

[Home Page](#)

[Title Page](#)



Page 1 of 43

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

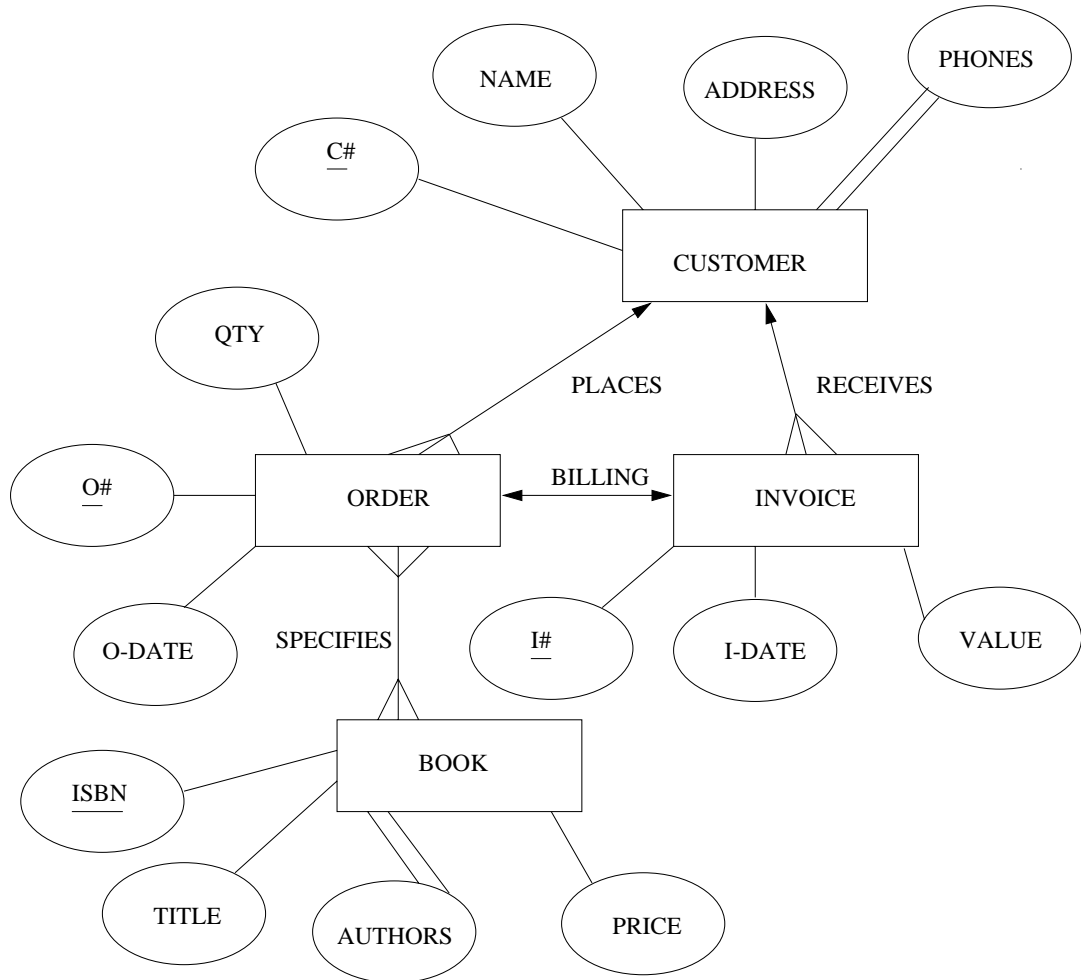
The Entity Relationship Model

- The building blocks of an Entity-Relationship Diagram (ERD)
- Cyclic relationships
- Weak entity types
- An informal algorithm for constructing an ERD

The Components of the ER Model [Che76]

- Data structures -
⇒ entities and relationships.
- Integrity constraints -
⇒ primary keys for entities and relationships, and
⇒ cardinality constraints for relationships.
- The ER model is only a partial data model, since it has no standard manipulative part.

Example Entity Relationship Diagram



Home Page

Title Page

◀ ▶

◀ ▶

Page 3 of 43

Go Back

Full Screen

Close

Quit

Fundamental Concepts

- *aggregation* -
 - ⇒ a collection of **attributes** forms an **entity type** (or entity set).
 - ⇒ two entity types form a **relationship** type.
- **generalisation** (specialisation, ISA) -
 - ⇒ E.g. Employee ISA Person and Student ISA a Person.

