Lecture Note #2

COSC4820/5820 Database Systems

Department of Computer Science

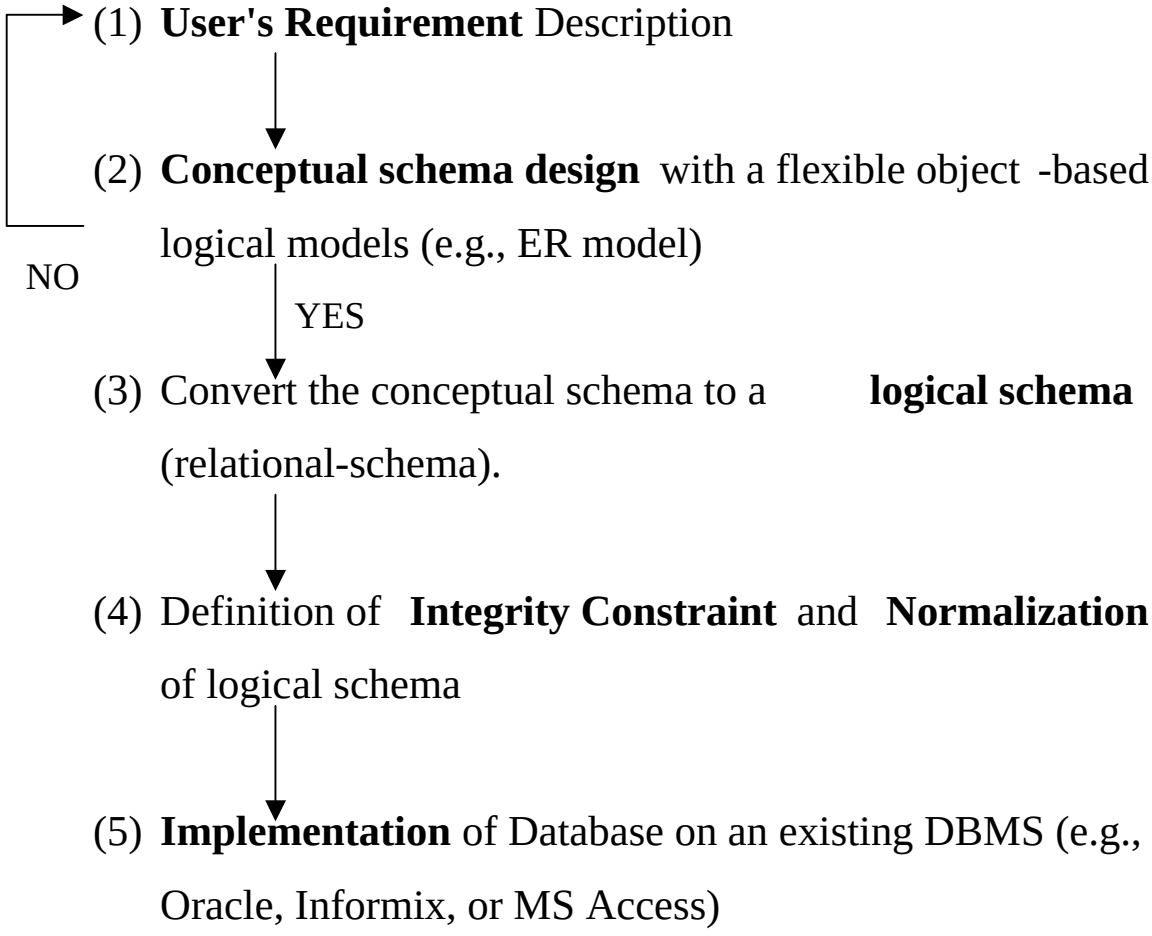
University of Wyoming

Byunggu Yu, 01/21/2002

ENTITY-RELATIONSHIP (ER) MODEL

(Originally designed by Peter Chen, 1976)

1. Background (Five Levels of Relational Database Design)



2. Entity sets

Entity: An entity is a concrete object (e.g., a person or a book) or an abstract object (e.g., a checking account, a holiday, or a concept) in the real world that can be uniquely identified.

Entity set: An entity set is a set of entities of the same type that have the same properties (attributes).

