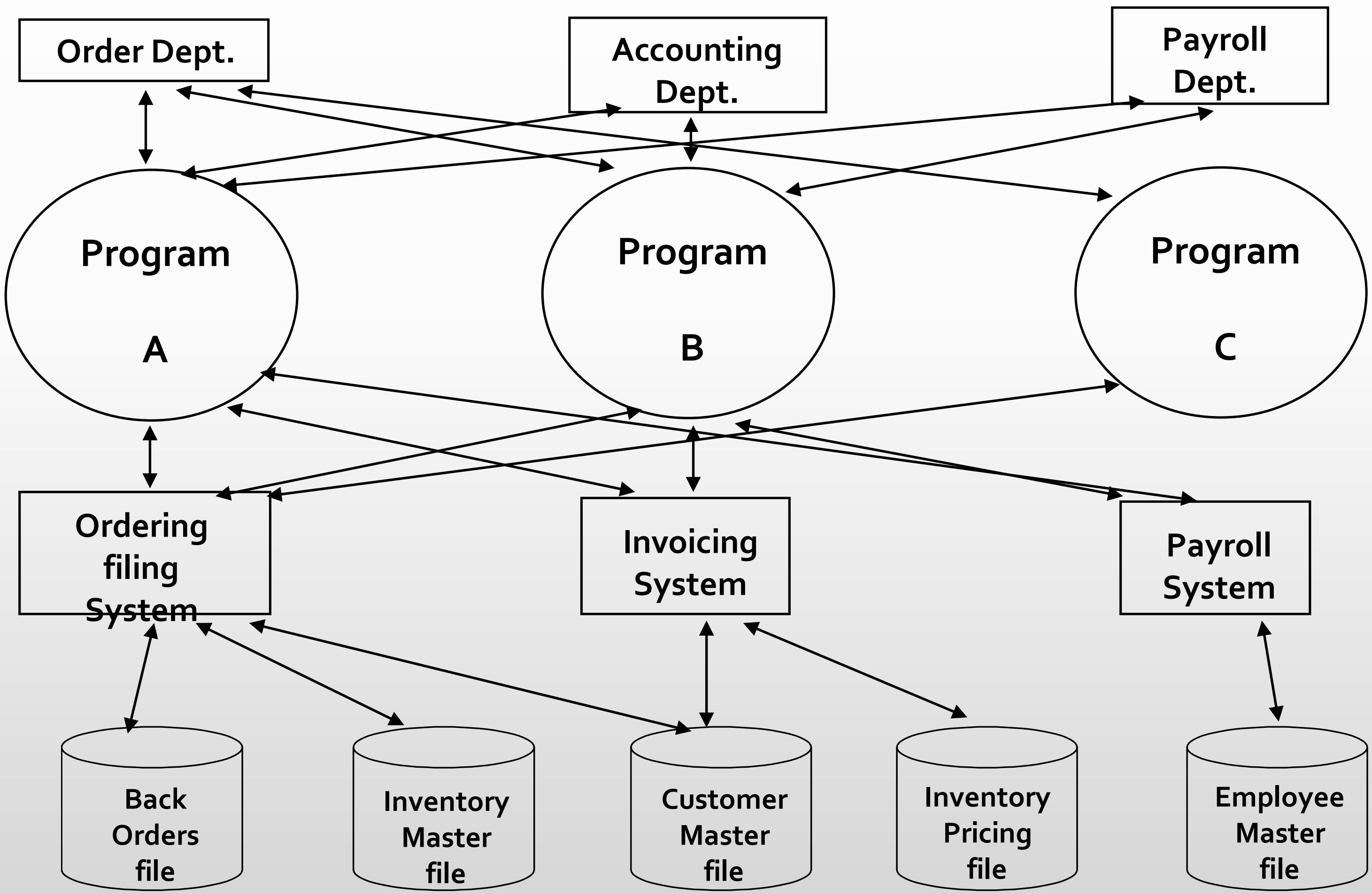
- Result:

  The Database and Database Management System (DBMS).

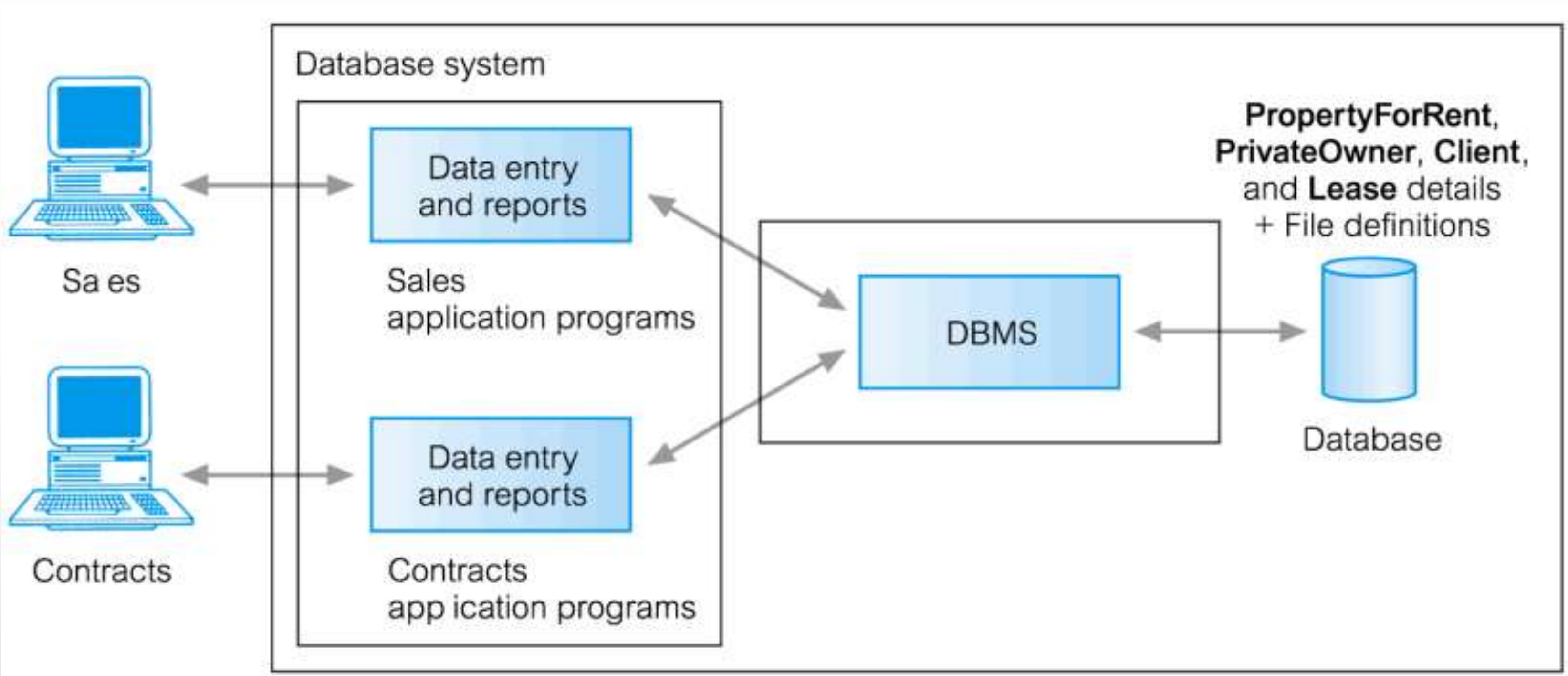
Cont.

SLIIT
Discover Your Future

Database Approach



Database Approach



PropertyForRent (propertyNo, street, city, postcode, type, rooms, rent, ownerNo)
PrivateOwner (ownerNo, fName, lName, address, telNo)
Client (clientNo, fName, lName, address, telNo, prefType, maxRent)
Lease (leaseNo, propertyNo, clientNo, paymentMethod, deposit, paid, rentStart, rentF nish)

Cont.

History in a Nutshell

- First DBMS: Bachman at General Electric, early 60's (*Network Data Model*). Standardized by CODASYL.
- Late 60's : IBM's IMS (Inf. Mgmt.Sys.) (*Hierarchical Data Model*).
- 1970: Edgar Codd (at IBM) proposed the *Relational Data Model*. Strong theoretical basis.
- 1980's -90's: Relational model consolidated. Research on query languages and data models => logic-based languages, OO DBMSs => Object-relational data model (extend DBMSs with new data types)

Cont.

Database system Environment

Database system environment normally can be considered to have five major parts.

- Hardware
- Software - Operating System Software, DBMS Software Application Program and Utilities
- People - System Administrators, DB Administrators, DB Designers, System Analyst and Programmers, End Users
- Procedures
- Data

Directed Reading *<u>Section 1.4, 1.5 and 1.6</u>* in Elmasri and Navathe.

Cont.

Database Management System

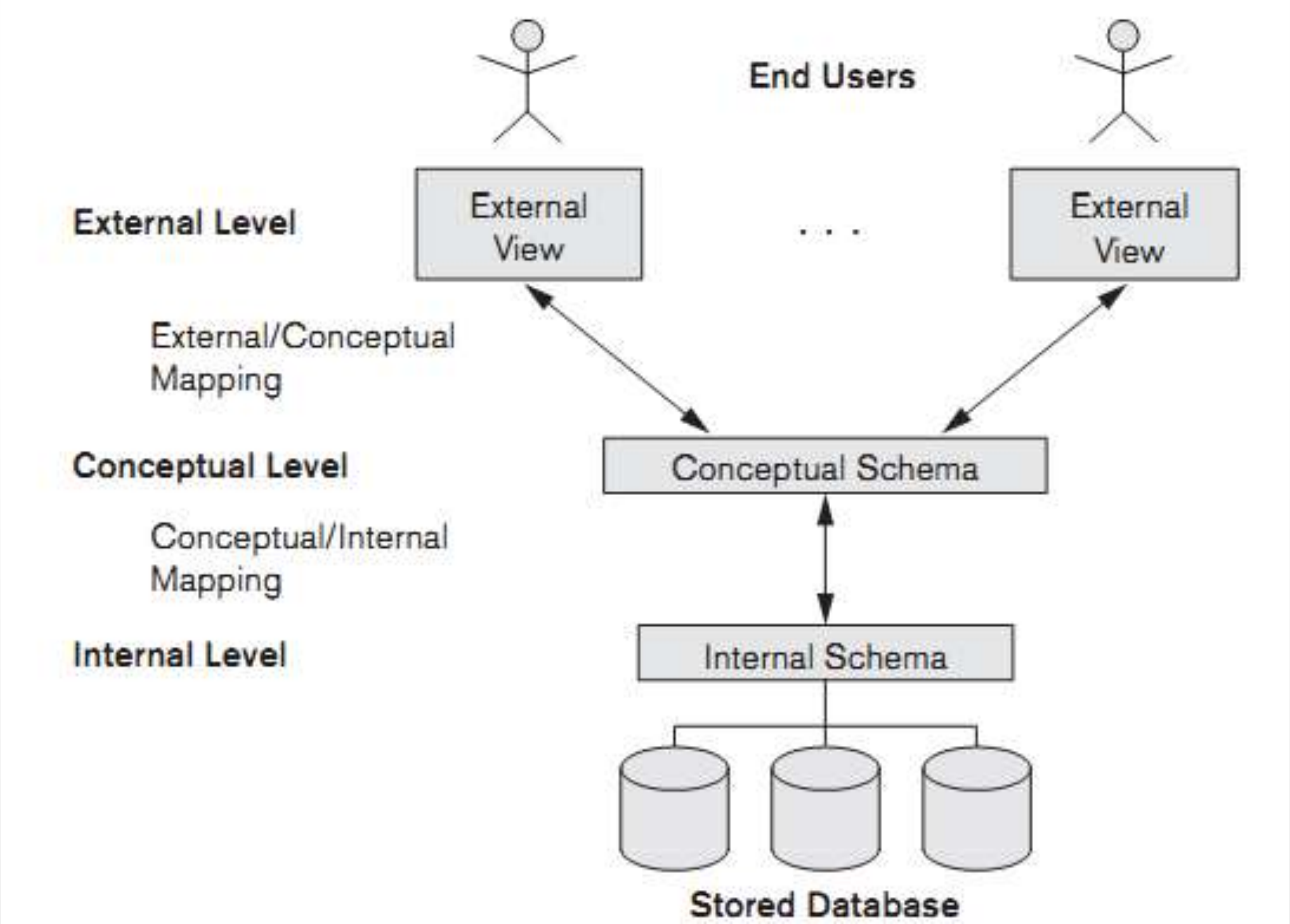
"Database is a logically coherent collection of data with some inherent meaning."

DBMS is a collection of programs that enables users to create and maintain a database

"DBMS is a general purpose software system that facilitates the processes of defining, constructing, manipulating, and sharing databases among various users and applications"

Cont.

Three-Tier Architecture of a DBMS



Three-Tier Architecture of a DBMS

The main objective of the three-schema architecture is to separate a user's views of the database from the way that the data is physically represented.

- All users should be able to access same data
- A user's view is immune to changes made in other views
- Users should not need to know physical database storage details
- DBA should be able to change database storage structures without affecting the users' views
- Internal structure of database should be unaffected by changes to physical aspects of storage
- DBA should be able to change conceptual structure of database without affecting all users

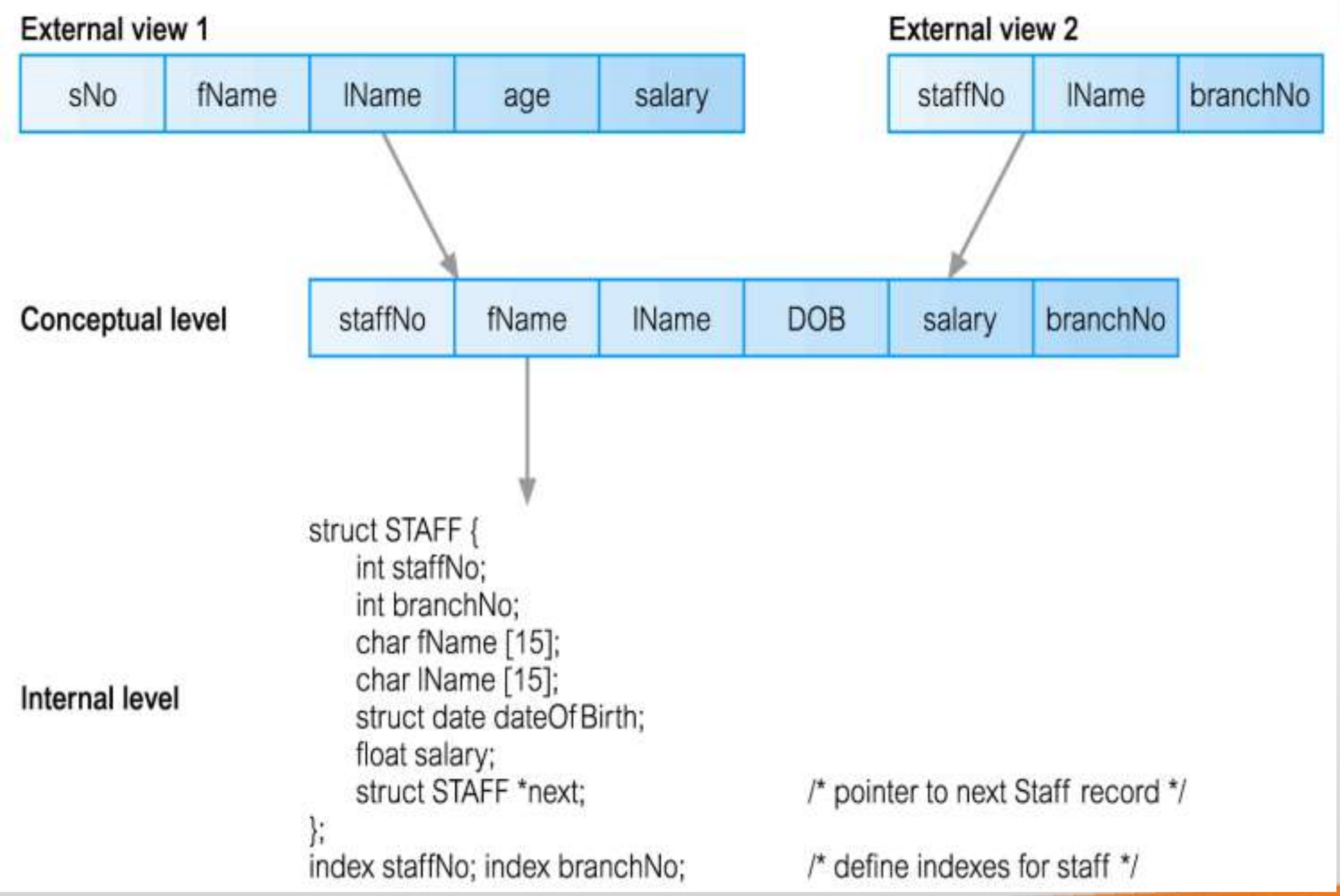
Three-Tier Architecture of a DBMS

Defines DBMS schemas at three levels:

- **Internal schema** at the internal level to describe physical storage structures and access paths. Typically uses a *physical* data model.
- **Conceptual schema** at the conceptual level to describe the structure and constraints for the *whole* database for a community of users. Uses a *conceptual* or an *implementation* data model.
- **External schemas** at the external level to describe the various user views. Usually uses the same data model as the conceptual level.

Cont.

Three-Tier Architecture of a DBMS

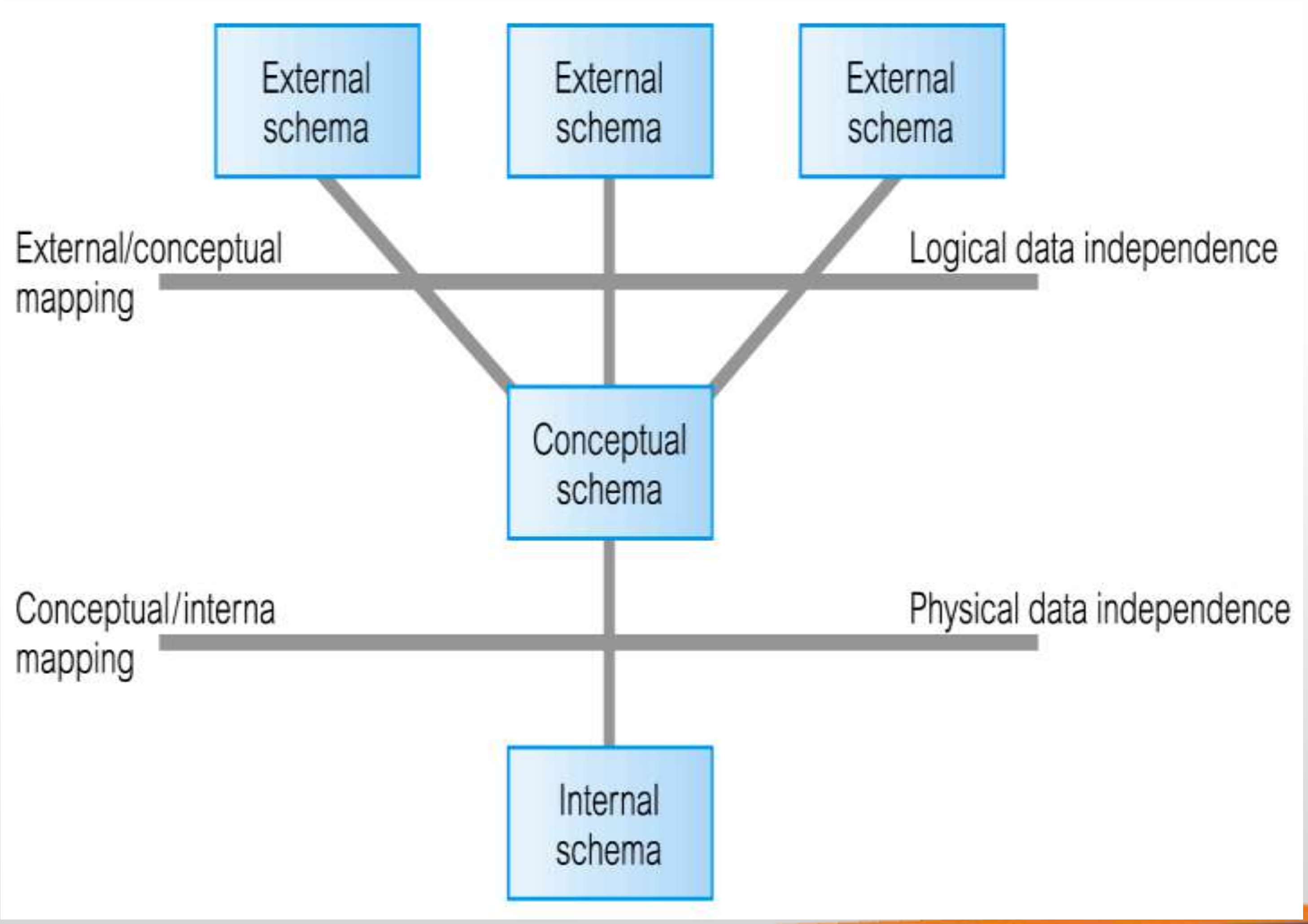


Data Independence

Data Independence is the capacity to change the schema at one level of a database system without having to change the schema at the next higher level

- **Logical Data Independence**: Change *conceptual schema* without having to change *external schemas* and their application programs.
- **Physical Data Independence**: Change *internal schema* without having to change *conceptual schema*.

Data Independence



Database Design

Why is Database Design important?

Without careful planning you may create a database that…

- contains unnecessary data which occupies the storage space
- omits data required to create critical reports
- takes a considerable time to respond to user questions
- produces results that are incorrect or inconsistent
- is unable to accommodate changes in the user's requirements

A poorly designed database will require more time in the long-term…!

Database Design process

1.Requirements Analysis

What does the user want?

2.Conceptual Database Design

Model Data requirements using a Conceptual Data Model --> ER model

3.Logical Database Design

Model Data requirements using a Logical Data Model → Relational Data Model

4.Schema Refinement

Fine tune the result

5.Physical Database Design

Implementation of the design using a Database Management System

6.Security Design

Implement Controls to ensure security and integrity

Data Modeling

What is a Data Model?

- A data model focuses on what data should be stored in the DB and how it should be organized
- Without representing the data as a database would see it, a data model represents the data as the user sees it in the ‘real world’
- A data model can be considered similar to an architect's building plan

The goal of the data model is to make sure that ***all data objects*** *required by the database are completely and accurately represented*

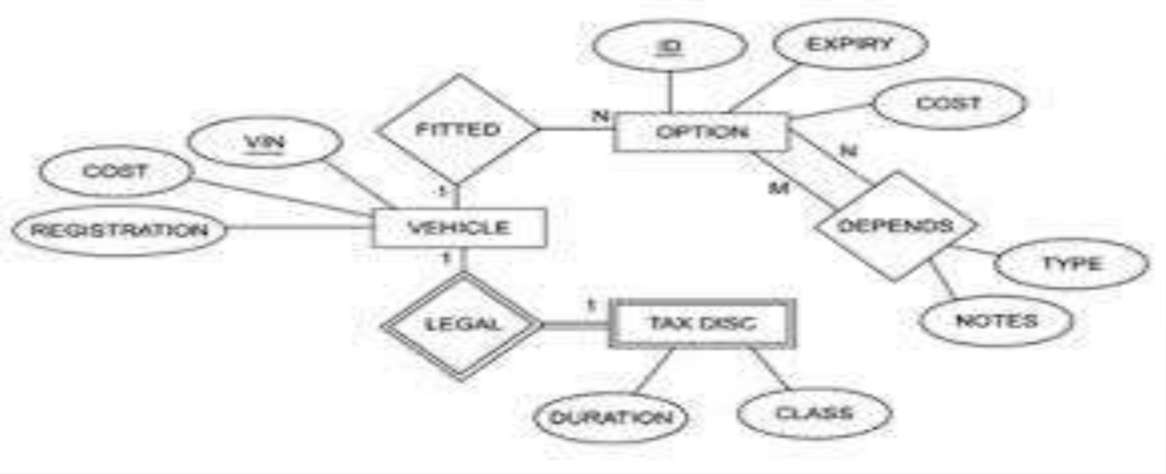
Types and Examples of Data Models

1) High Level Conceptual Data Models

- Provide concepts for presenting data in ways that are close to the way people perceive data.

Ex: Entity Relationship Model/Diagram

Entity Relationship (ER) Model



Represents data as 'real world' Objects (Entities)

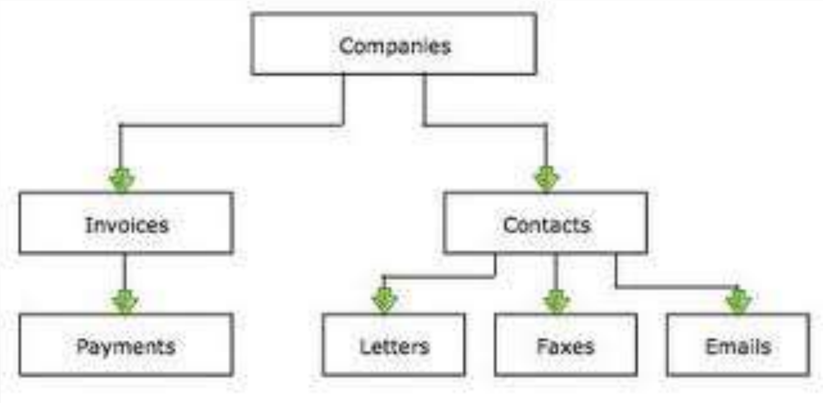
Types and Examples of Data Models …

2) Record-based Logical Data Models

- Provide concepts users can understand but are not too far from the way data is stored in the computer

Ex: Hierarchical Model,
Network Model,
Relational Model,
Object Oriented Model, etc…

Hierarchical Model



Represents data as a hierarchical tree structure. Each branch of the hierarchy represents a number of related records.

