Segment 5
Database Review

Dr. Abdul Kadar Muhammad Masum
Associate Professor of CSE
Department of Computer Science & Engineering
International Islamic University Chittagong
Email: akmmasum@yahoo.com

Data modeling

- Data modeling is often the first step in database design as the designers first create a conceptual model of how data items relate to each other.
- Data modeling involves a progression from conceptual model to logical model to physical schema.
- The Data Model typically evolves through the following three general stages:

A) conceptual data model

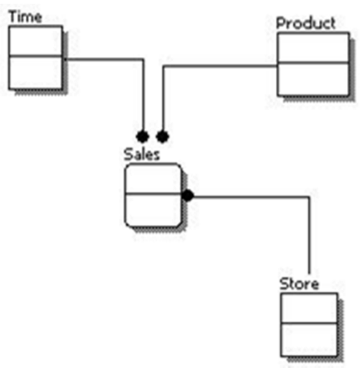
B) logical data model

C) physical data model

Conceptual Data Modeling

- A conceptual data model identifies the highest-level relationships between the different entities.
- Designed and developed to be independent of DBMS, data storage locations or technologies.
- Features of conceptual data model include:
  - Important entities and the relationships among them.
  - No attribute is specified.
  - No primary key is specified.

The figure below is an example of a conceptual data model.



From the figure above, we can see that the only information shown via the conceptual data model is the entities that describe the data and the relationships between those entities. No other information is shown through the conceptual data model.

Logical data model

A logical data model is a fully-attributed data model that is independent of DBMS, technology, data storage or organizational constraints. It typically describes data requirements from the business point of view.

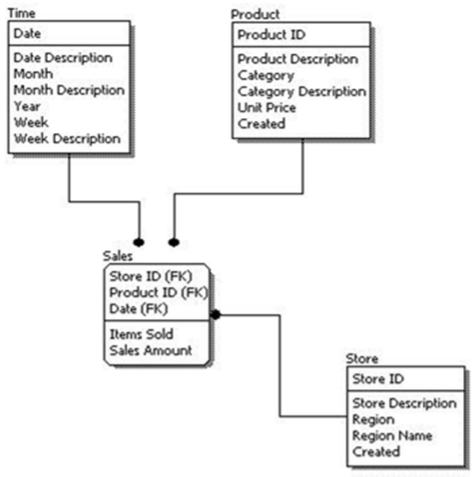
A logical data model describes the data in as much detail as possible. Features of a logical data model include:

- ✓ All entities and relationships among them.
- ✓ All attributes for each entity are specified.
- ✓ The primary key for each entity is specified.
- ✓ Foreign keys (keys identifying the relationship between different entities) are specified.
- ✓ Normalization occurs at this level.

Logical Model Design

The steps for designing the logical data model are as follows:

1. Specify primary keys for all entities.
2. Find the relationships between different entities.
3. Find all attributes for each entity.
4. Resolve many-to-many relationships.
5. Normalization.



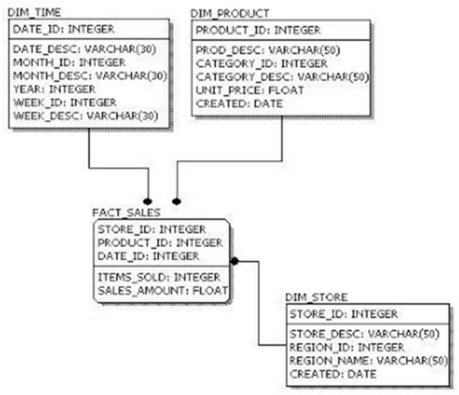
Physical data model

- ✓ A physical data model is a fully-attributed data model that is dependent upon a specific version of a data persistence technology.
- ✓ The target implementation technology may be a relational DBMS, an XML document, a NoSQL data storage component, a spreadsheet or any other data implementation option.
- ✓ Features of a physical data model include:
  - Specification all tables and columns.
  - Foreign keys are used to identify relationships between tables.
  - Denormalization may occur based on user requirements.
  - Physical considerations may cause the physical data model to be quite different from the logical data model.
  - Physical data model will be different for different RDBMS. For example, data type for a column may be different between Oracle, DB2 etc.

