System Analysis & Design

PART 6 – Systems Design and Modelling

BY

DR ABDULLAH ALZAHRANI

د. عبدالله بن عبدالرحمن الزهراني

AAHZAHRANI@UQU.EDU.SA

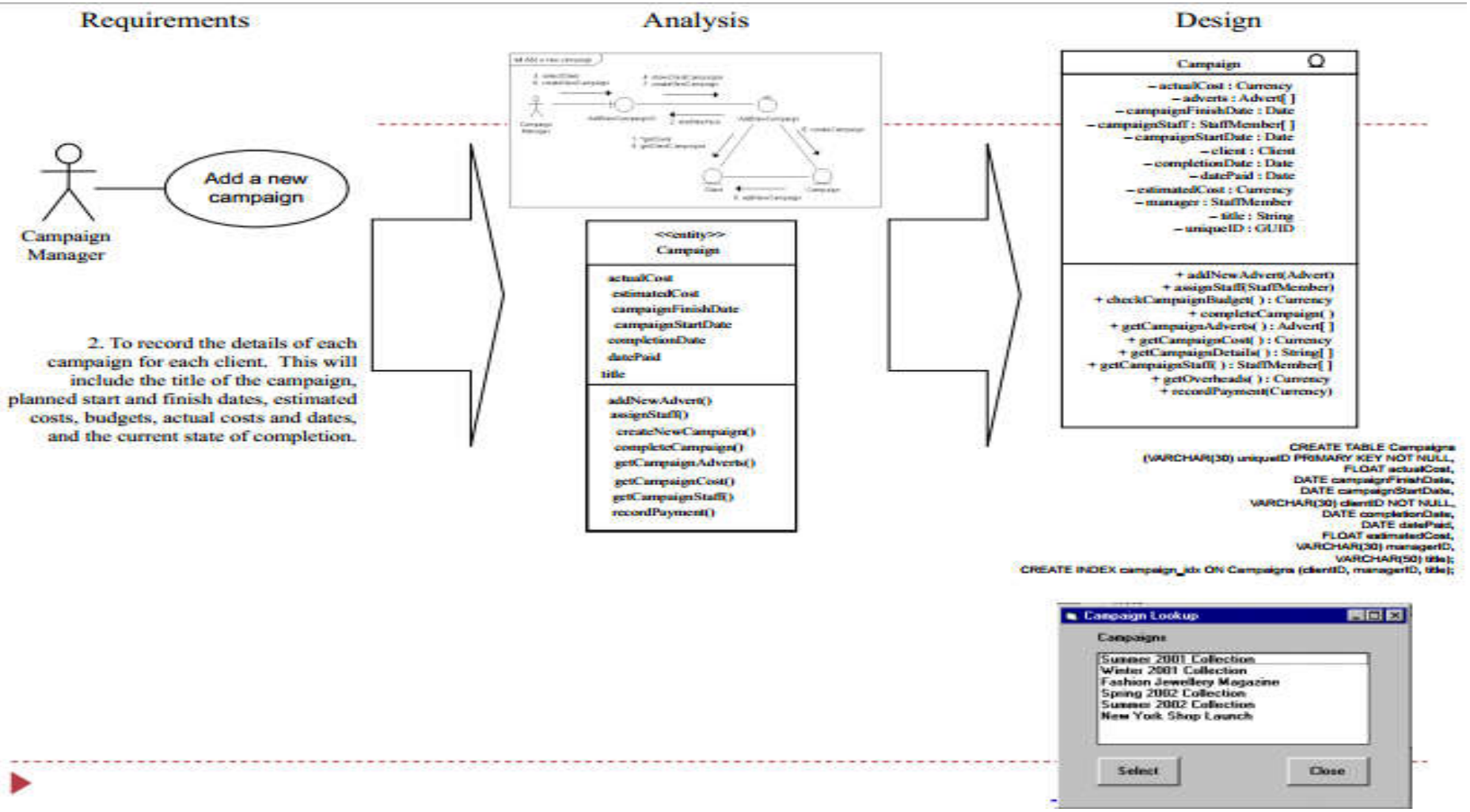
Systems Design and Modelling

- Systems Design
- Process Modelling
- Data Flow Diagrams
- Database Design

Systems Design (1)

When Does Analysis Stop and Design Start?

- In a waterfall life cycle there is a clear transition between the two activities
- In an iterative life cycle the analysis of a particular part of the system will precede its design, but analysis and design may be happening in parallel
- It is important to distinguish the two activities and the associated mindset
- We need to know ‘what’ before we decide ‘how’



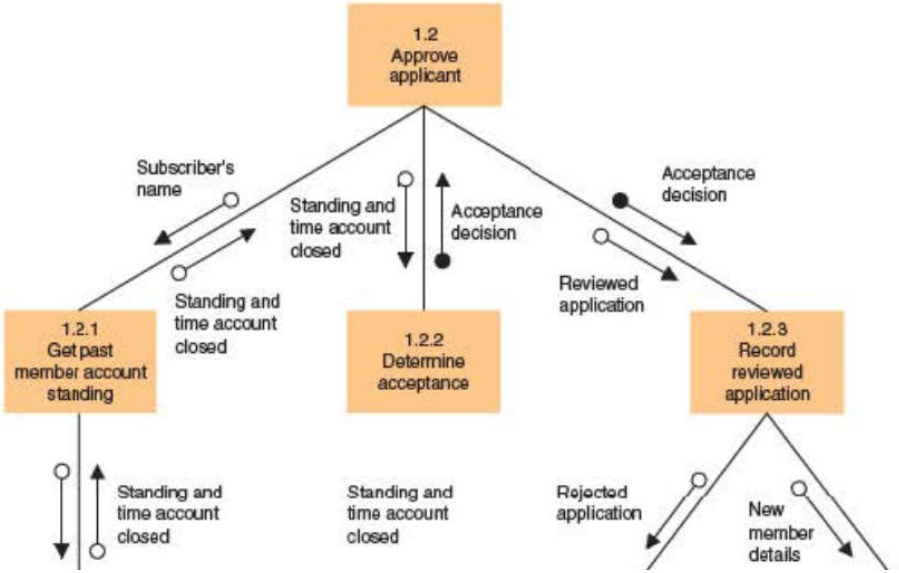
Systems Design (2)

Systems Design (3)

- ***SYSTEM DESIGN*** the specification or construction of a technical, computer-based solution for the business requirements identified in a system analysis. (Note: Increasingly, the design takes the form of a working prototype.)