Network Model

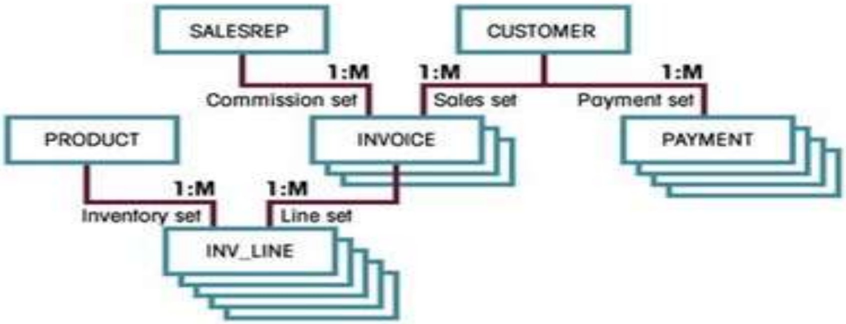


FIGURE 1.10 A NETWORK DATABASE MODEL

Represents data as record types

Relational Model

PubID	Publisher	PubAddress
03-4472822	Random House	123 4th Stree, New York
04-7733903	Wiley and Sons	45 Lincoln Blvd, Chicago
03-4859223	O'Reilly Press	77 Boston Ave, Cambridge
03-3920886	City Lights Books	99 Market, San Francisco

AuthorID	AuthorName	AuthorBDay
345-28-2938	Haile Selassie	14-Aug-92
392-48-9965	Joe Blow	14-Mar-15
454-22-4012	Sally Hemmings	12-Sep-70
663-59-1254	Hannah Arendt	12-Mar-06

ISBN	AuthorID	PubID	Date	Title
1-34532-482-1	345-28-2938	03-4472822	1990	Cold Fusion for Dummies
1-38482-995-1	392-48-9965	04-7733903	1985	Macrame and Straw Tying
2-35921-499-4	454-22-4012	03-4859223	1852	Fluid Dynamics of Aquaducts
1-38278-293-4	663-59-1254	03-3920886	1967	Beads, Baskets & Revolution

Represents data as relations/tables

Properties of a good Data Model

- Completely and accurately represents the **data requirements** of the end users/application
- Uses simple notations and natural language so it can be easily understood by the end user
- Detailed enough to be used by a database designer to build the database.
- Eliminates redundant data
- Independent of any hardware and software constraints
- Adapts to changing requirements with minimum of effort

SLIIT
Discover Your Future

Database Design process (Revisited)

1.Requirements Analysis

What does the user want?

2.Conceptual Database Design *

Model Data requirements using a Conceptual Data Model --> ER model

3.Logical Database Design *

Model Data requirements using a Logical Data Model → Relational Data Model

4.Schema Refinement *

Fine tune the result

5.Physical Database Design

Implementation of the design using a Database Management System

6.Security Design

Implement Controls to ensure security and integrity

* Will be discussed in this module

Step 1: Requirements Analysis

The goals of the requirements analysis are:

- to determine the **data requirements** of the database in terms of objects
- to identify and describe the information about these objects
- to identify the relationships among these objects
- to identify different transactions that will be performed on the database
- to identify performance, integrity, security or administrative constraints to be imposed on the database
- to identify design and implementation constraints if any (ex: specific technologies, hardware and software, programming languages, policies, standards, etc…)

Step 1: Requirements Analysis…

Information needed for the requirements analysis can be gathered in several ways:

- **Review of existing documents** –

Can become familiar with the organization/ activity you need to model by reviewing the existing documentation. Ex: existing forms and reports, written guidelines, job descriptions, personal narratives, etc…

- **Interviews with end users** –

Can organize individual/group meetings with the end users. Can use a blackboard, flip chart, or overhead transparencies to record information gathered from the interviews.

- **Review of existing automated systems** –

Can review the system design specifications and documentation, if the organization already has an automated system

Step 2: Conceptual Database Design

The information gathered in the requirements analysis phase is used to create a:

high-level description of the data in a conceptual data model or *Semantic Data Model.*

Eg. ER Model

Step 3: Logical Database Design

- In this step, we determine the DBMS to implement the database & also the data model.
- We utilize the conceptual schema created in the previous step and convert it into a schema of a particular data model* (e.g. Relational Data Model)
- *we will cover this in the next two lectures.

Step 4: Schema Refinement

The schema created by the logical database design phase is further refined for potential problems such as redundancies.

Step 5: Physical Database Design

In this step, performance criteria are taken into consideration and further enhancements to the schema & creation of indexes are considered.

Step 6: Security Design

Different user groups and their roles are identified.

Appropriate levels of access are then provided to the data ensuring that users have access to only the necessary data.

Eg. Bank

- Customer – read access
- Teller – read/update (limited)
- Manager – read/update

End of Lecture - 04

Questions ?

Data Modelling using ER diagrams

Lecture - 05

Learning Outcome

- **LO3**: Model data requirements using data models

Conceptual Modeling
ER Diagrams

- Many versions of ER-diagrams; differ both in their appearance and in their meaning
- We will use the version appearing in the book (Database Management Systems by Elmasri Navathe)
- Have a formal semantics (meaning) that must be thoroughly understood, in order to create correct diagrams

Entity-Relationship (ER) Model

Requirements gathering

↓

Conceptual database design

↓

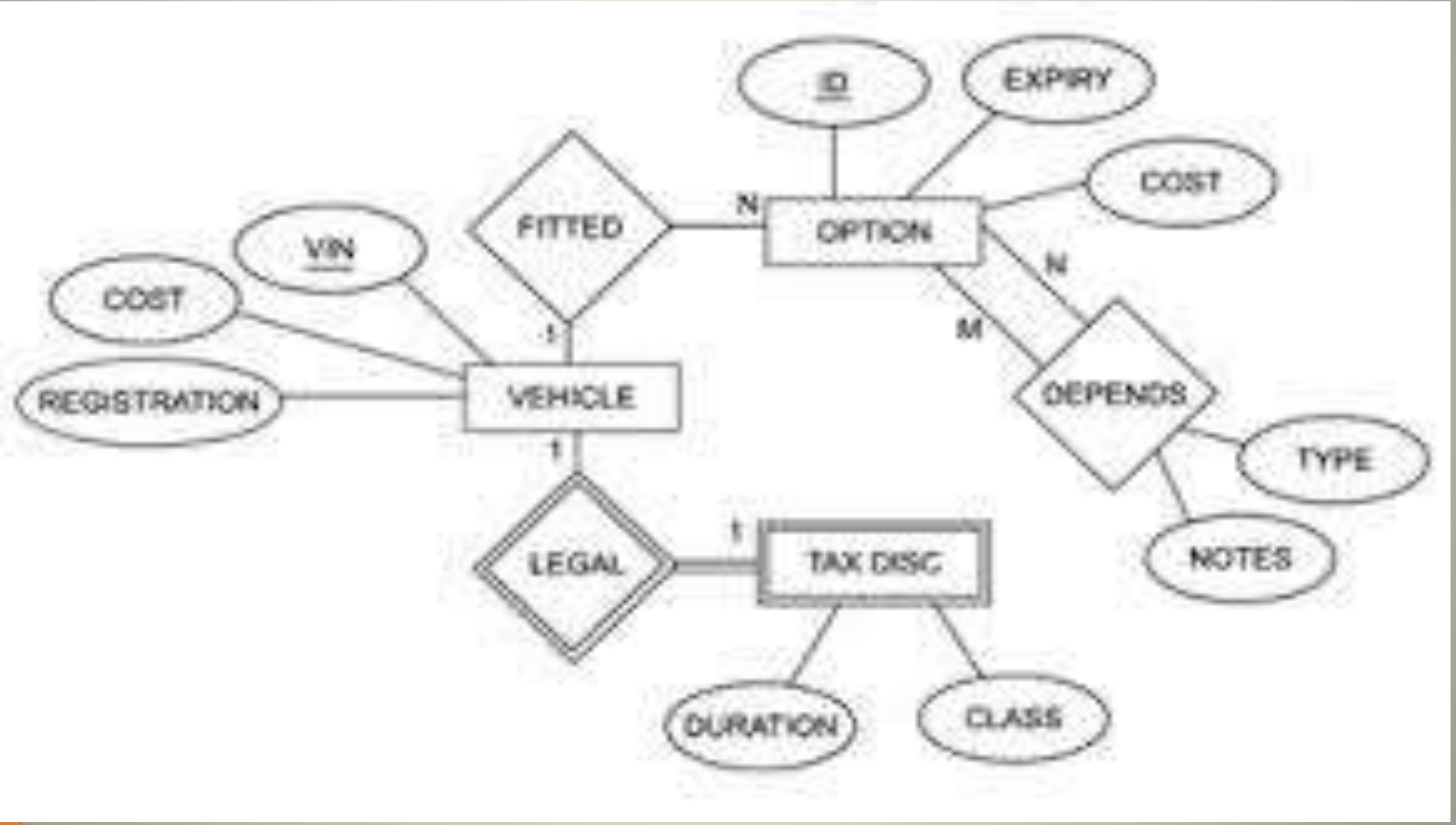
high level data model(ER)created

The 2 main constructs of the ER model:

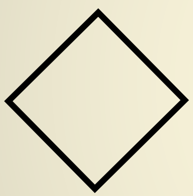
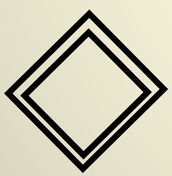
Entities ,Relationships.

different books use different graphical notation.

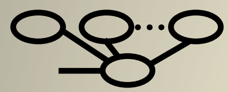
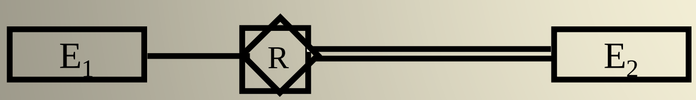
ER Diagrams


ID
EXPIRY
COST
FITTED
N
OPTION
N
VIN
COST
1
M
VEHICLE
DEPENDS
REGISTRATION
1
TYPE
LEGAL
1
TAX DISC
NOTES
DURATION
CLASS

E-r Notation

SYMBOL	MEANING
	ENTITY
	WEAK ENTITY
	RELATIONSHIP
	IDENTIFYING RELATIONSHIP
	ATTRIBUTE
	KEY ATTRIBUTE

E-r Notation

SYMBOL	MEANING
	MULTIVALUED ATTRIBUTE
	COMPOSITE ATTRIBUTE
	DERIVED ATTRIBUTE
	TOTAL PARTICIPATION OF E_2 IN R
	CARDINALITY RATIO 1:N FOR E_1:E_2 IN R
	STRUCTURAL CONSTRAINT (min, max) ON PARTICIPATION OF E IN R

Entity

- An **entity** is an object in the real world with an independent existence (place, person, car, house, company, job, university, etc).

- A collection of similar entities is called an **entity set**.

- Graphically,

ENTITY

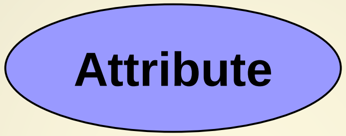
STUDENT

  - First letter of each word in the entity name is uppercase
    - E.g., Student

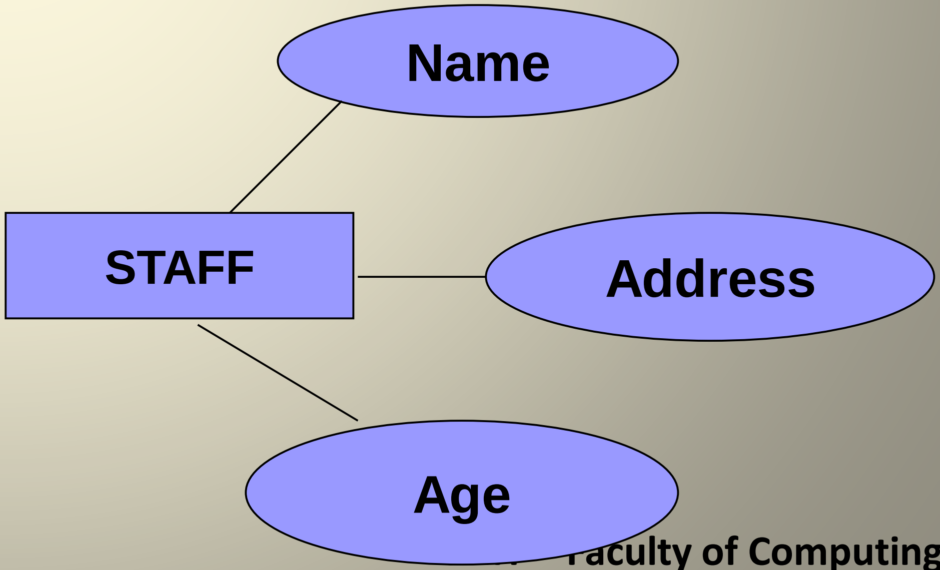
  - Normally an entity is named using a '**noun**' or 'noun phrase'

Attributes

- An entity is described using a set of **attributes**
- Graphically, **Oval**.



- For example staff has 3 attributes
  - Name
  - Position
  - Salary



BOC Network

- BOC is the largest bank network having over 5,000 branches and over 500,000 customers. Customers can open accounts in any bank branch.
- Identify
  - Entities
  - At least 3 attributes /entity

SLIIT Student Registration

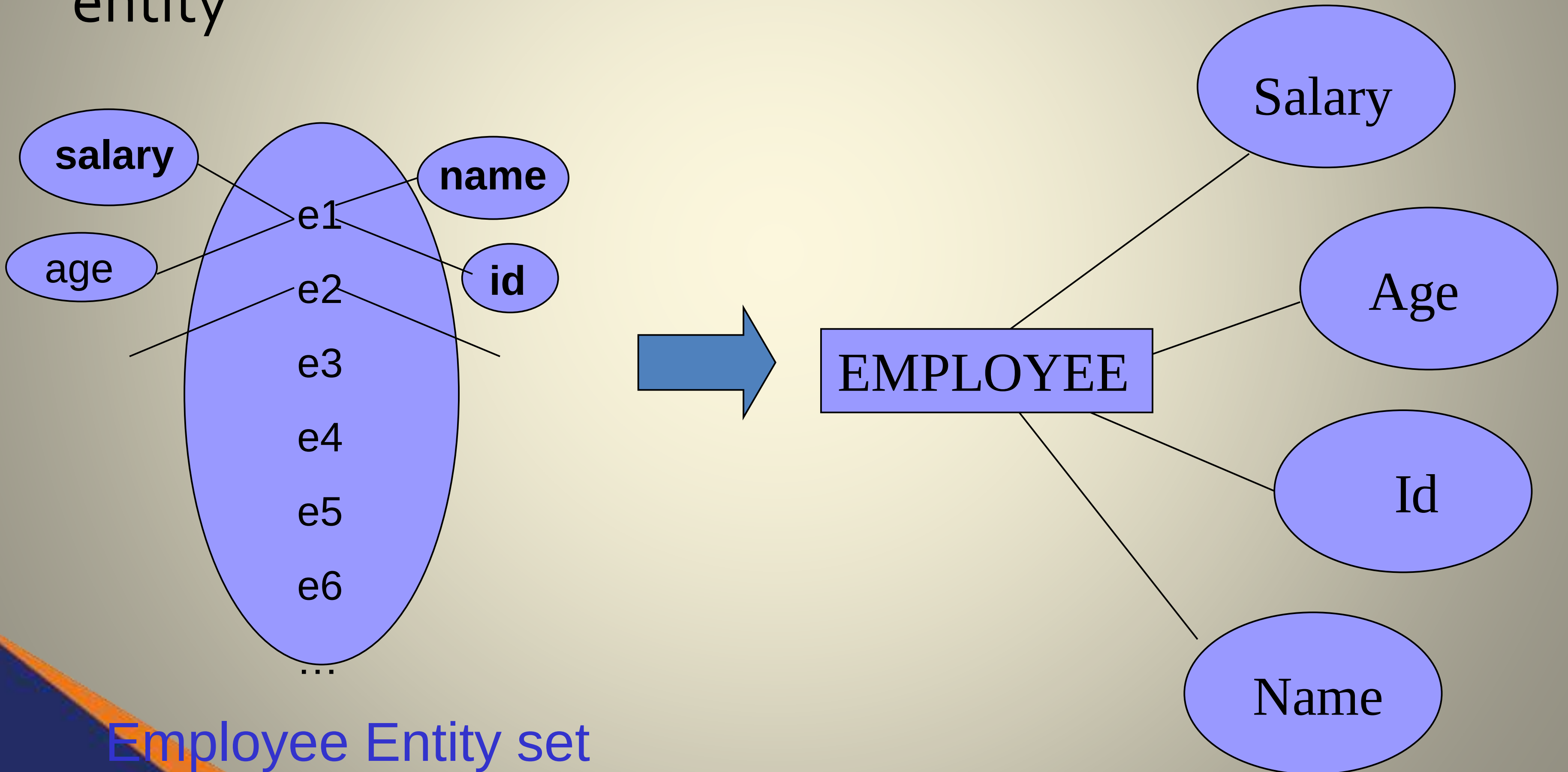
New students are required to produce their national ID card number, name, address, age, date of birth and gender during the registration process. they are also required to select the degree program of their choice.

The programs are selected based on the entry criteria, specialty, duration, fees etc.

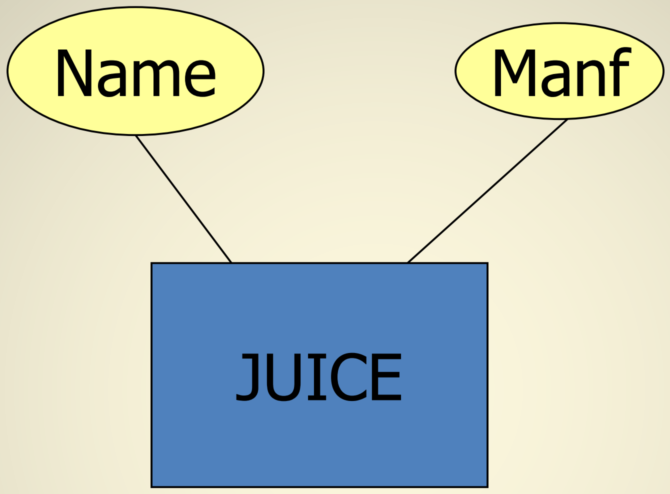
- Entity ?
- Attributes ?

Entity Set

Example: *name*, *id*, *age* & *salary* are attributes in Employee entity



Example



- Entity set Juice has two attributes, name and manf (manufacturer).
- Each Juice entity has values for these two attributes, e.g. (JustJUice, Anheuser-Busch)

Domain of an Attribute

The **domain** of an attribute specifies the set of possible values that the attribute can have.

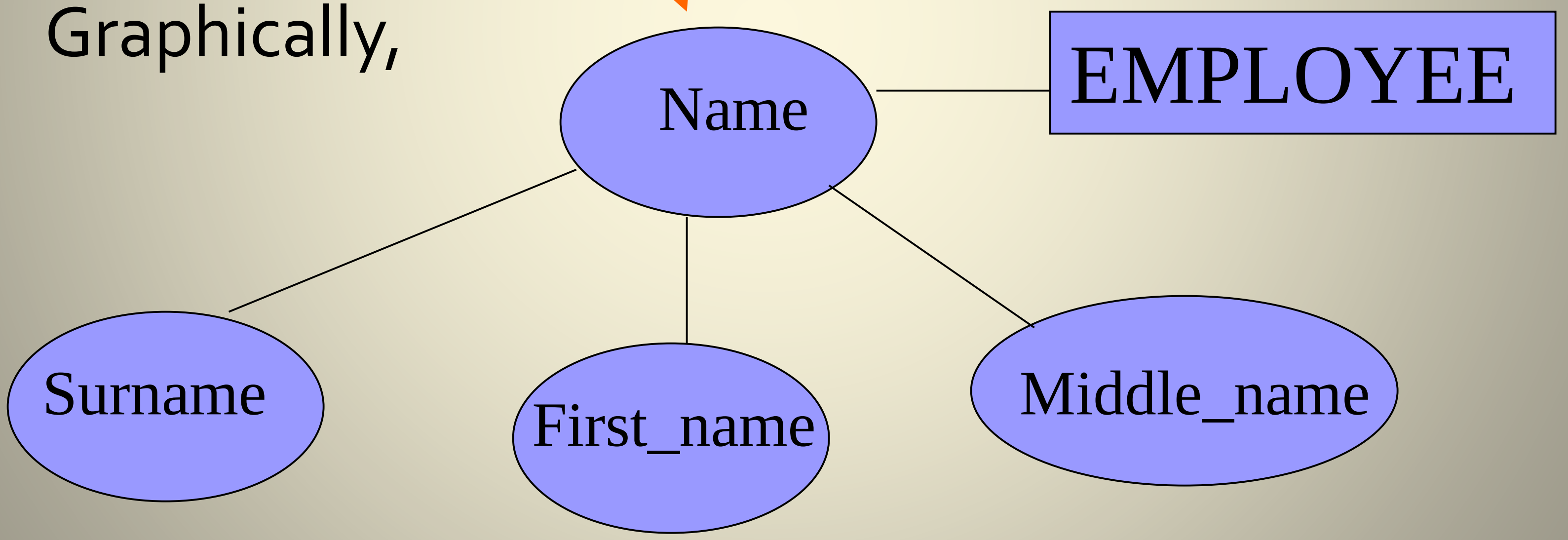
age : integer values from 0 - 120

name: 20 character string

Composite Attributes

- **Composite attributes** can be divided into smaller subparts.

Graphically,



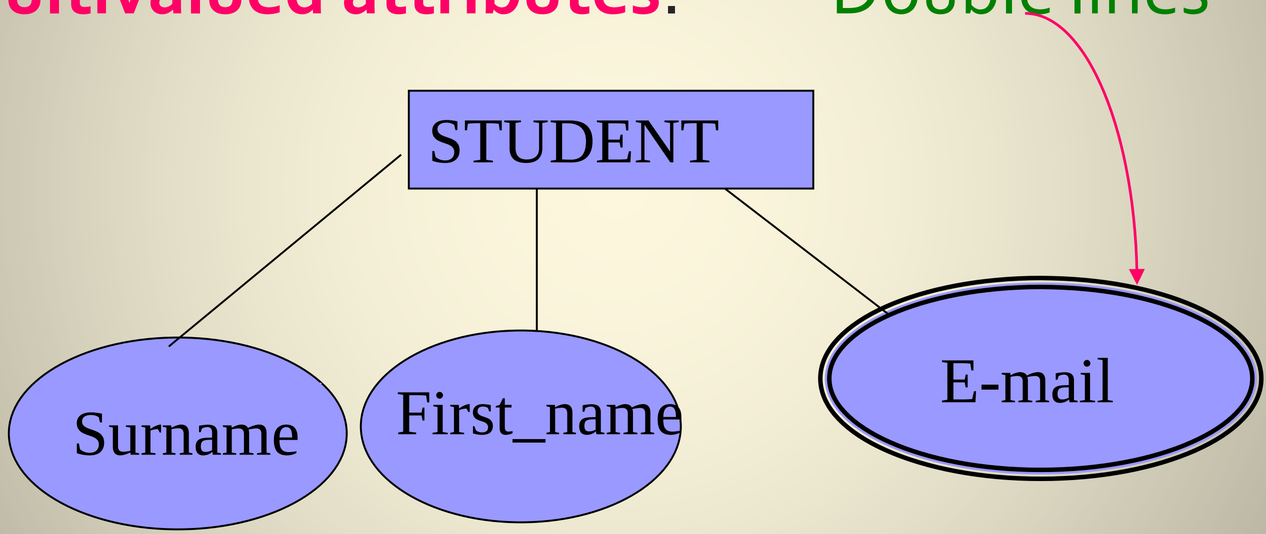
Your Turn!!

Think about the library database.

- Identify entities , attributes and domain of attributes

Multi valued Attributes

- Attributes containing multiple values are called **multivalued attributes**. Double lines



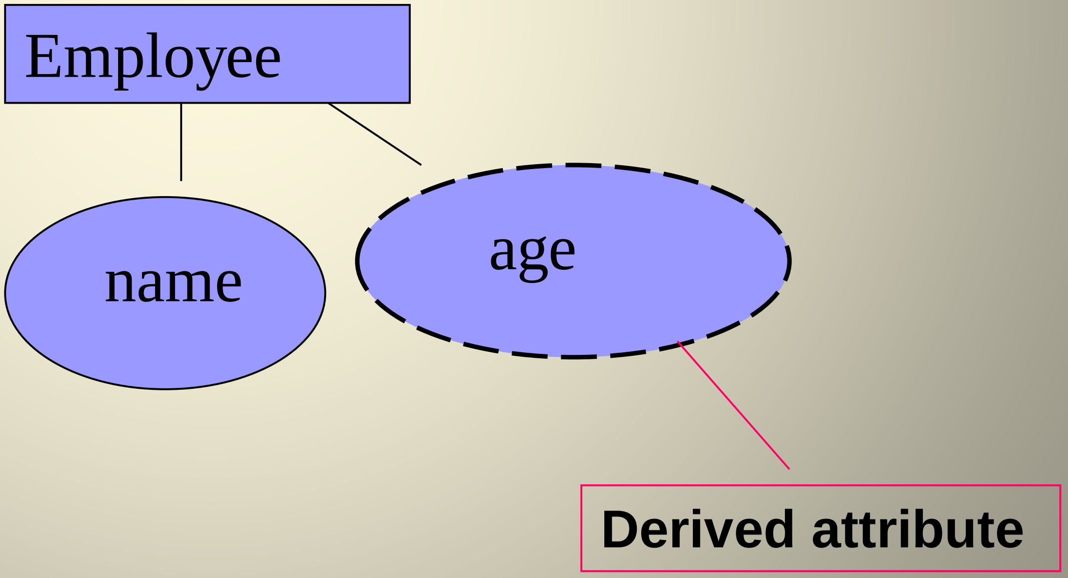
gmail , yahoo, SLIIT e-mail address

Your Turn!!

