Module 3
<u>Contents:</u>
Conceptual Modelling: High level Conceptual Data Model for Database Design, Entity Types, Entity Sets, Attributes and Keys, Relationship Types, Relationship sets, Roles, Structural Constraints, Weak Entity Types, ER Diagrams, Naming Conventions Relationship Types of Degree higher than Two. Design ER model for Real applications. **Enhanced ER model** :Specialization, Generalisation, Aggression
Relational Database Design: ER model to Relational Model Mapping

----

Entity Relationship Model (E-R model)

ER model – is a high level conceptual data model diagram.

ER model views the real world as a set of basic objects known as entities, their characteristics known as attributes and association among these objects known as relationships. Entities, attributes and relationship are basic constructs of ER diagram.

Entity- is distinguishable object that has independent existence in the real world. **Attributes** – Each

entity is represented by a set of attributes.

Entity type- a set or collection of entity that share same attribute but different values.

Entity set -collection of all instance of particular entity type in the database at any point of time.

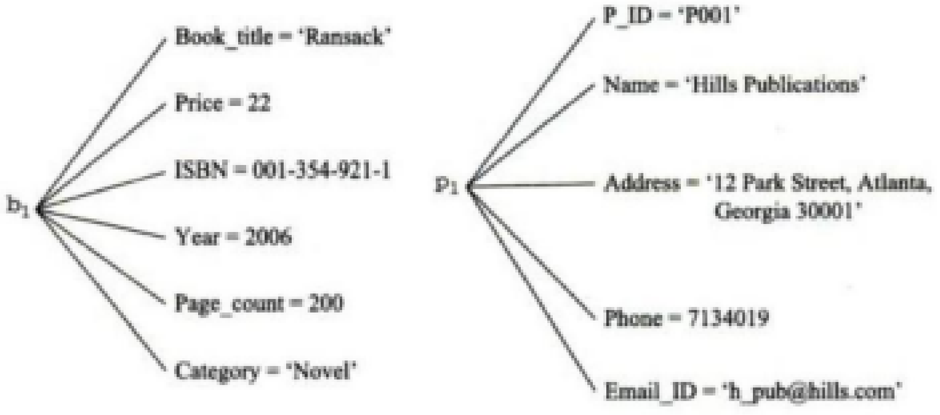
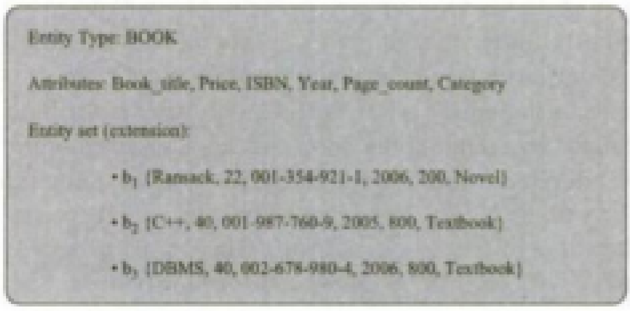


Fig. 2.1 *Entity instances b_1 (BOOK) and p_1 (PUBLISHER), and their attributes*

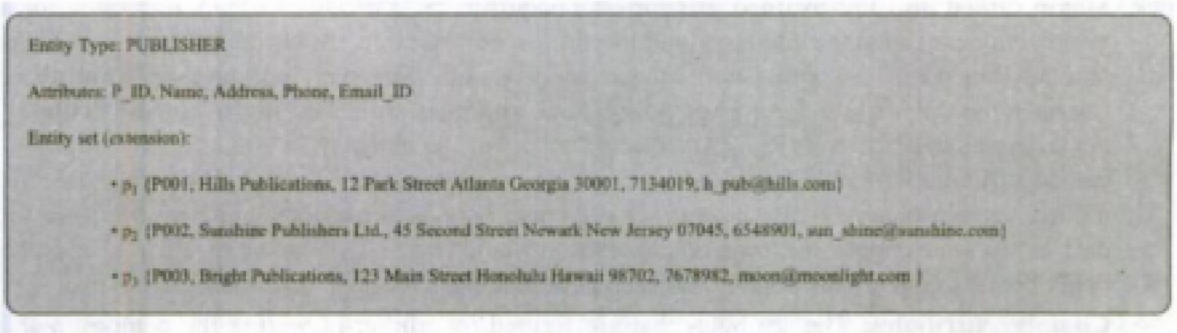


Entity Type: BOOK

Attributes: Book_title, Price, ISBN, Year, Page_count, Category

Entity set (extension):

- b_1 {Ramack, 22, 001-354-921-1, 2006, 200, Novel}
- b_2 {C++, 40, 001-987-760-9, 2005, 800, Textbook}
- b_3 {DBMS, 40, 002-678-980-4, 2006, 800, Textbook}