Physical data model

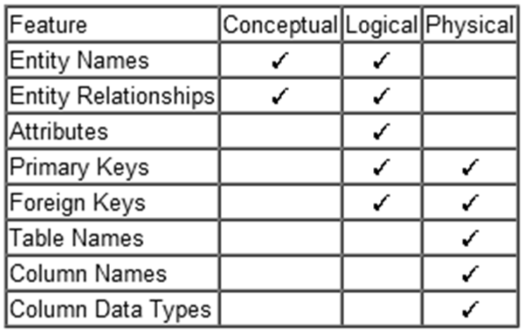
The steps for physical data model design are as follows:

1. Convert entities into tables.
2. Convert relationships into foreign keys.
3. Convert attributes into columns.
4. Modify the physical data model based on physical constraints / requirements.



Data Models

Here we compare these three types of data models. The table below compares the different features:



Feature	Conceptual	Logical	Physical
Entity Names	✓	✓	
Entity Relationships	✓	✓	
Attributes		✓	
Primary Keys		✓	✓
Foreign Keys		✓	✓
Table Names			✓
Column Names			✓
Column Data Types			✓

Figure 9-2 Relationship between data modeling and the systems development life cycle

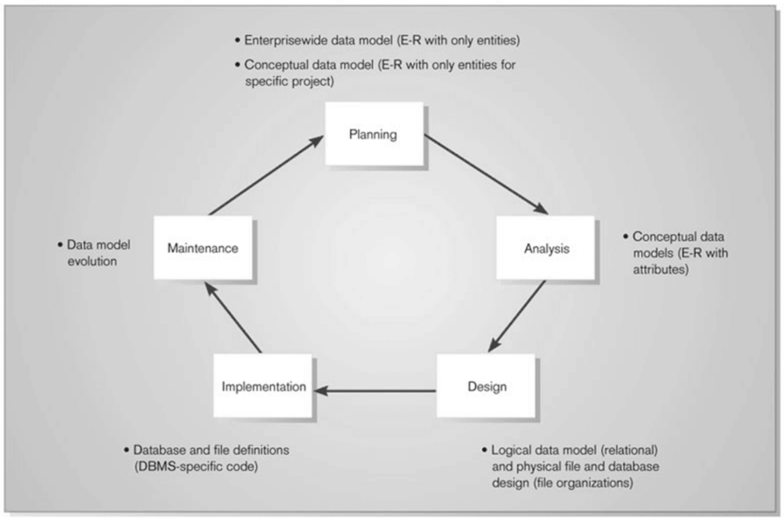

• Enterprisewide data model (E-R with only entities)
• Conceptual data model (E-R with only entities for specific project)
Planning
Analysis
• Conceptual data models (E-R with attributes)
Design
• Logical data model (relational) and physical file and database design (file organizations)
Implementation
• Database and file definitions (DBMS-specific code)
Maintenance
• Data model evolution

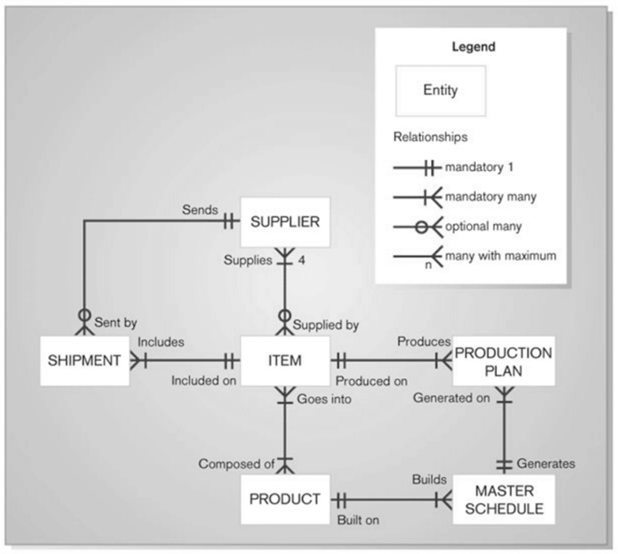
Fig.9.2 shows the different kinds of data modeling and database design that go on during the whole systems development cycle.

Conceptual Data Modeling and SDLC

- In the analysis phase, develop a new conceptual data model that includes all data requirements of the new system
- In the design stage, the final conceptual data model is translated into a physical design (a format from which physical data storage decisions can be made)
- Primary deliverable is an entity-relationship (E-R) diagram or class diagram
- Second deliverable is a set of entries about data objects to be stored in repository or project dictionary.
- Project repository links all design and data modeling steps performed during SDLC

Note: Each data store in a process model must relate to business objects represented in the data model.