ER Diagram:

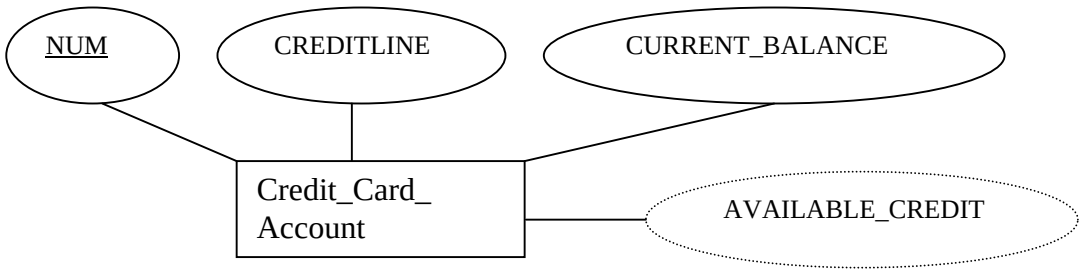


Figure 1 ER diagram of simple employee entity set

The primary key attribute is underlined. The primary key of an entity set is a candidate key that is chosen by the database designer as the principle means of identifying entities within an entity set (see Sectiown 4).

Dashed Circle: A dashed ellipse represents a derived attribute. The value "AVAILABLE_CREDIT" in Figure 1 can be der ived (i.e., CREDITLINE – CURRENT_BALANCE)

3. Relationship sets

Relationship: A relationship is an association among several entities.

A relationship r is an ordered n -tuple (e1, e2, …,en), where e *i* is a member of entity set E*i* for 1<= *i* <= n.

We say, the entities e1, e2, …,en participate in a relationship r.

Relationship set: A relationship set is a set of relationships of the same type that have the same properties.

A relationship set R is {(e1, e2, …,en) | e1 is a member of E1, e2 is a member of E2, …, en is a member of En}.

We say, the entity sets E1, E2, …,En participate in a relationship set R.

ER Diagram:

The following diagram shows that the mapping cardinality of the relationship set "DEPT -EMPLOYEE" is "one -to-many". That is, each department has many employees and each employee is associated with at most one department. The line from a relationship set to a participating entity set will be directed with an arrow pointing to the entity set if the entity set is on the "one" side. You can write the role of the participating entity member on the line between the entity set and the relationship set.

Double line ═══ represents Total Participation. That is, each entity in the entity set must participate in one or more relationships in the relationship set. For example, if a double line is used between "EMPLOYEE" and "DEPT -EMPLOYEE", each employee must be associated with one department.