Entity Type: PUBLISHER

Attributes: P_ID, Name, Address, Phone, Email_ID

Entity set (extension):

- p_1 {P001, Hills Publications, 12 Park Street Atlanta Georgia 30001, 7134019, h_pub@hills.com}
- p_2 {P002, Sunshine Publishers Ltd., 45 Second Street Newark New Jersey 07045, 6548901, sun_shine@sunshine.com}
- p_3 {P003, Bright Publications, 123 Main Street Honolulu Hawaii 98702, 7678982, moon@moonlight.com }

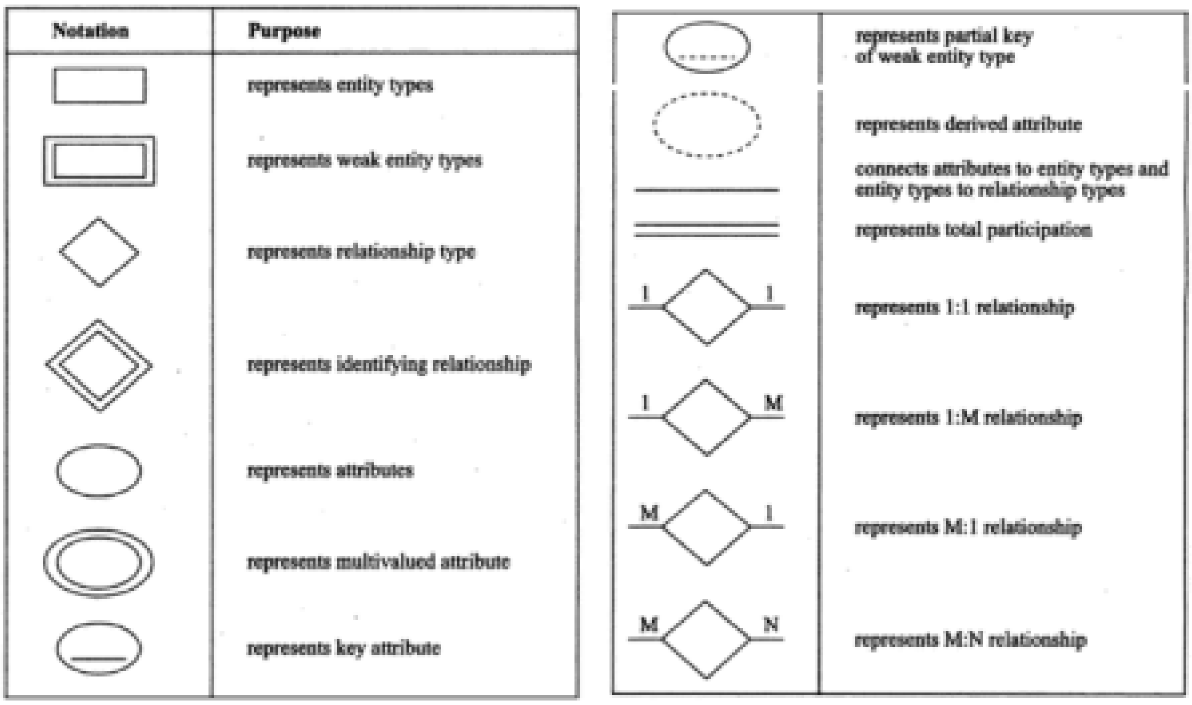
Types of attributes

1. Identifying and descriptive attributes

The attribute that is used to uniquely identifying an instance of an entity is known as identifying attribute(identifier)

A descriptive attribute (or descriptor) describes the non-unique characteristics of an entity instance.

Ex: for an entity type BOOK , ISBN is the identifier and other attributes are descriptive attributes.



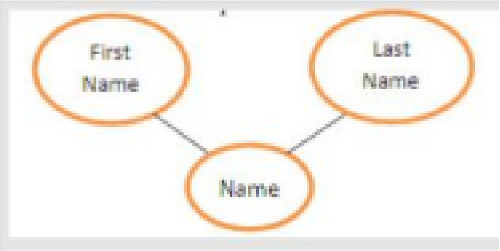
Notation	Purpose
	represents entity types
	represents weak entity types
	represents relationship type
	represents identifying relationship
	represents attributes
	represents multivalued attribute
	represents key attribute
	represents partial key of weak entity type
	represents derived attribute
	connects attributes to entity types and entity types to relationship types
	represents total participation
1 1	represents 1:1 relationship
1 M	represents 1:M relationship
M 1	represents M:1 relationship
M N	represents M:N relationship

Fig. 2.9 *E-R Diagram notations*

2. Simple and composite attributes

The attributes that are indivisible are known as simple attributes

Composite attributes can be divided into smaller subparts.

