Chapter -1

What are Integrated Enterprise Information Systems?

Enterprise; A business, an industrious effort, especially one directed toward making money
Information System; A set of interconnected channels for communicating knowledge of specific events or situations
Integrated ; Joined together, united, made into a whole by having brought all parts together

Aren't all enterprise systems integrated? No! why not? (هل كُل المؤسسات انظمتها مدموجة مع بعض)

* Enterprise **"stove pipes" or "silos"** – As enterprises grow, they typically become divided based on functional areas (Department)
* Each functional area typically has its own system
* Even within functional areas, enterprises often develop different systems for different information needs

If existing systems lack functionality, additional systems are built to satisfy new needs

Common Integration Attempts

1) **Integrate the end results**
   – Let each functional area have its own system and require them to submit end results in a standardized format that can be merged with results from other areas
2) **Integrate similar types of systems**
   – All financial areas use same system
   – All manufacturing areas use same system
   – All areas associated with human resources use same system – Etc.
3) **Enterprise Systems**

The essence of what you do does not change; you simply changed the resources used to perform work.
   – May be created from scratch
   – May be based on packaged software (e.g. OracleApps, PeopleSoft, SAP)

Breaking Down Stovepipes by Re-engineering Business Processes

Reengineering Work: Don't Automate, Obliterate
– "It is time to stop paving the cowpaths. Instead of embedding outdated processes in silicon and software, we should obliterate them and start over use the power of modern information technology to radically redesign our business processes in order to achieve dramatic improvements in their performance." (p. 104)

What is the deference between Reengineering and Automation ?

Reengineering: use the power of modern information technology to radically to The redesign of business processes or systems to achieve a dramatic improvement in enterprise performance.
Automation: implies replicating or replacing traditional processes using computer hardware and software.

Summary Chapter 1 :

- Systems in enterprises are not integrated very well, for a variety of reasons
- There are many obstacles to overcome in integrating existing systems
- An integrated database is the central component of an integrated system, and thus should be our focal point
- An integrated database must support all functional areas within an enterprise
- To design, use, and audit integrated enterprise systems, we must understand

Businesses (all functional areas)
Representation (in general)
Conceptual modeling
Information Retrieval tools
Audit objectives and techniques

Chapter -2

What is a model ?

A model is a simplification of something in reality

- Created for a specific purpose
- Hides details that are not needed for that purpose
- Examples: model car, architect's models – paper-based and 3-dimensional

Why do we build models of enterprise systems?

We build models so that we can better understand the system we are developing.
Most enterprise systems are too large and complex for the average person to comprehend in entirety.

Representation in Information Systems

In modeling enterprise systems our goal is to create representations of the "reality" of the business in a form that is computer-readable
We start by making a paper-based model with a set of symbols that have understood meanings and for which there are agreed upon procedures for translation into computer-readable form
We then translate the paper-based representation into a computerized information system

- The information system is itself a representation of the business reality
- The paper-based model is also a representation of the business reality

Principles of Modeling

When creating or evaluating models, how do we determine what makes "good" models?
This is important because if a model provides an intermediate step toward a solution or developed tool, choices in modeling partially determine the solution or tool's effectiveness.
Good models resemble their underlying reality as completely as possible
Good models can be expressed at different levels of precision
They can be broken down into smaller pieces for closer examination of some features and aggregated for holistic views

- The first step in database design is to create a list of entities and relationships between them.
- Even when we don't have knowledge about something in reality, if we have knowledge (either first-hand or second-hand) about something similar in reality (and we know it is similar) we can apply our knowledge of the familiar object or situation to the unfamiliar object or situation This is **"pattern-based thinking"**

Object Patterns

In conceptual modeling, an object pattern is called a "stereotypical constellation of entities"
In other words, a group of entities and relationships between them that we expect to exist in the underlying reality
At the business process level, REA is such a pattern, specifically created to represent enterprises and to serve as a foundation for integrated enterprise information systems

Scripts and the REA Ontology

The business-entrepreneur script is also called a Value Chain, which we will discuss more later.
The value chain is a sequence (chain) of scenes
Each scene is a business process **(transaction cycle)**
Each scene is represented by a pattern **(REA)**
The REA ontology is a combination of script
patterns and object patterns that together enable us to model enterprises and to understand and work with existing enterprises models.