Figure 9-3 Sample conceptual data model



Requirements Determination Questions for Data Modeling

- What are subjects/objects of the business?
  - ➔Data entities and descriptions
- What unique characteristics distinguish between subjects/objects of the same type?
  - ➔Primary keys
- What characteristics describe each subject/object?
  - ➔Attributes and secondary keys
- How do you use the data?
  - ➔Security controls and user access privileges

Requirements Determination Questions for Data Modeling (cont.)

- Over what period of time are you interested in the data?
  - ➔Cardinality and time dimensions
- Are all instances of each object the same?
  - ➔Supertypes, subtypes, and aggregations
- What events occur that imply associations between objects?
  - ➔Relationships and cardinalities
- Are there special circumstances that affect the way events are handled?
  - ➔Integrity rules, cardinalities, time dimensions

Introduction to Entity-Relationship (E-R) Modeling

- Entity-Relationship (E-R) Diagram
  - A detailed, logical representation of the entities, associations and data elements for an organization or business
- Notation uses three main constructs
  - Data entities
  - Relationships
  - Attributes

ERD Vs DFD

- ❑ DFD shows how data enter a system, are transformed in that system, and how it is stored in it. On the other hand, ERD represents the entity model and will show what a system or a database will look like but not explain how to implement it.
- ❑ With DFD, each of the processes and storings should have at least one data flow going towards it and one leaving it. With ERD, all the entities should represent a group of similar things. All the definitions in ERD should be unambiguous.

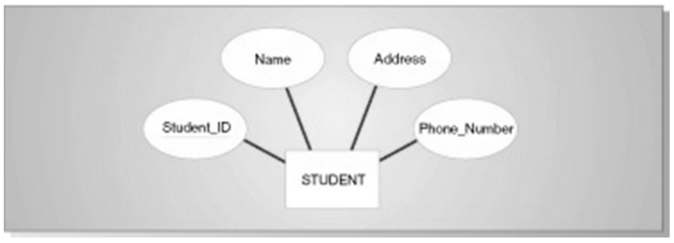
Entities

- An entity is a person, place ,object, event or concept about which an organization wishes to maintain data.
- Some Examples: EMPLOYEE, STUDENT,STORE, STATE, MACHINE, SALE, ACCOUNT
- **Entity type** (or entity class) collection of entities with common characteristics (Eg:STUDENT)
- **Entity instance**: A single occurrence of an entity type (E.g Eugene Ching)
- We use capital letters in naming an entity type and in an E-R diagram, the name is placed inside a rectangle representing the entity.

Attributes

- Each entity type has a set of attributes associated with it. An attribute is a property or characteristic of an entity that is of interest to the organization.
- Example: STUDENT: Student_ID, Student_Name, Home_ Address, Phone_Number,Major

E-R notation for STUDENT Entity type and Attributes

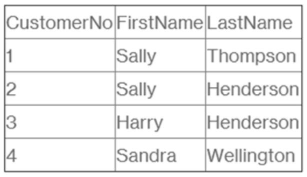


http://searchdatamanagement.techtarget.com
/answer/Definition-of-primary-super-foreign-
and-candidate-key-in-the-DBMS

Primary key

- A primary key is a column (or columns) in a table that uniquely identifies the rows in that table.

CUSTOMERS

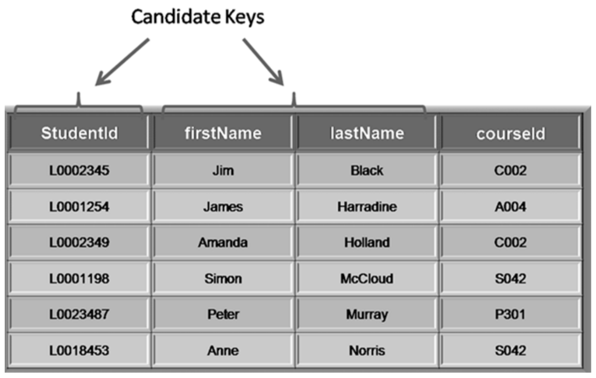


CustomerNo	FirstName	LastName
1	Sally	Thompson
2	Sally	Henderson
3	Harry	Henderson
4	Sandra	Wellington

- ✓ For example, in the table above, CustomerNo is the primary key.
- ✓ The values placed in primary key columns must be unique for each row: no duplicates can be tolerated. In addition, nulls are not allowed in primary key columns.

Candidate Keys and Identifiers

- Candidate key
  - A candidate key is a single field or the least combination of fields that uniquely identifies each record in the table.