- ***DESIGNER*** System designers are technology specialists for Information systems
  - system designers are interested In information technology choices and in the design of systems that use chosen technologies. Today's system designers tend to focus on technical specialities.

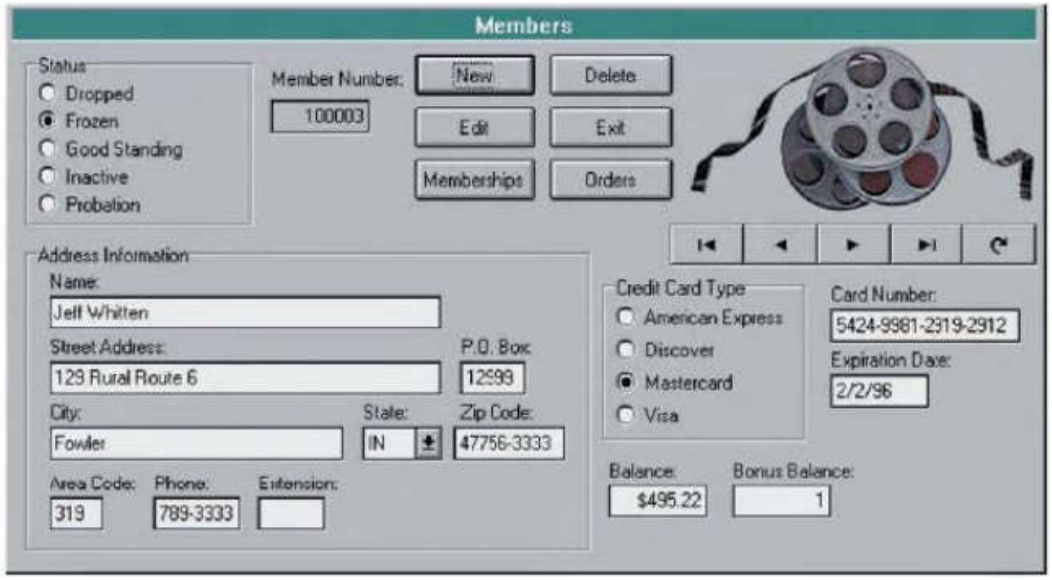
Systems Design (4)



❖ *METHODS*:

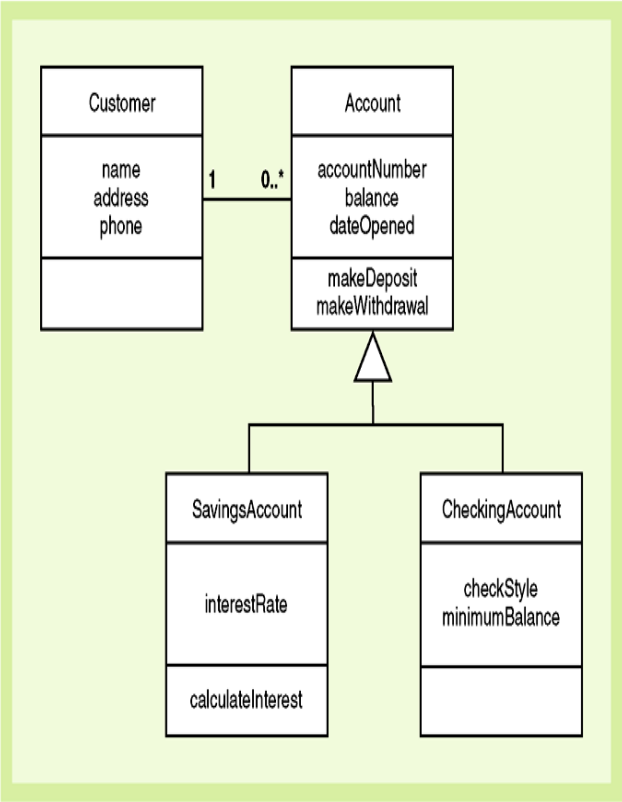
1. Modern Structured Design
   - ✓ Structured design techniques help developers deal with the ***size*** and ***complexity*** of programs
   - ✓ A system design technique that focuses on ***processes*** and ***decomposes*** the system's processes into manageable components.

Systems Design (5)



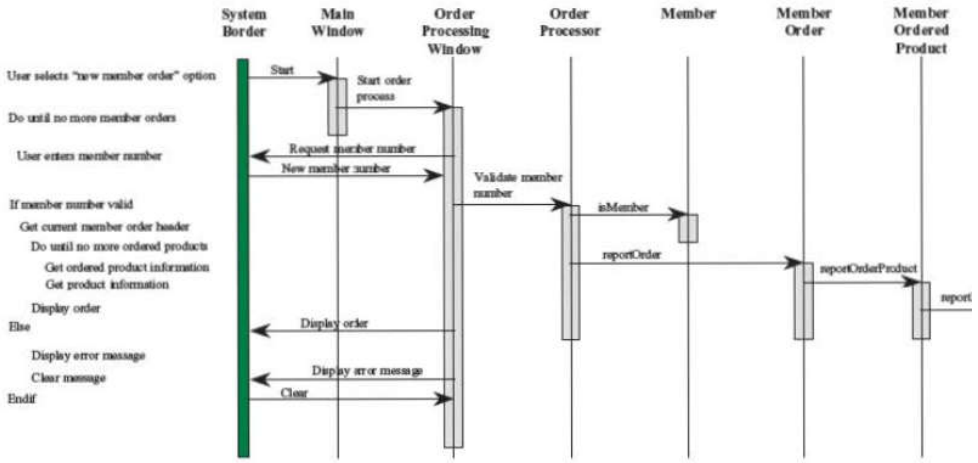
❖ *METHODS*:

2. Prototyping
   - ✓ The prototyping approach is an iterative process involving a close working relationship between the designer and the users.
   - ✓ doesn't necessarily fulfil all design requirements.



Systems Design (4)

- ***METHODS***:
  3. Object-Oriented Design
     - ✓ an approach used to specify the software solution in terms of collaborating objects, their attributes, and their methods
     - ✓ the newest design strategy.