What is an "ontology"?

An attempt to define what things exist in the world in general; a branch of metaphysics dealing with the nature of being

What is an "enterprise ontology"?

An attempt to define what kinds of things in enterprises need to be represented

Why do we need ontologies?

Ontologies improve communication, sharing, and reuse of information

For current information systems and e-business, these three concepts are very important!

REA Ontology Levels (مهم ارجعوا لرسمة كُل مستوى لفهم الفكرة)

1. Value System Level (object-based pattern)
2. Value Chain Level (script-based pattern)
3. Business Process Level (object-based pattern)
4. Task Level (script-based pattern)

Value System Level (object-based pattern)

– Examines enterprise in context of its external business partners

– The combination of value systems of business partners forms a supply chain

Value Chain Level (script-based pattern)

– Connects business processes of an enterprise via the resource flows between the processes

Business Process Level (object-based pattern)

- A pattern to which the reality of most (perhaps all) enterprises conform
- The key is mapping the objects in the enterprise to the pattern in order to generate the model from which a database is designed

Business Process Level

Entities Resources - Economic Events - Agents (internal and external)

Relationships

Stockflow (relationships between resources and events – increase or decrease)

Duality (relationships between increment and decrement economic events)

Control (relationships between events and the agents that participate in them)

Task Level (script-based pattern)

- Many different possible scripts exist
- REA does not dictate specific tasks to be performed in achieving an enterprise's business processes

Task Level

- May be depicted in various formats such as a fishbone diagram (with tasks listed in an ordered
- sequence), a system flowchart, a data flow diagram, a process model (with swim lanes), etc.
- No pattern is available, tasks are enterprise specific

Summary Chapter 2 :

- Modeling is a useful tool for minimizing complexity and enabling us to develop enterprise wide system solutions
- Good models use symbols that represent reality as closely as possible
- Object Patterns are stereotypical constellations of things and relationships between them
- Script Patterns are stereotypical sequences of events, and can be thought of in terms of scenes, actors, props, and roles
- A combination of object and script patterns can be used to model enterprise systems
- the REA Enterprise Ontology provides such an approach, modeling enterprises at the value system, value chain, business process, and task levels.

Chapter -3

Porter's Value Chain

Primary value activities

– **Inbound logistics** - activities associated with receiving, storing, and disseminating inputs to the products or services
– **Operations-** activities associated with transforming inputs into the final products or services
– **Outbound logistics** - activities associated with collecting, storing, and physically distributing the products or services
– **Marketing and sales** - activities associated with providing a means by which customers can buy produce and the means for inducing them to buy
– **Service** - activities associated with providing service to enhance or maintain the value of the products or services

Support value activities

– **Procurement -** the function of purchasing inputs to firm's value chain
– **Technology Development** - the know-how, procedures, or technology embedded in processes that are intended to improve the product, services, and/or process
– **Human Resource Management** - activities involved in recruiting, hiring, training, developing, and compensating all types of personnel
– **Firm Infrastructure** - activities that support the entire value chain (e.g. general management, planning, finance, accounting, legal, government affairs, quality management, etc.)