Candidate Keys

StudentId	firstName	lastName	courseId
L0002345	Jim	Black	C002
L0001254	James	Harradine	A004
L0002349	Amanda	Holland	C002
L0001198	Simon	McCloud	S042
L0023487	Peter	Murray	P301
L0018453	Anne	Norris	S042

Candidate Keys and Identifiers

- Identifier
  - A candidate key that has been selected as the unique identifying characteristic for an entity type

Criteria for Selecting Identifiers

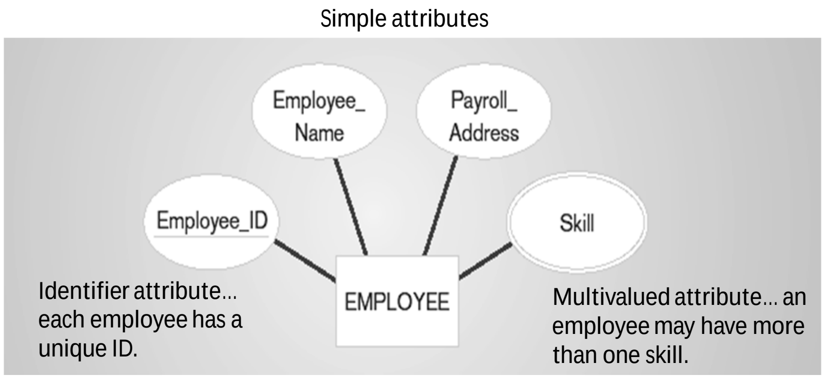
1. Choose a candidate key that will not change its value.
2. Choose a candidate key that will never be null.
3. Avoid using intelligent keys*.
4. Consider substituting single value surrogate keys for large composite keys.

* A key is a field (or column) used to identify a specific record in a database. An intelligent key is a key that also contains information about the record it is the key for. A fairly common one is an invoice number. It is a key because it uniquely distinguishes one invoice from another.

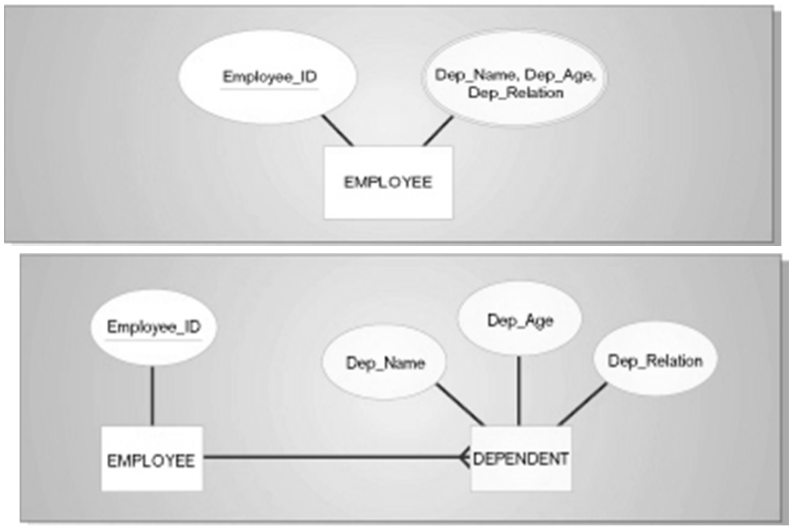
Multivalued Attributes

- An attribute that may take on more than one value for each entity instance
- Represented on E-R Diagram in two ways:
  - – double-lined ellipse
  - – weak entity

Example: Multivalued attribute (shown as double-lined ellipse)


Simple attributes
Employee_Name
Payroll_Address
Employee_ID
Skill
EMPLOYEE
Identifier attribute… each employee has a unique ID.
Multivalued attribute… an employee may have more than one skill.