- Give an example for a multi valued attribute in library database

Derived Attributes

- Some attributes can be derived - called **derived attributes**. Dotted line
- For example, *age* is derived from date of birth & current date.



An exercise

stuNum	stuName	stuMajor	stuDob	stuHrs	stuYr	stuGpa	stuAge
2103456	Aunt Annie	IT, Fine Art	14 Feb 1990	23	2	3.26	26
3006381	Bert Baxter	IT	23 Jun 1992	33	3	2.59	24
2007824	Corinne Creevey	IT, Education	5 Oct 1991	27	1	3.59	25
7650802	David Doo	Arts	28 Aug 1994	16	1	0	22
9635962	Eli Edwards	Arts, Teaching	15 Nov 1992	35	2	2.73	24
2113843	Fiona Fawcett	IT	30 Jul 1990	28	2	3.27	26

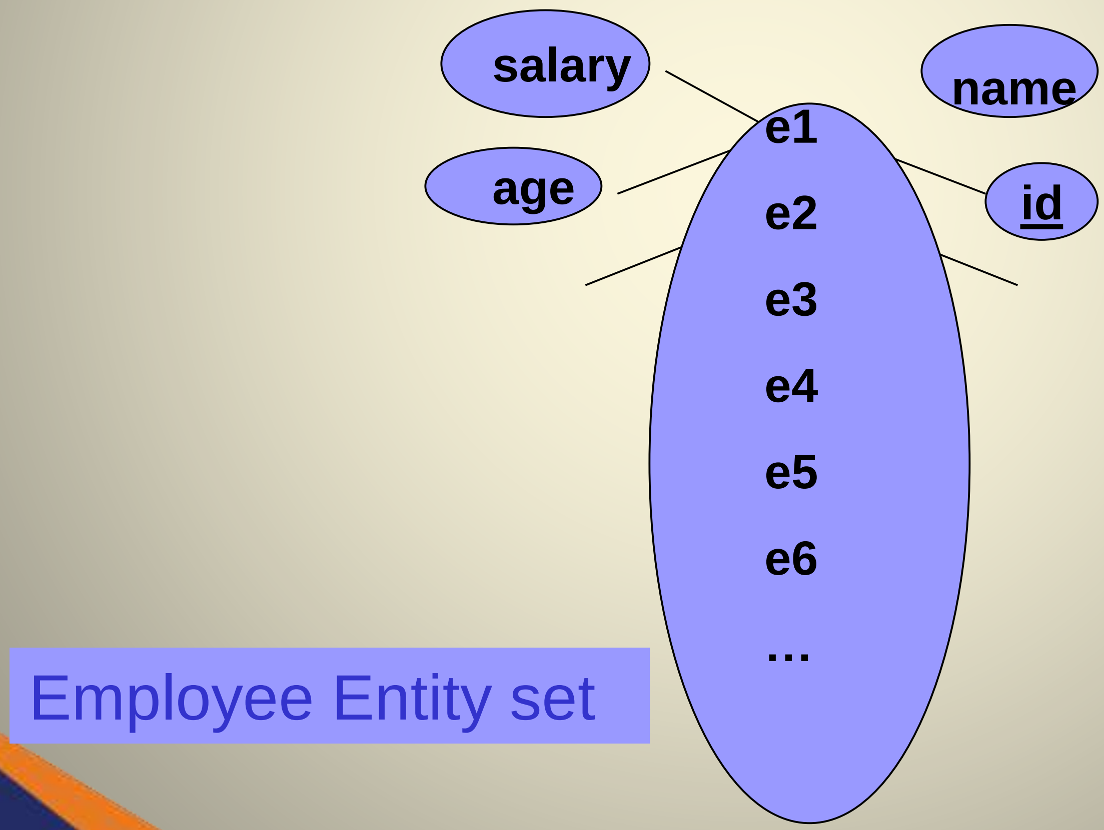
Tick where applicable

Attribute	simple	single-valued	composite	multivalued	derived
stuNum	✓	✓	✗		
stuName	(✓)		(✓)		
stuMajor				✓	
stuDob	✓				
stuHrs		✓			✓
stuYr		✓			
stuGpa					✓
stuAge					✓

Key Attributes

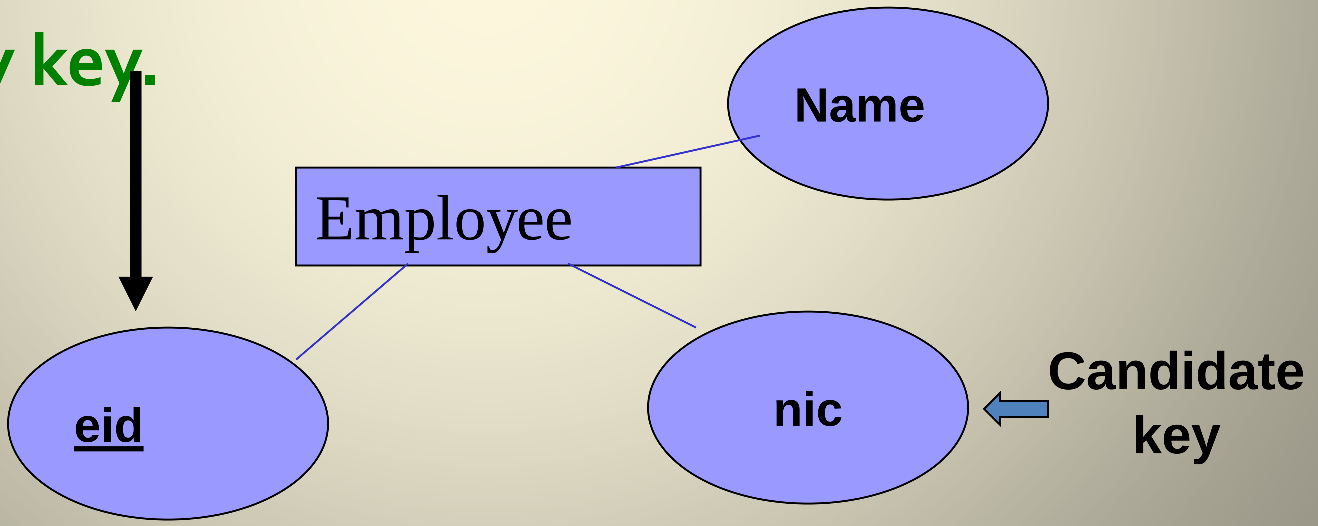
- **key attribute** - Minimal set of attributes which uniquely identify an entity in the entity list.

Underlined



Primary Key

- There can be multiple key attributes called **candidate keys** in a single entity.
- A single candidate key is designated as the **primary key.**



Composite Key

- Sometimes, a group of attributes make up the key. This is called a **composite key**.

- Example :

Composite key = (student no + Unit Number + marks)

ST ID	Unit ID	Marks
IT1601	IT103	85
IT1601	IT104	78
IT1602	IT103	72
IT1603	IT104	82

Super Key

- Any other set of attributes that uniquely identify a tuple is called the **superkey** of a relation

Student (SID, Name, Address, Contact, GPA)

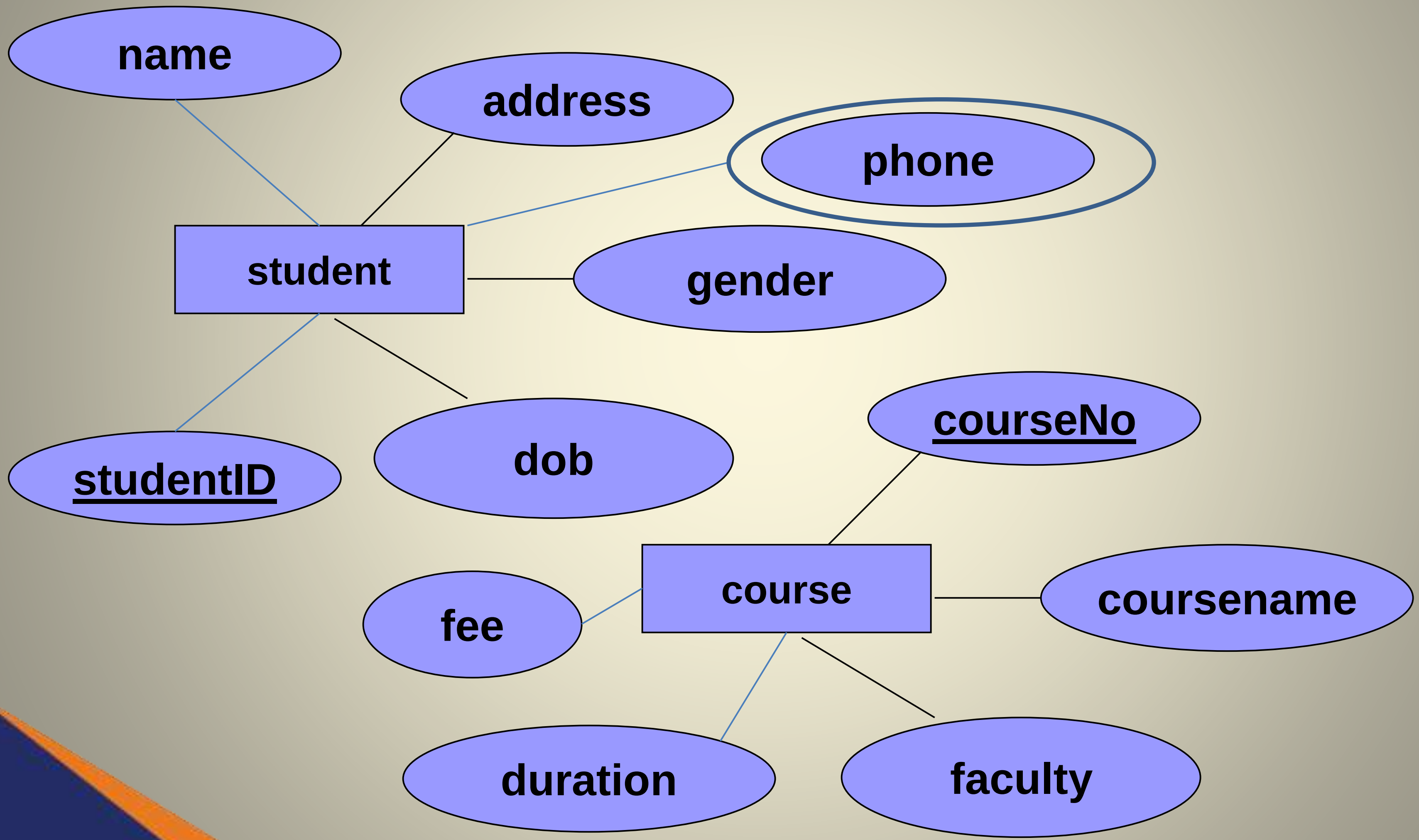
- What is the minimal set of attributes that uniquely identify the relation ?
  - SID =Referred to as Primary Key

- (SID + Name) Is this unique?
  - Yes, but NOT the minimal set
- Referred to as Super Key

Students should enroll for courses. Student need to provide studentID number, name, address , date of birth, gender, contact number when he/she is going to register for a course. University keeps track of course number , course name , offering faculty ,duration and course fee of each course.

How do you represent this scenario ?

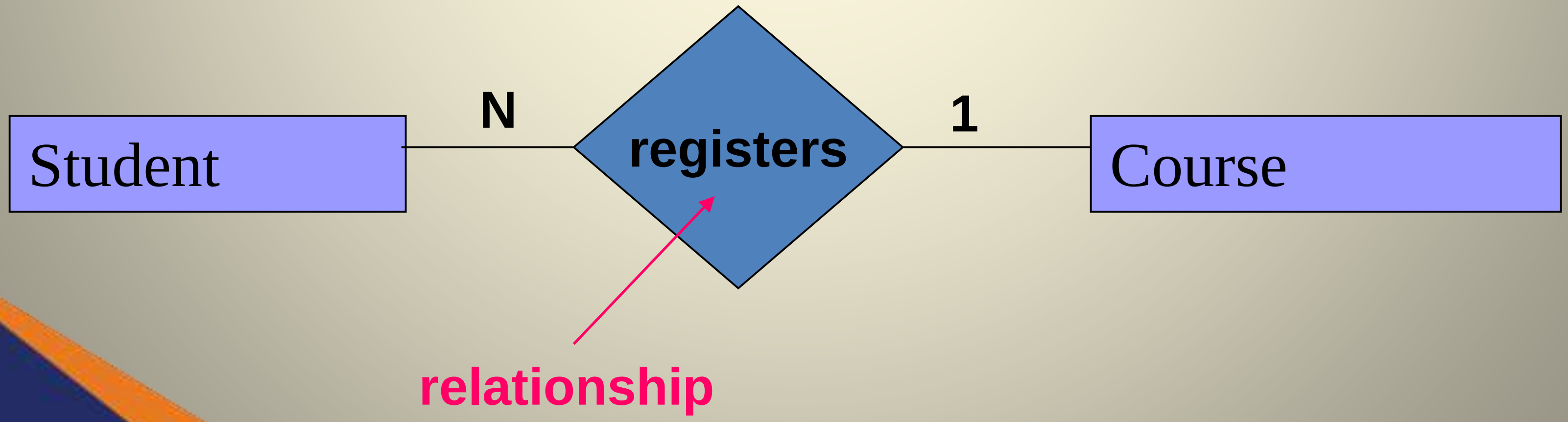
How do we find the course name of the student IT16001



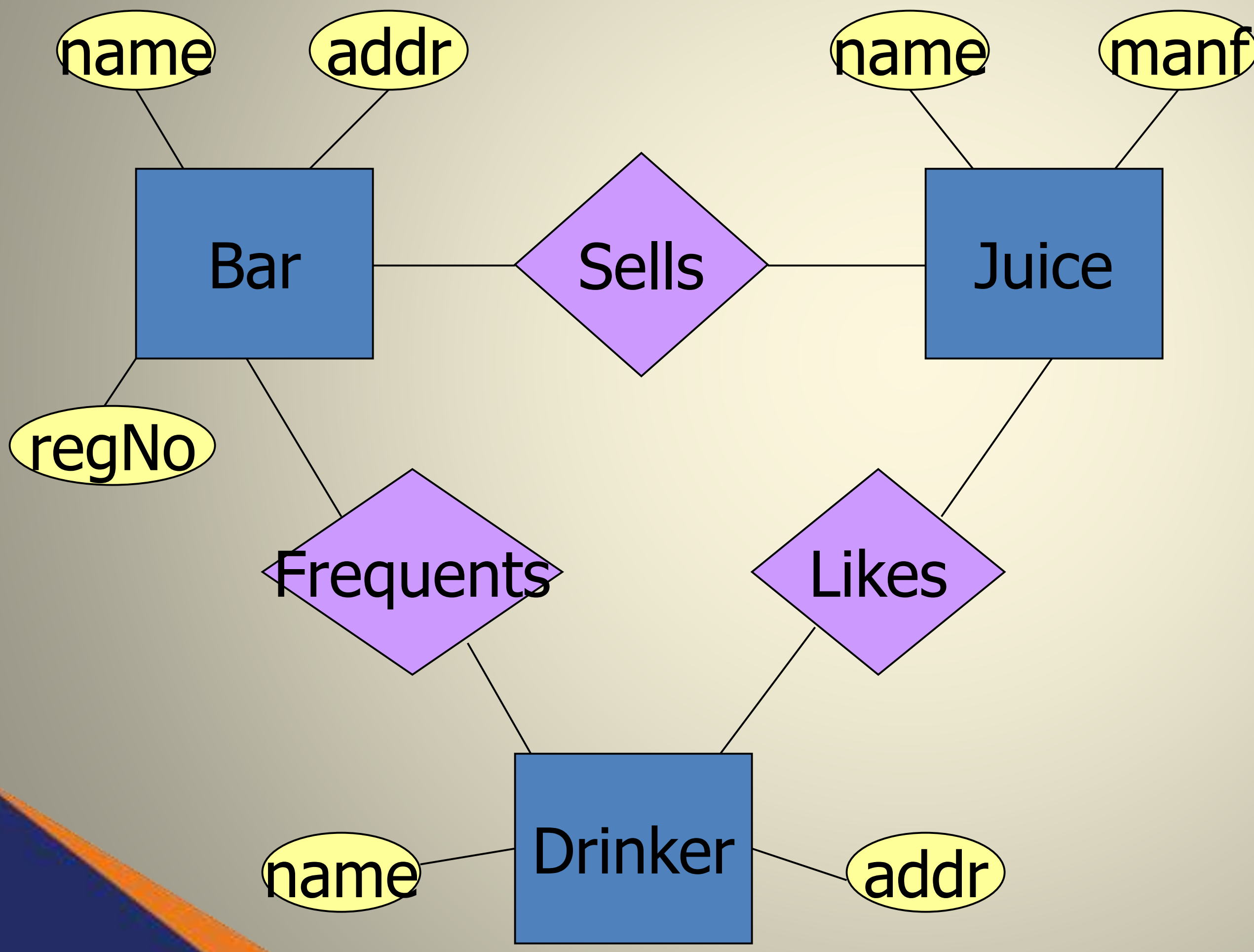
Relationship

- **Relationship** is an association among two or more entities.
- Collection of similar relationships - **relationship set** .
- Shown as a line connecting the associated entities, labelled with the name of the relationship.

  - Normally a relationship is named using a 'verb' or 'verb group'.



Example: Relationships



Bars sell some juices.

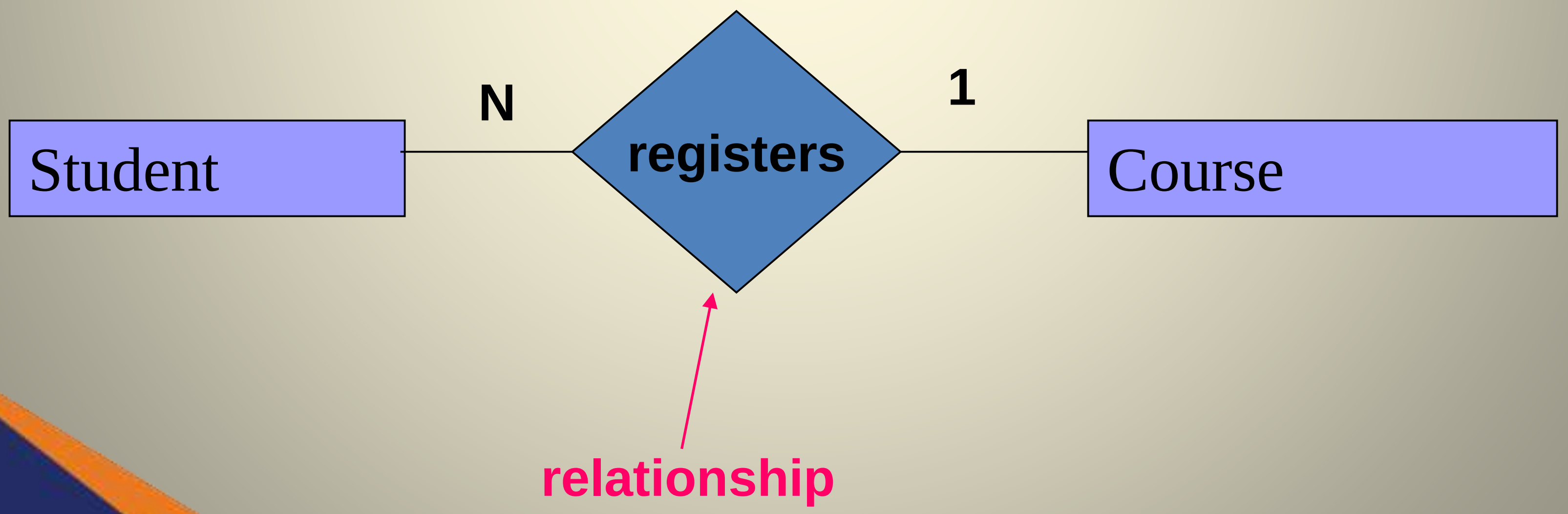
Drinkers like some juices.

Drinkers frequent some bars.

Degree of a Relationship

Degree of a relationship is the number of participating entities in the relationship.

- Degree / Number of Entities = 2 Or binary



Degree of Relationship

- = no of participating entities
- Relationships can be classified based on their degree into
  - **Binary – relationship with two participants-**

    Degree/No of Entities = 2
  - **Ternary – relationship with three participants**

    Degree/No of Entities = 3
  - **Quaternary – relationship with four participants**

    Degree/No of Entities = 4

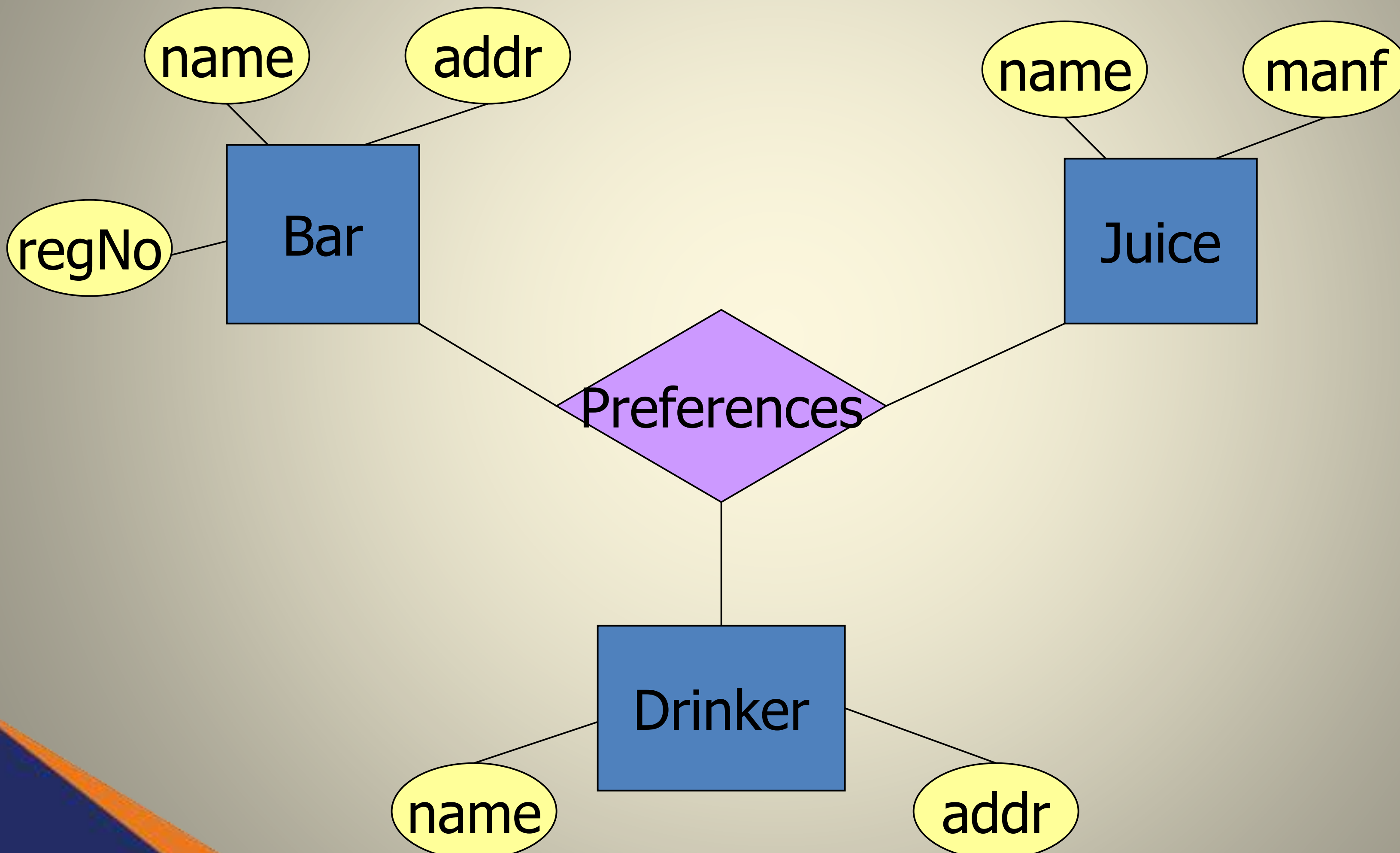
Ternary Relationships

- Sometimes, we need a relationship that connects more than two entity sets.
- Suppose that drinkers will only drink certain juices at certain bars.
  - Our three binary relationships Likes, Sells, and Frequents do not allow us to make this distinction.
  - But a 3-way/ ternary relationship would.