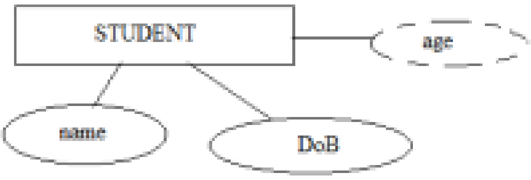


3. **Stored and derived Attributes:** Attribute that cannot be derived from other attributes are called as stored attributes.

- Example: Birth date of an employee is a stored attribute.

Derived Attributes: These attributes are derived from other attributes.

- Example: Age is a derived attribute . Age can be derived by the difference between current date and date of birth.



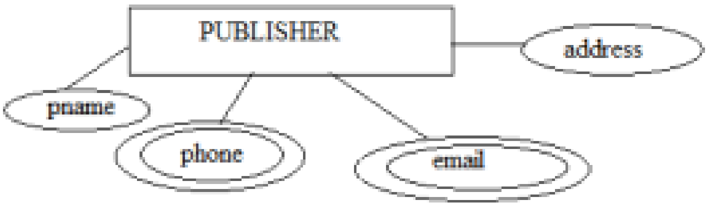
4. Single valued and multivalued attributes

The attributes that can have only one value for a given entity are called single valued attribute

Example: book_title, price, ISBN

The attributes that can have multiple values for a given entity are known as multivalued attribute.

Example: phone ,email



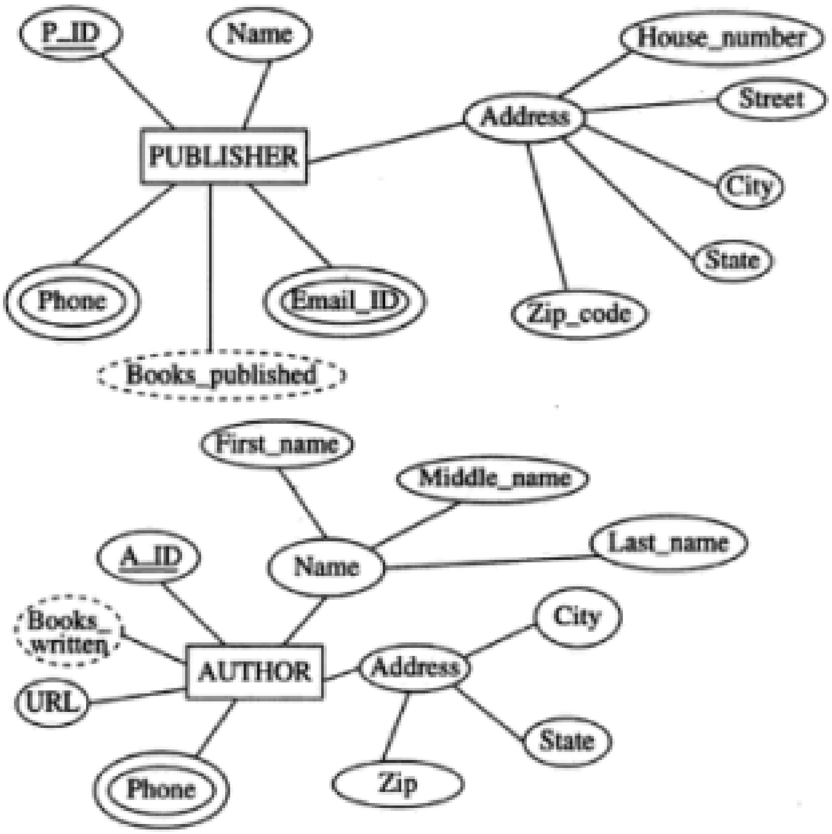


Fig. 2.11 *Multivalued, composite, and derived attributes*

5. Complex attributes

The attributes that formed by nesting the composite and multivalued attributes are called complex attribute

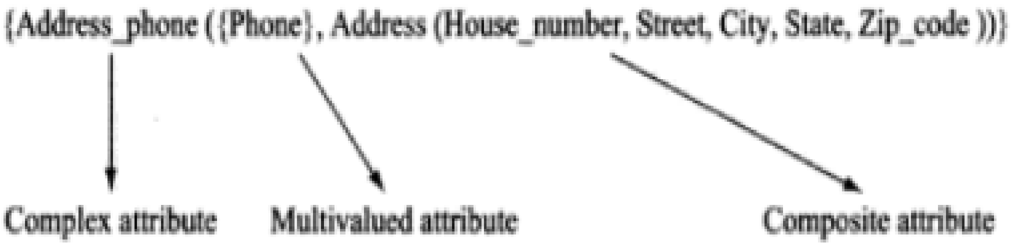

{Address_phone ({Phone}, Address (House_number, Street, City, State, Zip_code))}
Complex attribute
Multivalued attribute
Composite attribute

Fig. 2.4 *Complex attribute* — Address_phone

address{house name , state, city} is composite attribute, phone is multivalued attribute and address_phone is complex attribute.