Importance of Studying Value System and Value Chain Levels in REA

• Understanding an enterprise's activities at the value system and value chain levels in the REA ontology
– Helps keep perspective (gives the ability to "see the forest" without getting mired in the detail of the trees)
– Provides the structure to guide lower levels of analysis
– Requires consideration of the enterprise's mission and strategy, which should ensure that business processes and activities are constructed in a manner consistent with the mission and strategy

Value System Modeling طريقة وخطوات رسمها من سلايد ٩ الى سلايد ١٣

• Identify an enterprise's resource inflows and outflows
– Focusing on the cash flows and then identifying the "reasons" for those cash flows is a good way to start
– Although non-cash resource flows are rare, they are still important to consider
• Identify the external business partners to which and from which the resources flow

Value Chain Level طريقة وخطوات رسمها من سلايد ١٧ الى سلايد ٢٣

Duality relationships consist of paired increment economic events and decrement economic events
– Increment economic events increase resources (stock in-flows)
– Decrement economic events decrease resources (stock out-flows)
"Duality relationships are the glue that binds a firm's separate economic events together into rational economic processes, while stock-flow relationships weave these processes together into an enterprise value chain." -- Geerts & McCarthy 1997

Each economic event in each cycle in the value chain corresponds to a resource in or out flow.

– If there is a resource flowing into the cycle, there must be an event in the cycle that uses that resource
– If there is a resource flowing out of the cycle, there must be an event in the cycle that provides that resource

SUMMARY

• Modeling enterprise systems at the value system and value chain levels provides a valuable overview of the strategy and stock flows of the enterprise
• Keep in mind that resource flows at the value system and value chain levels need not be physical; they indicate a shift in responsibility or ownership from one agent or transaction cycle to another agent or transaction cycle

Chapter -4

Conceptual Modeling Constructs

Entities Sets of real world objects – things that have a separate existence, either physical or conceptual - **Note SETS**
Relationships Sets of associations between entities - Again, note SETS Degrees of relationships
Attributes Characteristics or elementary properties of entities and/or relationships
– Primary key attribute uniquely and universally identifies each instance of an entity or relationship set

Type of Attributes

1) Simple versus composite attributes
2) Derivable attributes

Type of Derivable attributes

- **Static –** will not change if new data is entered into system
- **Volatile** – will change if new data is entered into system

Abstraction Mechanisms

1) **Typification** **Specification of a relationship between a set of objects and a category** to which the objects could be assigned on the basis of shared characteristics
– Allows storage of characteristics that apply at the category level of detail
– E.g. for student category, characteristics such as tuition rate per hour and maximum number of credits allowed per semester

2) **Generalization** **Specification of subclass-superclass relationships**, I.e., "Is-A" relationships
Subclass entities include contain more specific instances of superclass entities
– E.g. Instrument as a supertype, Woodwind Instrument, Brass Instrument, and Stringed Instrument as subtypes
– Are used when similar entity sets have enough shared attributes to warrant grouping them together, but they also have non-shared attributes that need to be stored
– The non-shared attribute values are null if the attribute is assigned to the superclass entity; generalization avoids this problem.

Conceptual Modeling Notations قراءة فقط

– Multiple notations are used to communicate the same conceptual modeling concepts
– Just as people across the globe speak multiple languages to communicate the same general concepts – that's life!

– People who are multi-lingual in life have an advantage over those who only speak one language
– People who learn multiple conceptual modeling notations have an advantage over those who only learn one

Entity-Relationship (ER) Grammar notation

- **Entity:** ENTITY NAME
  Attributes: ATTRIBUTE NAMES
  Identifiers: PRIMARY KEY ATTRIBUTE
- **Relationship**: RELATIONSHIP NAME
  Connected entities: (min,max) ENTITY1
  (min,max) ENTITY2

  Attributes: ATTRIBUTE NAMES (if any)
- **Objects are often stored in alphabetical order according to entity and relationship names, with all entities first, then all relationships**

– Notation is sometimes called "BNF Grammar" format (Backus-Naur Form), after its creators, Backus and Naur.

Entity-Relationship (ER) Diagram Notation

- The **ER Diagram is** an alternative means for representing the same constructs as in the **ER Grammar**
- Most conceptual models are created with some version of the ER Diagram

Several variations exist, and notation (especially for cardinalities) differs between the variations

Conceptual Modeling Notation: Entities

- ER Grammars **list entities** (usually alphabetically) by name, each name is preceded by the word Entity and a colon, i.e. Entity: name of entity
- ER Diagrams portray entities as **rectangles** (each rectangle is labeled with the entity name)

Examples: Customer, Sale Slides 13

Conceptual Modeling Notation: Relationships

- ER Grammars **list relationships** (usually alphabetically) by name, with the word Relationship and a colon, i.e., Relationship: name of relationship, followed by the words Connected Entities: and the names of the related entities
- ER Diagrams portray relationships as **diamonds** attached by straight lines to the entities for which the relationship depicts an association.