A Typical Relationship Set

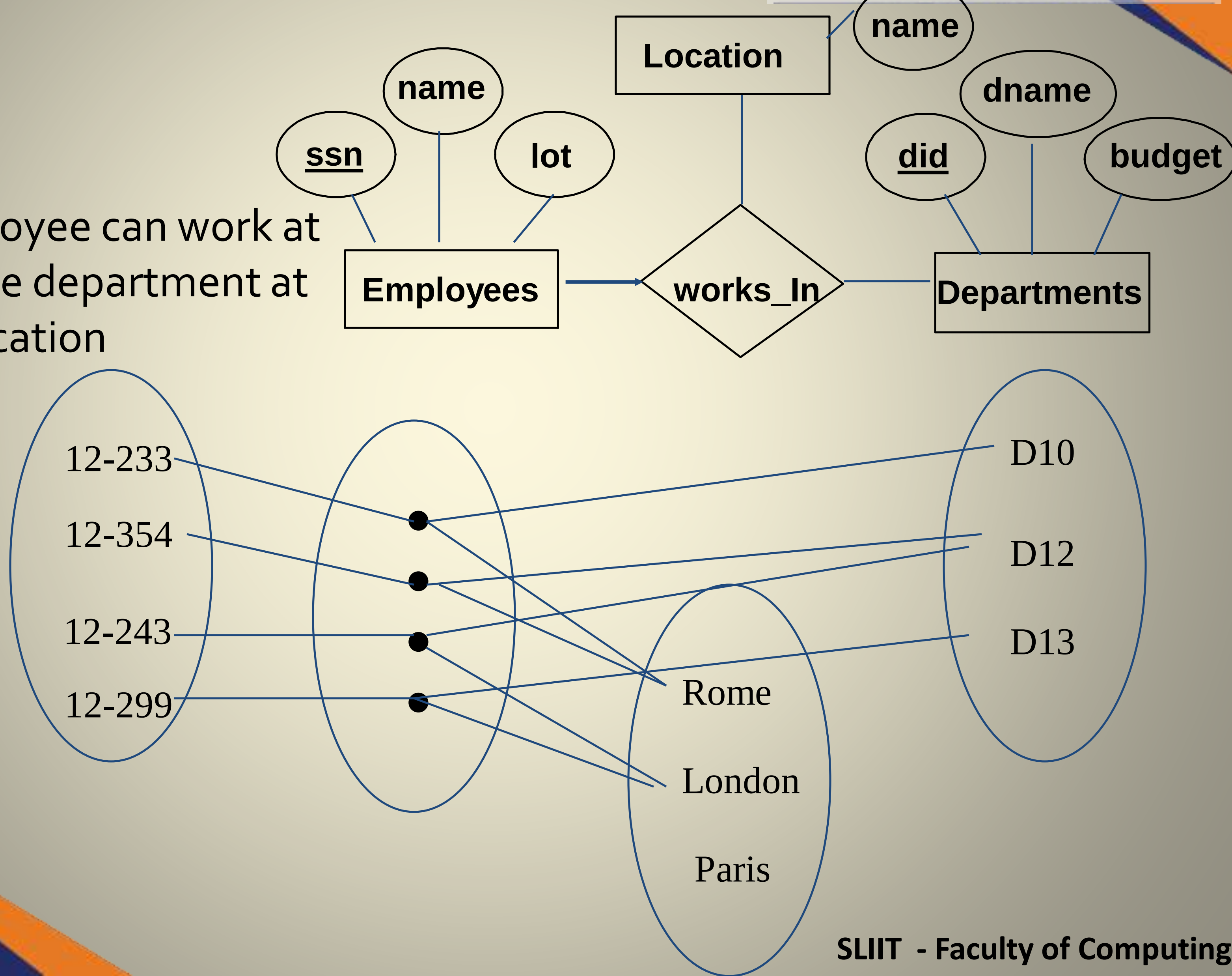
Bar	Drinker	Juice
Joe's Bar	Ann	Avocado super
Sue's Bar	Ann	Apple Lite
Sue's Bar	Ann	Watermelon cool
Joe's Bar	Bob	Mango Lite
Joe's Bar	Bob	Avocado super
Joe's Bar	Cal	Avocado super
Sue's Bar	Cal	Apple Lite

Ternary Relationship



Ternary relationship

Each employee can work at most in one department at a single location



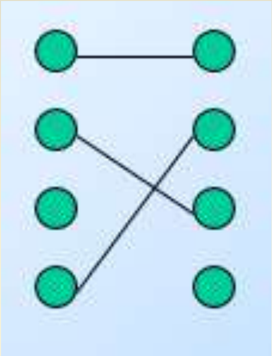
Cardinality

The **cardinality ratio** for a binary relationship specifies the number of relationship instances that an entity can participate in.

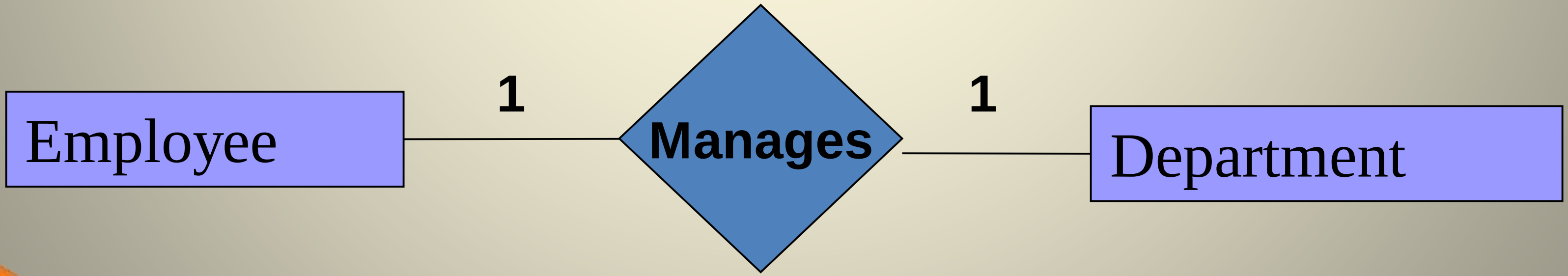
- There are three types of cardinality ratios for binary relationships.

  - **one-to-one (1: 1)**
  - **one-to-many (1 : N)**
  - **many-to-many. (N : M)**

One-to-One relationship



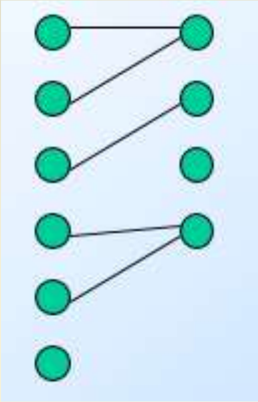
An employee manages at most one department. A department can have only one manager managing it.



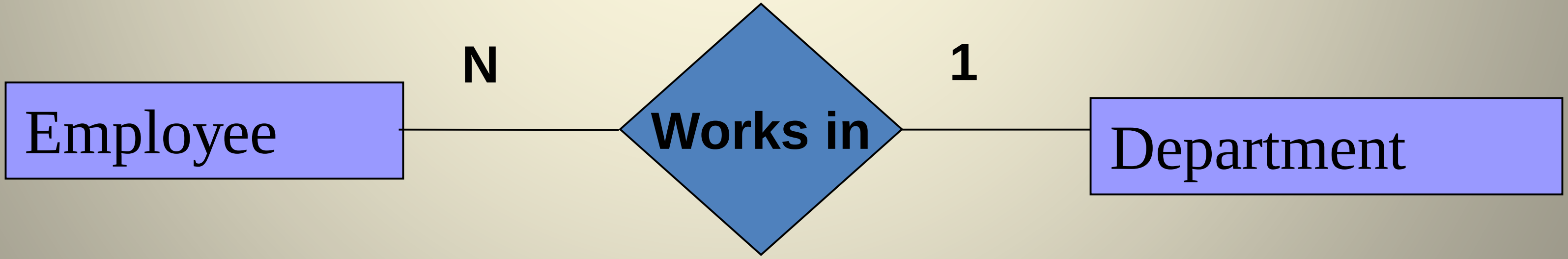
Your Turn !!

- Think about the Library database and give an example of a 1:1 relationship

ONE-TO-MANY RELATIONSHIP



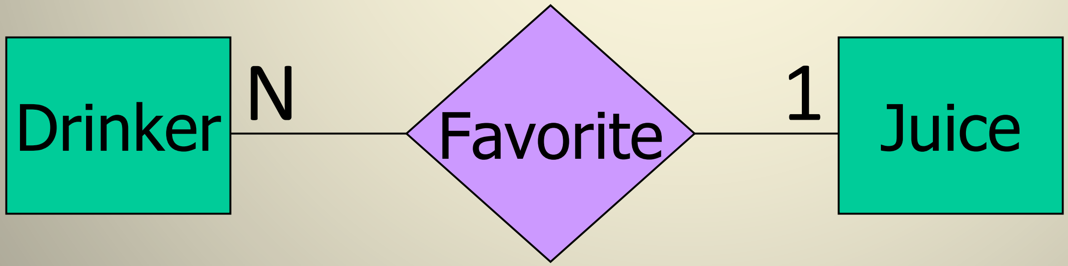
- An employee works in at most one department.



- This is a **one-to-many** (or **many-to-one**) relationship

Example: One - Many Relationship

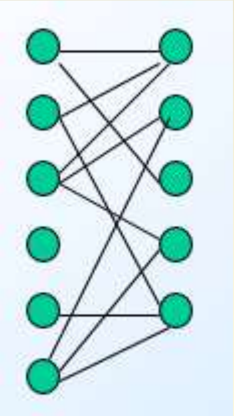
- Favorite, from Drinker to Juice is one-many.
- A drinker has at most one favorite juice.
- But a juice can be the favorite of any number of drinkers, including zero.



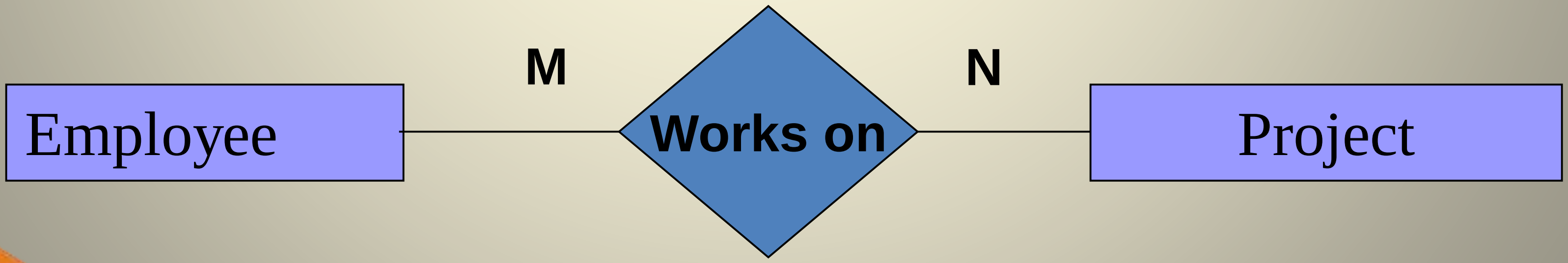
Your Turn !!

- Think about the Library database and give an example of a 1:N relationship

Many-to-many relationship

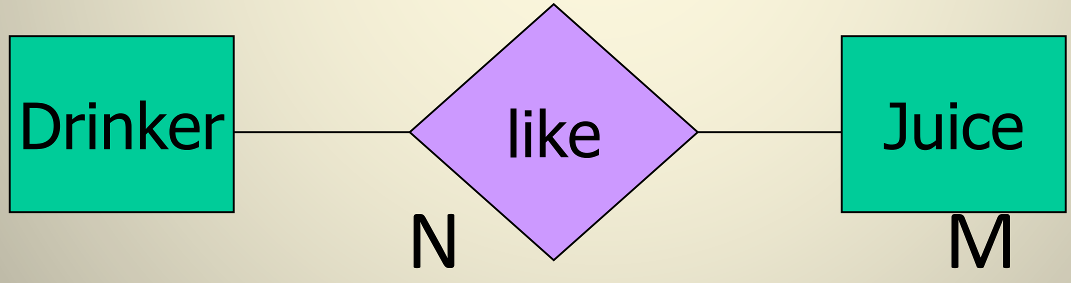


- An employee can work on several projects. A project can have many employees working on it.



Example: MANY - Many Relationship

- Likes, from Drinker to Juice is many-many.



Your Turn !!

- Think about the Library database and give an example of a N:M relationship

Exercise

Identify the Cardinality

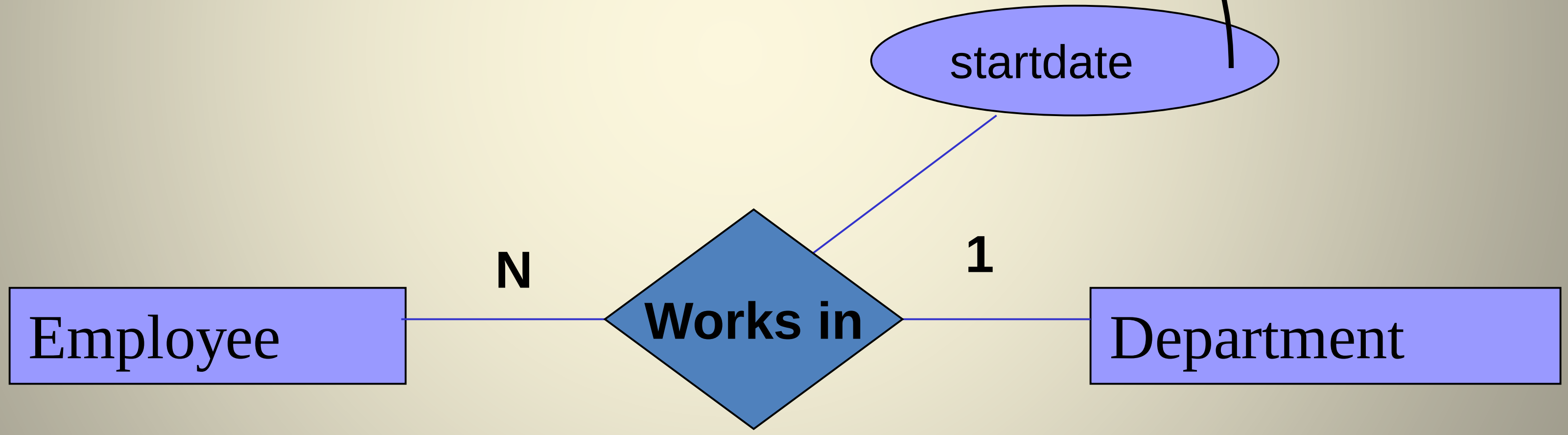
- Doctor patient
- Principal School
- Mother child
- Husband wife
- Teacher Student

Rosell, 23 years old, a diploma holder joined the 'ABC' company as a *management trainee*. She joined the ***Production department*** on 01/01/2005. When Rosell successfully completed her Commerce degree, she was promoted as *Finance Executive* and transferred to the ***Finance department*** from 02/02/2006.

Considering her MBA qualification, the management of the company promoted her as *Assistant Manager - Business Development* and transferred to the ***Business Development department*** with effect from 03/03/2008.

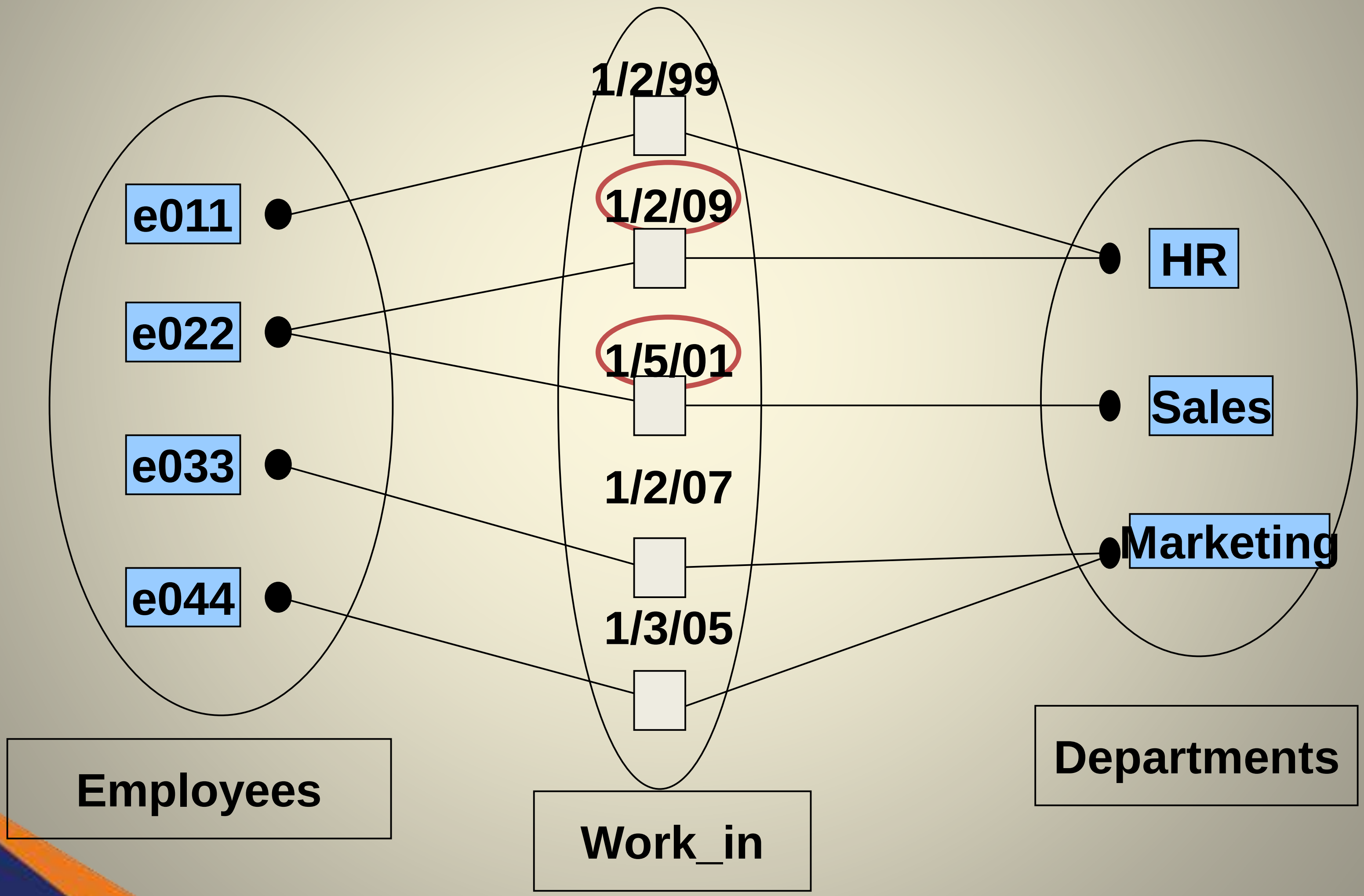
Descriptive Attributes

A relationships can also have **descriptive attributes**. Used to record information about the relationship rather than any one of the participating entities.

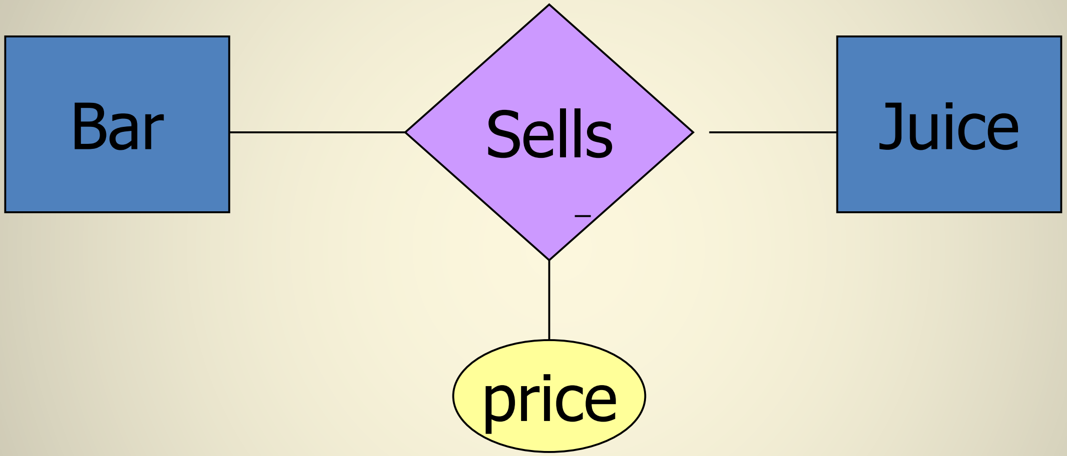


Nimal works in accounts dept since 2008

Descriptive Attributes

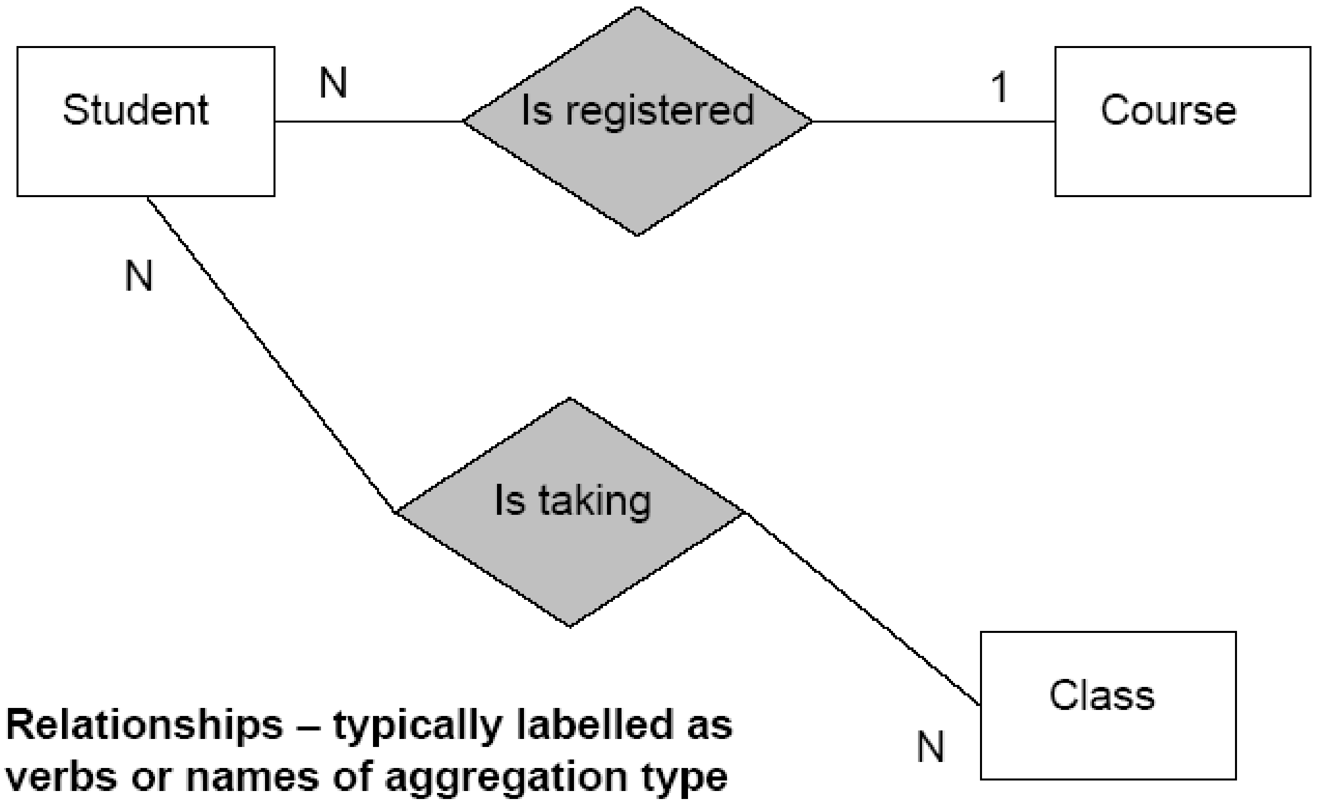


Attribute on Relationship



Price is a function of both the bar and the juice, not of one alone.

ER Diagram



Relationships – typically labelled as verbs or names of aggregation type

Your Turn !

An organisation employs people. These people are assigned to departments and they will work on a range of projects. Every project is the responsibility of a particular department but may utilise employees from across the organisation. (Only include a few ‘critical’ attributes.)

Your Turn !!!

- Identify the entities -> nouns
- Identify the relationships -> verbs
- Determine
  - Cardinality
  - Participation (total / partial)

Restrictions- constraints

What is the criteria to become a student at SLIIT?

Register for a degree

Mandatory /compulsory

Student MUST be registered in a degree

Restriction - Constraint

How do we present this information in the data base??