Example: Multivalued attribute (shown as weak entity)



weak entity

- In a relational database, a weak entity is an entity that cannot be uniquely identified by its attributes alone; therefore, it must use a foreign key in conjunction with its attributes to create a primary key

10.27

Relationships

- Relationship
  - An association between the instances of one or more entity types that is of interest to the organization
  - Association indicates that an event has occurred or that there is a natural link between entity types
  - Relationships are always labeled with verb phrases

10.28

Degree of Relationship

- Degree: number of entity types that participate in a relationship
- Three cases
  - **Unary:** between two instances of one entity type
  - **Binary:** between the instances of two entity types
  - **Ternary:** among the instances of three entity types

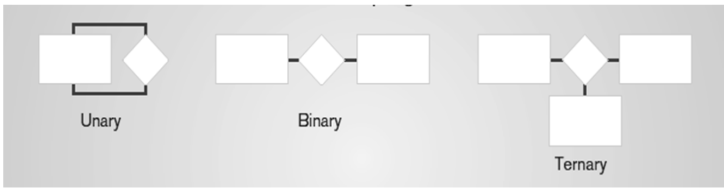
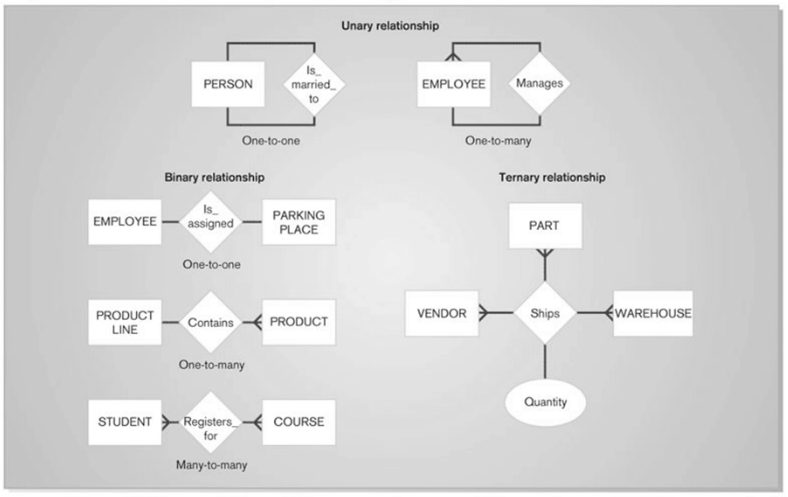


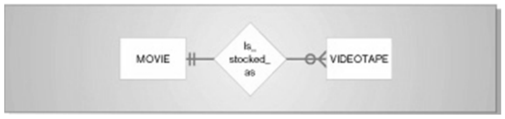
Figure 9-6 Example relationships of different degrees



Cardinality

- The number of instances of entity B that can or must be associated with each instance of entity A
- Minimum Cardinality
  - The minimum number of instances of entity B that may be associated with each instance of entity A
- Maximum Cardinality
  - The maximum number of instances of entity B that may be associated with each instance of entity A
- Mandatory vs. Optional Cardinalities
  - Specifies whether an instance must exist or can be absent in the relationship

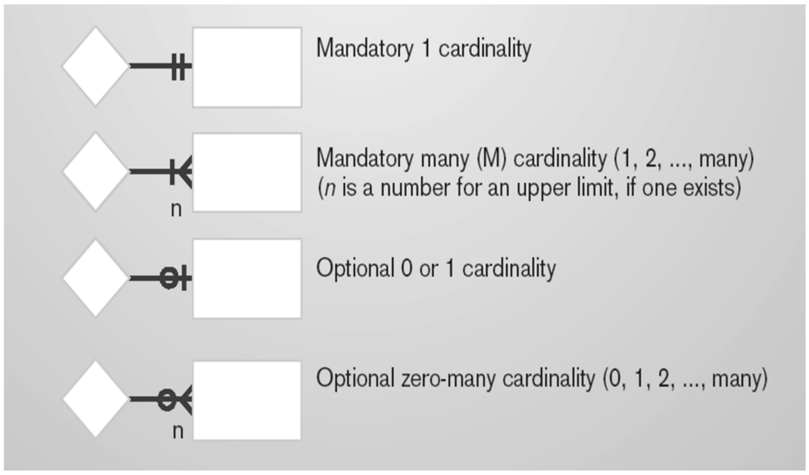
Cardinality-Example



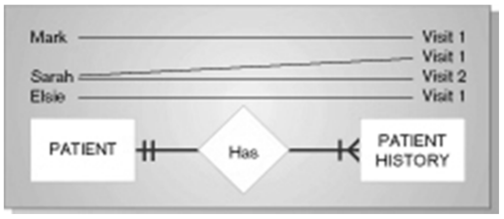
Minimum and Maximum cardinalities

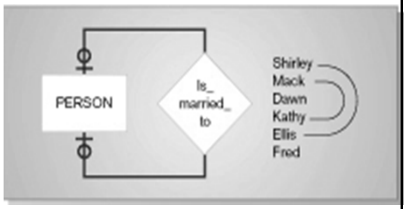
- In the preceding example, the minimum number of videotapes available for a video tape is zero, in which case we say that VIDEOTAPE is an- *optional participant* -in the relationship.
- If the minimum cardinality of a relationship is one , then we say entity B is -*mandatory participant* -in the relationship.
- For the preceding example, the maximum is “many” (an unspecified number greater than one)
- The zero through the line near the VIDEOTAPE entity means a minimum cardinality of zero, while crow’s foot notation means a “many” cardinality.