Some important terms

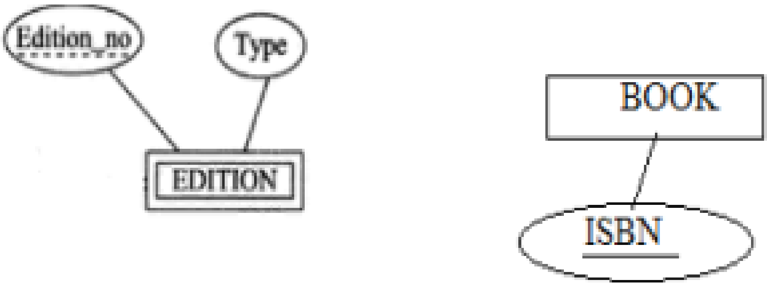
- *Relation:* a relation database consists of collection of tables or relations. Relation is a mathematical term for a table. Each row in a table represents relationship among a set of values.
- *Domain*: is a set of permissible values of same data type for an attribute. • *Field or Attribute*: the column of a relation represents field or attribute. The value of each attribute is taken from domain.

- *Tuple or record*: rows of a relation represents tuple.

- *Degree or arity*: number of attributes in a relation. The relation with degree one is called unary relation and with two is called binary and with three is called ternary. The degree is n, then known as n-ary relation.
- *Cardinality*: number of tuples in a relation.
- *Key*: any subset of a relation.
- *Super Key*: a key is called super key if it is sufficient to identify a unique tuple of a relation.
- *Candidate key*: minimal super key is called candidate key or irreducible key..i.e no proper subset of a candidate key is a superkey.
- *Primary key*: a candidate key is chosen to uniquely identify a tuple in a relation is called primary key. Other candidate key that are not chosen as primary key is known as alternate key.
- *Foreign key*: a key of a relation which is a primary key of some other relation in the relational schema.
- *Composite key* – if primary key is formed by the combination of two or more attributes then it is called composite key

Weak and strong entity type

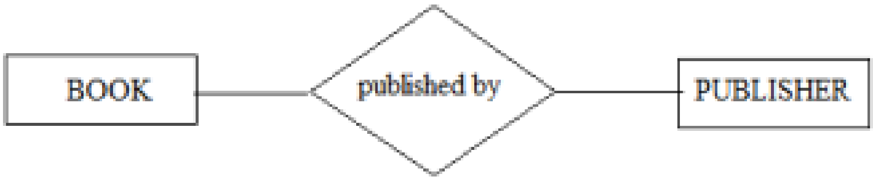
An entity type that does not have any key attribute of its own is called weak entity type[dependent entity].
An entity that has a key attribute is called a strong entity type [independent entity].
A week entity has a partial identifying key is known as partial key.



BOOK – Strong entity , primary key –ISBN
EDITION- weak entity, partial key – edition no

Relationship

An association between two or more entities is known as relationship.



Relationship type R among an entity type defines a set of association among instance from these entity types.
Relationship instances represented an association between individual entity instances.

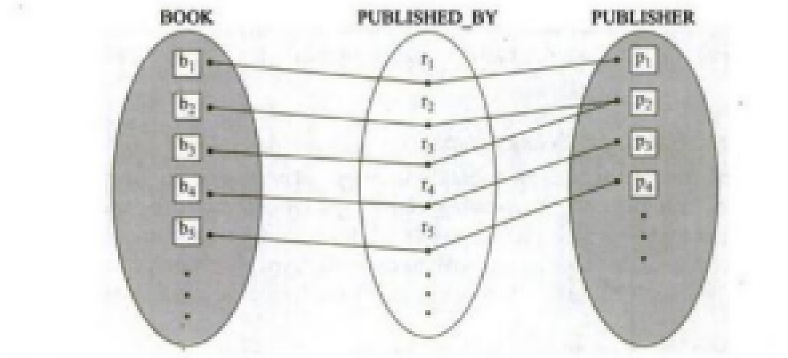


Fig. 2.5 Relationship PUBLISHED_BY between two entity types BOOK and PUBLISHER

Relationship may have descriptive attributes.