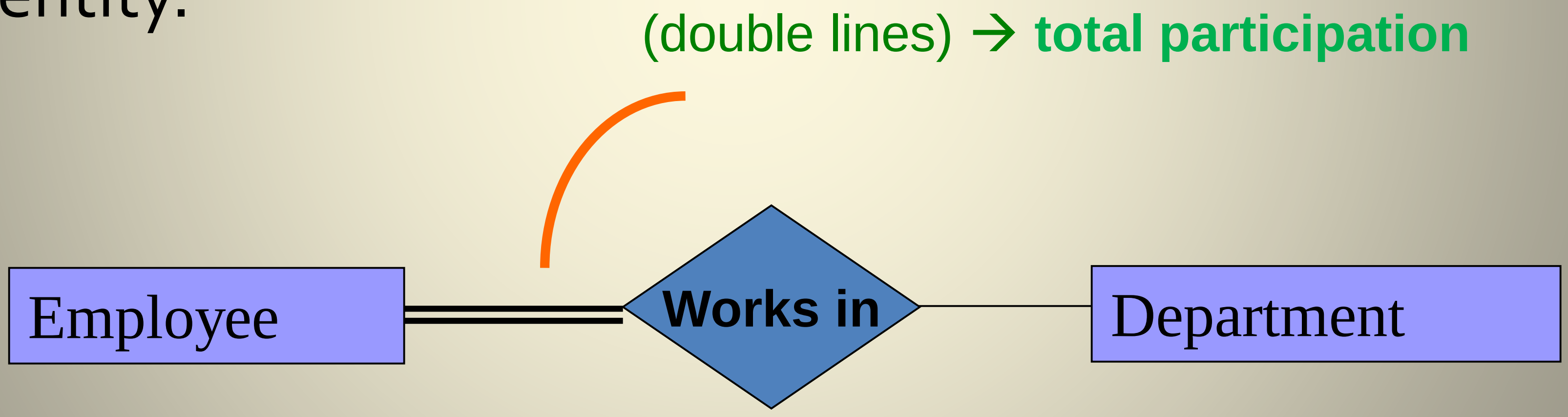
- Student must be registered in a degree.

Participating Constraints

- **Participating constraint** specifies whether the existence of an entity depends on its being related to another entity via the relationship type.

Participation Constraints

- For example, if we specify that an employee **must** always work for a department.
- Then we say that the relationship "works in" is in **total participation** from employee entity to department entity.

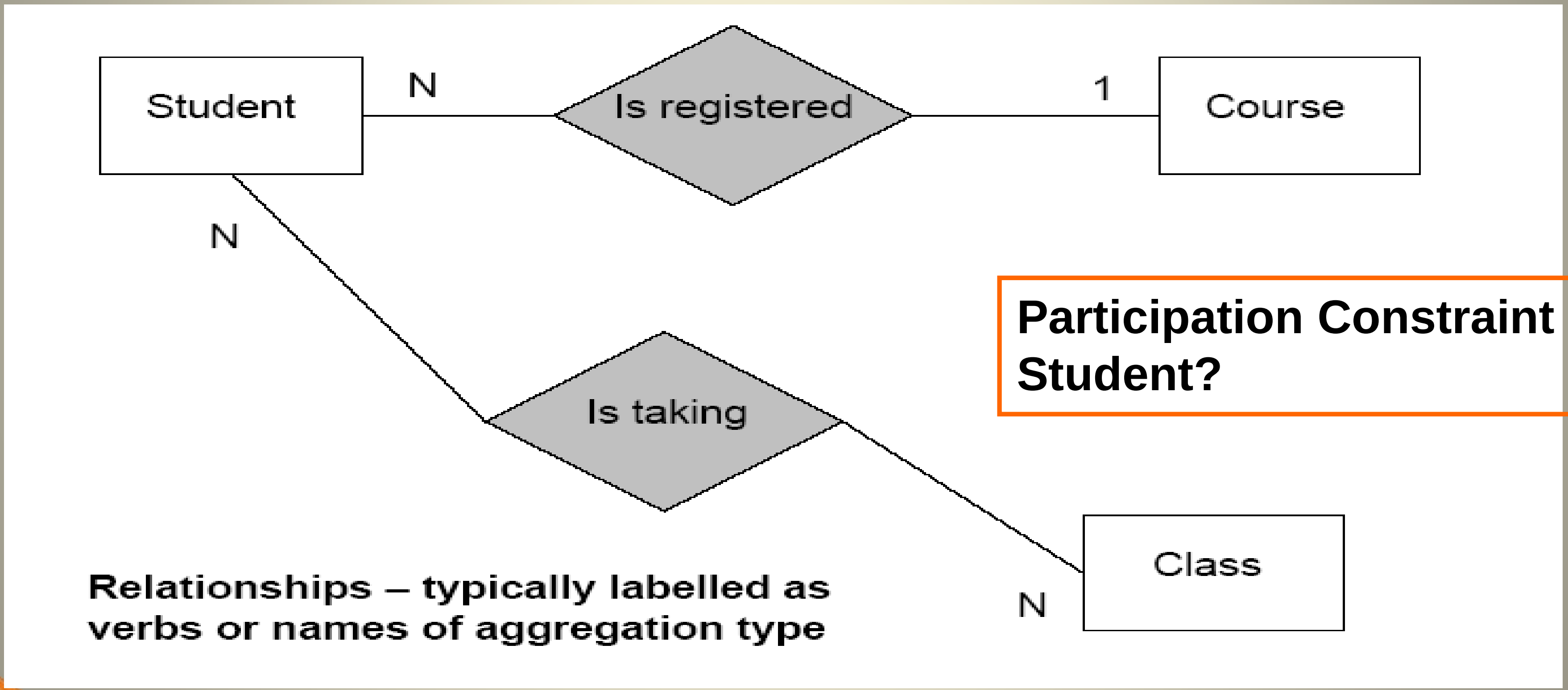


Employee must work in a department
Department may or may not have employees

E-R Model (contd.)

- If the relationship is not in total participation, then it is known as in **partial**.
- For example, from Department to Employee
- Department may or may not have Employees

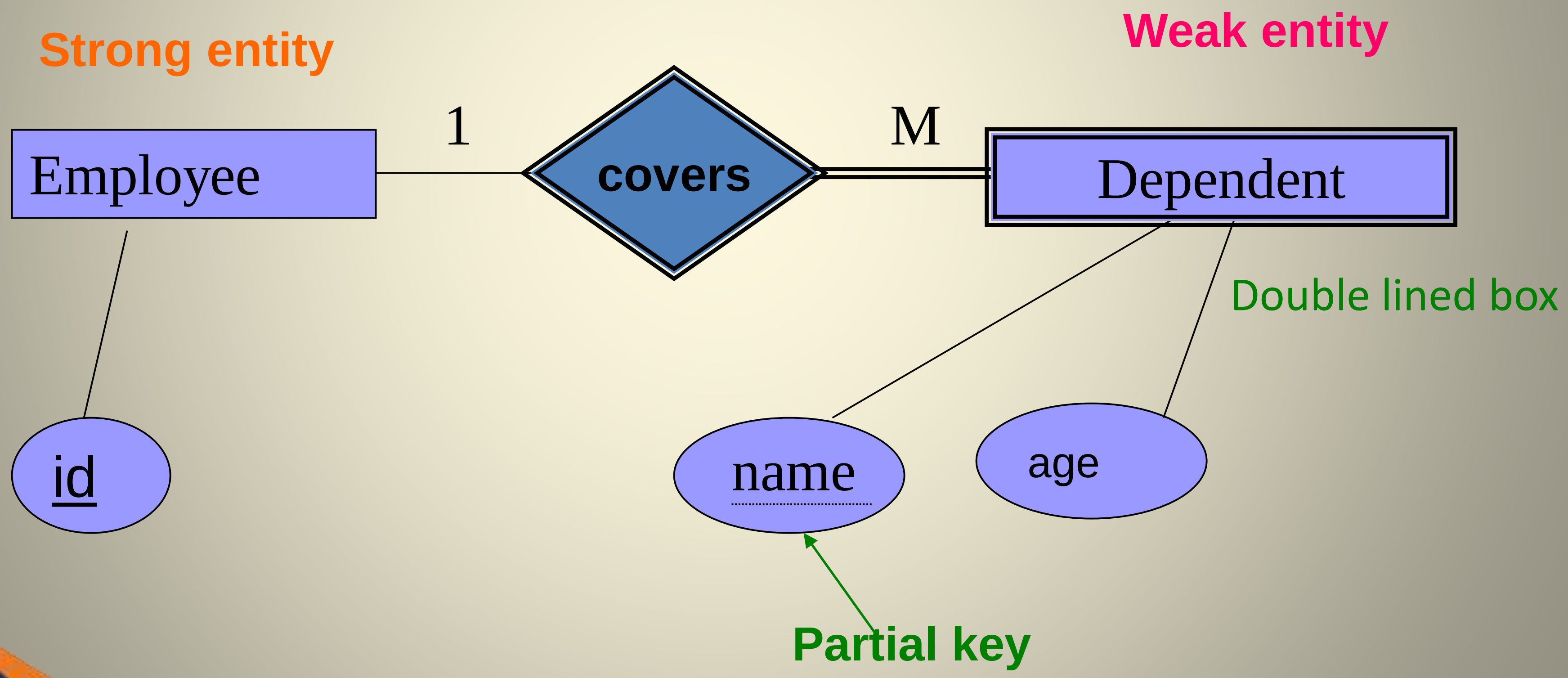
Participation Constraints



Weak Entity

- Parents employed?
- Does the company cover THE CHILDRENS medical insurance?
- How do you claim your medical bills
  - Can you get it reimbursed or through your parents
- Is the same coverage given to children after resignation

Weak Entity


Strong entity
Weak entity
1
M
Employee
covers
Dependent
Double lined box
id
name
age
Partial key

Weak Entity

- Some entities can't exist on its own.
- Its' existence-dependent on another entity, i.e., it cannot exist without the entity with which it has a relationship.
- It inherits the part of the primary key from the entity to which it is related.
- Entity types without any key attributes is called **weak entity types**.

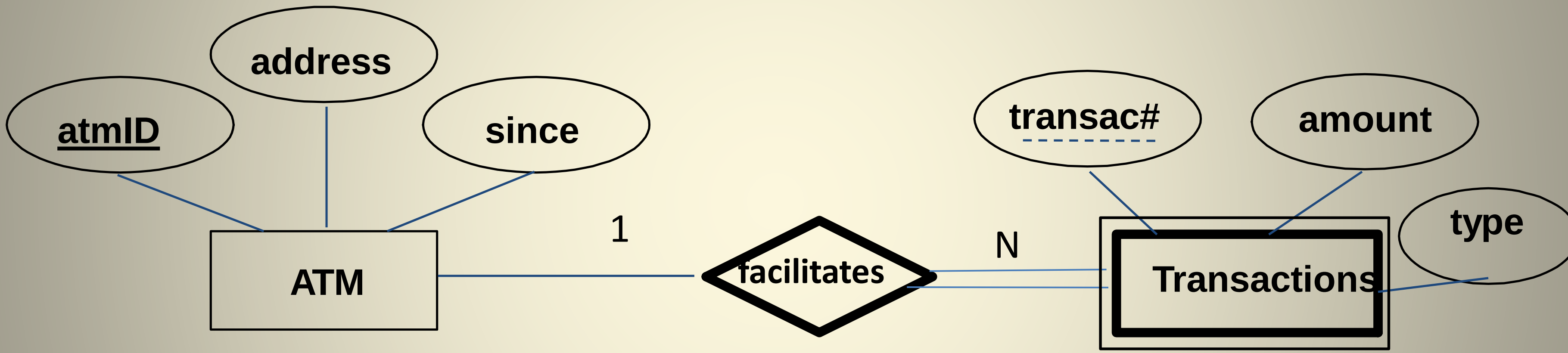
- The attributes in the weak entity participating in the key are called **partial keys.**

- The owner entity and the **weak entity** participates in an **identifying relationship**.

- The cardinality of the identifying relationship is either **one-to-one** or **one-to-many** from owner entity to weak entity.

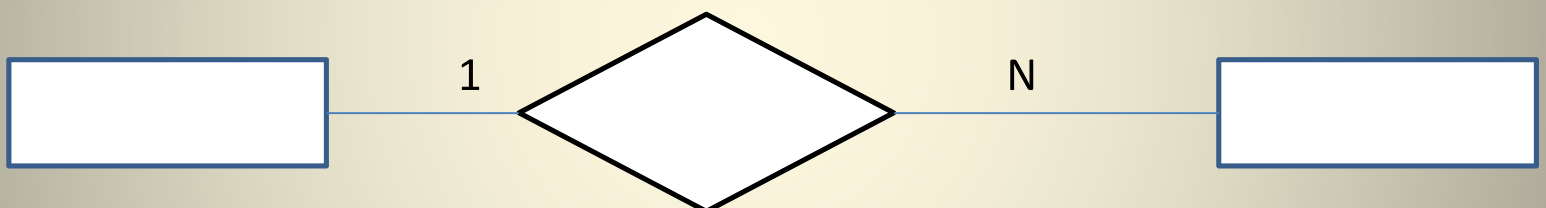
- The weak entity must have **total participation** in the identifying relationship.

EXAMPLE- Weak Entity

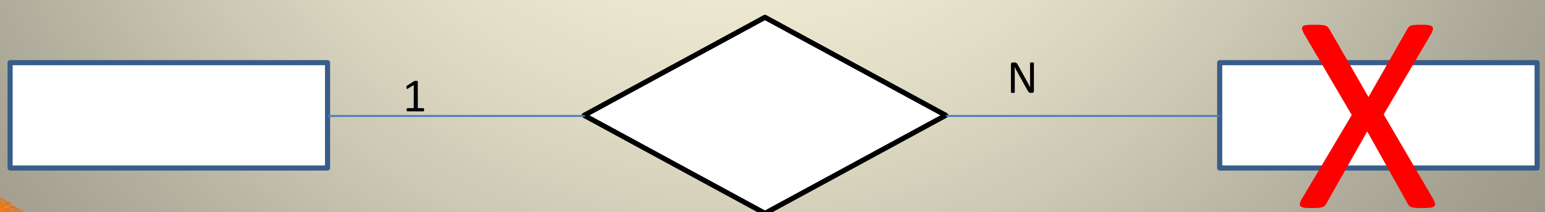


Recursive Relationship

- Have you formed your project groups
- What is the structure?
  - Leader and members

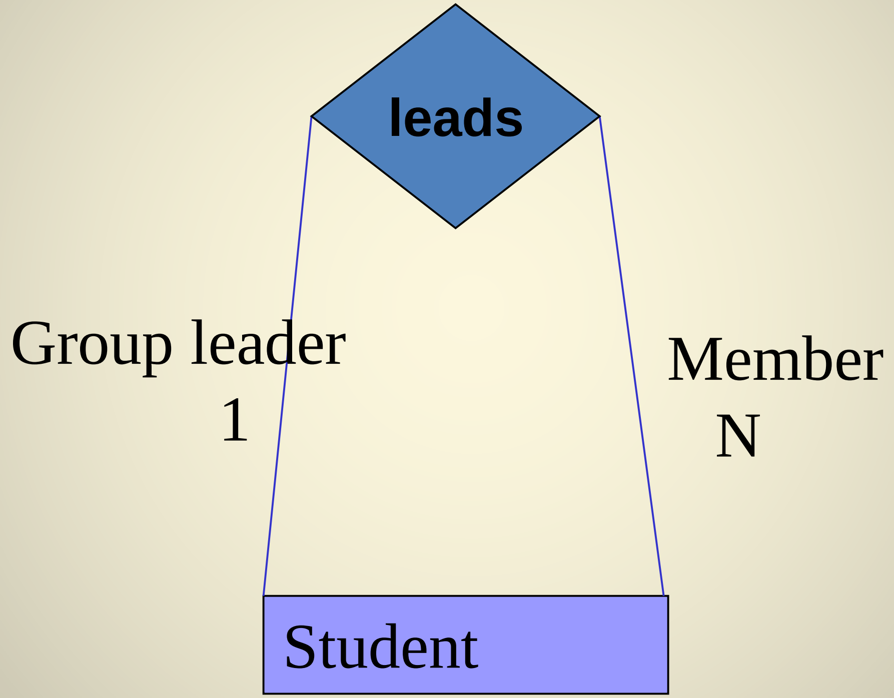


  - Both leader and members are students



E-R Model (contd.)

Student leads set of students in group assignments.



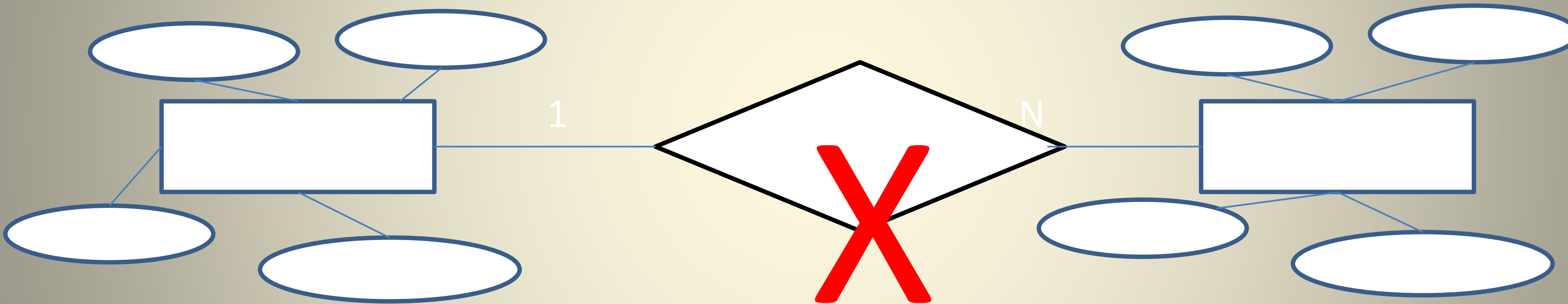
Recursive Relationship

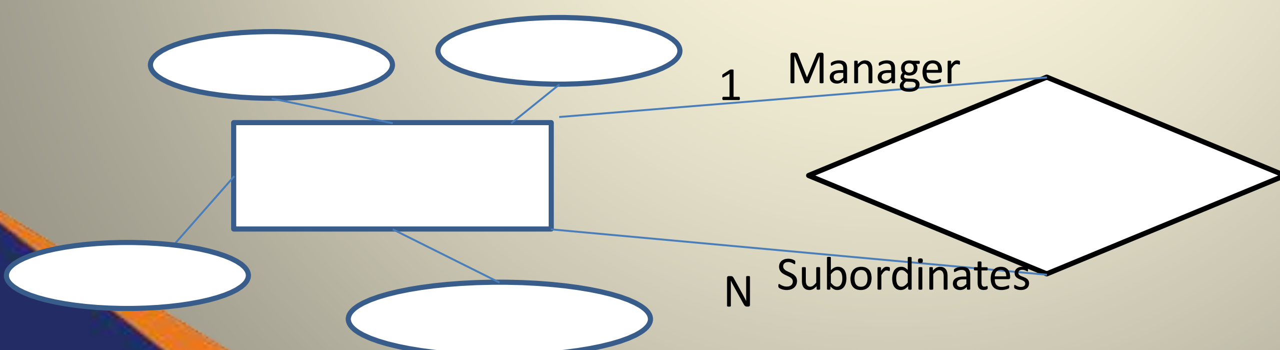
- Entities participating in a relationship need not be distinct. Such relationships are called **recursive relationships**.

- Each entity in the relationship play a **role** in the relationship. It is recommended to state the role in recursive relationships.

Recursive Relationships

- In most companies, each employee (except the CEO) is supervised by one manager. Of course, not all employees are managers.





Your Turn !

- Draw an ER diagram for the following requirements.

- A company database needs to store information about employees (identified by NIC, salary, position, phone, office);
- departments (identified by dno, with department name and annual budget);
- children of employees (with name, and age as attributes).

- Employees work in departments; each department is managed by an employee; a child must be identified uniquely by name when the parent (who is an employee; assume that only parent works for the company) is known. We are not interested about a child once the parent leaves the company.

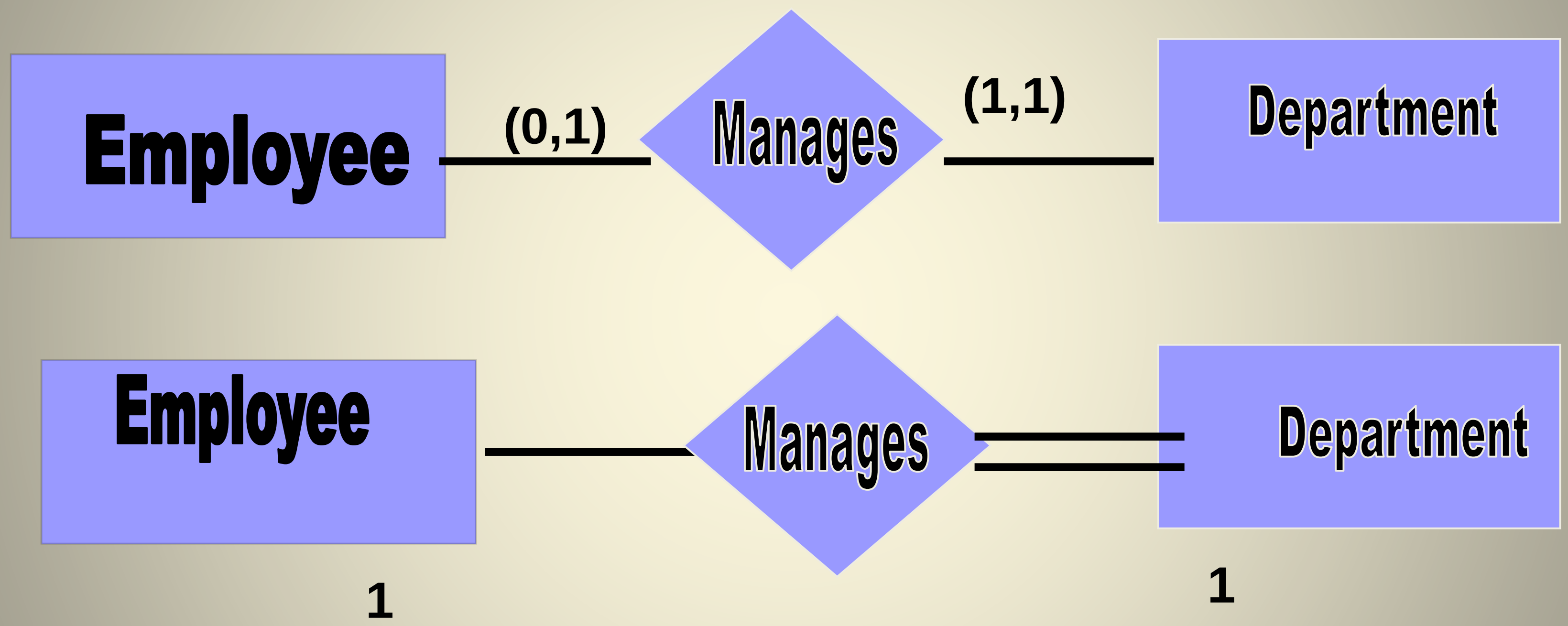
Alternate Notation – Min: Max

- Used to specify the number of possible occurrences of each participating entity type in a relationship.