Example: Sale IS-MADE-TO Customer Slides 15

Conceptual Modeling Notation: Attributes

- ER Grammars **list attributes by name** within entity and/or relationship descriptions. The name of the **primary key** attribute(s) is also included after the word Identifiers and a colon.
- ER Diagrams depict attributes with small circles attached by short lines to the applicable entities and/or relationships. The circles for **primary key** attributes are **darkened.**

Example: Customer Characteristics Slides 17

Conceptual Modeling Notation: Participation Cardinalities

- ER Grammars incorporate participation cardinalities into the relationship descriptions. The participation cardinalities that apply to each entity in the relationship appear next to each entity name in the **Connected Entities: statement.**
- ER Diagrams include participation cardinalities as labels on the lines between the entities and the relationships for which the participation cardinalities apply.

Example (continued): Employee/Department Cardinalities Slides 20

Comparison of Cardinality Notations جدول سلايد ٢١ مروا عليه
Steps to Create a REA Business Process Level Model الخطوات من سلايد ٢٩ الى ٣٤
REA business process level with extensions

- **Entities**

Resources and Resource Types
Events

- **Instigation Event**

An event that initiates activities in the business process; may be internally instigated (e.g. a marketing event) or externally instigated (call from supplier's salesperson)

- **Mutual Commitment Event**

An event in which commitments are made by the enterprise and one of its external business partners for a future economic exchange

- **Economic Exchange Event**

An event in which a resource is either given up or taken
Increment economic event results in resource inflow
Decrement economic event results in resource outflow

- **Entities**

Agents

- Internal agents act on behalf of the enterprise
- External agents are external business partners

- **Relationships**

Event-Event relationships

- Duality (link increment and decrement economic events)
- Reciprocal (link increment and decrement commitment events)

Is the commitment equivalent of duality

- Fulfillment (link commitment and economic events)

Event-Resource relationships

- Stock flow (link economic events and resources or resource types)
- Reservation (link commitment events and resources or resource types)

REA business process pattern

- **Relationships**

Event-Agent relationships

- Participation (link events and the agents that participate in the events)

Agent-Agent relationships

- Assignment (link internal agent to external agent)

Use only when relationship between internal agent and external agent exists independently of their mutual participation in an event

- Responsibility (link internal agent to internal agent)

Use when one internal agent is responsible for another, independent of their mutual participation in an event

- **Relationships**

Resource-Agent relationships

- Custody (link resource and internal agent)

Use when an internal agent's responsibility for a resource needs to be tracked independently of any event

Resource-Resource relationships

- Linkage (link two resources)

Use to identify resource made up of another resource

Typification

- Each resource, event, or agent can be related to a resource type, event type, or agent type

Generalization

- Each resource, event, agent, and commitment can also participate in a relationship with a sub-class or super-class via a generalization relationship.

Summary

- The primary purpose of creating a conceptual REA model at the business process level is to design the database architecture for an enterprise system using a specified object pattern
- The database architecture components included in REA modeling are entities, attributes, relationships, and cardinalities
- Three main entity types in the REA ontology are Resources, Events, and Agents
- Three main relationship types in the REA ontology are duality, stockflow, and participation; additional types are assignment, custody, fulfillment, linkage, reciprocal, reservation, and responsibility. The REA ontology also allows for generalization and typification relationships

Chapter -5

Introduction

- Tasks are the individual steps involved in accomplishing events in an enterprise
- Events are tasks; however, not all tasks should be represented as events
- Tasks that are activities that may be changed or eliminated without substantially changing the nature of the enterprise should not serve as foundational elements in an enterprise information system database
- **The purpose of task level models is NOT to design a database; rather, it is to document the flow of data through an enterprise**
- No pattern has yet been identified at the task level