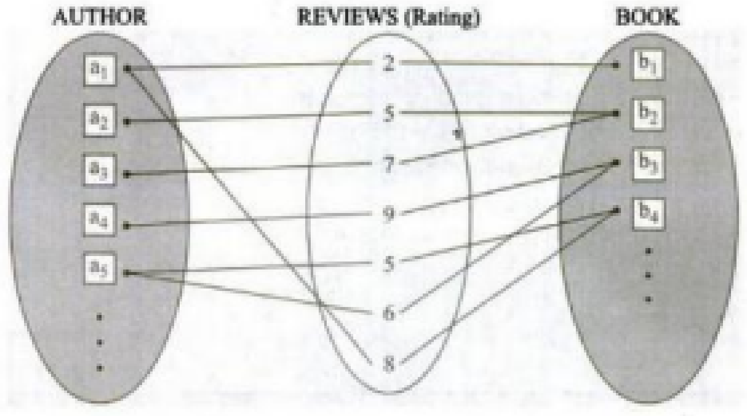


Fig. 2.6 Relationship type REVIEWS with attribute Rating

Here rating is the descriptive attribute

Identifying relationship

The relationship between a weak entity type and its identifying entity is known as identifying relationship of the weak entity type.

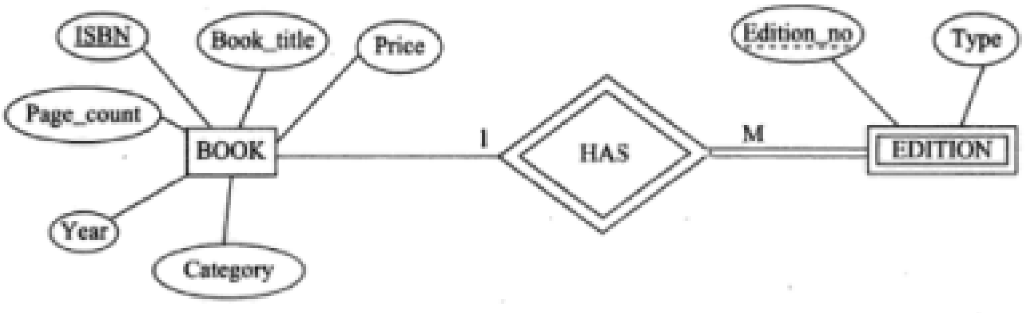


Fig. 2.14 E-R diagram showing weak entity, identifying relationship, and partial key