Process Modelling

- ***MODEL***
  - a graphic representation of reality.
- ***LOGICAL MODEL***
  - a non-technical pictorial representation that depicts what a system is or does. Synonyms are **essential model, conceptual model** and **business model**
- ***PHYSICAL MODEL***
  - a technical pictorial representation that depicts what a system is or does and how the system is implemented. Synonyms are **implementation model** and **technical model**.
- ***PROCESS MODELLING***
  - involves graphically representing the functions, or processes that capture, manipulate, store and distribute data between a system and its environment and among system components

Data Flow Diagrams (DFD)

- ***Data flow diagrams (DFD)***
  - A common and traditional form of process modelling technique
  - Graphically illustrate movement of data between external entities and the processes and data stores within a system

- ***DFD's*** are not as good as ***flowcharts*** to depict details of physical systems

Data Flow Diagrams (DFD) (2)

❖ ***CONTEXT DIAGRAM***

- The highest-level view of an organizational system that shows the system boundaries, external entities that interact with the system and the major information flows between the entities and the system
- All context diagrams have only one process labeled **"0"**
- No data stores appear on a context diagram

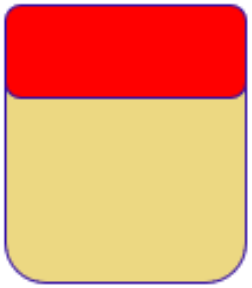
❖ ***LEVEL-0 DIAGRAM***

- A data flow diagram (DFD) that represents a system's major processes, data flows and data stores at a high level of detail
- Each process has a number that ends in **.0**

❖ DFD hides many physical characteristics of system

- We do not know **timing** of when data flow is produced, how **frequently** it is produced, what **size** of data is sent

process



data store

source/sink

data flow

Data Flow Diagrams (DFD) (3)

- Symbols
  - ***Four*** symbols are used to represent both physical and logical information systems

Symbols definitions (1)

Data Flow

- Depicts **data in motion** and moving from one place to another in the system.
- Example: results of query of database, contents of printed report
- Data flow is data that move together
  - Data flow can be composed of many individual pieces of data that are generated at the same time and flows together

Data Store

- Depicts **data at rest**
- May represent one of many different physical locations for data:
  - File folder
  - Computer-based file
  - Notebook
- Might contain data about customers, students, customer orders

Symbols definitions (2)

Process

- Depicts **work** or **action performed on data** so that they are transformed, stored or distributed

Source/Sink

- Depicts the **origin** and/or **destination** of the **data**
- Sometimes referred to as an external entity so they are outside system and define boundaries of system
- Because they are external, many characteristics are not of interest to us
- Data must originate from outside a system from one or more sources and system must produce information to one or more sinks
- consist of – another organization, a person inside or outside business, another information system

Data Flow Diagramming Symbols

Data flow is shown as an **arrow** labeled with a meaningful name for data (*all elements of data moving as part of one packet*) in motion – sales receipt, customer order.

Source/Sink is shown as a **square** and has a name that states what external agent is – customer, teller.

Data store is shown as **rectangle** *without its right vertical side* and *left side* has a small box used to number the data store and inside the main part of rectangle is a meaningful label – student file.

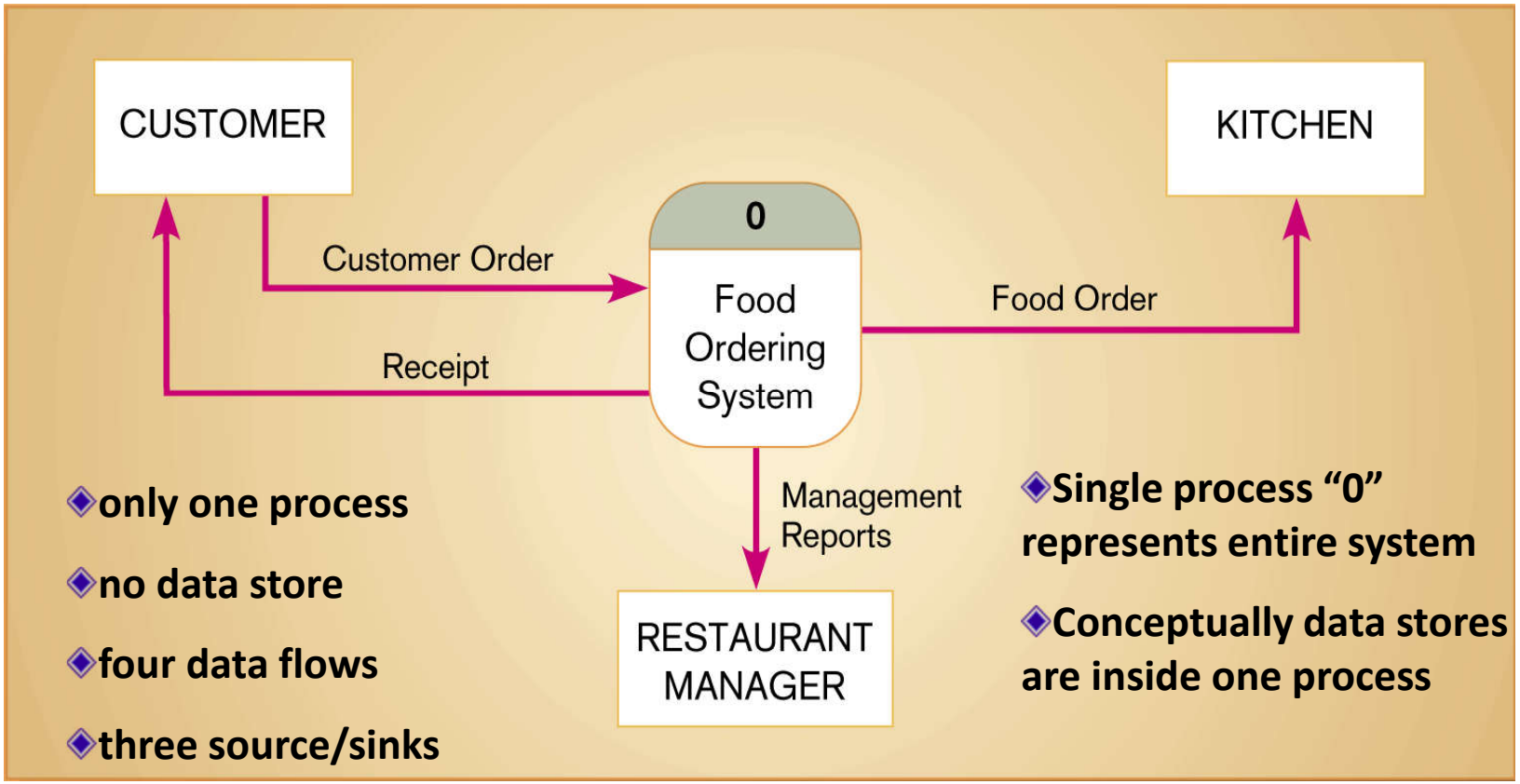
Process is shown as a **rectangle** with rounded corners with a line dividing it into two parts – upper part has the *number of process* and lower part has *name of process*