Next

Example Entity Sets

LECTURER

\$MARK	\$SANDY
\$JOHN	\$PAUL
\$PETER	\$CHRIS

CUSTOMER

\$C1	\$C2
\$C3	\$C4

STUDENT

\$JAMES	\$PAULB
\$DAN	\$MARKB

ORDER

\$01	\$04
\$02	\$05
\$03	\$06

SQUASH-PLAYERS

\$SANDY	\$MARKB
\$DAN	\$PAUL

INVOICE

\$I1	\$I2
\$I3	\$I4
\$I5	

BOOK

\$B1	\$B2	\$B3
\$B4	\$B5	\$B6
\$B7	\$B8	\$B9

Home Page

Title Page

◀▶

◀▶

Page 5 of 43

Go Back

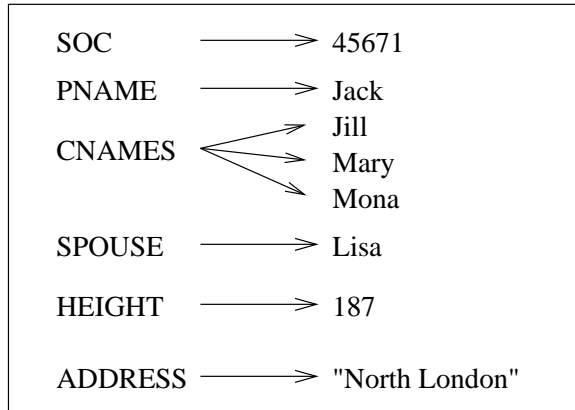
Full Screen

Close

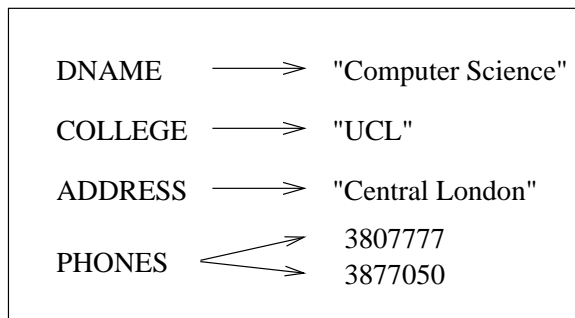
Quit

Example Entities

\$P1



\$D2



Home Page

Title Page

◀ ▶

◀ ▶

Page 6 of 43

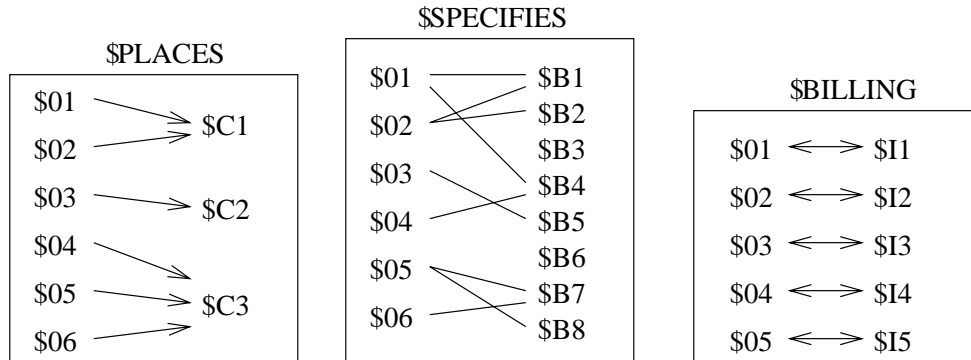
Go Back

Full Screen

Close

Quit

Example Relationships



[Home Page](#)

[Title Page](#)

[◀](#) [▶](#)

[◀](#) [▶](#)

Page 8 of 43

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

The Main Advantages of ERDs

- They are relatively simple
- They are user friendly
- They can provide a unified view of data, which is independent of any data model.

Graphs

A graph is an ordered pair (N, E) where

- N is a finite set of **nodes**, and
- E is a finite set of **edges**.

Each edge $e = \{u, v\}$ in E is an unordered pair of nodes in N .

★ Nodes and edges can be **labeled**.