- Contains two parts
  - Min
  - Max
  - For example, (0,1)
    - Min = 0
    - Max = 1

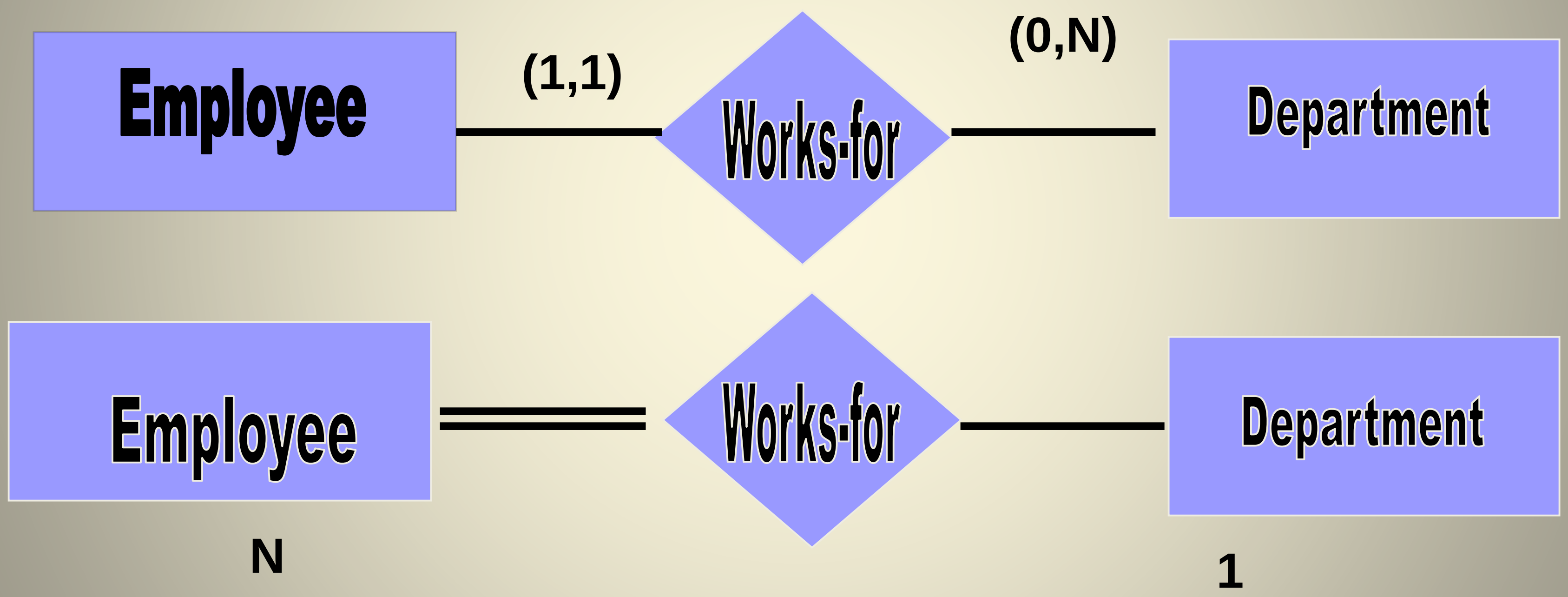
- Max of a multiplicity range denotes Cardinality
- Min of the range denotes Participation constraint

THE (MIN,MAX) NOTATION



Employee may or may not mange a department. He can at most manage one department.
A department must be managed by an employee

THE (MIN,MAX) NOTATION



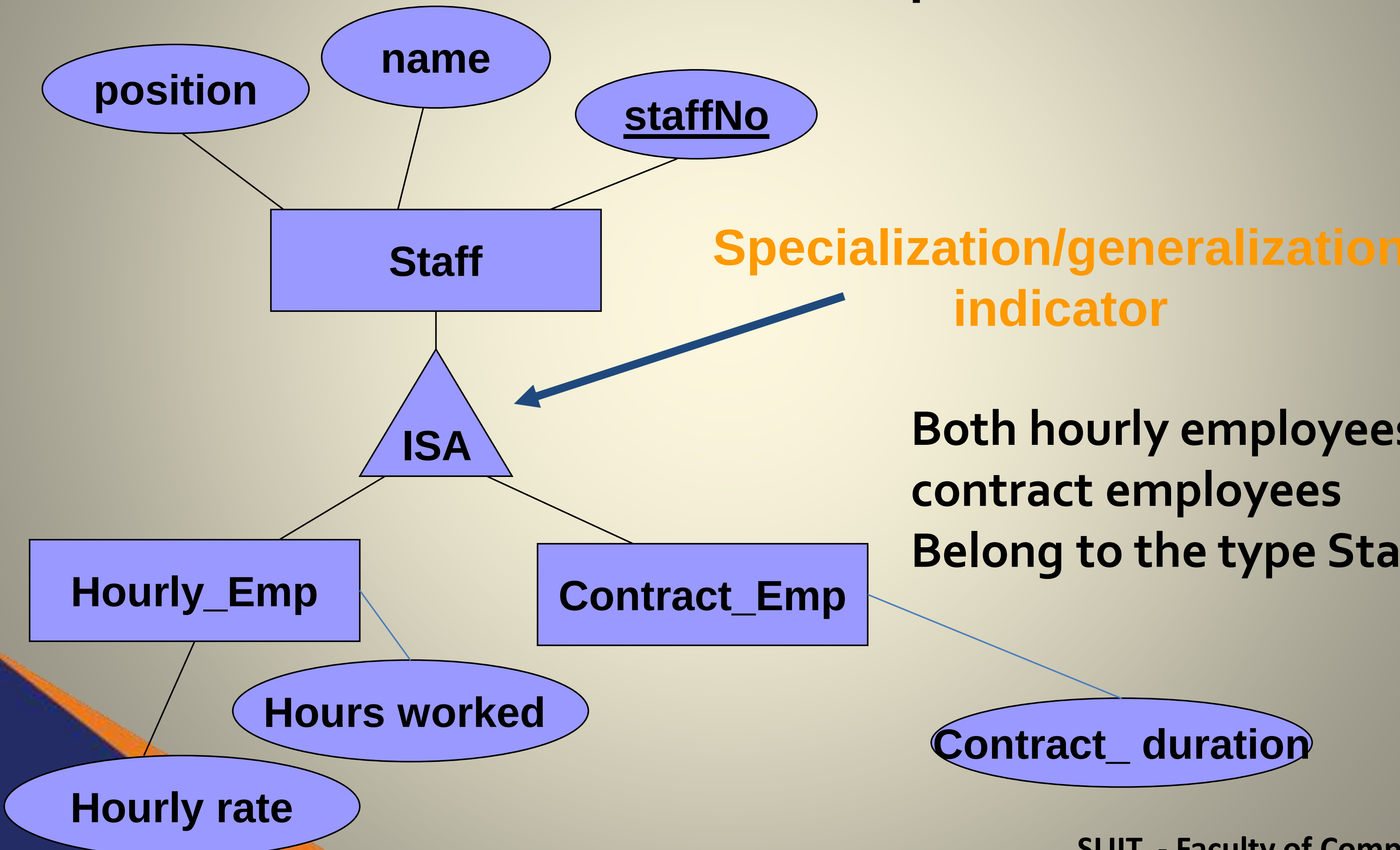
Employee must work for a department. .
A department may or may not have any employee

Enhanced ER Modelling

- ER modelling does not capture all the semantics of client's domain, such as
  - 'ISA' ('is a') relationship or specialization-generalization
    - 'Manager' entity type 'is a' sub entity of 'Staff' entity.

- Enhanced ER models represent the above relationships, hence capture client's domain more comprehensively.

Diagrammatic Representation of 'ISA' relationship



Both hourly employees & contract employees Belong to the type Staff

End of Lecture 05

Questions ?

Logical Database Design & Relational Model

Lecture - 06

Learning Outcome

- **LO3**: Model data requirements using data models

In this lecture you will learn

- What is logical database design?
- How to derive a logical model from the information represented in the ER model (conceptual model)
- We focus on one type of logical model which is relational model

Relational Model

- Data model on which most DBMS implementations are based. **CODD 1970**
- Simple & elegant mod:
  - Everything is a relation (= table)

    Every relation is a table with rows & columns
- There are standard ways to convert from

  the E-R model (**conceptual model**)

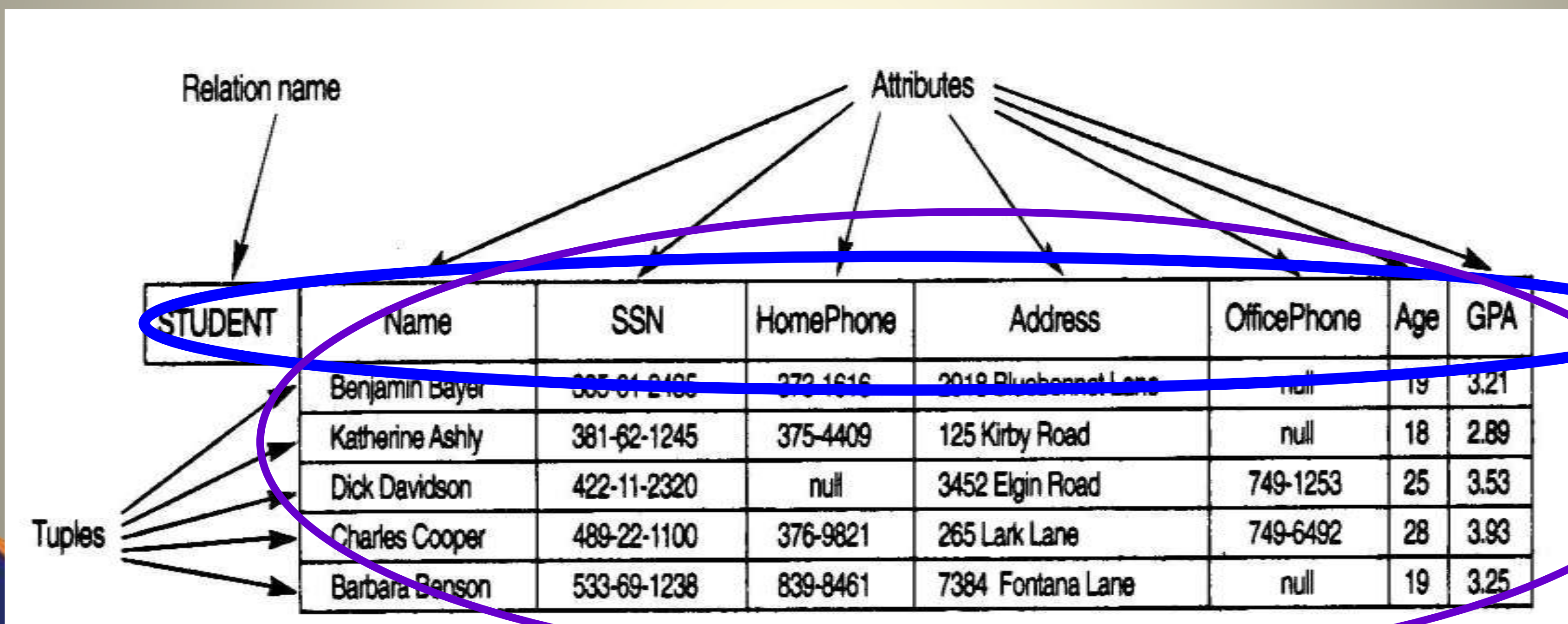
  Relational model (**logical model**)

Relational Model (contd.)

- The relational model represents the database as a collection of **relations.**
- Relation consists of
  - Relation schema
  - Relation instance (table)

Relation

- RELATION:
  - Schema
  - Instance



STUDENT	Name	SSN	HomePhone	Address	OfficePhone	Age	GPA
	Benjamin Bayer	305-61-2435	373-1616	2918 Bluebonnet Lane	null	19	3.21
	Katherine Ashly	381-62-1245	375-4409	125 Kirby Road	null	18	2.89
	Dick Davidson	422-11-2320	null	3452 Elgin Road	749-1253	25	3.53
	Charles Cooper	489-22-1100	376-9821	265 Lark Lane	749-6492	28	3.93
	Barbara Benson	533-69-1238	839-8461	7384 Fontana Lane	null	19	3.25

Relation Schema

- Describes the column heads (attributes) of the relation
  - name of the relation,
  - name of each filed,
  - domain of each field
    - Domain : is described by domain name and set of associated values

Relation Schema

Schema Name(relation name)

- Students

(sid:*string*,
name:*string*,
login:*string*,
age :*integer*,
gpa :*real*)

Field name

domain

- Domain GPA : real (0-4)

Relation Instance

- Set of tuples or records or rows :
- Each tuple has the same number of fields as the relation schema

Example : Relation Instance

Relation name

Attributes

STUDENT	Name	SSN	HomePhone	Address	OfficePhone	Age	GPA
	Benjamin Bayer	305-61-2435	373-1616	2918 Bluebonnet Lane	null	19	3.21
	Katherine Ashly	381-62-1245	375-4409	125 Kirby Road	null	18	2.89
	Dick Davidson	422-11-2320	null	3452 Elgin Road	749-1253	25	3.53
	Charles Cooper	489-22-1100	376-9821	265 Lark Lane	749-6492	28	3.93
	Barbara Benson	533-69-1238	839-8461	7384 Fontana Lane	null	19	3.25

Tuples

Degree of a relation

- The **degree** of R is the number of attributes in R
- (ID,Name,Address,Phone)=4

ID	Name	Address	Phone
100	Sampath	Moratuwa	01992883
110	Amali	Colombo -3	01983733
120	Sanath	Negombo	null

Example

Make a list of students in the class, keeping their ID, name and phone number.

ID	Name	Phone
S01	Mike	111
S02	Elisa	222

This is the basic structure of the relation model, a table or relation.

Formalizing : Relations

- Definition: A relation is a named table of data

– Table is made up of rows (records or tuples), and columns (attributes or fields)

- Not all tables qualify as relations. Requirements:

1. Every relation has a unique name
2. Every attribute value is atomic (not multivalued, not composite)
3. Every row is unique (can't have two rows with exactly the same values for all their fields)
4. Attributes (columns) in tables have unique names
5. The order of the columns is irrelevant
6. The order of the rows is irrelevant

Formalizing Key Fields

Primary key (PK)

- Minimal set of attributes that uniquely identifies a row
- This is how we can guarantee that all rows are unique

Foreign key (FK)

Set of attributes in a table that serves as a reference to the primary key of another table

Foreign Key

- A constraint involving *two* relations
  - **referencing relation**
  - **referenced relation.**
- Tuples in the *referencing relation* have attributes **FK** (called **foreign key** attributes) that reference the primary key attributes **PK** of the *referenced relation*

referencing Enrolled (cid ,grade, studid) **FK**

referenced Students (sid, name ,login, age, gpa)

PK

- Display the foreign keys by drawing an arrow from the **foreign key** to the **primary key**

Integrity Constraints IC

- DBMS must prevent entry of incorrect information
- To prevent : Constraints / conditions are specified on a relational schema = ICs
- Database which satisfies all constraints specified on a database schema is a legal instance.
- DBMS enforces constraints - permits only legal instances to be stored
- When the application is run the DBMS checks for the violation and disallows the changes to the data that violates the specified IC

Integrity Constraints

- Specified and enforced at different times.
  - **Specified :** When the DBA /end user defines the data base schema
  - **Enforced :** When database application is run
    - DBMS checks for violations
    - Disallow violating entries

Integrity Constraints

Many kinds of ICs:

- Domain constraints
- Key constraints
- Entity integrity constraints
- Referential integrity constraints

Domain Constraints

- **Domain constraints**: value in the Column must be drawn from the domain associated with that column

- **Restricts the :**
  - Type
  - Values that can appear in the field

Eg.

- Name Char (25)
- GPA (real >=0, =<4)

Key constraints

- **Is a statement that;**
  - A certain minimal subset of the fields of a relation is a unique identifier for a tuple.
- **Which Means**
  - Two tuples in a legal instance cannot have identical values in all the fields of a key.

Constraints…

- **Entity Integrity Constraints:** states that **primary key** values cannot be null
  - This is because primary key values are used to *identify* the individual tuples.
- **Referential Integrity Constraints**
  - Some times information stored in one relation is linked to information stored in another relation.
  - If one is modified the other must be modified to keep the data consistent.
  - An IC involving both relations must be specified
  - IC involving 2 relations is a **foreign key constraint.**
  - Foreign keys enforce referential integrity constraints

SLIIT Discover Your Future

Referential Integrity

- The value in the foreign key column (can be either:
  - a value of an existing primary key in the referenced relation or a null

FK

cid	grade	stuid
Carnatic101	C	53666
Reggae203	B	53666
Topology112	A	53650
History105	B	53666

Enrolled

PK

Students

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8

SLIIT Discover Your Future

DB operations & constraints

- IC are specified when a relation is created and enforced (checked) when a relation is modified
- 3 types of modifications to the relation :
  - Insert : inserts a new tuple(s) into a relation.
  - Delete : delete tuple(s) in a relation.
  - Update : changes the values of some attributes in existing tuples .

Insert operation

The insert operation can violate the following constraints:

- Domain constraints (invalid value)
- Key constraints (duplicate key values)
- Entity integrity constraints (null primary key value)
- Referential integrity constraint (non-existing primary key value)

examples

Students

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8

- Domain constraints (invalid value)

  Insert <'abc','Tom','tom@lk','17',3.2>into Students

  – sid value is not in the domain

Enrolled

cid	Grade	sid
IT	A	53666
IS	B	53650

examples

Students

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8

- Key constraints (duplicate key values)

  Insert <'53666','Tom','tom@lk','17',3.2>into Students

Enrolled

cid	Grade	sid
IT	A	53666
IS	B	53650

examples

Students

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8

- Entity integrity constraints (null primary key value)

  Insert <null,'Tom','tom@lk','17',3.2>into Students

Enrolled

cid	Grade	sid
IT	A	53666
IS	B	53650

examples

Students

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8

- Referential integrity constraint (non-existing primary key value)

  Insert <'IT','A','53900'>into Enrolled

Enrolled

cid	Grade	sid
IT	A	53666
IS	B	53650

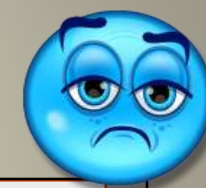
Delete operation

- Delete operation can violate referential integrity.

Enrolled

cid	grade	stuid
Carnatic101	C	53666
Reggae203	B	53666
Topology112	A	53650
History105	B	53666

Students



sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8

- Two options:
  - Reject the deletion
  - Cascade the delete

Update operation

- Update operation can be considered as a deleting a tuple and re-inserting the tuple with new values
- All constraints discussed in Insert & Delete need to be considered
  - Domain constraints (invalid value)
  - Key constraints (duplicate key values)
  - Entity integrity constraints (null primary key value)
  - Referential integrity constraint (non-existing primary key value)

ER to Relational Mapping…

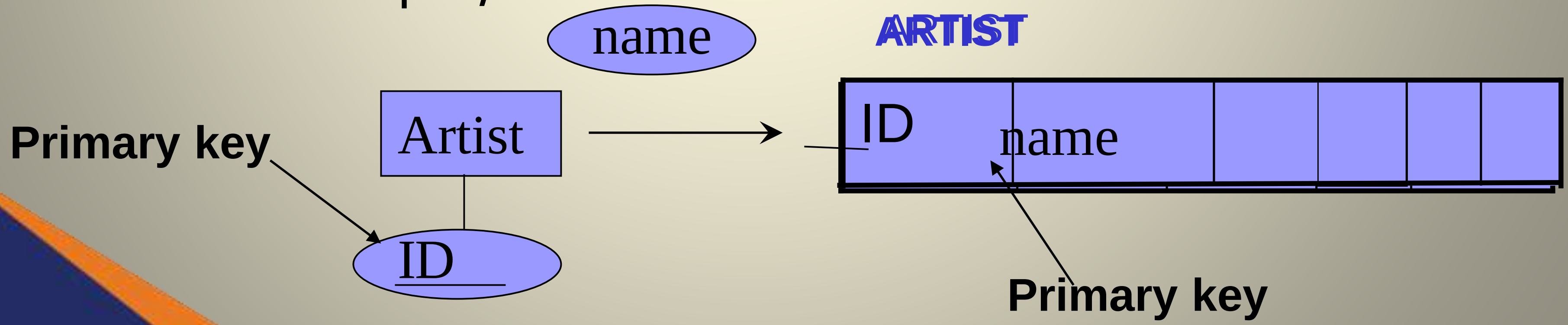
- In the Database Design process, we firstly derive a **conceptual model** (ER Diagram)
- This model needs to be mapped to the **relational model** in order to be implemented using a relational DBMS (RDBMS). Moving from Conceptual (ER) to lower level Logical Model (Relational)
- ER is independent of the details of the implementation (relational, network or OO)_
- This section discusses the rules that can be used for this process

Mapping: Regular Entity

ER Model **Relational Model**

– Entity (strong) → Relation

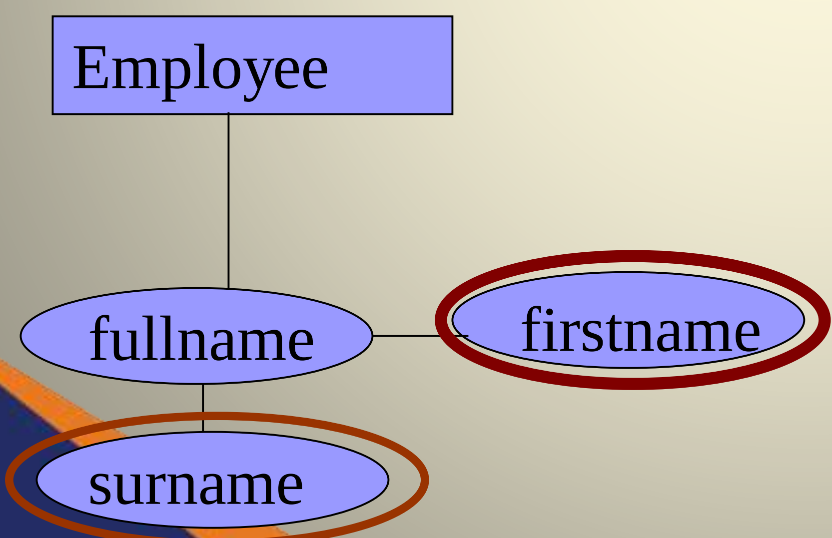
– For example,



Mapping: Composite Attribute

ER Model

- Composite attributes → attributes



Relational Model

Set of simple atomic

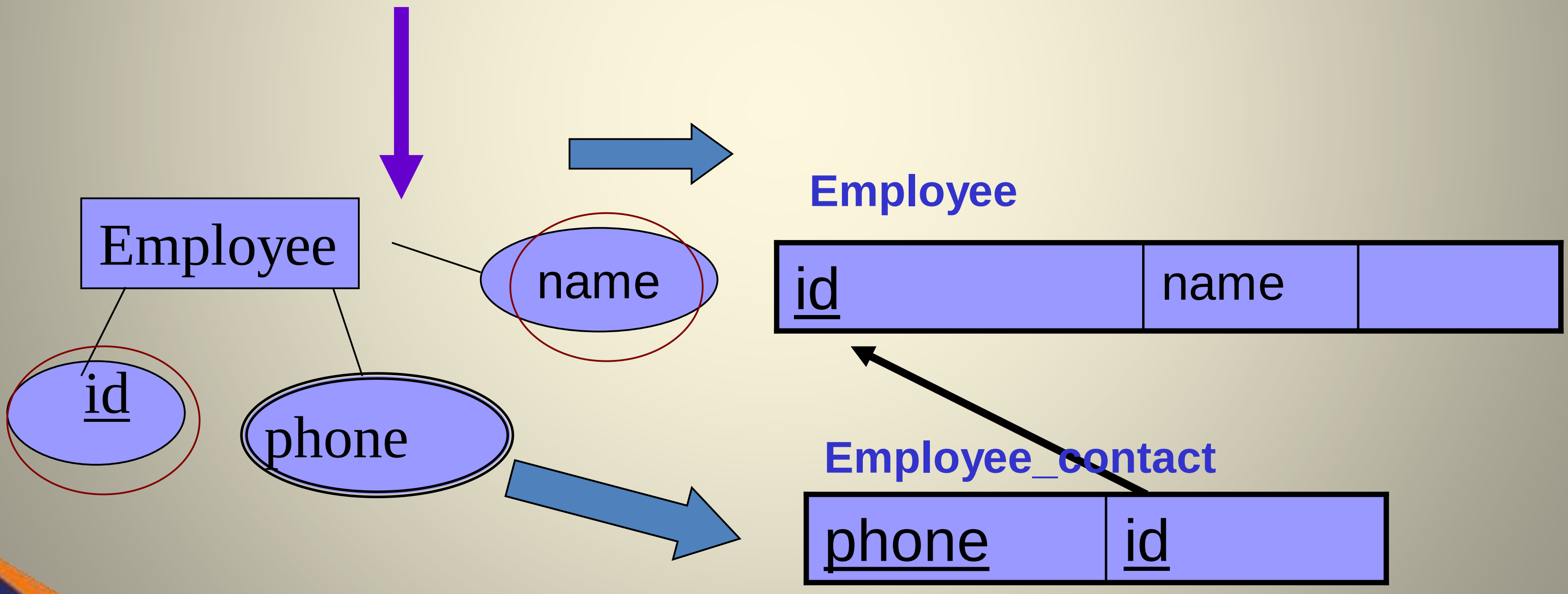
EMPLOYEE

surname	firstname

Mapping: Multivalued Attributes

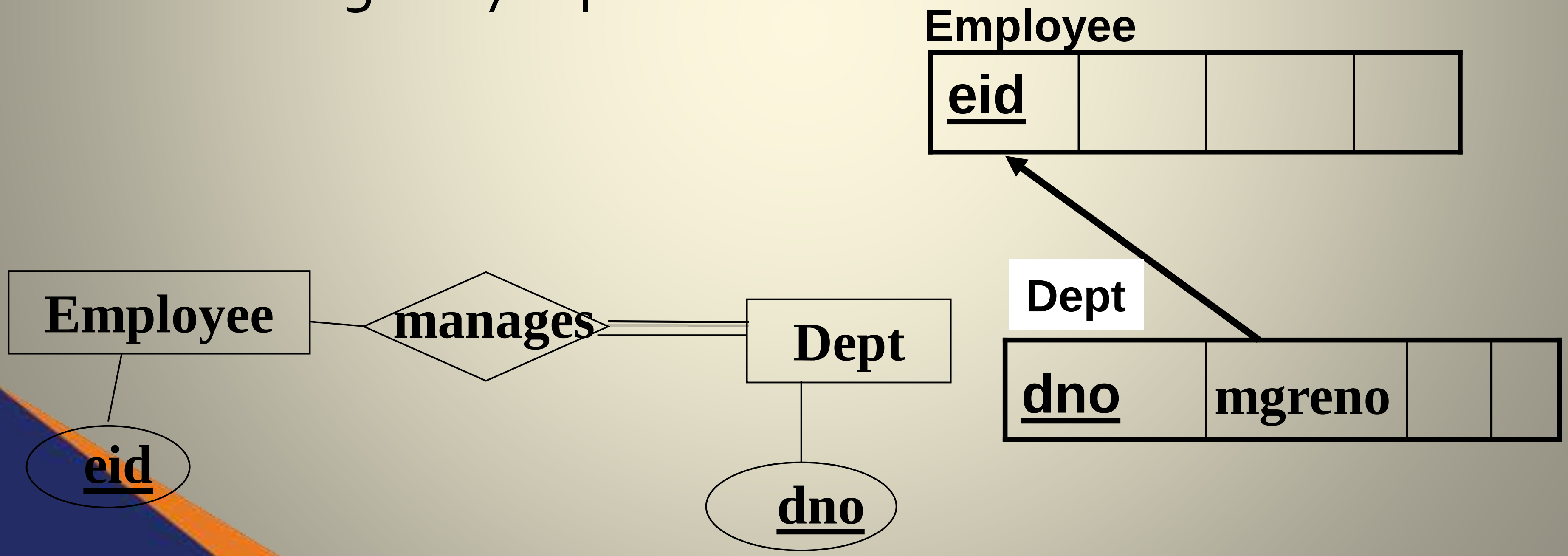
ER Model **Relational Model**

- Multivalued attribute → Relation & Foreign Key



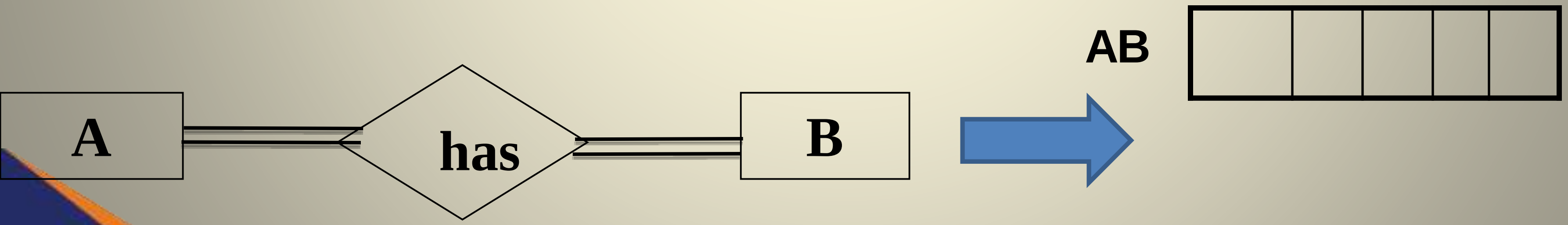
Mapping:1:1 with total & Partial participation