Role name and recursive relationship

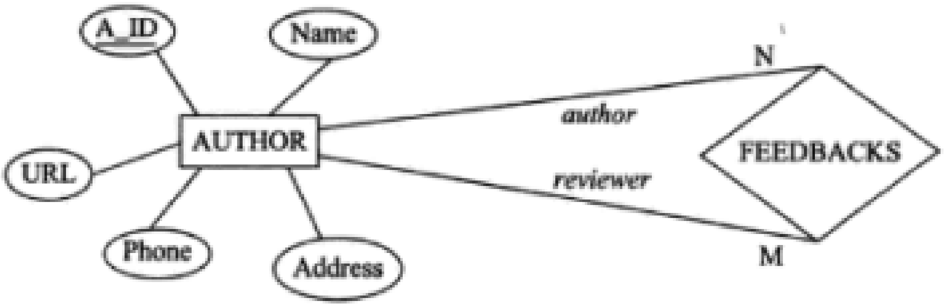


Fig. 2.13 *E-R diagram showing role names and recursive relationship*

Constraints on Relationship type

Two types of constraints

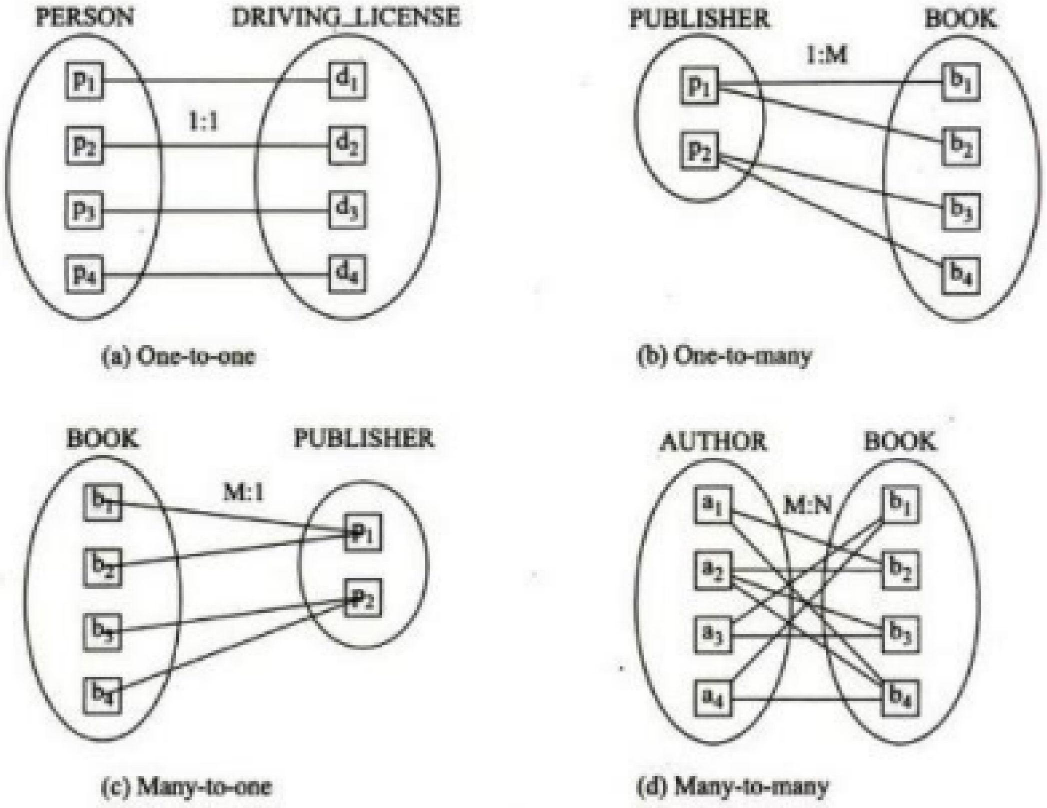
Mapping cardinalities
Participation constraints

Mapping cardinalities

1.one to one (1:1)
2. one to many(1:M)
3. Many to one (M:1)
4. many to many(M:N)

one to one (1:1)

In one to one mapping each instance of entity type E1 is associated with atmost one instance of entity type E2 and vice versa.



(a) One-to-one

(b) One-to-many

(c) Many-to-one

(d) Many-to-many

Fig. 2.8 *Mapping cardinalities*

Participaion Constraints

1. Total ( )
2. Partial( )

If every instance of entity type E participates in atleast one relationship instance of type R, the the participation is said to be total participation otherwise partial participation.

Enhanced ER model(Extended ER model or EER model)

ER models represent basic concepts of database schema. However some aspects such as inheritance among entity types cannot be expressed using the basic ER model. These aspects can be expressed by enhancing the ER model. The resulting diagrams are known as enhanced ER model (EER diagram).

Specialization and generalization

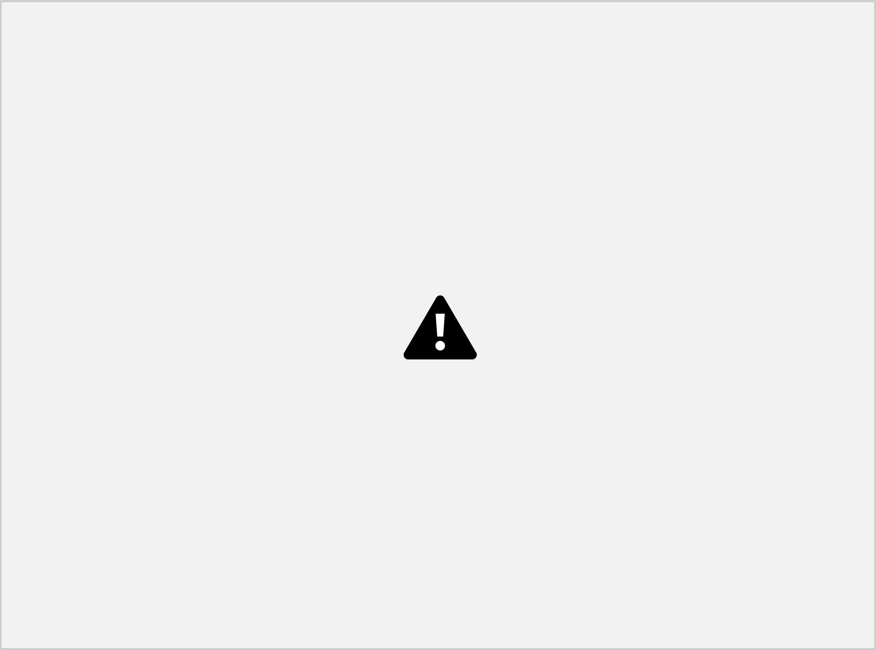
The entity type BOOK can be classified into three types: TEXTBOOK, LANGUAGE BOOK and NOVEL. These entity types are described by a set of attributes that are differentiating them from each other. The additional attributes are known as local or specific attributes.