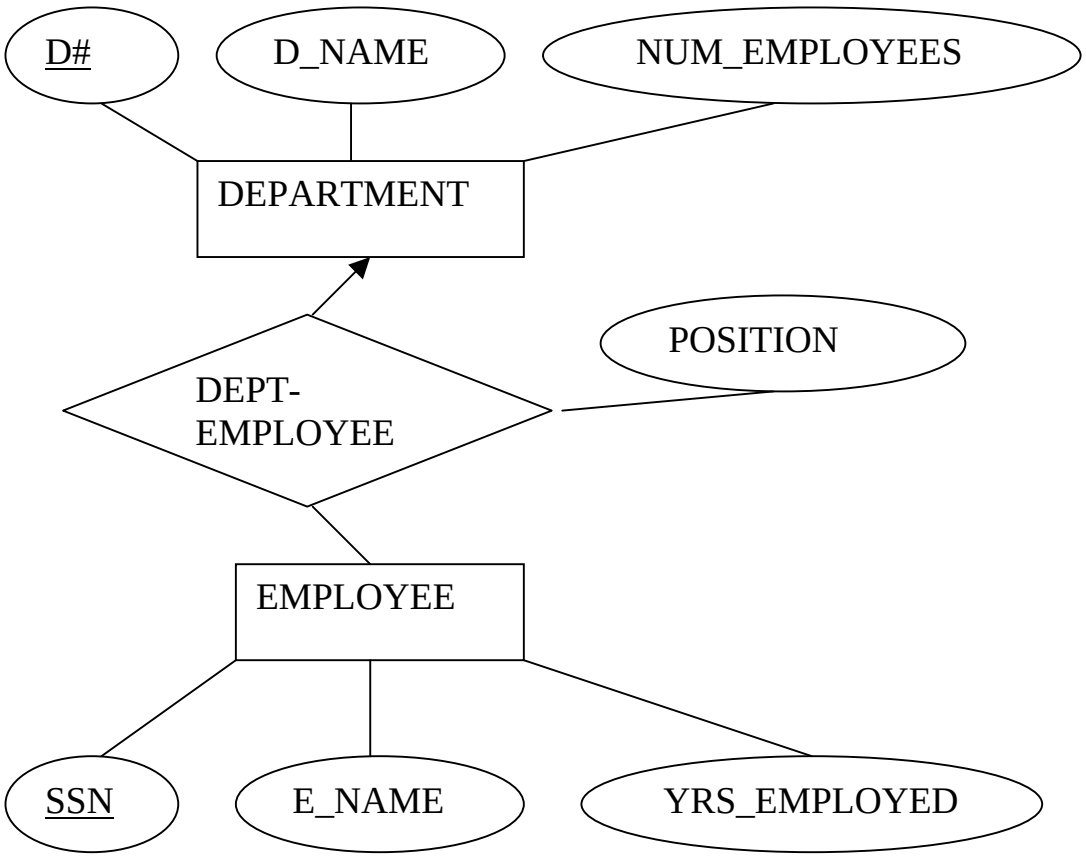


Figure 2 ER diagram including a one-to-many binary relationship set

4. Keys in ER model

- **Superkey:** A superkey is a set of one or more attributes that identifies uniquely an entity in the entity set.
- **Candidate key:** a superkey may contain extraneous attributes. A candidate key is a superkey for which no proper subset is a superkey (i.e., a minimal superkey containing no extraneous attribute).
- **Primary key:** A candidate key that is chosen by the database designer as the principle means of identifying entities within the entity set.

5. Mapping Cardinalities and the Placement of Relationship Attributes

The mapping cardinality of a binary relationship set must be one of the following:

- one-to-one
- one-to-many or many-to-one
- many-to-many

Note, as introduced, the line between the relationship set and any "one" side participating entity set is directed with an arrow pointing to the entity set.

6. Strong Entity and Weak Entity

- Strong Entity Set: If an entity set has a primary key then it is a strong entity set.
- Weak Entity Set: If an entity set has no key, i.e. we cannot identify uniquely an entity member, then it is a weak entity set.
- Example:

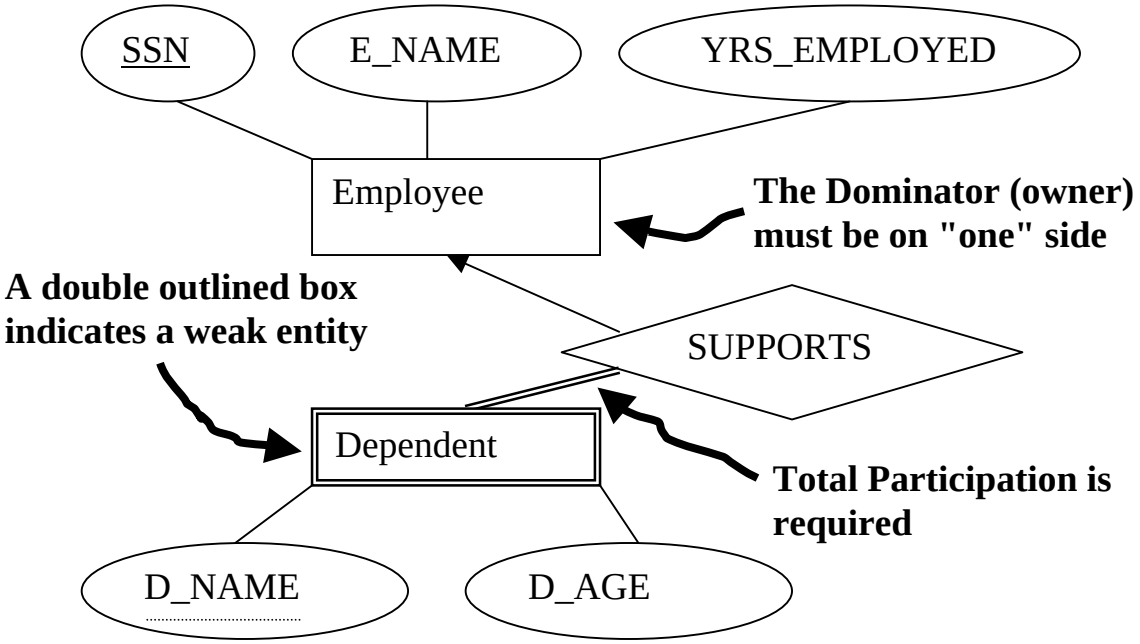


Figure 3 Strong entity and Weak entity

The domin ating entity set is usually a strong entity set. The relationship between a weak entity set and the dominator must be "many-to-one". If it is "one -to-one" every attributes of the weak entity set will be placed in the dominator as its attributes and the weak entity can be safely eliminated. The weak entity set has

"discriminator" and we can identify uniquely each entity in the weak entity set with { *dk*, *wd*}, where *dk* is the primary key of the dominating entity set and *wd* is the discriminator (dashed underlined attribute) of the weak entity set. Because of this, the cardinality of the relationship between a weak entity set and its dominator cannot be "many-to-many".

Weak entity set can be represented by a multivalued composite attribute of the dominator (a dou ble ellipse  represents a multi-valued attribute). Each multivalued attribute must have the maximum number of values. However, multivalued attributes are barely used in relational database design.

TIP: If there is a multivalue attribute, you may want to convert it to a weak entity set.

7. Extended ER Feature #1: Specialization & Generalization

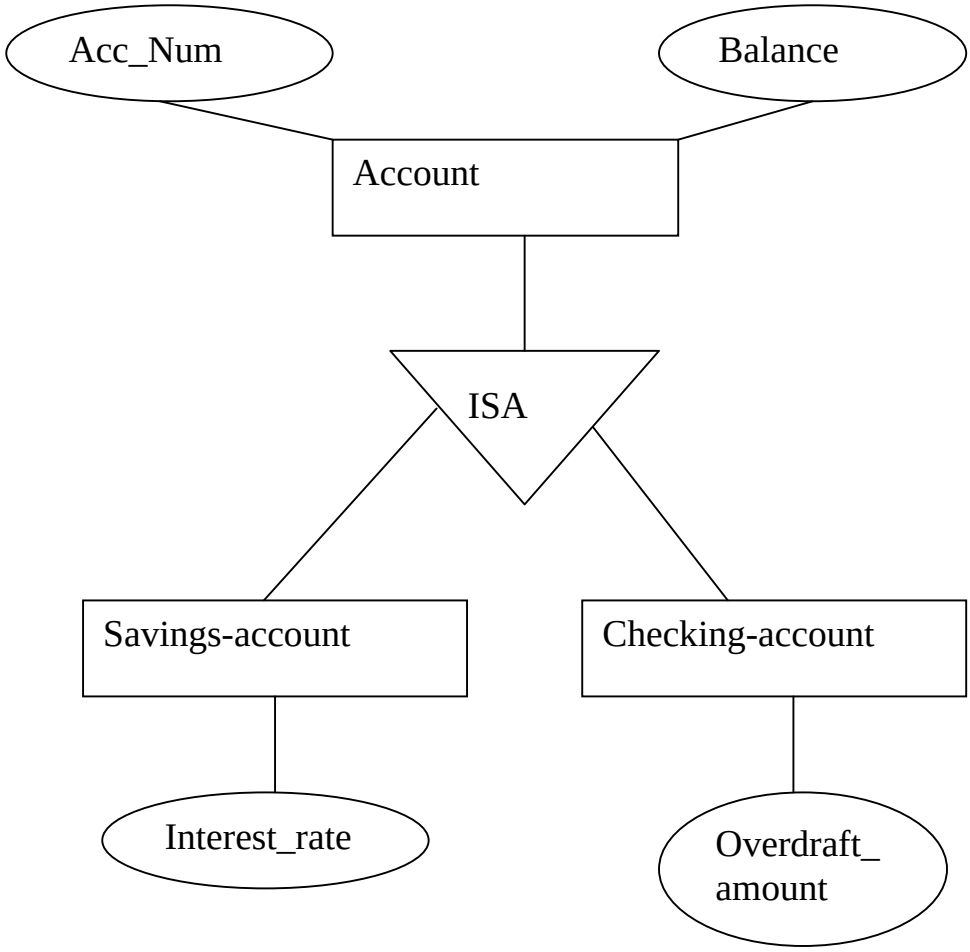


Figure 4 Specialization (Generalization) example

Both specialized entity sets (subclasses) "Savings -account" and "Checking-account" inherit all the attributes of the g eneralized entity set (superclass) "Account"

8. Extended ER Feature #2: Aggregation

Example: E1 and E2 participate in R1. Each relationship in R1 has zero or more associated entities in E3.