System Flowcharts

- Graphically document information systems
- Summarize pages of narrative in diagrammatic format
- Focus on the physical aspects of information flows

System Flowcharts: Basic Elements

- System flowcharts may be drawn **freehand** or with **a plastic flowchart** template/stencil tool.
- System flowcharts may be prepared using software created for that purpose.
  – E.g. **SmartDraw, Visio**, ABC Flowcharter
  – Microsoft Word and Powerpoint (but they are not as automated)
- System flowcharts combine three simple graphical elements to represent various types of physical information flows and processes

Flow Lines

- Flow lines are used to connect the symbols on the document flow chart.
- A solid line indicates the flow of a document or object
- A dotted or dashed symbol indicates a flow of information rather than the physical document
- Some flowcharts also show communication flows such as by telephone modem or satellite
- Arrows are used when the documents or information flow is not left-to-right or top-to-bottom

Areas of Responsibility

- Areas of responsibility are displayed to enable the flowchart reader to clearly identify changes in responsibility as the documents flow through the system.
- They are represented on flowcharts by segmenting and labeling columns.
- Areas of responsibility may be departments, sections within a department, or individual employees within a department.
- Judgment must be used in choosing the level of subdivision that one column should represent.

Flowchart Preparation Conventions

- Left-to-right, Top-to-bottom
- All documents must have an origin and termination (“cradle to grave documentation)
  - **each copy of the document must flow to**
    - a permanent file symbols
    - a symbol denoting an exit from the system, or
    - an off-page connector
    - a document destruction symbol (small black box)
    - Corner of originating symbol may be darkened to indicate its introduction to the system

• Keep flowcharts uncluttered
  – Place responsibility areas with frequent interchange in adjacent columns
  – Enter narrative only within symbols
  – Avoid unnecessary explanation with narrative
• Make sure progress of each document is clear.
  – **Diagram a document**
    • before and after each process
    • entering or leaving a file
    • entering or leaving a page or area of responsibility
    • Ensure flowchart is complete

System Flowchart Summary

The Good

- Flowcharts are relatively easy for information customers and managers to understand
- Flowcharts help auditors understand business and systems controls

The Not-So-Good

- Flowcharts are tied to physical information flows and system characteristics that hide procedural essence of
- Flowcharts may be artifactual and tied to outdated information technology

File Types

Master files Contain the balance or status of entities
• E.g. vendors, credit customers, inventory, assets, employees
Transaction files Contain activity data. E.g. orders, sales, payments
History or archive files Contain inactive past or historical data
Reference files Contain information needed for reference purposes
e.g., rates, prices, zip codes, chart of accounts
Suspense files Contain items awaiting action, errors, missing information

Storage and Access of Data

Sequential Storage and Sequential Access

  – Records are stored in order
  – To access a record, the access device must read through all records that are stored previous to the desired record
  – Tape cartridges and open reel tapes require sequential storage and sequential access of data

Random Storage and Direct Access

  – Records are stored in any order
  – Any record can be retrieved directly regardless of physical position on the media; the access device need not read all the records prior to the desired record
  – Computer hard disks, floppy disks, zip disks, CD-ROMs, and DVD-ROMs allow random storage and direct access of data

Media

1) **Paper**
  – Most common form of media
  – Most easily used by people
  – Doesn't depend on electricity to access
  – **Disadvantages**
    • Bulk (for storage)
    • Lack of search and automated processing capability

- Susceptibility to destruction

– Sequential storage (may be indexed for indexed-sequential access)
– May update on same physical input media
  - Add information to existing document

2) **Magnetic tape**

– Audiocassette tapes, VHS videotapes, and 8mm video camera cassettes use magnetic tape
– Sequential storage and sequential access
– Sorting is important for processing (transaction file must be sorted to match the order of
– the master file)
– Separate physical media must be used for input and output in an update process
  - Old master file, new master file
– Easy backups
– Dependent on electricity and on hardware
  - Cannot be read or processed directly by a person