- If there is a 1:1 relationship R from entity A to B and if B is in total participation with A on R then the foreign key is placed in B



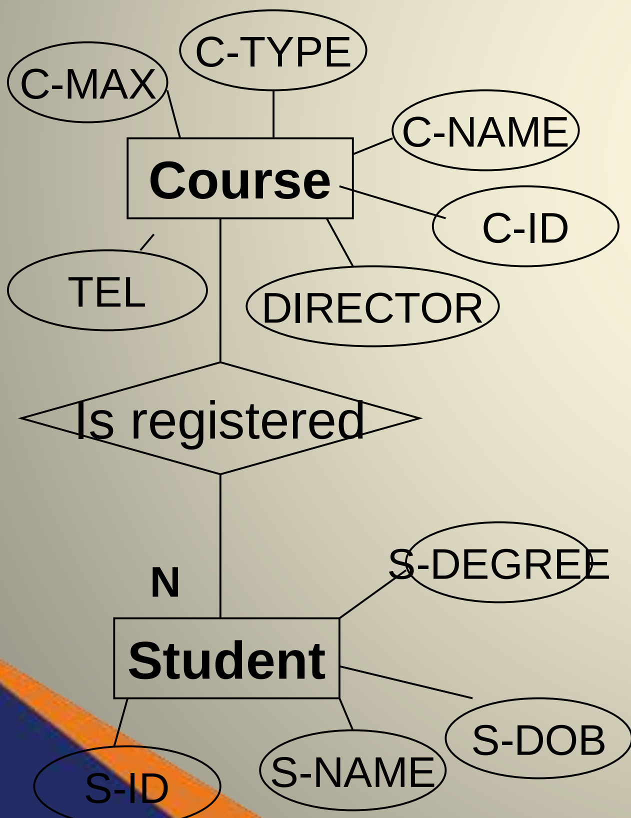
Mapping:1:1 with total participation

- If there is a 1:1 relationship R from entity A to B and if A and B are both in total participation with R then A & B can be collapsed as 1 table



Mapping :1:N Conversion

- 1:N relationships, post the identifier (Primary Key) from the 'one' side as an attribute into the 'many' side



Course

C-ID	C-Name	C-Type	C-Max	Director	Tel
1001	ILS	PG/MSc	75	P Burton	3906

Student **FK**

S-ID	S-Name	S-DofB	S-Degree	C-ID
2005123	M Mouse	1/1/1982	Economics	1001
2005987	D Duck	1/8/1984	Aquatic Eng.	1001

Mapping : M:N Conversion

- For N:M relationships, create a new table and post the identifiers from each of the linked entities as attributes in this table

FK

Class

Cl-ID	Cl-Name	Cl-Type
CS504	IR	Lectures
CS520	DBMS	Lectures

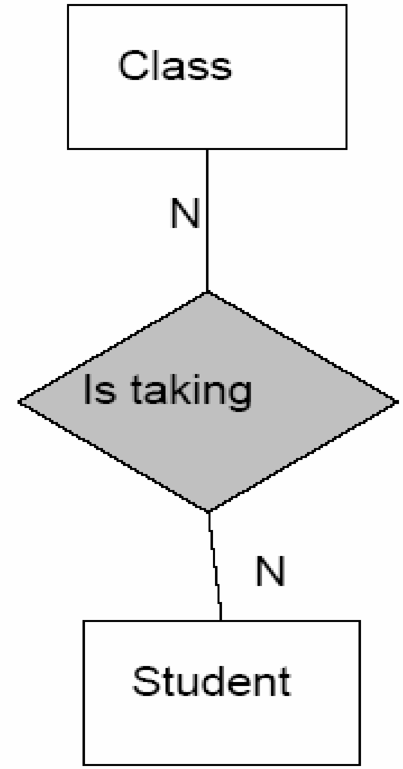
Class-Student

Cl-ID	S-ID	Mark
CS504	2005123	67%
CS504	2005987	59%

FK

Student

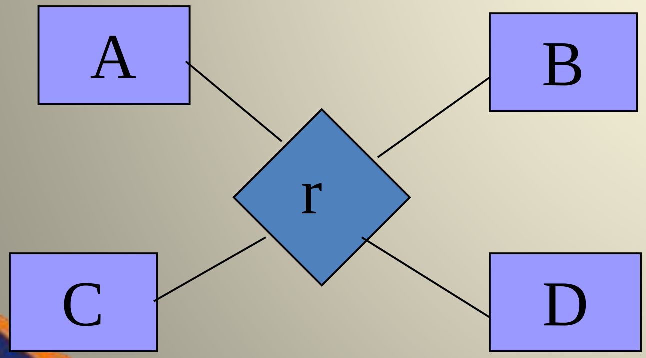
S-ID	S-Name	S-DofB	S-Degree
2005123	M Mouse	1/1/1982	Economics
2005987	D Duck	1/8/1984	Aquatic Eng.



Mapping ..N-array Relationships

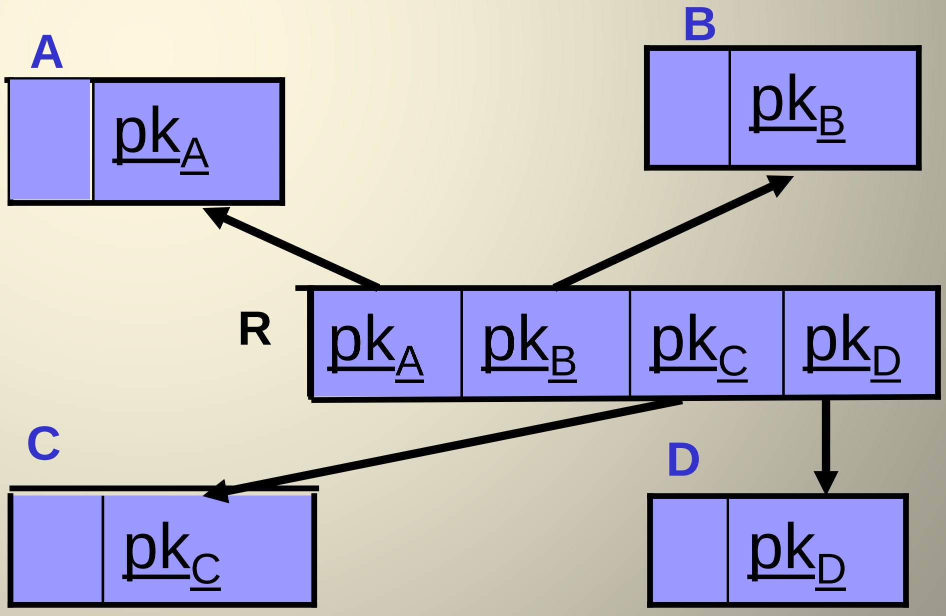
ER Model

– N-ary relationship relation and n foreign keys



Relational Model

"Relationship"



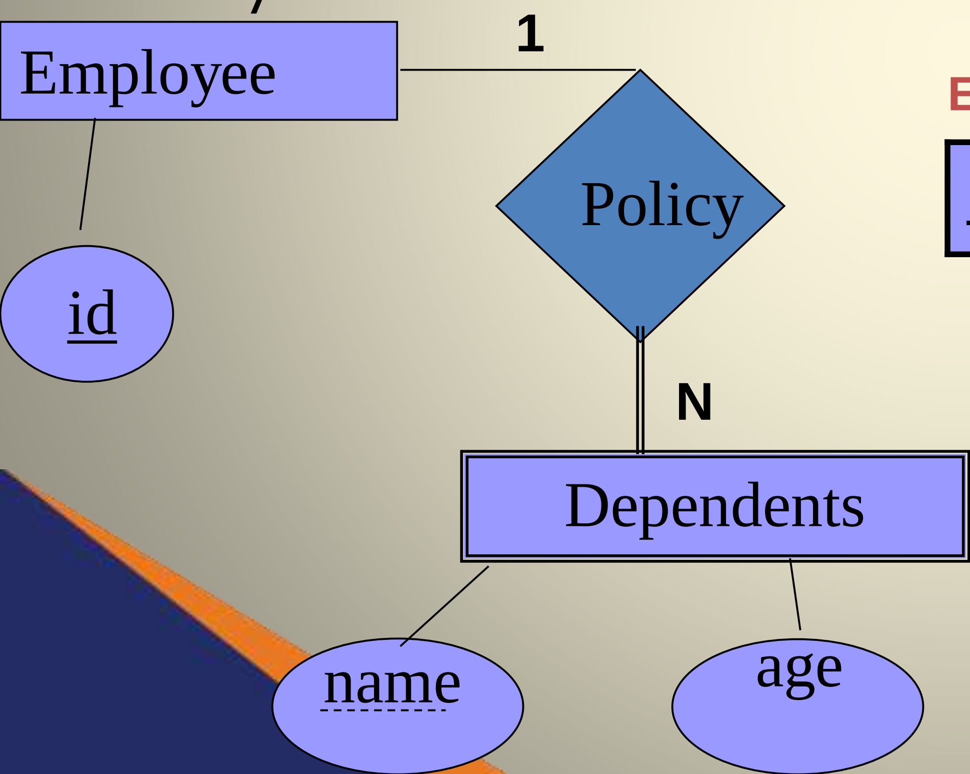
Mapping Weak Entities … (contd.)

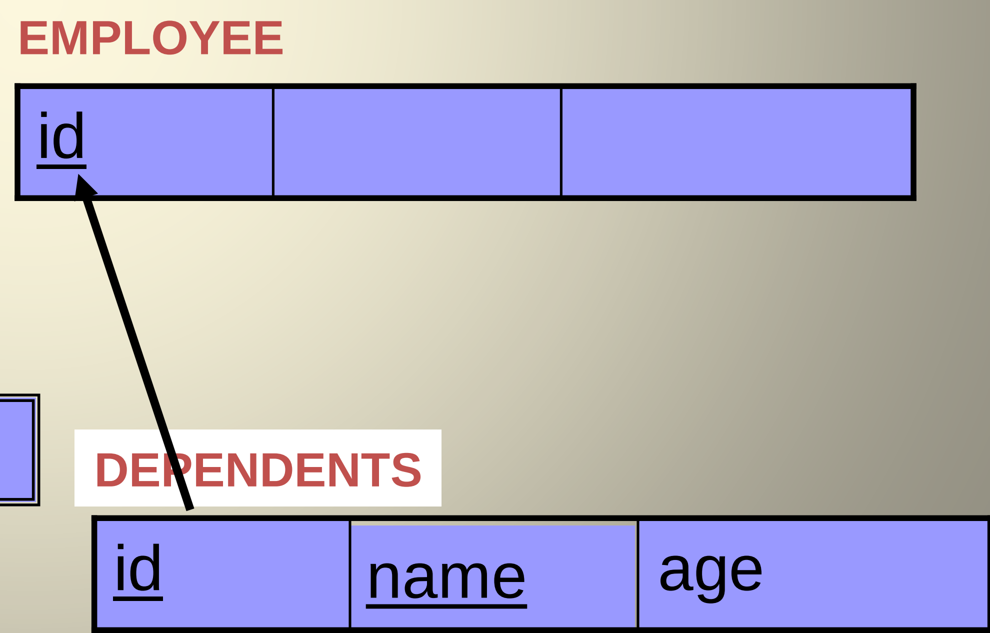
ER Model

Weak Entity

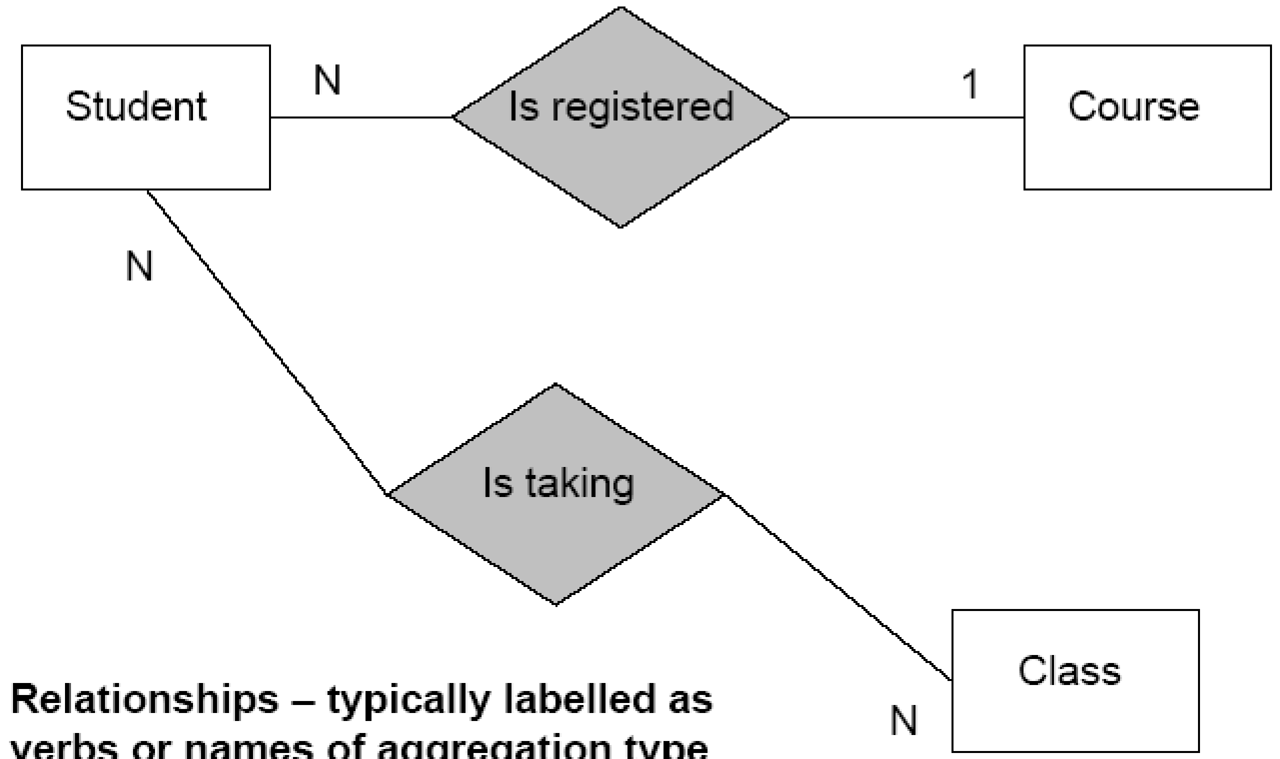
Relational Model

Relations and combination of partial and primary keys





Your Turn – Map to a Relational Schema



Relationships – typically labelled as verbs or names of aggregation type

Lecture Outline

- **ER-to-Relational Mapping Algorithm**
  - Step 1: Mapping of Regular Entity Types
    - Multivalued attributes.
    - Composite attributes
  - Step 2: Mapping of Weak Entity Types
  - Step 3: Mapping of Binary 1:1 Relation Types
  - Step 4: Mapping of Binary 1:N Relationship Types.
  - Step 5: Mapping of Binary M:N Relationship Types.
  - Step 6: Mapping of N-ary Relationship Types.

Summary of Mapping constructs and constraints

Table 7.1 Correspondence between ER and Relational Models

ER Model	**Relational Model**
Entity type	"Entity" relation
1:1 or 1:N relationship type	Foreign key (or "relationship" relation)
M:N relationship type	"Relationship" relation and two foreign keys
n-ary relationship type	"Relationship" relation and n foreign keys
Simple attribute	Attribute
Composite attribute	Set of simple component attributes
Multivalued attribute	Relation and foreign key
Value set	Domain
Key attribute	Primary (or secondary) key

End of Lecture 06

Questions ?

SLIIT

Discover Your Future

Information systems and Data Modeling (IT 1090)

Lecture 07 – Schema Refinement

SLIIT
FACULTY OF COMPUTING

Learning Outcome

- **LO3**: Apply formal methods to refine the data model

Database Design process

1.Requirements Analysis

What does the user want?

2.Conceptual Database Design

Defining the entities and attributes, and the relationships between these

--> The ER model

3.Logical Database Design (Map ER to Relational Schema)

4.Schema Refinement (fine tune)

5.Physical Database Design

Implementation of the design using a Database Management System

6.Security Design

Implement Controls to ensure security and integrity

Normalization

- Conceptual Modeling is a subjective process
- Therefore, the schema after the logical database design phase may not be very good (contain redundant data)
- However, there are formalisms to ensure that the schema is good.
- This process is called Normalization

Normalization

- Relational database schema = set of relations
- Relation = set of attributes
- How we group the attributes to relations is **very important**
- Normalization or Schema Refinement help determine “**GOOD**” relations

Purpose of Normalization

- To avoid redundancy by storing each ‘fact’ within the database only once.
- To put data into a form that conforms to relational principles (e.g., single valued attributes, each relation represents one entity) - no repeating groups.
- To put the data into a form that is more able to accurately accommodate change.
- To avoid certain updating ‘anomalies’.
- To facilitate the enforcement of intergrity constraints.

Redundancy and Data Anomalies

Redundant data is where we have stored the same 'information' more than once. i.e., the redundant data could be removed without the loss of information.

Example: We have the following relation that contains staff and department details:

staffNo	job	dept	dname	city
SL10	Salesman	10	Sales	Stratford
SA51	Manager	20	Accounts	Barking
DS40	Clerk	20	Accounts	Barking
OS45	Clerk	30	Operations	Barking

staffNo	job	dept	dname	city
SL10	Salesman	10	Sales	Stratford
SA51	Manager	20	Accounts	Barking
DS40	Clerk	20	Accounts	Barking
OS45	Clerk	30	Operations	Barking

Such ‘redundancy’ could lead to the following ‘anomalies’

Insert Anomaly: We can’t add a new a dept without inserting a member of staff that works in that department

Update Anomaly: Change the name of the Accounts dept to Finance dept. We have to change all other records to avoid update anomaly.

Deletion Anomaly: Employee SL10 resigns. We remove the record. With that we lose all information pertaining to the Sales dept.

Repeating Groups

Is an attribute (or set of attributes) that can have more than one value

staffNo	job	dept	dname	city	contact number
SL10	Salesman	10	Sales	Stratford	018111777, 018111888, 079311122
SA51	Manager	20	Accounts	Barking	017111777
DS40	Clerk	20	Accounts	Barking	
OS45	Clerk	30	Operations	Barking	079311555

Repeating Groups are not allowed in a relational design, since all attributes have to be ‘atomic’ - i.e., there can only be one value per cell in a table!

Schema Refinement

A relation with redundancy can be refined by

Decomposing the relation into smaller relations…

- contain the same information
- with no redundancy

Problems related to decompositions

What problems (if any) does a given decomposition cause?

To help:

Two properties of decompositions:

- Loss-less join property
- Dependency preserving property

Normal forms have been proposed to preserve above properties.

Loss-less join property

Loss-less join property: we might lose information if we decompose relations…

S

S	P	D
S1	**P1**	**D1**
S2	**P2**	**D2**
S3	**P1**	**D3**

R_1

S	P
S1	**P1**
S2	**P2**
S3	**P1**

R_2

P	D
P1	**D1**
P2	**D2**
P1	**D3**

Loss-less join property (contd.)

Joining them together, we get spurious tuples…

$R_1 \bowtie R_2$

S	P	D
S1	**P1**	**D1**
S1	**P1**	**D3**
S2	**P2**	**D2**
S3	**P1**	**D1**
S3	**P1**	**D3**

Dependency-preserving property

- Consider previous example
- Dependency-preserving property: The set of dependencies in relation S can be verified by a set of dependencies in relation R_1 and relation R_2

S

S	P	D
S1	**P1**	**D1**
S2	**P2**	**D2**
S3	**P1**	**D3**

R_1

S	P
S1	**P1**
S2	**P2**
S3	**P1**

R_2

P	D
P1	**D1**
P2	**D2**
P1	**D3**

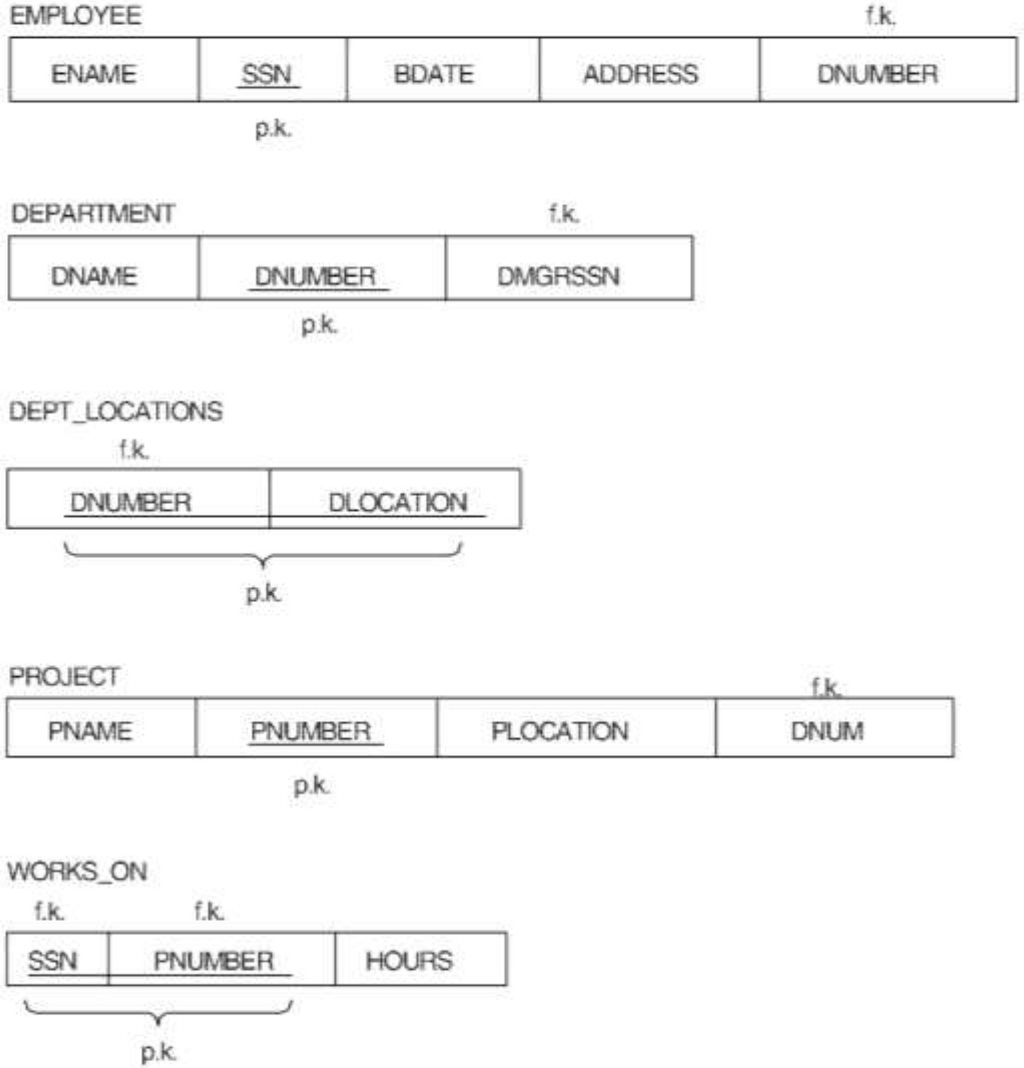
Informal Guidelines

Guideline 1 : *The relation's semantics should be clear and easy to explain.*

Attributes of different entities (EMPLOYEEs, DEPARTMENTs, PROJECTs) should not be mixed in the same relation
Only foreign keys should be used to refer to other entities

Do not combine attributes from multiple entity types and relationship types in to a single relation

Figure 14.1 Simplified version of the COMPANY relational database schema.



Informal Guidelines

Guideline 2 : *Minimize the storage space used by the base relations and design a schema that does not suffer from the insertion, deletion and update anomalies*

Relations suffer from anomalies

EMP_PROJ

		redundancy		redundancy	
SSN	PNUMBER	HOURS	ENAME	PNAME	PLOCATION
123456789	1	32.5	Smith,John B.	ProductX	Bellaire
123456789	2	7.5	Smith,John B.	ProductY	Sugarland
666884444	3	40.0	Narayan,Ramesh K.	ProductZ	Houston
453453453	1	20.0	English,Joyce A.	ProductX	Bellaire
453453453	2	20.0	English,Joyce A.	ProductY	Sugarland
333445555	2	10.0	Wong,Franklin T.	ProductY	Sugarland
333445555	3	10.0	Wong,Franklin T.	ProductZ	Houston
333445555	10	10.0	Wong,Franklin T.	Computerization	Stafford
333445555	20	10.0	Wong,Franklin T.	Reorganization	Houston
999887777	30	30.0	Zelaya,Alicia J.	Newbenefits	Stafford
999887777	10	10.0	Zelaya,Alicia J.	Computerization	Stafford
987987987	10	35.0	Jabbar,Ahmad V.	Computerization	Stafford
987987987	30	5.0	Jabbar,Ahmad V.	Newbenefits	Stafford
987654321	30	20.0	Wallace,Jennifer S.	Newbenefits	Stafford
987654321	20	15.0	Wallace,Jennifer S.	Reorganization	Houston
888665555	20	null	Borg,James E.	Reorganization	Houston

Informal Guidelines (contd.)