Design #1:

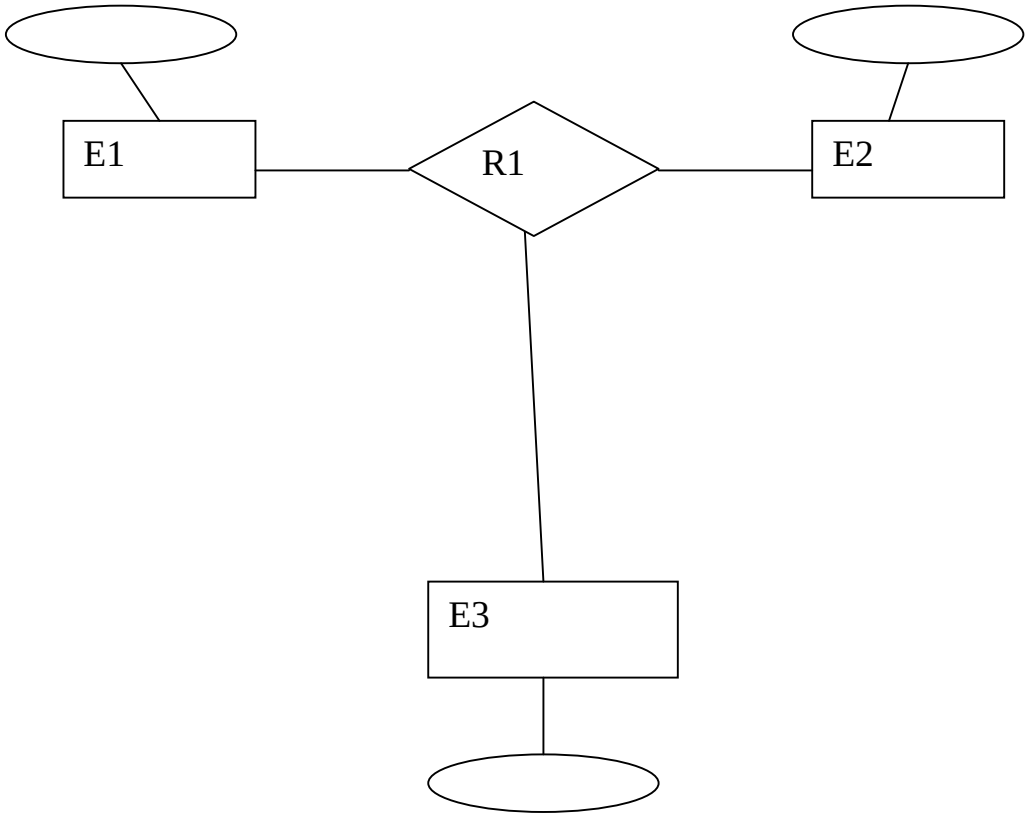


Figure 5 ER diagram without aggregation #1

➔ Problem: Each relationship in R1 must have at least one participating entity in E3.

Design #2:

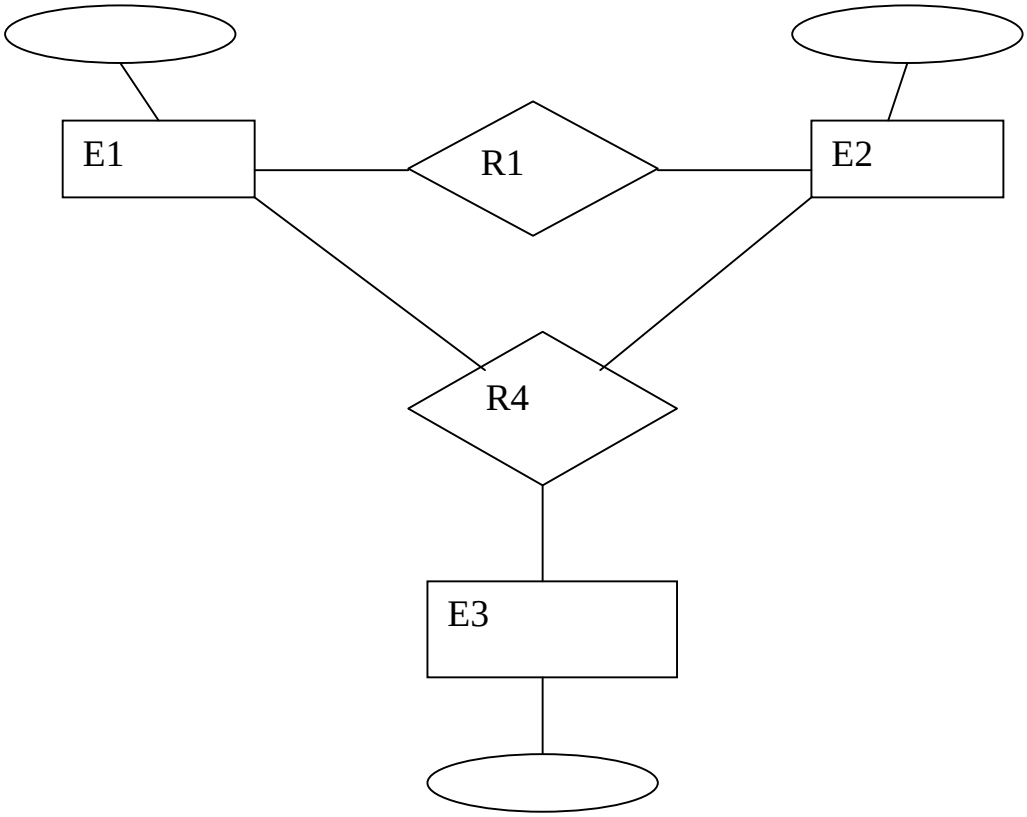


Figure 6 ER diagram without aggregation #2

➔ Problem: Some relationships in R1 will be duplicated in R4 (information redundancy/inconsistency problem).

Design #3 with the "Aggregation" feature of the extended ER model

If we can represent the relationship between a relationship set and an entity set or between two relationship sets, the above problems can be solved. Extended ER model allows us to *aggregate* a sub ER model into a single entity set.