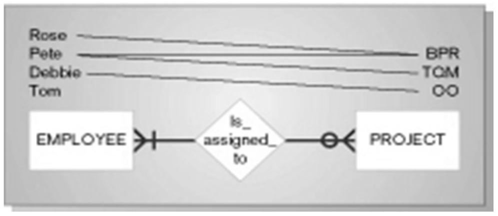
Cardinality Symbols


Mandatory 1 cardinality
Mandatory many (M) cardinality (1, 2, ..., many)
(*n* is a number for an upper limit, if one exists)
n
Optional 0 or 1 cardinality
Optional zero-many cardinality (0, 1, 2, ..., many)
n

Examples of cardinalities in relationships







Naming and Defining Relationships

- Relationship name is a verb phrase
- Avoid vague names
- Guidelines for defining relationships
  - Definition explains what action is being taken and why it is important
  - Give examples to clarify the action
  - Optional participation should be explained
  - Explain reasons for any explicit maximum cardinality

10.36

Guidelines for defining relationships (Cotnd)

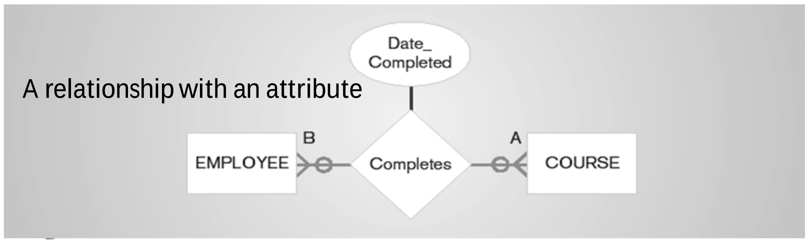
- Explain any restrictions on participation in the relationship
- Explain extent of the history that is kept in the relationship
- Explain whether an entity instance involved in a relationship instance can transfer participation to another relationship instance

Associative Entity

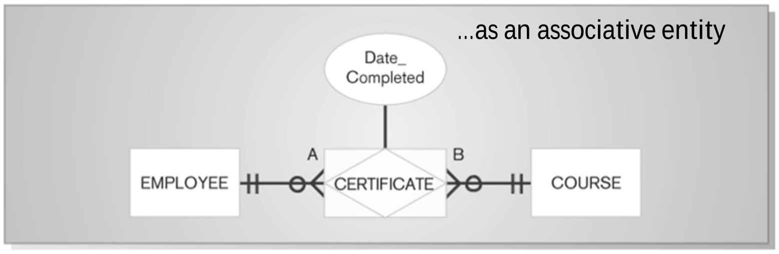
- An entity type that associates the instances of one or more entity types and contains attributes that are peculiar to the relationship between those entity instances.
- Ex: Suppose an organization wishes to record the date when an employee completes each course. The attribute Date-Completed is not a property of the entity EMPLOYEE nor a property of COURSE . It is a property of the relationship between EMPLOYEE and COURSE.
- In this example, an associative entity called CERTIFICATE can replace the 'completes' relationship. In the new E-R diagram, we have one-to-many relationships between EMPLOYEE and CERTIFICATE and also between COURSE and CERTIFICATE
- In representing an associative entity, the diamond symbol is included within the entity rectangle as a reminder that the entity was derived from a relationship.