3) Digital (Disk) media

– Computer hard disks, floppy disks, zip disks, CDs, DVDs, and memory cards
– Random storage
  - Information may be stored anywhere on the media; may be broken up (fragmented) and stored in multiple places
  - "Defragging" a hard drive is the process of sorting the data to reconnect all the fragments for more efficient processing
– Direct access
  - From index, hardware can jump directly to the desired information and proceed with processing
– Same physical media may be used for input and output in an update process (unless disk is full)
– Easy backups
– Dependent on electricity and on hardware
  - Cannot be read or processed directly by a person

Processing Methods

– **Batch:** accumulates transaction data for a period of time. Then all of the transactions in the transaction file are posted to the master file in one processing run. (Tape processing is always batch)
– **Online:** means the computer-input device is connected to the CPU so that master files are updated as transaction data are entered
– **Real-time:** denotes immediate response to an information user; transaction data are entered and processed to update the relevant master files and a response is provided to the person conducting the business event fast enough to affect the outcome of the event
– **Report-time:** the data used to generate the requested report is processed as the report is created

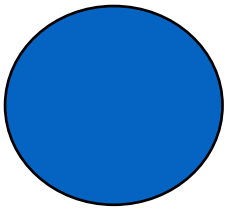
Data Flow Diagrams (DFD)

– DFD symbols are used for a variety of system analysis purposes, including graphically displaying the logical flows of data through a process.
– Unlike flowcharts which represent the physical components of an information system, DFDs can provide a more conceptual, nonphysical display of the movement of data through a system.
– DFDs disregard such things as organizational units, the computer on which the data are processed, and the media on which the data are stored.
– Movements of data across offices or departments within a particular system are not represented.

Data Flow Diagram Symbols

Process

- Circles are used to represent processes that take data inflows
- and transform them to information outflows
- Each circle contains two labels
  - a process numbers
  - a process name
- Alternate notations is rectangular box with rounded corners



(A)
Process

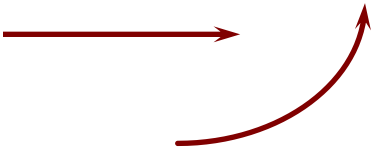
Data Sources and Sinks

- Rectangles (or squares) represent data (inflow) sources
- and (information outflow) sinks
- Rectangle is labeled with the name of the data source
- or sink/destination (e.g. Customer, Vendors, Government Agency).
- Sources and sinks are agents external to
- (i.e. outside the scope of) the system represented on the diagram
  - Delineate the boundaries of the system

(B)
Data inflow sources, information outflow destinations

Data Flow Lines

- Data flow lines display the route of data inflow and information outflow
- Lines can be straight or curved
- Data flows are generally labeled with the name of the data (e.g. a customer order, a bill, a financial analysis)
- Arrow indicates the direction of the data flow



(D)
Data flow lines

Data Stores

- Two parallel straight lines are used to display a store or collection of data
- Some people refer to data stores as data at rest
- A description of the data store contents is entered on the symbol
- Data stores are used anytime it is necessary to store the output from a process before sending it on to the next process
- Alternative notation uses a rectangular box that is open at one end

(C)
Data store

Constraints: General Rules

1) All processes should have unique names
   - If two data flow lines (or data stores) have the same label, they should both refer to the exact same data flow (or data store)
2) The inputs to a process should differ from the outputs of a process
3) Any single DFD should not have more than about seven processes
4) No process can have only outputs. (This would imply that the process is making information from nothing.) If an object has only outputs, then it must be a source.
5) No process can have only inputs (referred to as a "black hole")
   - If an object has only inputs, then it must be a sink
6) Process should be labeled with a verb phrase
7) Data must be moved by a process; data cannot move directly from one data store to another
8) Data must be moved by a process that receives data from the source and places the data in the data store; data cannot move directly from an outside source to a data store
9) Data must be moved by a process; cannot move directly to an outside sink from a data store