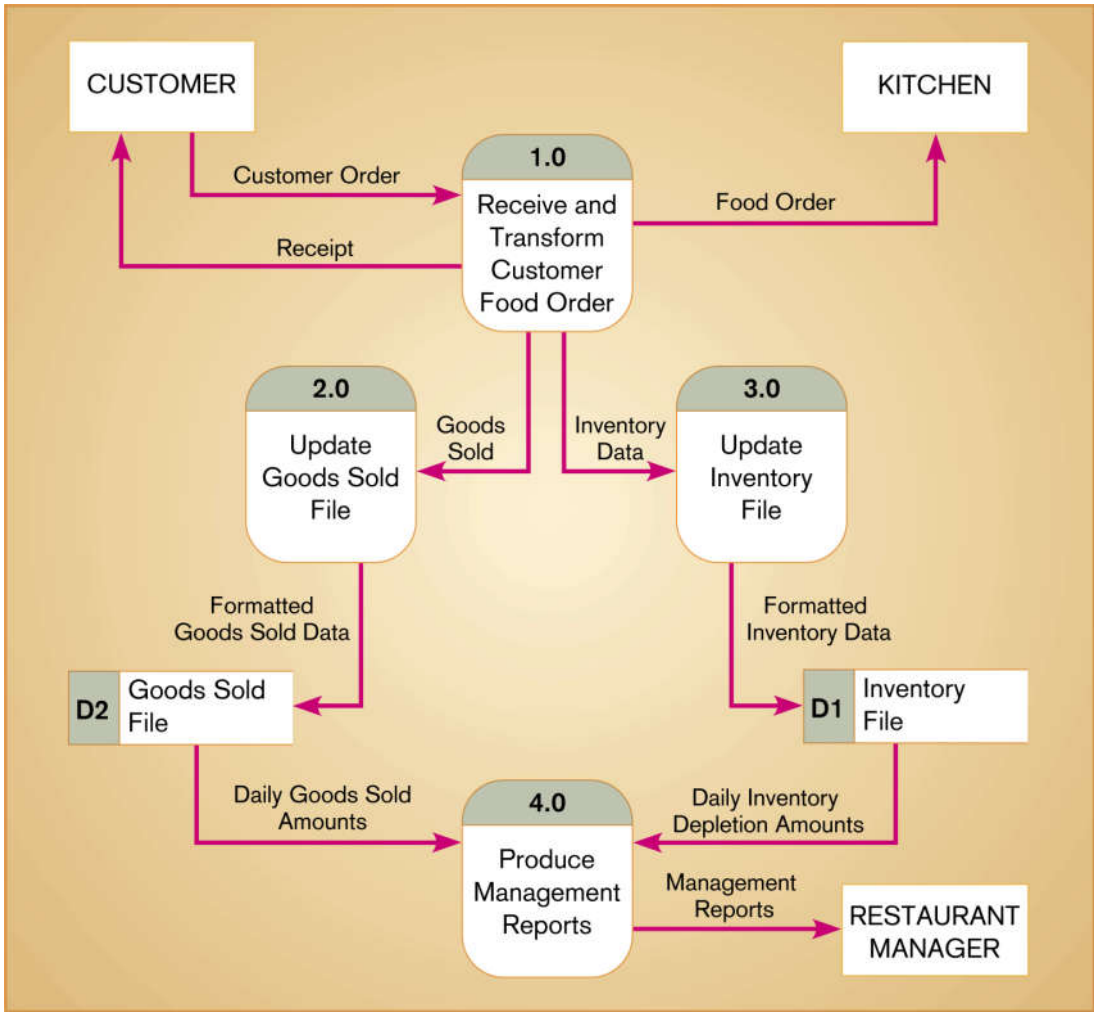
Developing DFDs: An Example

- **Process names**
  - should be clear yet concise
  - begin with an action verb, such as receive, generate, calculate, merge, sort, read, write…….
  - should capture essential action of the process in just few words yet describe the process' action so that reading its name explains what process does

- **An Example**
  - Hoosier Burger's automated food ordering system
  - ***CONTEXT DIAGRAM*** contains no data stores
  - Next step is (***LEVEL-0***) to expand the context diagram to show the breakdown of processes

CONTEXT DIAGRAM of Hoosier Burger's food ordering system





LEVEL-0 DFD of Hoosier Burger's food ordering system

Data Flow Diagramming Rules (1)

Basic rules that apply to all DFDs

- Inputs to a process are always different than its outputs – purpose of a process is to transform inputs to outputs
- Objects on a DFD always have a **unique name**
  - In order to keep the diagram uncluttered, you can repeat data stores and sources/sinks on a diagram

Process:

A. No process can have only outputs (we can't make data from nothing). Having only outputs means it must be a source.

B. No process can have only inputs. Having only inputs means it must be a sink.

C. A process has a **verb** phrase label

Data Flow Diagramming Rules (2)

Data store:

D. Data must be moved by a process and cannot move directly from one data store to another data store

E. Data cannot move directly from an outside source to a data store. Data must be moved by a process that receives data from the source and places data into data store.

F. Data cannot move directly to an outside sink from a data store. Data must be moved by a process.

G. A data store has a **noun** phrase label

Source/Sink:

H. Data cannot move directly from source to sink and has to be moved by a process else data flow is not shown on the DFD.

I. A source/sink has a **noun** phrase label

Data Flow Diagramming Rules (3)

Data flow:

J. A data flow has only one direction of flow between symbols. It may flow in both directions between a process and a data store usually indicated by two separate arrows as this happens at separate times

K. A **fork** in a data flow means that exactly the same data goes from a common location two or more different processes, data stores, or sources/sinks.