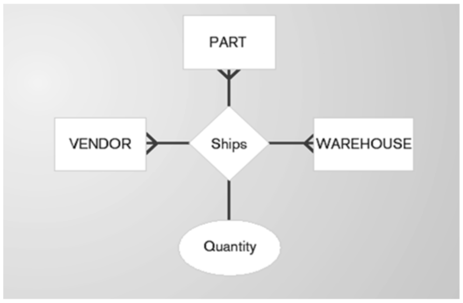
10.38

A relationship with an attribute

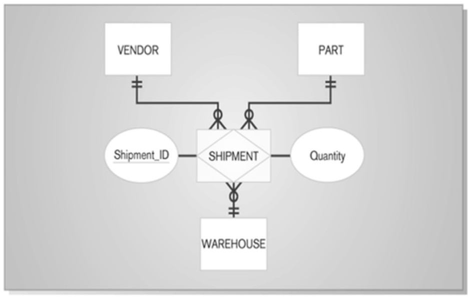


...as an associative entity





Ternary relationship



...as an associative entity

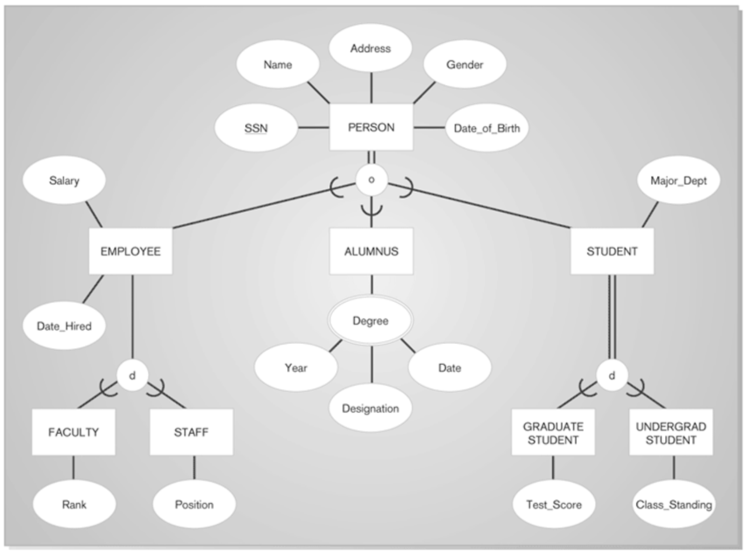
Supertypes and Subtypes

- **A Subtype is** a subgrouping of the entities in an entity type that is meaningful to the organization and that shares common attributes or relationships distinct from other subgroupings.
- Ex: STUDENT is an entity type in a university. Two subtypes of STUDENT are GRADUATE STUDENT and UNDERGRADUATE STUDENT
- **A Supertype is a** generic entity type that has a relationship with one or more subtypes.

Rules for Supertype/Subtypes Relationships

- Total specialization: an entity instance of the supertype must be an instance of one of the subtypes
- Partial specialization: an entity instance of the super type may or may not be an instance of one of the subtypes
- Disjoint: an entity instance of the supertype can be an instance of only one subtype
- Overlap: an entity instance of the supertype may be an instance of multiple subtypes

Figure 9-12 Example of supertype/subtype heirarchy



Business rules

- Specifications that preserve the integrity of a conceptual or logical data model.
- Four basic types of business rules are:

1.Entity integrity – each instance of an entity type must have a unique identifier that is not null.

2. Referential integrity constraints – rules concerning the relationship between entity types.

3. Domains – constraints on valid values for attributes.

4.Triggering operations – other business rules that protect the validity of attributes.

Data Dictionary

- a set of information describing the contents, format, and structure of a database and the relationship between its elements, used to control access to and manipulation of the database.

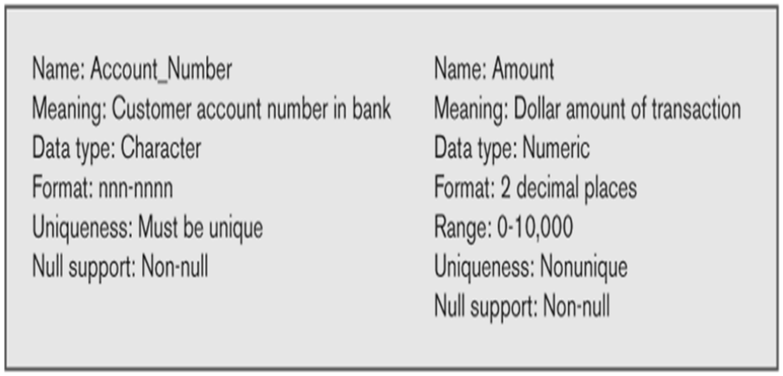
Domains

- The set of all data types and ranges of values that an attribute can assume.
- Domain definitions specify some or all of the following characteristics:
- data type, length, format, range, allowable values, meaning, uniqueness, and null support.
- **Several advantages**
- Domains verify that the values for an attribute are valid
- Domains ensure that various data manipulation operations are logical
- Domains help conserve effort in describing attribute characteristics