Guideline 3 : *Relations should be designed such that their tuples will have as few NULL values as possible*

Attributes that are NULL frequently could be placed in separate relations (with the primary key)

- Reasons for nulls:
  attribute not applicable or invalid
  attribute value unknown (may exist)
  value known to exist, but unavailable

Guideline 4 : *Design schemas so that they can be joined with equality conditions on attributes that are primary key , foreign key (this will avoid spurious tuples generated).*

Formal Process

Formal process for good relational schema:

- To avoid the above mentioned issues in the relational schema, we can apply a formal process called Normalization
- Normalization is based on functional dependencies

Functional Dependencies

- FDs are used to specify *formal measures* of the "goodness" of relational designs
- FDs and keys are used to define **normal forms** for relations

Functional dependency

- A functional dependency, is a constraint between two sets of attributes
- denoted by $X \rightarrow Y$,
  - X functionally determines Y
  - Y is functionally dependent on X
- where X and Y are sets of attributes in relation R, specifies the following constraint:

  Let t_1 and t_2 be tuples of relation R for any given instance

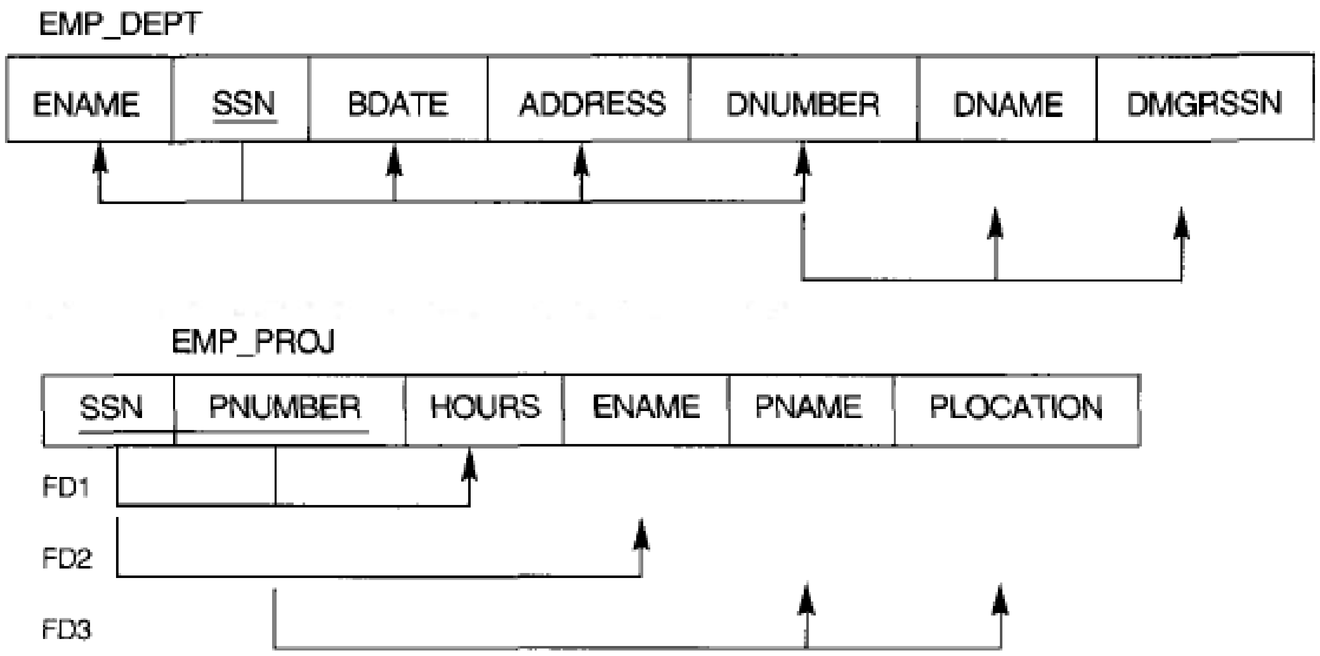
  Whenever $t_1[X] = t_2[X]$ then $t_1[Y] = t_2[Y]$

  where $t_i[X]$ represents the values for X in tuple t_i

Functional dependency

- Describes the relationship between attributes in a relation.
- If A and B are attributes of relation R, B is functionally dependent on A (denoted A → B), if each value of A is associated with exactly one value of B.
- (A and B may each consist of one or more attributes)

Functional dependency



(Ssn,Pnumber) -> Hours (SSN & PNUMBER detrmines hrs emp work on a project)
Ssn -> Ename
Pnumber -> (pname, plocation) (PNUMBER determines pname & location)

Functional dependency

redundancy redundancy

EMP_PROJ

SSN	PNUMBER	HOURS	ENAME	PNAME	PLOCATION
123456789	1	32.5	Smith,John B.	ProductX	Bellaire
123456789	2	7.5	Smith,John B.	ProductY	Sugarland
666884444	3	40.0	Narayan,Ramesh K.	ProductZ	Houston
453453453	1	20.0	English,Joyce A.	ProductX	Bellaire
453453453	2	20.0	English,Joyce A.	ProductY	Sugarland
333445555	2	10.0	Wong,Franklin T.	ProductY	Sugarland
333445555	3	10.0	Wong,Franklin T.	ProductZ	Houston
333445555	10	10.0	Wong,Franklin T.	Computerization	Stafford
333445555	20	10.0	Wong,Franklin T.	Reorganization	Houston
999887777	30	30.0	Zelaya,Alicia J.	Newbenefits	Stafford
999887777	10	10.0	Zelaya,Alicia J.	Computerization	Stafford
987987987	10	35.0	Jabbar,Ahmad V.	Computerization	Stafford
987987987	30	5.0	Jabbar,Ahmad V.	Newbenefits	Stafford
987654321	30	20.0	Wallace,Jennifer S.	Newbenefits	Stafford
987654321	20	15.0	Wallace,Jennifer S.	Reorganization	Houston
888665555	20	null	Borg,James E.	Reorganization	Houston

Functional dependency

An FD $A_1A_2 \dots A_n \rightarrow B_1B_2 \dots B_m$ is

- *trivial* if the B's are a subset of the A's, i.e., $\{B_1, B_2, \dots B_n\} \subseteq \{A_1, A_2, \dots A_n\}$.
- *non-trivial* if at least one B is not among the A's, i.e., $\{B_1, B_2, \dots B_n\} - \{A_1, A_2, \dots A_n\} \neq \emptyset$.

Trivial :- staffNo, sName → sName
staffNo, sName → staffNo

NonTrivial :- staffNo, sName → salary

Functional dependency

Given a set of FDs F, we can *infer* additional FDs that hold whenever the FDs in F hold

Armstrong's inference rules:

IR1. (**Reflexive**) If Y *subset-of* X, then X → Y
IR2. (**Augmentation**) If X → Y, then XZ → YZ
(Notation: XZ stands for X U Z or {X, Z})
IR3. (**Transitive**) If X → Y and Y → Z, then X → Z

Additional inference rules that are useful:

(**Decomposition**) If X → YZ, then X → Y and X → Z
(**Union**) If X → Y and X → Z, then X → YZ
(**Psuedotransitivity**) If X → Y and WY → Z, then WX → Z

Functional dependency

Closure of a set F of FDs is the set F^+ of all FDs that can be inferred from F

Closure of a set of attributes X with respect to F is the set X^+ of all attributes that are functionally determined by X

X^+ can be calculated by repeatedly applying IR1, IR2, IR3 using the FDs in F

Normalization

- Key points:
  - Redundancy is based on functional dependencies
  - Therefore, *normalization* is based on functional dependencies
  - Therefore, relational database schema need to be refined
- Schema Refinement Steps:
  - Determine Functional dependencies for relation
  - Find all keys in relation
  - Normalize the relation

Database basics - review

Review of some terms

- **Key:** A key is a superkey with the additional property that removal of any attributes from the key will not satisfy the key condition (minimal set of attributes)
    - eg : Student-No
- **Superkey:** Set of attributes S in relation R that can be used to identify each tuple uniquely.
    - eg : (Student-No, name)

Database basics - review

- **Candidate Key:** Each key of a relation is called a candidate key
- **Primary Key:** A candidate key is chosen to be the primary key
- **Prime Attribute:** an attribute which is a member of a candidate key
- **Nonprime Attribute:** An attribute which is not prime

Stages of Normalization

- There are many Normal Forms proposed to reduce redundancies

Unnormalised **(UDF)**

Remove repeating groups

First normal form **(1NF)**

Remove partial dependencies

Second normal form **(2NF)**

Remove transitive dependencies

Third normal form **(3NF)**

Remove remaining functional dependency anomalies

Boyce-Codd normal form **(BCNF)**

Un-normalized Normal Form (UNF)

A relation is un-normalized when it has not had any normalization rules applied to it, and it suffers from various anomalies

Normalization - 1^{st} Normal Form

- A relation R is in first normal form (1NF) if domains of all attributes in the relation are *atomic* (simple & indivisible).
- Avoid multi valued & composite attributes
- Remove repeating groups into a new relation

Steps from UNF to 1NF:

- Remove the outermost repeating group and create a new relation
- Add to this relation a copy of the PK of the original relation.
- Name the new relation
- Determine the PK of the new relation
- Repeat steps until no more repeating groups.

Normalization - 1st Normal Form

For example:

DEPARTMENT (Dname,<u>Dnumber</u>, DMGRSSN, (DLocation))

DEPARTMENT

DNAME	<u>DNUMBER</u>	DMGRSSN	DLOCATIONS
Research	5	333445555	{Mathara, Kandy, Colombo}
Administration	4	987654321	{Malabe}
Headquarters	1	888665555	{Colombo}

- Department is in UNF
- Department relation not in 1NF
- How to take into 1NF ?

Solution

: Create a separate DEPT_LOCATION relation with foreign key

DEPT_LOCATIONS

DNUMBER	DLOCATIONS
1	Colombo
4	Malabe
5	Mathara
5	Kandy
5	Colombo

DEPARTMENT

DNAME	DNUMBER	DMGRSSN
Research	5	333445555
Administration	4	987654321
Headquarters	1	888665555

- Remove the attribute DLOCTION and place it in a separate relation DEPT_LOCTIONS along with the primary key DNUMBER of DEPARTMENT.
- The PK is the combination {DNUMBER, DLOCTION}
- This decompose the non-INF relation into two INF relation.

Full Functional Dependency

A functional dependency, X → Y is a full functional dependency if removal of any attribute A from X means that the dependency does not hold

(i.e. (X –{A}) → Y does not hold)

TEACH

STUDENT	COURSE	TEACHER	CAMPUS
Narayan	**Database**	**ABC**	**Colombo**
Smith	**Database**	**XYZ**	**Malabe**
Nalin	**Operating Systems**	**Samantha**	**Colombo**
Kamal	**Operating Systems**	**ABC**	**Malabe**
Janith	**Database**	**ABC**	**Colombo**
Ranil	**Operating Systems**	**Samantha**	**Colombo**
Saman	**Operating Systems**	**ABC**	**Malabe**
Ruwan	**Database**	**XYZ**	**Malabe**

{Teacher, Campus} → Course

Teacher does NOT determine course

Normalization – 2nd Normal Form

- A relation R is in second normal form (2NF) if every nonprime attribute A in R is **not partially dependent** on any key of R
- Remove partial functional dependencies into a new relation

Example: Not in 2NF

TEACHER	CAMPUS	COURSE	ADDRESS
Kapila	**Colombo**	**Database**	**BoC Merchant Tower**
Nuwan	**Malabe**	**Database**	**New Kandy Road**
Samantha	**Colombo**	**Operating Systems**	**BoC Merchant Tower**
Kapila	**Malabe**	**Operating Systems**	**New Kandy Road**

Partial dependency

Normalization – 2nd Normal Form

Steps from 1NF to 2NF:

- Remove the attributes that are only partially functionally dependent on the composite key, and place them in a new relation.
- Add to this relation a copy of the attribute(s) which are the determinants of these offending attributes. These will automatically become the primary key of this new relation.
- Name the new relation
- Rename the original relation

Normalization – 2nd Normal Form

Example: After normalized into 2NF

TEACHER	CAMPUS	COURSE
Kapila	**Metro**	**Database**
Nuwan	**Malabe**	**Database**
Samantha	**Metro**	**Operating Systems**
Kapila	**Malabe**	**Operating Systems**

CAMPUS	ADDRESS
Metro	**BoC Merchant Tower**
Malabe	**Malabe Campus**

EMP_PROJ

<u>SSN</u>	<u>PNUM</u>	HOURS	ENAME	PNAME	LOC

FD1

FD2

FD3

FD1

<u>SSN</u>	<u>PNUM</u>	HOURS

FD2

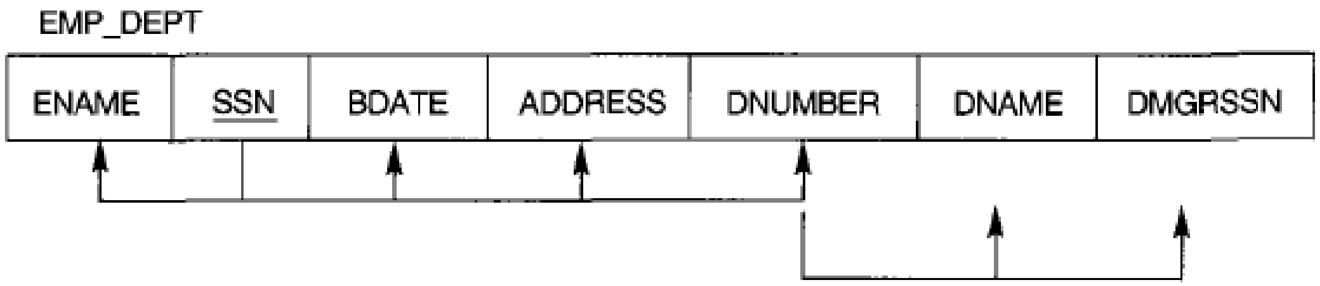
<u>SSN</u>	ENAME

FD3

<u>PNUM</u>	PNAME	PLOC

Normalization – 3rd Normal Form

- A relation R is in 3rd normal form (3NF) if every
  - R is in 2NF, and
  - No nonprime attribute is transitively dependent on any key
  - Remove transitive dependencies into a new relation



Normalization – 3rd Normal Form

Steps from 2NF to 3NF:

- Remove the offending attributes that are transitively dependent on non-key attribute(s), and place them in a new relation.
- Add to this relation a copy of the attribute(s) which are the determinants of these offending attributes. These will automatically become the primary key of this new relation.
- Name the new relation.
- Rename the original relation.

Normalization – 3rd NormalForm

FD1

ENAME	SSN	BDATE	ADD	DNUM

FD2

DNUM	DNAME	DMGR

Boyce-Codd Normal Form

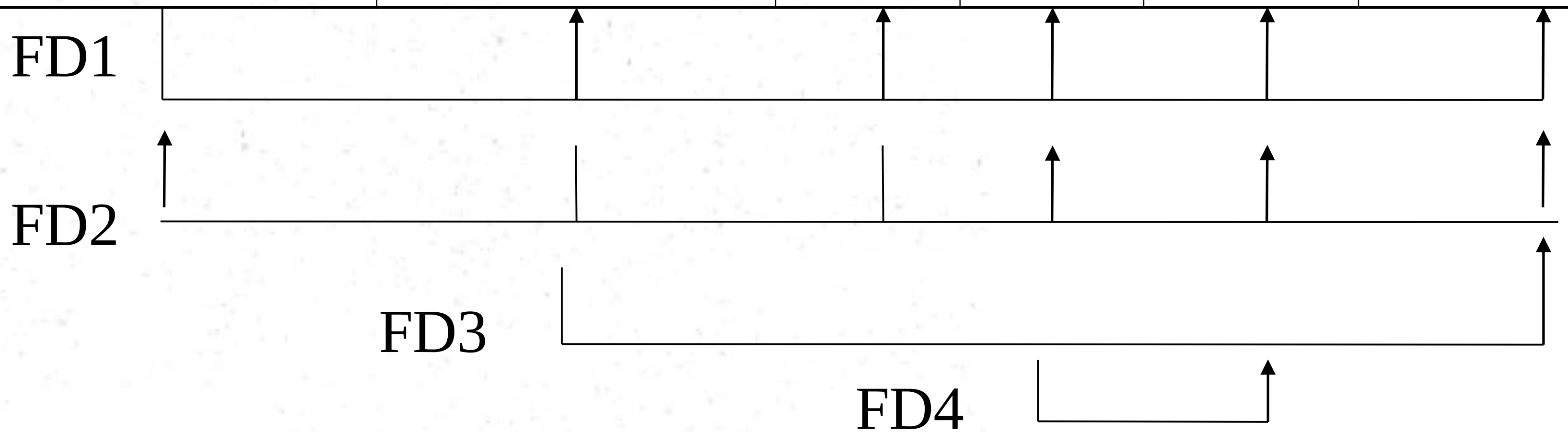
A relation schema is in Boyce-Codd Normal Form

- If every nontrivial functional dependency X→A hold in R, then X is a superkey of R
- A relation is in BCNF if and only if, every determinant is a candidate key
- Every relation in BCNF is also in 3NF

- A relation is in BCNF, if and only if, every determinant is a candidate key

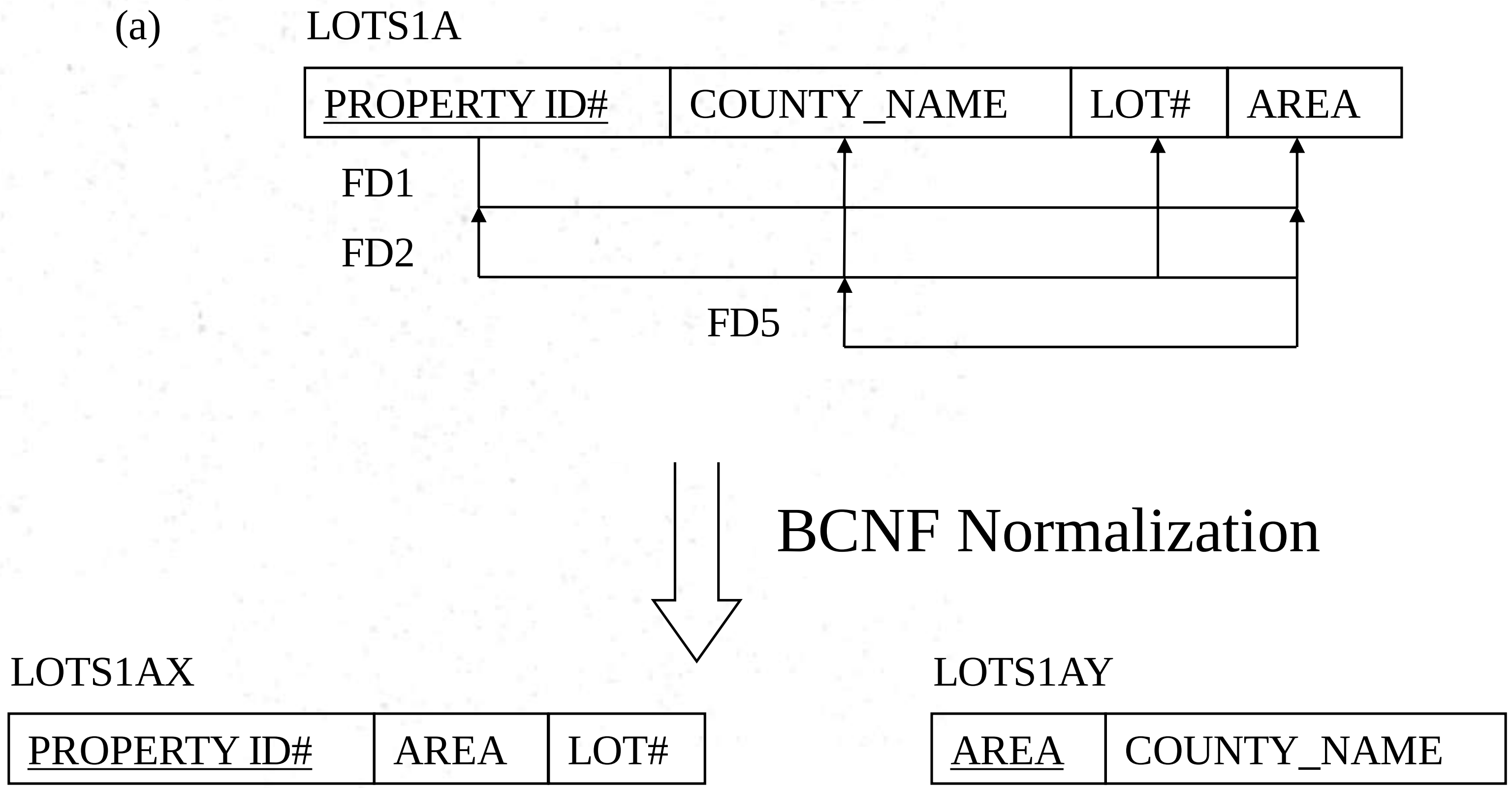
Normalization –(BCNF)

PROPERTY_ID	COUNTY_NAME	LOT#	AREA	PRICE	TAX_RATE



Keys: PropertyID, (County_Name, Lot#)

Normalization –BCNF



Normalization

- 1NF, 2NF, 3NF & BCNF guarantee to preserve lossless join property

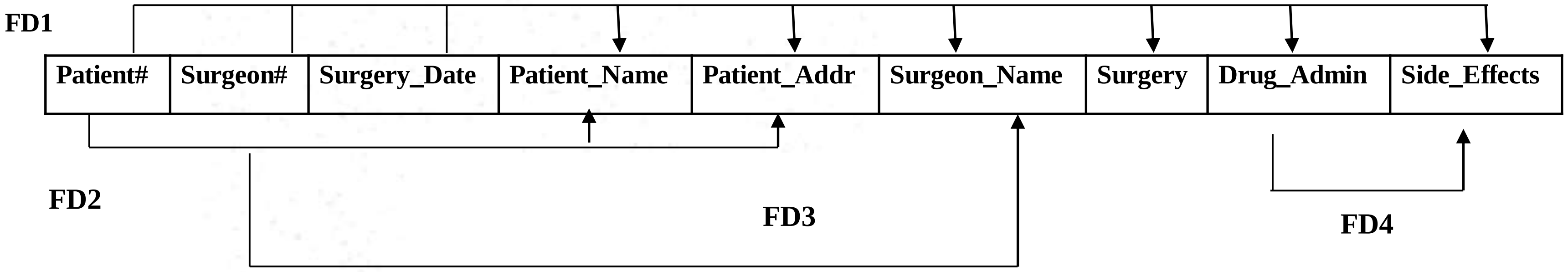
- However, BCNF does not guarantee to preserve dependency preserving property

De-normalization

Sometime for performance reasons, database designer may leave the relation in a lower normal form. This process is known as de-normalization.

Exercise 1

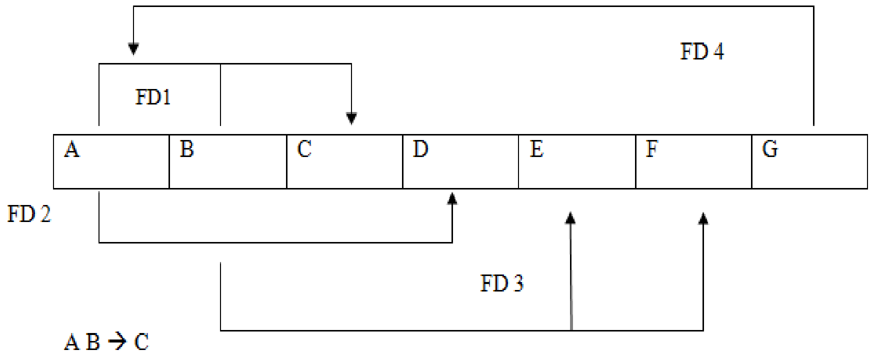
Patient #	Surgeon #	Surgery Date	Patient Name	Patient Addr	Surgeon Name	Surgery	Drug admin	Side Effects
1111	311	12-Jun-95	John White	15 New St. New York, NY	Michael Diamond	Kidney stones removal	none	rash
1234	243	05-Apr-94	Mary Jones	10 Main St. Rye, NY	Charles Field	Eye Cataract removal	Tetracyclin e	Fever
1234	467	10-May-95	Mary Jones	10 Main St. Rye, NY	Patricia Gold	Thrombos is removal	none	none
2345	189	08-Jan-96	Charles Brown	Dogwood Lane Harrison, NY	David Rosen	Open Heart Surgery	Cephalosp orin	none



1. What normal form is the relation in?
2. If it is not in 3NF, convert it to 3NF. Explain your answer.

Exercise 2

Consider the following relational schema for R:
R(A, B,C,D,E,F,G)
AB is the primary key in the relation. Assume that the following dependencies exist:



A B → C

A → D

B → E F

G → A

1. What normal form is the relation in?
2. If R is not in BCNF, convert it to BCNF. Explain your answer.

End of Lecture 07

Questions ?