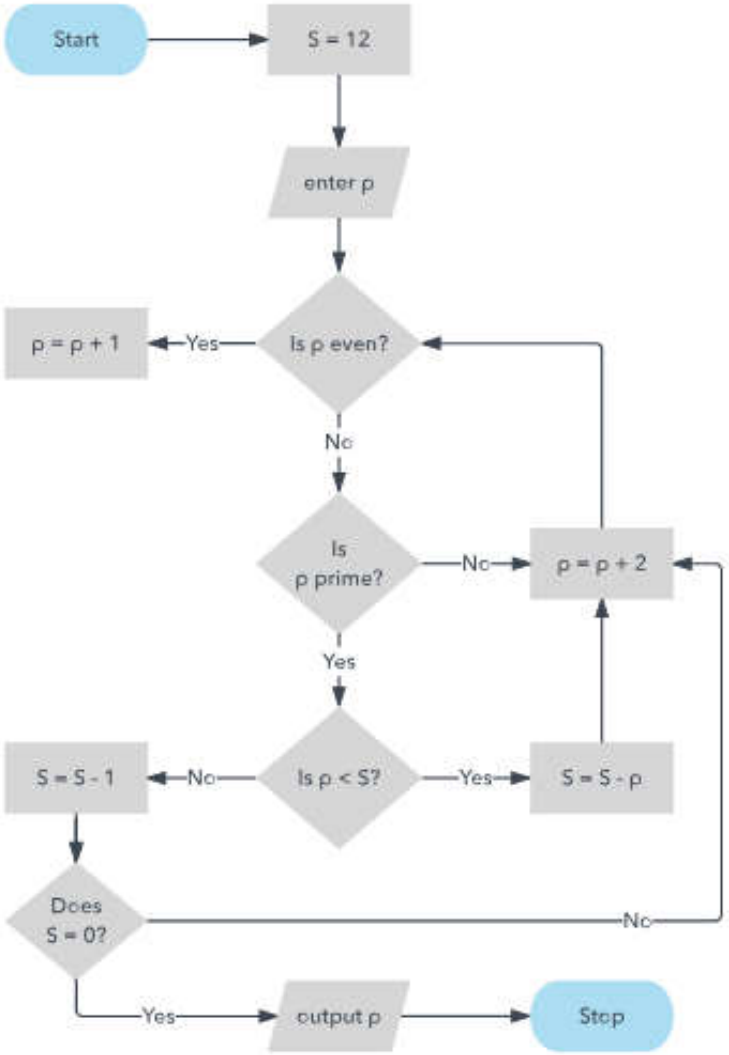
L. A **join** in a data flow means that exactly the same data comes from any two or more different processes, data stores, or sources/sinks to a common location

M. A data flow cannot go directly back to the same process it leaves.

N. A data flow to a data store means **update** (delete or change)

O. A data flow from a data store means **retrieve** or **use**.

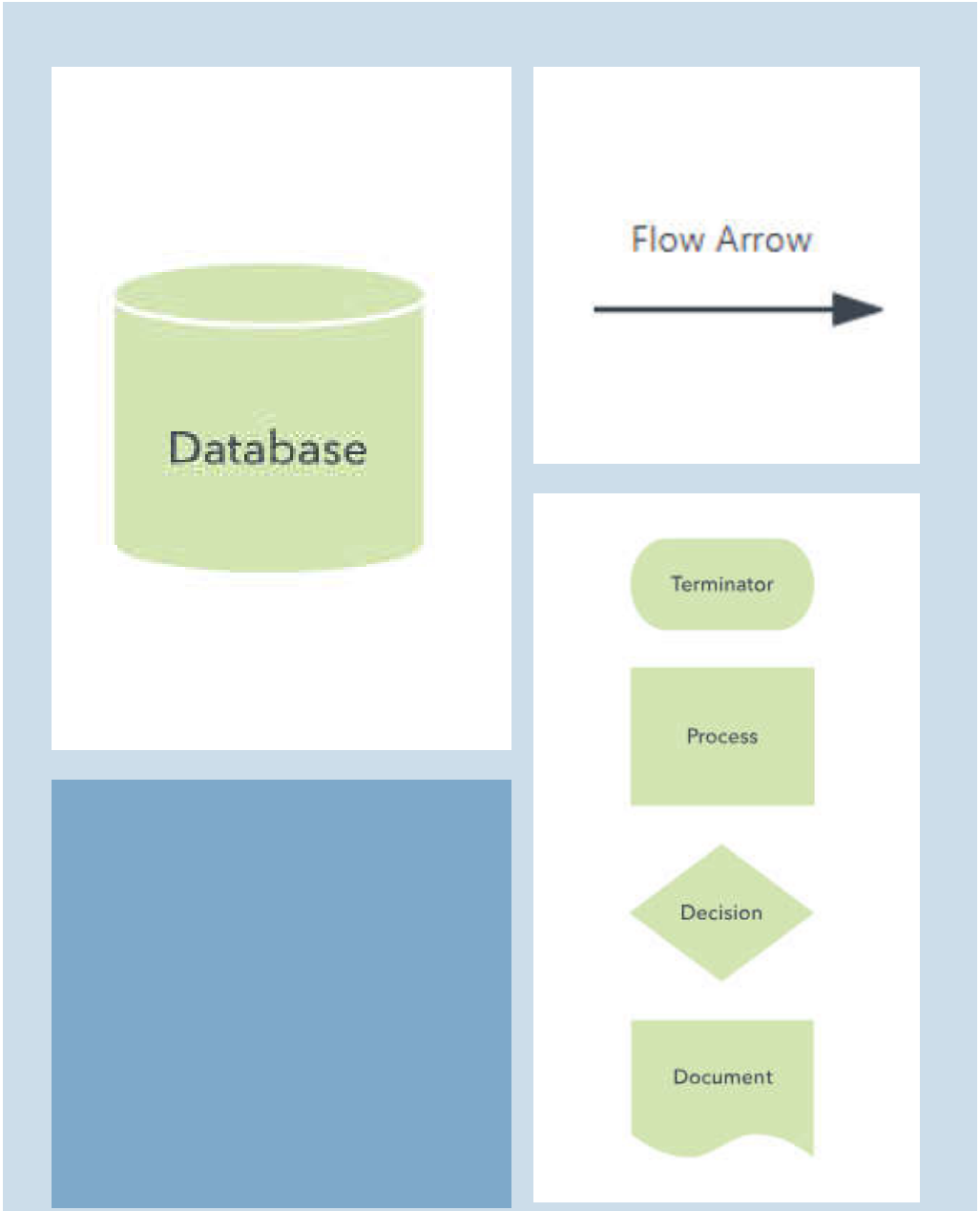
P. A data flow has a **noun** phrase label.