10) Data store should be labeled with a noun phrase
11) Data cannot move directly from a source to a sink. It must be moved by a process if the data are of any concern to the system. If data flows directly from a source to a sink (and does not involved processing) then it is outside the scope of the system and is not shown on the system data flow diagram DFD.
12) A source/sink has a noun phrase label.
13) A data flow has only one direction between symbols. It may flow in both directions between a process and a data store to show a read before an update. To effectively show a read before an update, draw two separate arrows because the two steps (reading and updating) occur at separate times.
14) A fork in a data flow means exactly the same data (e.g. different copies of an invoice) goes from a common location to two or more different processes, data stores, or sources/sinks
15) A join in a data flow means exactly the same data comes from two or more different processes, data stores, or sources/sinks, to a common location.
16) A data flow cannot go directly back to the same process it leaves. At least one other process must handle the data flow, produce some other data flow, and return the original data flow to the originating process.
17) A data flow to a data store means update (i.e., delete, add, or change).
18) A data flow from a data store means retrieve or use.
19) A data flow has a noun phrase label. More than one data flow noun phrase can appear on a single arrow as long as all of the flows on the same arrow move together as one package.

DFD Levels

- DFDs are divided into levels to keep their size and complexity manageable
- Context level shows least detail
- Each subsequent level (Level Zero, Level One, Level Two, etc.) subdivides one process on the previous diagram into more detail
- Balance must be maintained between levels
  - All inflows from and outflows to external sources/sinks must be the same from one level to the next

Context Level DFD

- The context diagram shows one process (representing the entire system) and the sources/sinks that represent the boundaries of the system.

Summary

- Task level modeling represents workflow activities within enterprise information systems
- Task level modeling is useful for representing the individual steps that make up events but which are subject to change and therefore should not serve as foundational elements in the enterprise database architecture
- System flowcharts and data flow diagrams are alternative means for representing task level models
  - Each has advantages and disadvantages
- To create task level models, one must understand physical media, file types, and processing methods

Chapter -6

Database Model Levels

1. A Conceptual model represents reality in an abstracted form that can be used in developing an information system in a wide variety of formats (e.g. relational, object-oriented, flat-file, etc.)
   a. It is hardware and software independent
   b. It is independent of any logical model type
2. A Logical model represents reality in the format required by a particular database model (e.g. relational or object-oriented)
   a. Is still hardware and software independent
   b. Depends on the chosen logical model type
3. A Physical model is created specifically for a particular database software package
   a. Is dependent on hardware, software, and on the chosen logical model type

Relational Database Model

- The relational model is a type of logical database model that was conceived by E.F.
- The relational model is based on set theory and predicate logic
- It is well formalized, so its behavior is predictable
- A relational database consists of tables (relations) that are linked together via the use of primary and foreign keys
- A FOREIGN KEY in a table is a primary key from a different table that has been posted into the table to create a link between the two tables

Relational database tables are made up of rows and columns

- Rows are called the table extension or tuples (The ordering of rows in a table does not matter)
- Columns are called the table intension or schema the ordering of columns in a table does not matter
  - ❖ All values in a column must conform to the same data format (e.g. date, text, currency, etc.)
- Each cell in a database table (a row-column intersection) can contain only one value no repeating groups are allowed.

Converting Conceptual to Relational

Step 1: Create a separate table to represent each entity in the conceptual model
– 1A: Each attribute of the entity becomes a column in the relational table
– 2A: Each instance (member) of the entity set will become a row in the relational table

Steps 2-4 (detailed in the next few slides) involve determining whether each relationship in the conceptual model should be represented as a separate table or as a posted foreign key
– Redundancy and Load are important determinants

• **Redundancy =** one fact in multiple places or multiple facts in one place
• **Load =** the percentage of non-null values in a column
– Participation Cardinalities communicate some of the information regarding redundancy and load