10.46

Typical Domain Definitions

Figure 9-13b Examples of business rules - Typical domain definitions



Name: Account_Number	Name: Amount
Meaning: Customer account number in bank	Meaning: Dollar amount of transaction
Data type: Character	Data type: Numeric
Format: nnn-nnnn	Format: 2 decimal places
Uniqueness: Must be unique	Range: 0-10,000
Null support: Non-null	Uniqueness: Nonunique
	Null support: Non-null

Triggering Operations

- **An assertion or rule that governs the validity of data manipulation operations such as insert, update and delete**
- **Includes the following components**:
- **User rule**

Statement of the business rule to be enforced by the trigger

- **Event**

Data manipulation operation (insert, delete, update) that initiates the operation

- **Entity Name**

Name of entity being accessedand/ or modified

- **Condition**

Condition that causes the operation to be triggered

- **Action**

Action taken when the operation is triggered

Typical Triggering operation

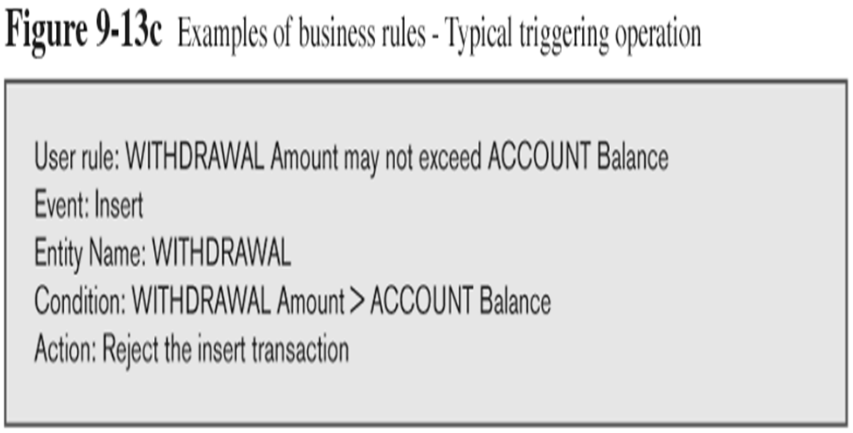


Figure 9-13c Examples of business rules - Typical triggering operation

User rule: WITHDRAWAL Amount may not exceed ACCOUNT Balance
Event: Insert
Entity Name: WITHDRAWAL
Condition: WITHDRAWAL Amount > ACCOUNT Balance
Action: Reject the insert transaction

Triggering Operations (Contd)

- With triggering operations, the responsibility for data integrity lies within the scope of database management system, not with application programs or human operators

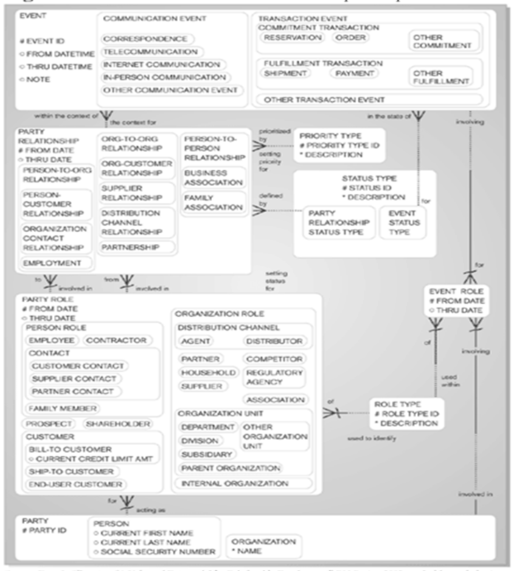
The Role of CASE in Conceptual Data modeling

- Business rules should be documented in the CASE repository. This has the following advantages:
- 1.Provides for faster application development
- 2.Reduces maintenance effort and expenditures
- 3.Facilitates end user involvement
- 4.Provides for consistent application of integrity constraints
- 5.Reduces time and effort required to train application programmers
- 6. Promotes ease of use of a database.

Packaged Data Models

- Generic data models that can be applied and modified for an organization
- Two categories
  - Universal
  - Industry-specific
- Benefits
  - Reduced implementation time and cost
  - High-quality modeling

Figure 9-14 A universal data model for relationship development



Source: From L. Silverston, "A Universal Data model for Relationship Development" *DM Review*, 2002, used with permission.)

Packaged data models provide generic models that can be customized for a particular organization's business rules