Flowchart Diagrams

- A flowchart is a diagram that depicts a process, system or computer algorithm.
- They are widely used in many fields to document, study, plan, improve and communicate often complex processes in clear, easy-to-understand diagrams.

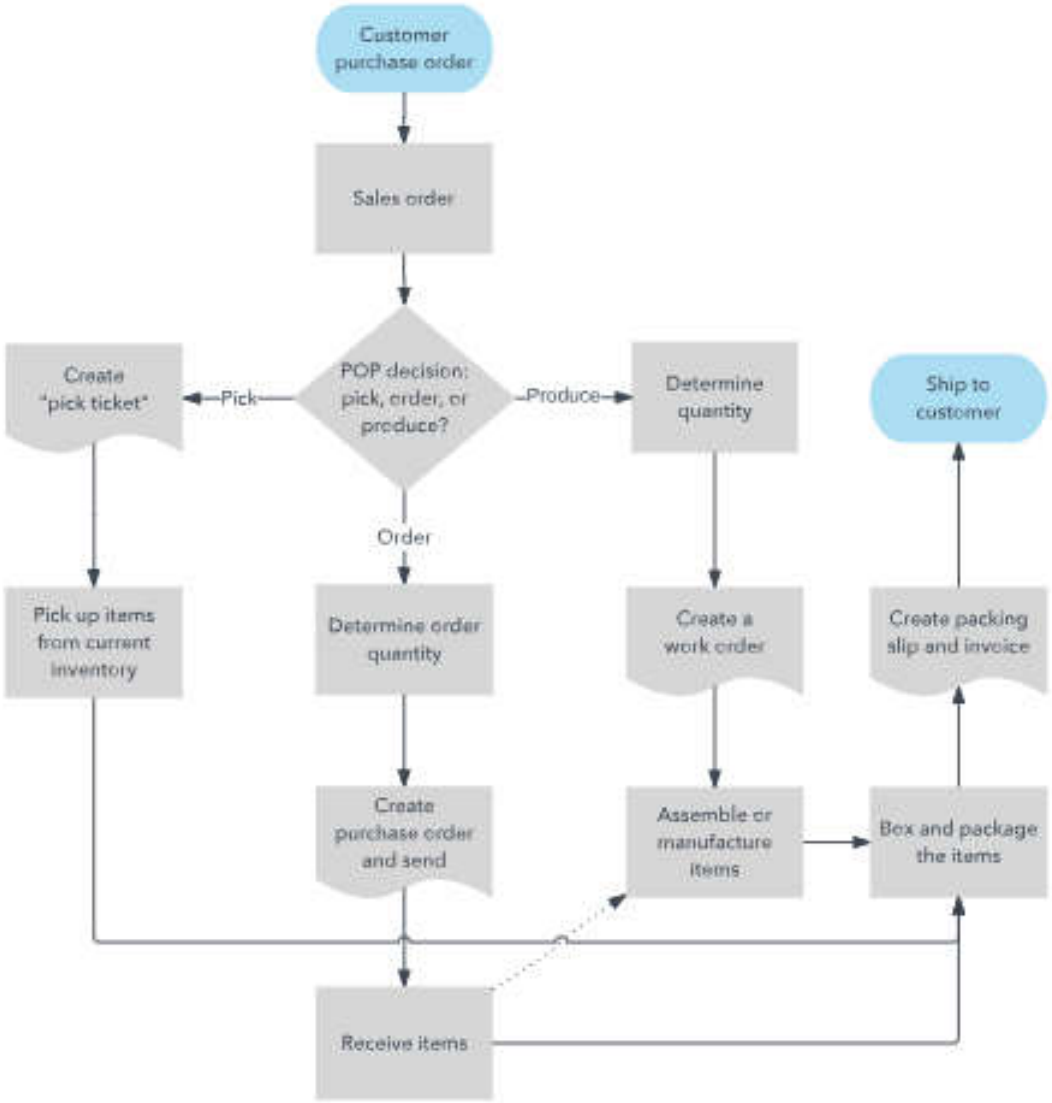
https://www.lucidchart.com/



Flowchart Diagrams symbols

https://www.lucidchart.com/

Flowchart Diagrams example



https://www.lucidchart.com/

Conceptual Data Modeling

- Conceptual data model is a representation of organizational data
- Purpose is to show as many **rules** about the *meaning* and *interrelationships* among data as are possible
- Entity-Relationship (E-R) diagrams are commonly used to show how *data are organized*
- Main goal of conceptual data modeling is to create accurate E-R diagrams
- Methods such as interviewing, questionnaires are used to collect information
- Primary deliverable is the **entity-relationship diagram**

Entity-Relationship Model

- Technique for carrying out the conceptual and logical design of the system
- A widely accepted data modelling approach
- 3 basic notions:
  1. Entities
  2. Attributes
  3. Relationships

What is an entity?

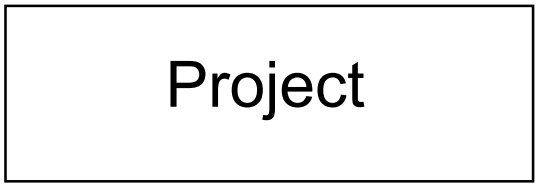
An *entity* is an object that can be identified in the users' work environment and that users want to track.

Entities

An entity is a thing or object in the real world (within the application context)

An entity has a set of properties which uniquely identify it.

An entity is represented as a rectangle in an ER diagram



What is an Attribute?