For example TEXTBOOK may have the additional attribute subject, LANGUAGE BOOK has language attribute and novel has type attribute.

The process of defining the subgroup of a given entity type is called specialization.

The entity type containing the common attributes is known as super class and the entity type which is a subset of the superclass is known as subclass.

For example BOOK is superclass and entity type NOVEL, TEXTBOOK and LANGUAGE BOOK are subclass of BOOK.



Specialization is a top-down design approach. The design process may also follow bottom up design approach in which multiple lower level entity types are combined on the basis of common features to form higher level entity types. This process is known as generalization.

Generalization is the reverse process of specialization.

Specialization and generalization are different in terms of their starting and ending point.

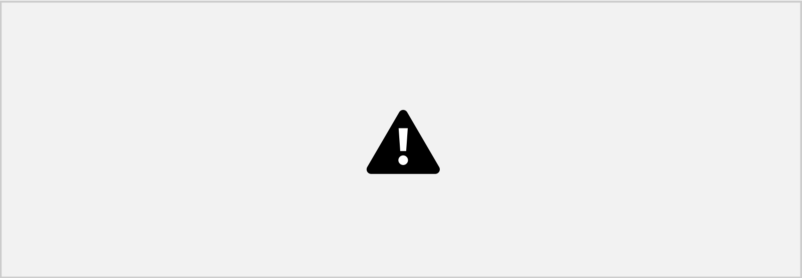
Specialization starts with a single higher level entity type and ends with a set of lower level entity types having some additional attributes that distinguish them from each other.

Generalization on the other hand starts with identification of number of lower level entity types and end with grouping of the common attributes to form a single higher level entity type.

Attribute inheritance

BOOK- year, page count, category, ISBN, book_title, Pid
TEXT BOOK- year, page count, category, ISBN, book_title, Pid , **subject**
Language BOOK- year, page count, category, ISBN, book_title,Pid, **Language**
Novel - year, page count, category, ISBN, book_title,Pid ,**type**

If a subclass is the subset of only one superclass, that is it has only one parent, it is known as single inheritance. And the resulting structure is known as specialization or generalization hierarchy. Fig above is the example of single inheritance.
If a subclass is formed from more than one superclass, it is known as multiple inheritance and the resulting structure is known as specialization or generalization lattice.



Constraints on Specialization and Generalization

Constraints on subclass membership

- Condition defined
- User defined

Condition defined

In condition defined, membership of entities is determined by placing a condition on the value of some attribute of the superclass.

For example, all instance of type BOOK can be evaluated based on the value of the attribute category. The entities that satisfy the condition “category=textbook” are allowed to belonging to the subclass TEXTBOOK only. Similarly “category=novel” are allowed to belonging to the subclass NOVEL.

User- defined

If the membership of the superclass entities in a given subclass is determined by database users and not by any membership function is called user defined specialization and their subclass that are formed are called user defined subclass.

<u>Disjoint and overlapping constraints</u>

<u>Disjoint constraint</u>: this constraint specifies that the same higher level entity instance cannot be belong to more than one lower level entity types. It is represented by a symbol „d‟ written in a circle.

<u>Overlapping constraints:</u> this constraint specifies that the same higher level entity instance can be belong to more than one lower level entity types. It is represented by a symbol „o‟ written in a circle.

<u>Completeness constraint</u>

1. Total specialization
2. Partial specialization

<u>Total specialization</u>

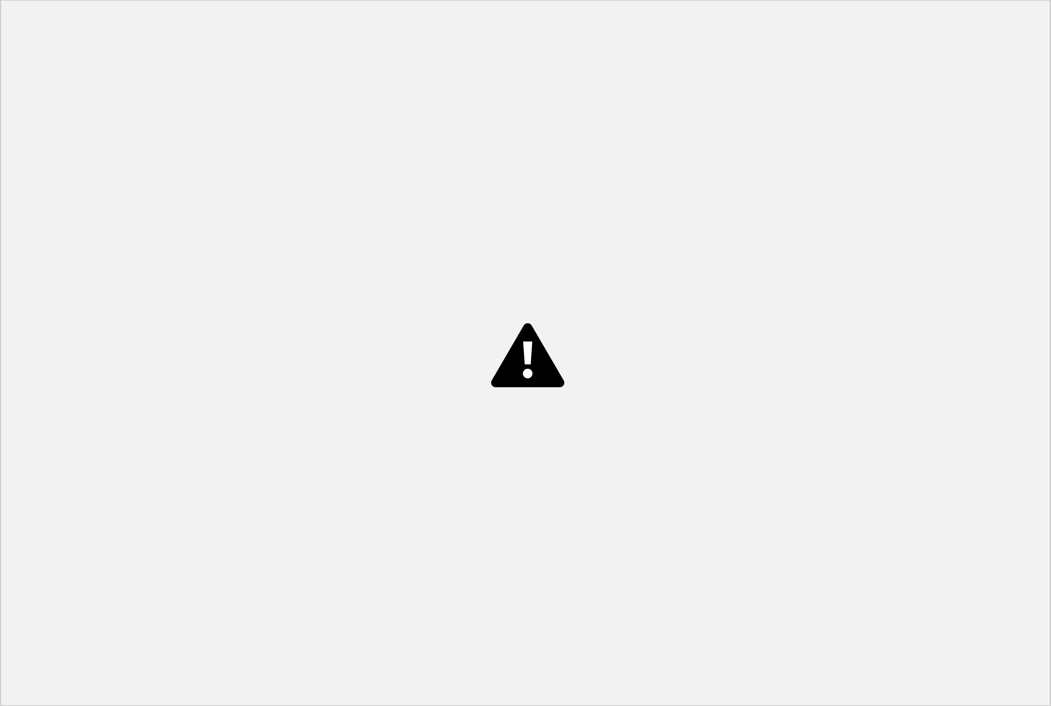
It specifies that each higher level entity must belong to atleast one of the lower level entity types in the specialization