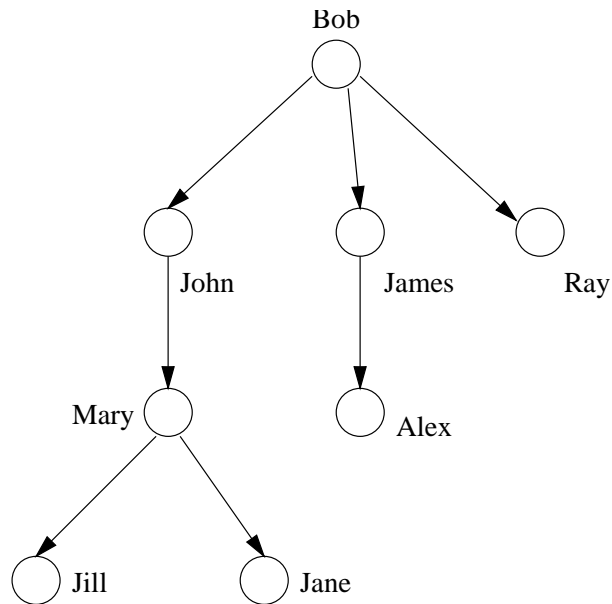
★ In a **directed graph** (or digraph) E is a set of **arcs**.

Each arc $e = (u, v)$ in E is an ordered pair of nodes in N .

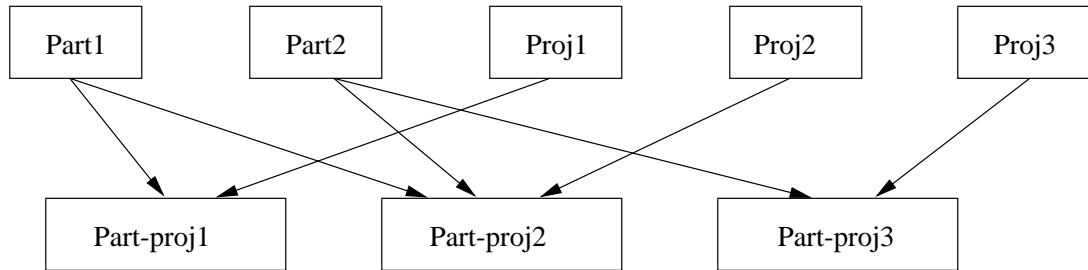
- Graphs and digraphs can either be **cyclic**, **hierarchical**, or **acyclic**.

Next

Hierarchy



Acyclic Directed Graph



[Home Page](#)

[Title Page](#)



Page 12 of 43

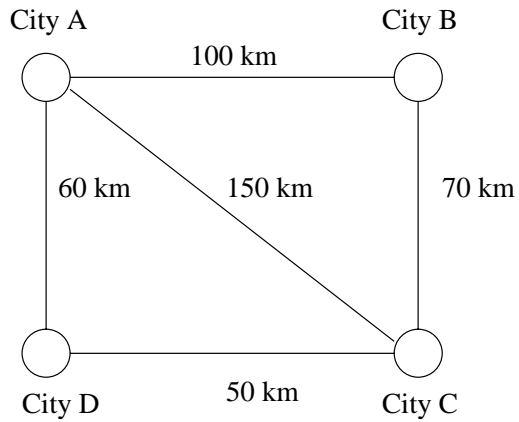
[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

Cyclic Undirected Graph



[Home Page](#)

[Title Page](#)

[◀](#) [▶](#)

[◀](#) [▶](#)

Page 13 of 43

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

The Building Blocks of an ERD

- **Entities** and **entity sets** (entity types).
- (Binary) **relationships** and (binary) **relationship types**.
- Domains, attribute values and attributes.

[Home Page](#)[Title Page](#)[◀](#) [▶](#)[◀](#) [▶](#)[Page 14 of 43](#)[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)

Entities

Definition. An *entity* is a “thing” that exists and can uniquely be identified.

Definition. An *entity type* (or *entity set*) is a collection of similar entities.

- An entity type consisting of a finite set of entities can be depicted by a **graph** having *no* edges.

[Home Page](#)[Title Page](#)[◀▶](#)[◀▶](#)[Page 15 of 43](#)[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)

Relationships and their Functionality

Definition. A *(binary) relationship type* is an association between two entity types.

- There may be more than one relationship type between two entity types.

E.g. Tutors and Teaches between Lecturer and Student.