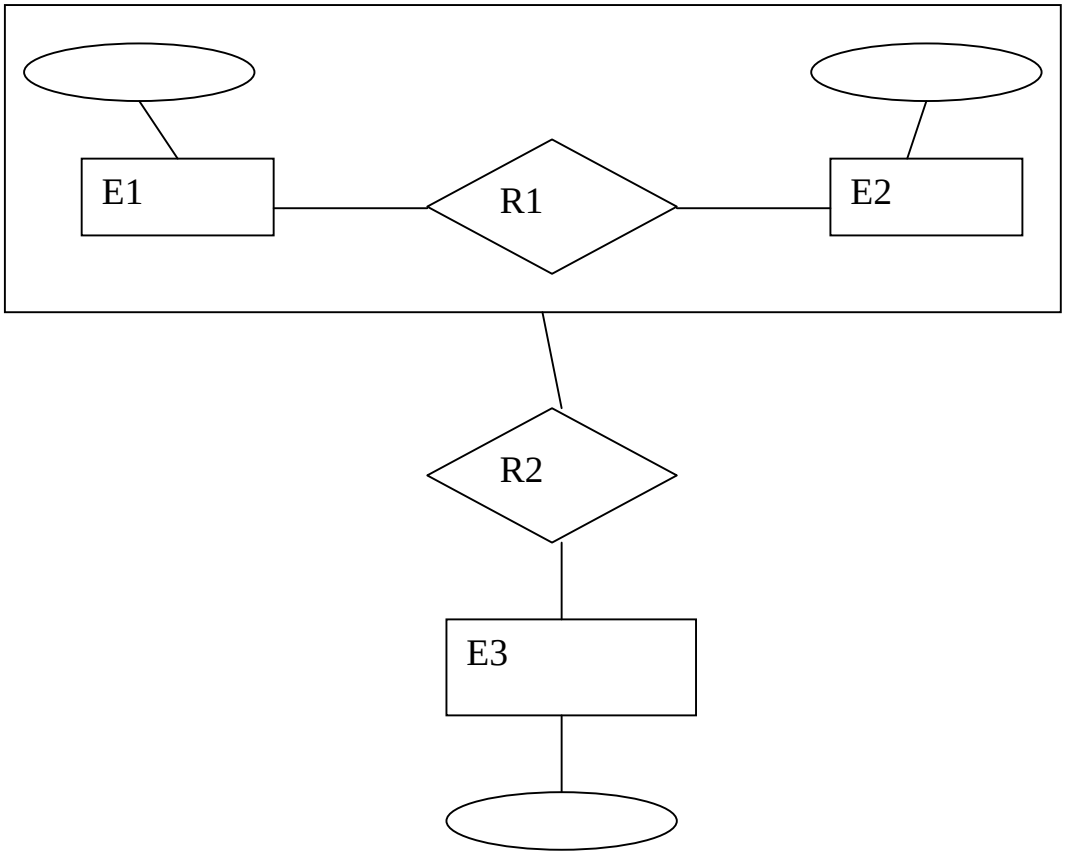


Figure 7 ER diagram with aggregation

➔ No contradiction, No duplication

9. Converting an ER Schema to Relational Schema (Tables)

- Strong entity sets:

A strong entity set A with attributes a1, a2, …, an is represented by a table called A with n distinct columns a1, a2, …, an. For example, the follo wing table “Employee” represents the entity set Employee in Figure2.

Employee

SSN	E_NAME	YRS_EMPLOYED

- Weak entity sets:

A weak entity set A with attributes a1, a2, …, an can be represented by a table called A with distinct columns {b1, b2, …, bm} ∪ {a1, a2, …, an}, where {b1, b2, …, bm} is the primary key of the dominating entity set B. The primary key of this table will be {b1, b2, …, bm} ∪{discriminator of A}. For example, the weak entity set "Dependent" in Figure 3 is converted to the following table "Dependent":

Dependelt

SSN	**D_NAME**	D_AGE

- Relationship sets:

A n -ary relationship set R, which has attributes r1, r2, …, rk is represented by a table called R with distinct columns {k1, k2, k3, …, kn}∪{r1, r2, …, rk}, where {k1, k2, …, kn} is the union of the primary keys of every participating entity sets. The primary key of this table is {k1, k2, …, kn}. For example, the relationship set "DEPT-EMPLOYEE" in Figure 2 will be represented by the following table "DEPT-EMPLOYEE"

DEPT-EMPLOYEE

D#	**SSN**	POSITION

- Many-to-one binary (or unary) Relationship set:

We don’t have to create a table for the relationship set. Instead, include the primary key of "one" side entity set into the table of the entity set on the "many" side. Now, the primary key of the "many" side table will be:

(1) When the "many" side entity set is a weak entity, then the key is the same as that of Weak Entity Conversion case.

(2) If the "many" side entity set is not a weak entity, then the key is the original primary key of the entity set.

- One-to-One binary (or unary) Relationship set:

(3) We don’t have to create a table for the relationship set. Instead, include the primary key of one entity set into the table of the other entity set.

- Generalization:

(1) Convert the parent entity set to a tabl e as we do with a strong entity set. For example, the "Account" in Figure 4 is represented by a table Account = {Account -number, Balance}

(2) Convert each child entity set into a table as (1). Then, add the primary key of the parent to the table as its prima ry key attributes. The "Savings -account" in Figure 4 is represented by a table Savings -account = {Account -number, Interest -rate}

(3) Repeat (2) until there is no remaining child.

If the generalization is disjoint (i.e., no entity is a member of more than one child entities) and complete (i.e., every entity in the parent entity set is also a member of one of the children). Then we need not create a separate parent table. In this case, every child table has all the attributes of the parent entity set. For exampl e, Figure 4 can be represented by two tables Savings -account = {Account-number, Balance, Interest-rate} and Checking-account = {Account-number, Balance, Overdraft-amount}.

- Aggregation:

Very straight forward. For example, The relationship set R2 in Figure 7 can be represented by a table R2 that consists of columns {the primary key of R1}$\cup${the primary key of E3}$\cup${the attributes of R2}.