

Entity Relationship (ER) Modeling

Entity Relationship Model (ERM)

...

- ERM forms the basis of an ERD.
- From the end-users' point of view, the ERD represents the conceptual database.
- ERDs depict the database's main components: entities, attributes, and relationships.

- **Required and Optional Attributes**

- Required attribute is an attribute that must have a value at the time the instance is added. Null is not allowed as a value.
- Optional attribute is an attribute that allows to be left empty at the time the instance is added.

- **Domains**

- It is the set of possible values that an attribute can contain
- It is written as (<minimum>, <maximum>)
- Examples: gender (M,F), QPA (0,4), years of service (0,30)
- Attributes may share domain

- **Identifiers (Primary keys)**

- Is composed of one or more attributes that is used to differentiate one entity instance from another.
- These identifiers are mapped to primary keys in tables.
- It is underlined in the ERD.

- **Composite identifiers**

- A primary key that is composed of more than one attribute.

- **Composite and Simple Attributes**

- A *composite attribute* (different from composite key) is an attribute that can be subdivided to new several attributes.
- *Simple attribute* is an attribute that cannot be subdivided into several new attributes

- **Single-valued Attributes**

- Is an attribute that can have only a single value.
- Examples: social security number, student id number.
- Note: a single-valued attribute is not necessarily a simple attribute.

- **Multivalued Attributes**

- Attributes that can have many values.
- Are shown (in Chen ERM) with a double line connecting the attribute and the entity.

- **Implementing Multivalued Attributes**

- Though multivalued attributes are allowed in the conceptual model do not implement them in the RDBMS.
- Resolve this problem with either:
 - a. Within the original entity, create several new attributes for the multivalued attribute.
 - b. Create a new entity of the original multivalued attributes' component.

- **Derived attributes**

- It is an attribute whose value are calculated from other attributes.
- Need not be stored within the database.
- In the Chen notation, a dashed line is used to connect the attribute and the entity.
- Sometimes known as computed attributes.
- The decision to store derived attributes depends on the processing requirements and the constraint placed on the application.

	DERIVED ATTRIBUTE	
	STORED	NOT STORED
Advantage	<p>Saves CPU processing cycles</p> <p>Saves data access time</p> <p>Data value is readily available</p> <p>Can be used to keep track of historical data</p>	<p>Saves storage space</p> <p>Computation always yields current value</p>
Disadvantage	<p>Requires constant maintenance to ensure derived value is current, especially if any values used in the calculation change</p>	<p>Uses CPU processing cycles</p> <p>Increases data access time</p> <p>Adds coding complexity to queries</p>

- **Relationships**

- Relationship name is an active verb or a passive verb.

- **Connectivity and cardinality**

- Cardinality expresses the minimum and maximum number of entity occurrences associated with one occurrence of the related entity.
- In the ERD, numbers in the format (x, y) placed near the entity indicates the cardinality
- Note: a DBMS cannot handle the implementation of the cardinalities at the table level – it is done in the application program.

- **Weak (non-identifying) Relationships**

- Exists if the primary key (PK) of the related entity does not contain a PK component of the parent entity.
- Example:
- *COURSE(CRS_CODE, DEPT_CODE, CRS_DESCRIPTION, CRS_CREDIT)*
- *CLASS(CLASS_CODE, CRS_CODE, CLASS_SECTION, CLASS_TIME, ROOM_CODE, PROF_NUM)*

- **Strong (Identifying) Relationships**

- exists when the PK of the related entity contains a PK component of the parent entity
- the order in which the tables are created and loaded is very important
- Example:
- *COURSE(CRS_CODE, DEPT_CODE, CRS_DESCRIPTION, CRS_CREDIT)*
- *CLASS(CRS_CODE, CLASS_SECTION, CLASS_TIME, ROOM_CODE, PROF_NUM)*

- **Weak Entity**

- is an entity type whose existence depends on some other entity type.
- A weak entity type has no business meaning in an E-R diagram without the entity on which it depends.
- It does not typically have its own identifier.
- Generally, on an E-R diagram, a weak entity type has an attribute that serves as a partial identifier.





- **Strong Entity**

- A strong entity type is one that exists independently of other entity type - most of the basic entity types

- **Relationship Participation**

- **Optional participation** means that one entity occurrence does not require a corresponding entity occurrence in a particular relationship.
 - Its minimum cardinality is 0
- **Mandatory participation** means that one entity occurrence requires a corresponding entity occurrence in a particular relationship.
 - Minimum cardinality is 1.

Crow's Foot Symbols

CROW'S FOOT SYMBOL	CARDINALITY	COMMENT
	(0,N)	Zero or many. Many side is optional.
	(1,N)	One or many. Many side is mandatory.
	(1,1)	One and only one. 1 side is mandatory.
	(0,1)	Zero or one. 1 side is optional.

Relationship Degree

- indicates the number of entities or participants associated with a relationship.
- A **unary relationship** exists when an association is maintained within a single entity - also known as recursive relationship.
- A **binary relationship** exists when two entities are associated in a relationship.
 - Most common form of relationship.

- A **ternary relationship** implies an association among three different entities.
 - whenever possible, most higher-order (ternary and higher) relationships are decomposed into appropriate equivalent binary relationships

DEVELOPING AN ER DIAGRAM

...

- The process of database design is an iterative rather than a linear or sequential process.
- Building an ERD usually involves the following activities:
 - Create a detailed narrative of the organization's description of operations.
 - Identify the business rules based on the description of operations.
 - Identify the main entities and relationships from the business rules.
 - Develop the initial ERD.
 - Identify the attributes and primary keys that adequately describe the entities.
 - Revise and review the ERD

- The above activities are repeated until the end users and designers agree that the ERD is a fair representation of the organization's activities and functions.

End of Presentation

Source:

Rob, Peter and Coronel, Carlos. *Database Systems : Design, Implementation and Management*, 7th Edition. Course Technology, Thomson Learning Inc.©2007