[Home Page](#)

[Title Page](#)



Page 16 of 43

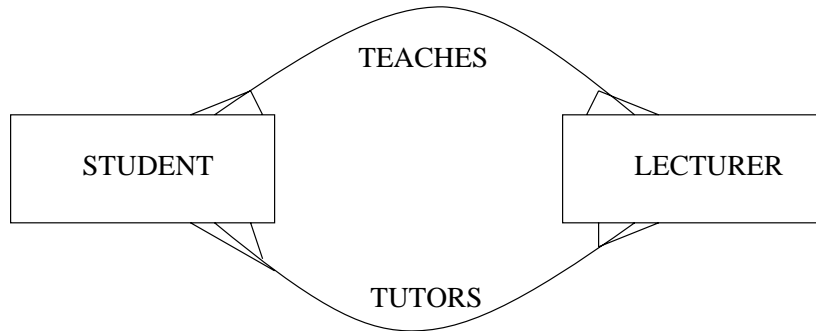
[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

Example of M to M relationship Types



[Home Page](#)

[Title Page](#)



Page 17 of 43

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

Example of M to One relationship Types



[Home Page](#)

[Title Page](#)



Page 18 of 43

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

Example of One to One relationship Types



Definition. A *relationship* is an instance of a relationship type.

⇒ In mathematical terms a relationship is a finite *binary relation*.

- A relationship can be depicted by a **bipartite** graph between entity sets.

Example relationships

Next

[Home Page](#)

[Title Page](#)

[◀◀](#) [▶▶](#)

[◀](#) [▶](#)

Page 20 of 43

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

Classification of Relationships

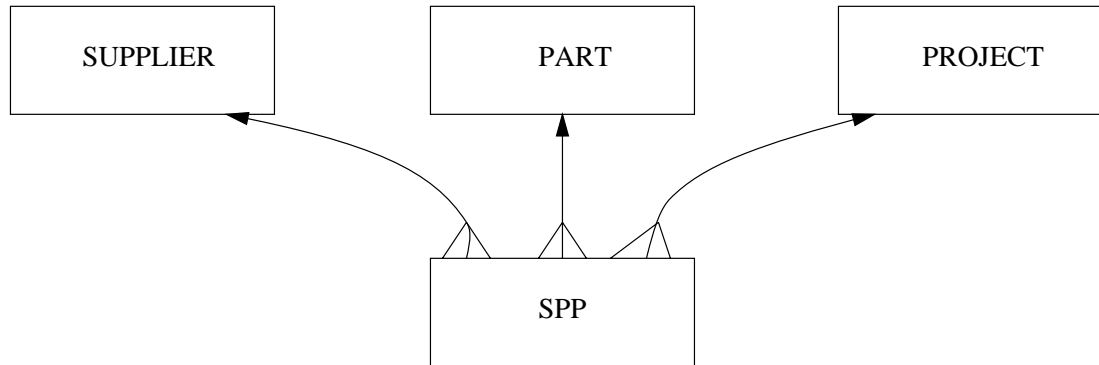
- **optional** relationship -
An Employee may or may not be assigned to a Department
- **mandatory** relationship -
Every Course must be Taught by at least one Lecturer.

Classification of Relationships - Continued

- **many-to-one** (or **one-to-many**) -
An Employee Works in one Department or a Department has many Employees.
- **one-to-one** -
A Manager Heads one Department and vice versa.
- **many-to-many** -
A Teacher Teaches many Students and a Student is Taught by many Teachers.

Example relationships

Binary Versus General Relationships



Attributes and Domains

Definition of attribute. *Attribute names* (or simply attributes) are properties of entity types.

- Attributes can be **single-valued** (e.g. pname and dname),
or
- **multi-valued** (e.g. cnames and phones).

See ERD

[Home Page](#)[Title Page](#)[◀](#) [▶](#)[◀](#) [▶](#)[Page 24 of 43](#)[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)

Definition of domain. The *domain* of an attribute of an entity type is the set of constant values associated with that attribute.

- Domains can be **atomic** such as the domain of integers and the domain of strings.
- **set-valued** such as the domain of finite sets of integers or of finite sets of strings.

Home Page

Title Page

◀ ▶

◀ ▶

Page 25 of 43

Go Back

Full Screen

Close

Quit

Definition of attribute value. An attribute, say att of an entity type E associates a *value* from its domain with each entity e of E .

This value is denoted by $att(e)$.

- The attribute values of entities can be depicted by a **bipartite graph** from attributes to their values.