<u>Partial specialization</u>

It allows some of the instance of higher level entity types not belong to lower level entity.

Aggregation

ER model cannot represent relationship among relationship. In EER model we can represent them by aggregation. The process through which one can *treat relationship as higher level entity* is known as aggregation. For example, in the Online book database, the relationship WRITES between entity types AUTHOR and BOOK can be treated as a single higher level entity called WRITES. i.e BOOK and AUTHOR are aggregated into a single entity type WRITES. Here we use the fact that once the author has written a book then only it gets published. Then the relationship PUBLISHED_BY can be shown between entity types PUBLISHER and WRITES.



Mapping of ER model to relational model

Step 1: Mapping of Regular Entity Types

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 Multivalued attributes.
Step 7: Mapping of N-ary Relationship Types.
Step 8: Mapping Specialization or Generalization.
Step 9: Mapping Aggregation

Step 1: Mapping of Regular Entity Types.

● For each regular (strong) entity type E in the ER schema, create a relation R that includes all the simple attributes of E.

● Key attributes of E becomes the primary key for R.

● If the chosen key of E is composite, the set of simple attributes that form it will together form the primary key of R.

Schema for STUDENT table is STUDENT(Roll_no, Name, Class, Subject)

Schema is

Step 2: Mapping of Weak Entity Types

- For each weak entity type W in the ER schema with owner entity type E, create a relation R and include all simple attributes (or simple components of composite attributes) of W as attributes of R.

- In addition, include as foreign key attributes of R the primary key attribute(s) of the relation(s) that correspond to the owner entity type(s).

- The primary key of R is the *combination of* the primary keys of the owners and the partial key of the weak entity type W, if any.

Partial key or discriminator – check_number

Step 3: Mapping of Binary 1:1 Relation Types

For each binary 1:1 relationship type R in the ER schema, identify the relations S and T that correspond to the entity types participating in R.

Foreign Key approach: Choose one of the relations-S, say-and include a foreign key in S the primary key of T. It is better to choose an entity type with *total participation* in R in the role of S. **Step 4: Mapping of Binary 1:N Relationship Types.**

● For each regular binary 1:N relationship type R, identify the relation S that represent the participating entity type at the N-side of the relationship type.

● Include as foreign key in S the primary key of the relation T that represents the other entity type participating in R.

● Include any simple attributes of the 1:N relation type as attributes of S.

Step 5: Mapping of Binary M:N Relationship Types.

● For each regular binary M:N relationship type R, *create a new relation* S to represent R.

● Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types; *their combination will form the primary key* of S.

● Also include any simple attributes of the M:N relationship type (or simple components of composite attributes) as attributes of S.

Step 6: Mapping of Multivalued attributes.

● For each multivalued attribute A, create a new relation R. This relation R will include an attribute corresponding to A, plus the primary key attribute K-as a foreign key in R-of the relation that represents the entity type of relationship type that has A as an attribute.

● The primary key of R is the combination of A and K. If the multivalued attribute is composite, we include its simple components.

Step 7: Mapping of N-ary Relationship Types.

- For each n-ary relationship type R, where n>2, create a new relationship S to represent R.

- Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types.

- Also include any simple attributes of the n-ary relationship type (or simple components of composite attributes) as attributes of S.

9.Mapping of aggregation

- Include primary key of the entities and descriptive attributes ●

COPYRIGHT relation with schema (ISBN,P_ID,A_ID, Copyright_date