An *attribute* describes a characteristic of an entity

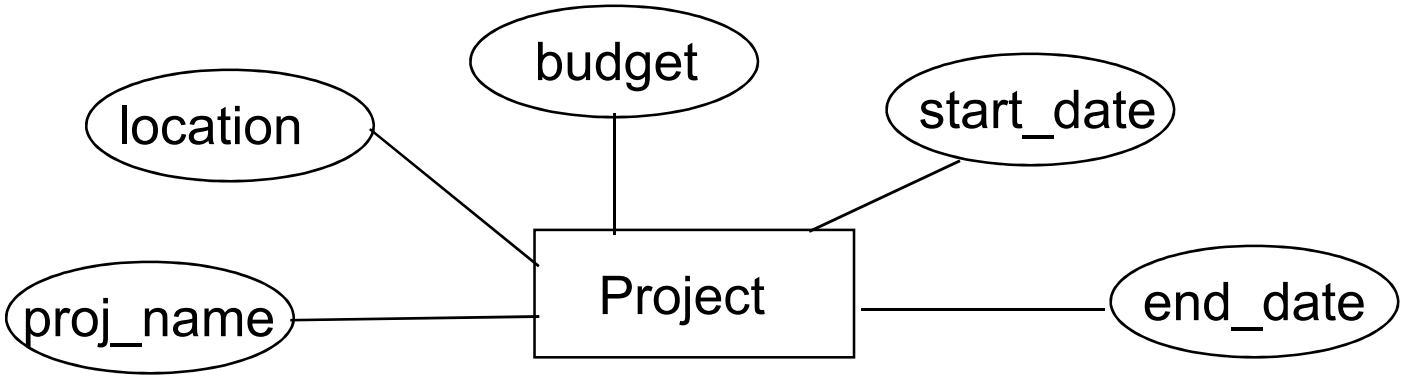
For example:

- An entity: Employee
- Has attributes:
  - Employee_Name
  - Extension
  - Date_Of_Hire

Attributes

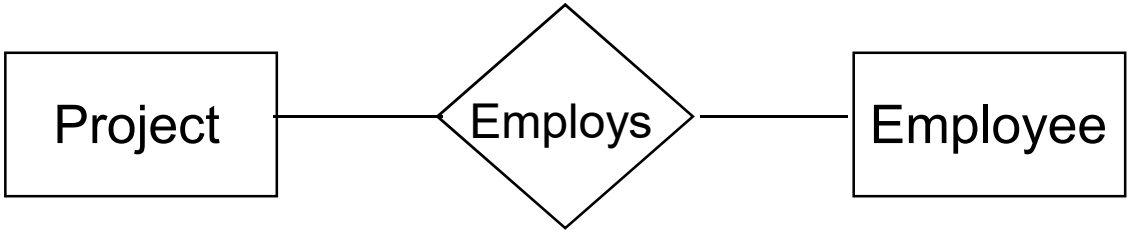
Example: *Project = (proj_name, location, budget, start_date, end_date)*

Represented as ellipses in an ER diagram



What are Relationships?

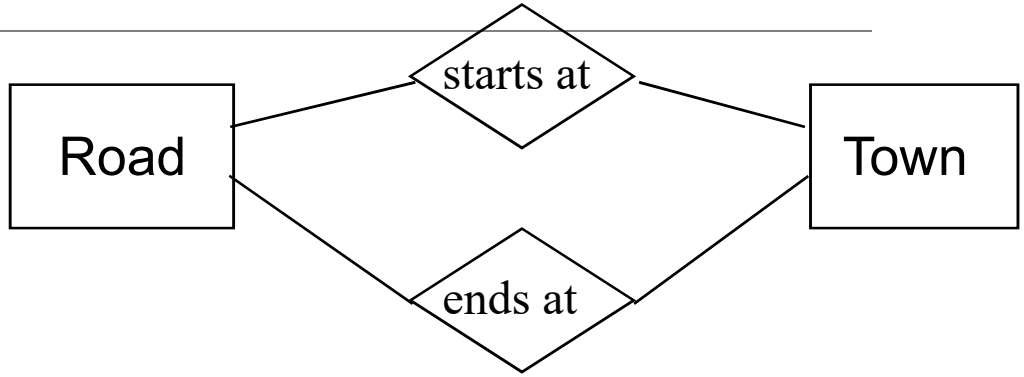
Relationships are associations between entities which express some real world relationship



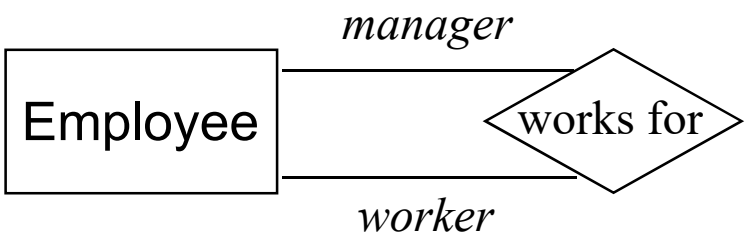
- *Project* and *Employee* **participate** in the employs relationship
- The function that an entity plays in a relationship is called that entity's **role**

Relationship Sets

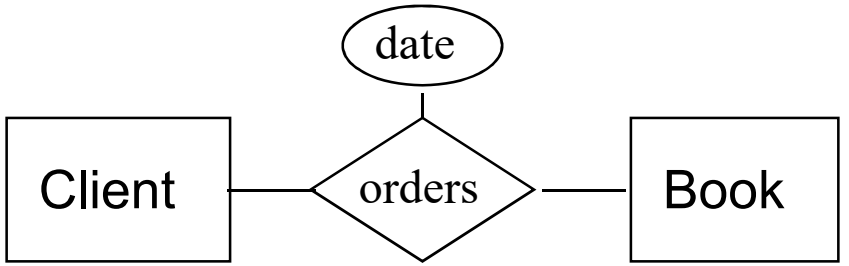
- There can be more than one relationship between entities.