Keys and Superkeys

Definition of a superkey. A set of attributes of an entity type is a *superkey* if for each entity, say e , over that type, the set of attribute values of the attributes in the superkey *uniquely* identify e .

Definition of a key. A *key* for an entity type is a superkey which is *minimal*.

- **simple** keys are single attribute keys, such as $E\#$ and $SOC\#$.
- **composite** keys have more than one attribute, such as $\{Dname, College\}$ and $\{Pname, Address\}$.

[Home Page](#)[Title Page](#)[◀◀](#) [▶▶](#)[◀](#) [▶](#)[Page 27 of 43](#)[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)

Definition of primary key of an entity type. A *primary key* is a key, which is designated by the database designer.

- The primary key guarantees logical access to every entity.

Definition of primary key of a relationship type \mathbf{R} .

- \mathbf{R} is a relationship type between E_1 with primary key K_1 and E_2 with primary key K_2 .
 1. If \mathbf{R} is many-to-many the primary key of \mathbf{R} is $K_1 \cup K_2$; (see **many to many relationship types**).
 2. If \mathbf{R} is many-to-one the primary key of \mathbf{R} is K_1 ; (see **many to one relationship types**).
 3. If \mathbf{R} is one-to-many the primary key of \mathbf{R} is K_2 ; (see **one to many relationship types**).
 4. If \mathbf{R} is one-to-one the primary key of \mathbf{R} is either K_1 or K_2 ; (see **one to one relationships**).

Cyclic Relationships

Definition. A **cyclic relationship** type (also called *recursive*) is a relationship type between two occurrences of the same entity type.

- Marriage between Person and itself.
- Parent-Child between Person and itself.
- Part-Sub-Part between Part and itself.

★ With each entity type in a cyclic relationship type we associate a *role*.

- We add the **roles** to the primary key attributes to form the primary key of the relationship

E.g. {Husband_Soc#, Wife_Soc#} is the primary key of Marriage assuming Soc# is the primary key of Person.

[Next](#)

[Home Page](#)

[Title Page](#)



Page 30 of 43

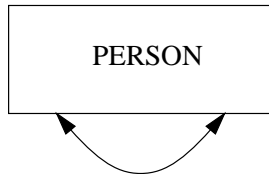
[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

Example of a Cyclic Relationship Type



[Home Page](#)

[Title Page](#)



Page 31 of 43

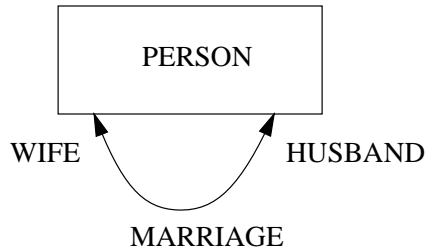
[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