Relationship Conversion

• Maximum Cardinalities
– The general rule is to post into a “1” entity table. This avoids “repeating groups” redundancy
– You can NEVER post into an “N” entity. This causes “repeating groups” redundancy
• Minimum Cardinalities The general rule is to post into a “1” (mandatory) entity table. This avoids null values in the foreign key column
This rule should be violated in some circumstances (to be discussed soon)

• **Step 2:** Create a separate table to represent each many-to-many relationship in the conceptual model, I.e., for the following participation cardinality patterns

(0, N)-(0, N) (0,N)-(1,N) (1,N)-(0,N) (1,N)-(1,N)

– You must create a separate table to represent the relationship

- The primary keys of the related entity tables are posted into the relationship table to form it primary key. This kind of primary key is called a composite or concatenated primary key
- This avoids redundancy
- There are no exceptions to this rule!!!

– If you post a foreign key in either direction, redundancy will be a problem for many-tomany Relationships

Step 3: For participation cardinality pattern (1,1) -(1,1),
consider whether the two entities are conceptually separate or whether they should be combined

• If they should remain separate, then

- **3A: Post the primary key from one entity's table into the other entity's table as a foreign key**
- **3B: It doesn't matter which entity's primary key is posted into the other entity's table, but DO NOT post both**
  - **DO NOT make a separate table**
  - **Redundancy is automatically avoided and load is not an issue when you post a foreign key into either table in a (1,1)-(1,1) relationship**

Step 4: For remaining relationships that have (1,1) participation by one entity set, post the related entity's primary key into the (1,1) entity's table as a foreign key
I.e., for the following participation cardinality patterns

(0,N)-(1,1) (1,N)-(1,1) (1,1)-(0,N) (1,1)-(1,N) (0,1)-(1,1) (1,1)-(0,1)

- **Do NOT make a separate table**
- **Post a foreign key INTO the (1,1) entity's table from the other entity's table**
- **Redundancy is avoided and load is not an issue if you follow this instruction**
- **If you post the opposite direction, either redundancy [for N maximums] OR load [for 0 minimums] will be a problem**

Step 5: For remaining relationships that have (0,1) participation by one or both of the entities, consider load I.e., for the following participation cardinality patterns

(0,N)-(0,1) (1,N)-(0,1) (0,1)-(0,N) (0,1)-(1,N) (0,1)-(0,1)

- **The rule for maximum cards requires posting into a (0,1) or making a separate table; you CANNOT post into the (0,N) or (1,N)**
- **The rule for minimum cards says you really shouldn't post into the (0,1) because it will create null values that waste valuable space in the database**
- **However, if a separate table would waste more space, then it is better to follow the maximum rule and break the minimum rule**
- **5A: Post the related entity's primary key into the (0,1) entity's table as a foreign key for any relationships for which that results in a high load**
- **5B: Create a separate table for any relationships for which posting a foreign key results in low load**
- **Note: For (0,1) -(0,1), step 5A, post whichever direction results in highest load; if neither direction yields high load, then follow step 5B**

Example: Load Considerations

• Some cash disbursements (13/26) pay for purchases
– If we post Receiving Report# into Cash Disbursement, 13 out of 26 will be non-null
– This is a medium load
– Might be worth breaking minimum rule
– Consider other posting option
• Most purchases (14/18) result in cash disbursements
– If we post Check# into Purchase, 14 out of 18 will be non-null
– This is a high load
– Worth breaking the minimum rule
Conclusion: Make a separate table to represent the "allowance for" relationship

Relationship Attribute Placement

• If relationship becomes a separate table, then relationship attributes are placed in that table
• If relationship can be represented by a posted foreign key, relationship attribute is posted alongside the foreign key

Relational Database Design Summary

- The relational model is based on set theory and predicate logic and the resultant relations (tables) can be manipulated for information retrieval purposes if they are properly constructed
- To create well-behaved tables, follow the rules we discussed

– Conversion rules for cardinality patterns
– One Fact-One Place

- Think at the data (extensional) level!!
- When creating physical databases, use the conceptual and logical models to help you realize the important issues and potential pitfalls