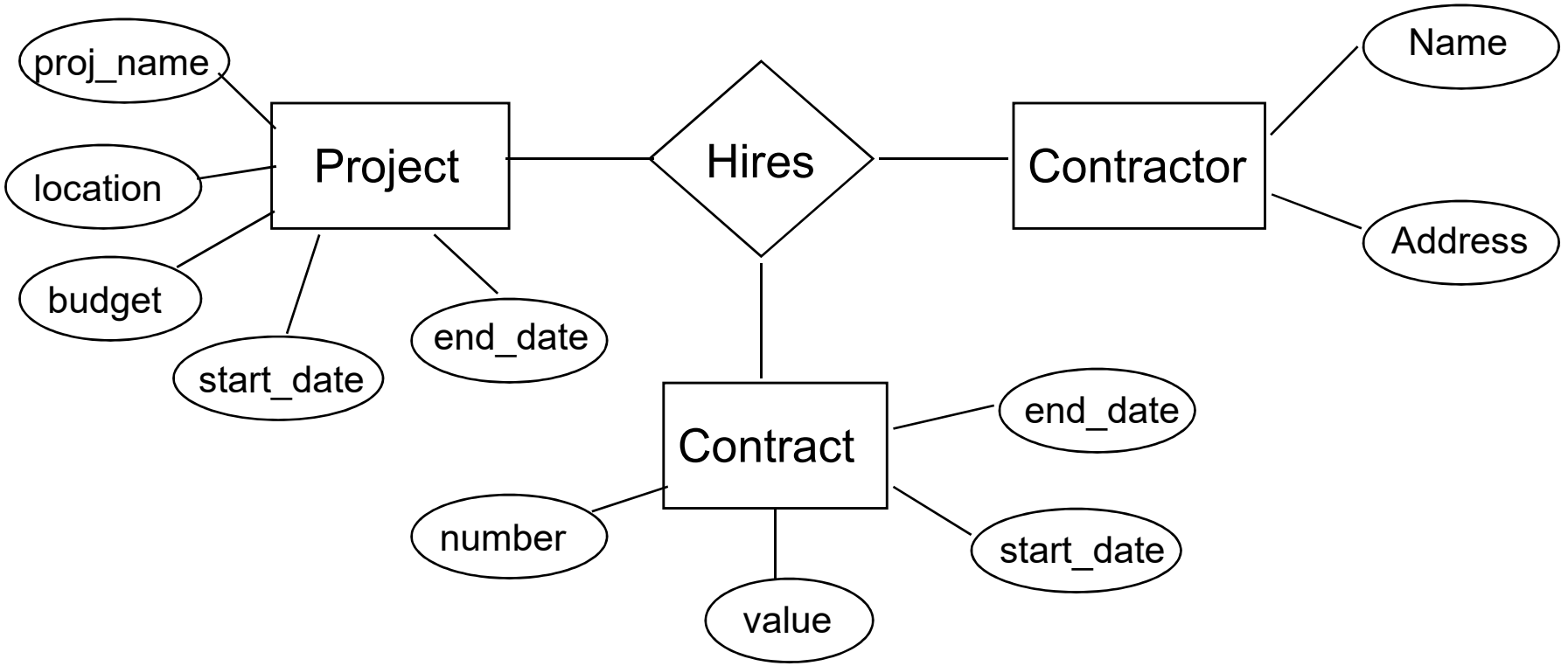
- There can be recursive relationships that can indicate roles for clarity.



- A relationship can also have descriptive attributes.



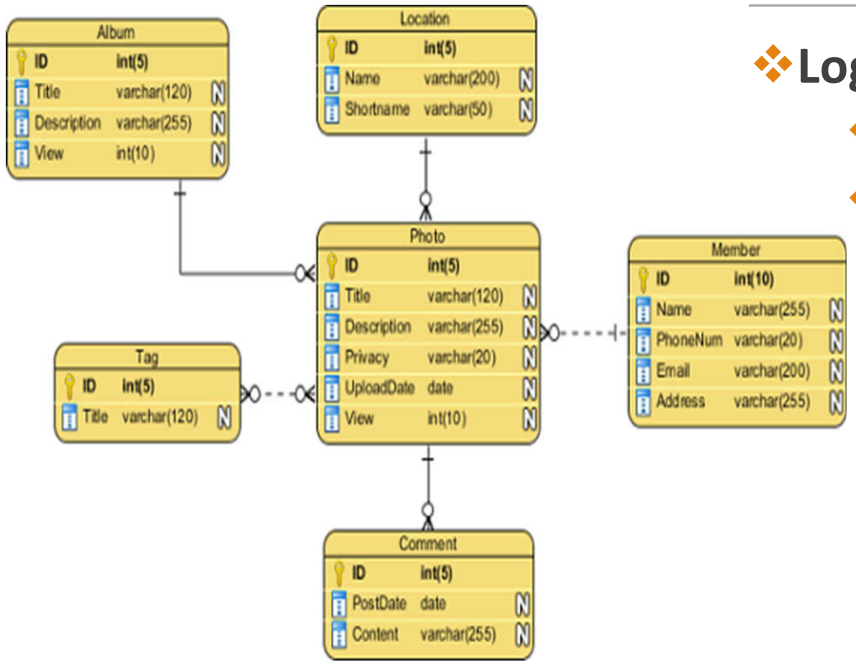
ER example


proj_name
location
budget
start_date
Project
end_date
Hires
Contractor
Name
Address
Contract
number
value
end_date
start_date

Database Design

- Purpose of Database Design
  1. ***Structure*** the data in ***stable structures***, called normalized tables
     - Not likely to change over time
     - Minimal redundancy
  2. ***Develop*** a logical database design that ***reflects*** actual ***data requirements*** that exists in ***forms*** and ***reports***
  3. ***Develop*** a ***logical*** database design from which a ***physical*** database design can be developed
  4. ***Translate*** a relational database ***model*** into a ***technical file*** and database design that balances several performance factors
  5. ***Choose*** data ***storage technologies*** that will efficiently, accurately and securely process database activities

Database Design (2)



- **Logical Design**
  - Based upon the ***conceptual data model***
  - Four key steps
    1. ***Develop*** a ***logical data model*** for each known user interface for the application using normalization principles
    2. ***Combine*** normalized ***data requirements*** from all user interfaces into one consolidated ***logical database model***
    3. ***Translate*** the conceptual ***E-R data model*** for the application ***into*** normalized ***data requirements***
    4. ***Compare*** the consolidated ***logical database design*** with the ***translated E-R model*** and ***produce*** one ***final logical database model for the application***

Database Design (3)

- **Physical Design**
  - Based upon results of ***logical database design***
  - Key decisions
    1. ***Choosing storage format*** for each attribute from the logical database model (txt, CSV, TSV, PSV, JSON)
    2. ***Grouping attributes*** from the logical database model into ***physical records***
    3. ***Arranging related records*** in secondary memory (***hard disks*** and magnetic tapes) so that records can be stored, retrieved and updated rapidly
    4. ***Selecting media*** and ***structures*** for ***storing data*** to make access more efficient (***DBMS***)

End

Sources:
1. Whitten, Jeffrey L., Lonnie D. Bentley, and Kevin C. Dittman. Systems Analysis and Design Methods 5e. McGraw-Hill Higher Education, 2000.
2. George, Joey F., and Joseph S. Valacich. Modern systems analysis and design. Prentice Hall, 2002.
3. https://www.lucidchart.com/