Example of Roles in a Cyclic Relationship Type



Weak Entity Types

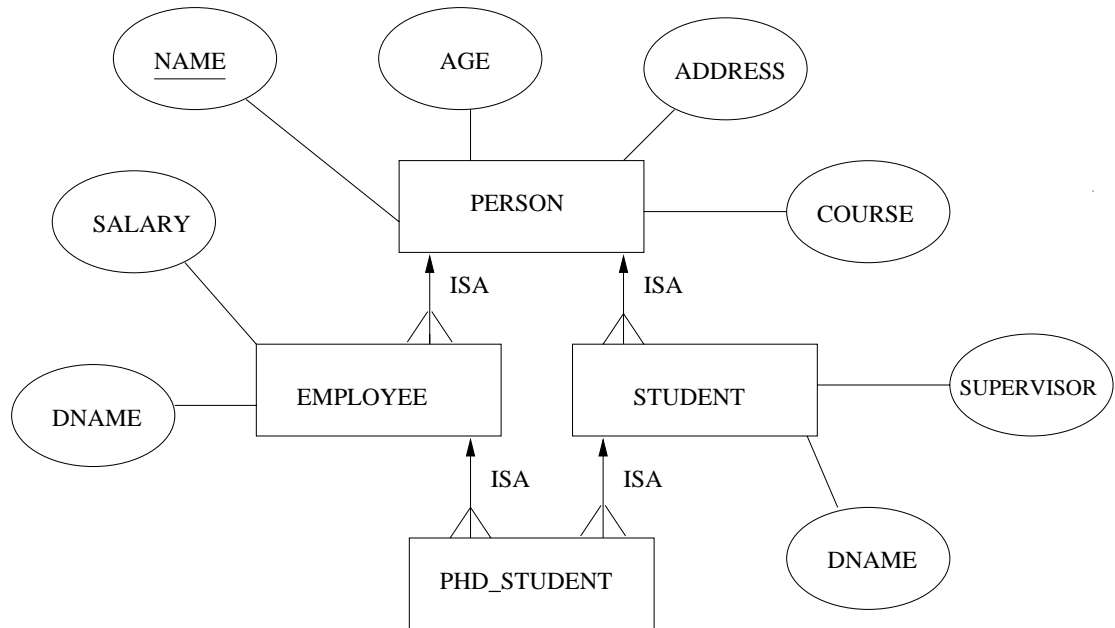
- ID relationship type -

In an employees database Child entities *exist* only if their corresponding Parent employee entity exists.

- ISA relationship type -

An Employee is a special case of Person and therefore the *existence* of an Employee entity implies the existence of a corresponding Person entity.

ISA Relationships



[Home Page](#)

[Title Page](#)



Page 34 of 43

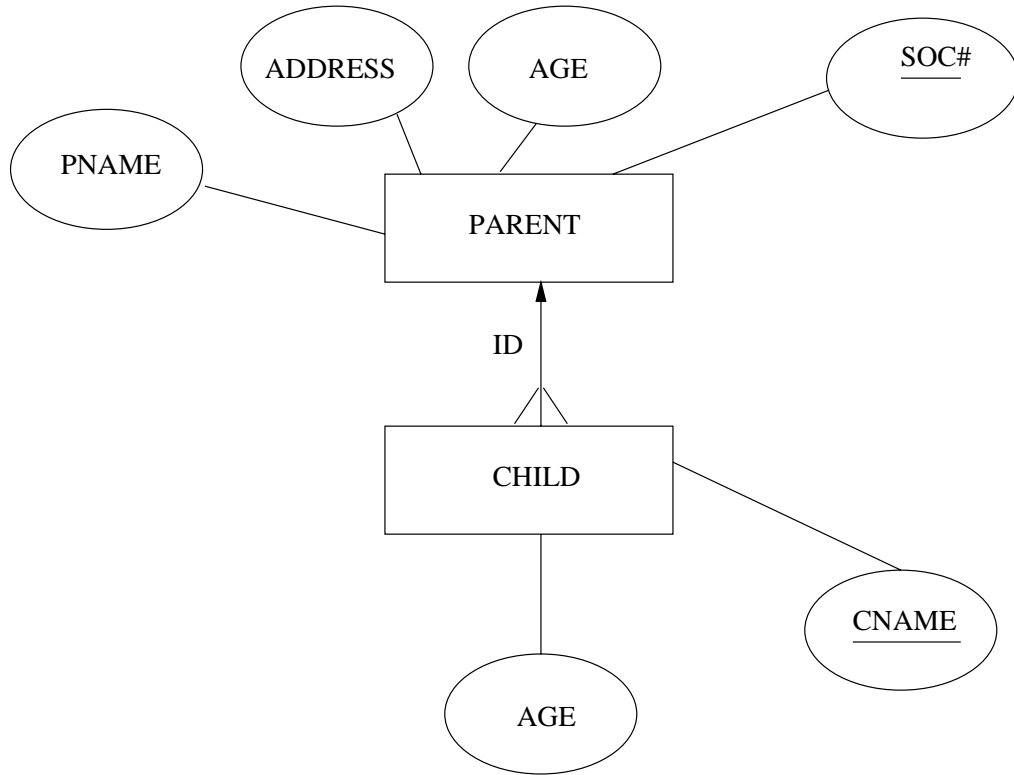
[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

ID Relationships



Let E_1 and E_2 be entity types.

Definition of ID relationship type. E_1 ID E_2 if the primary key of E_1 is composed of the primary key of E_2 plus one or more attributes of E_1 .

Definition of ISA relationship type. E_1 ISA E_2 if the primary key of E_1 is also the primary key of E_2 .

In addition, if in the database I_1 is the set of instances of E_1 and I_2 is the set of instances of E_2 , then I_1 is a subset of I_2 .

- If Employee ISA Person then Employee *inherits* all the attributes of Person.

The Universal Relation Schema Assumption (URSA)

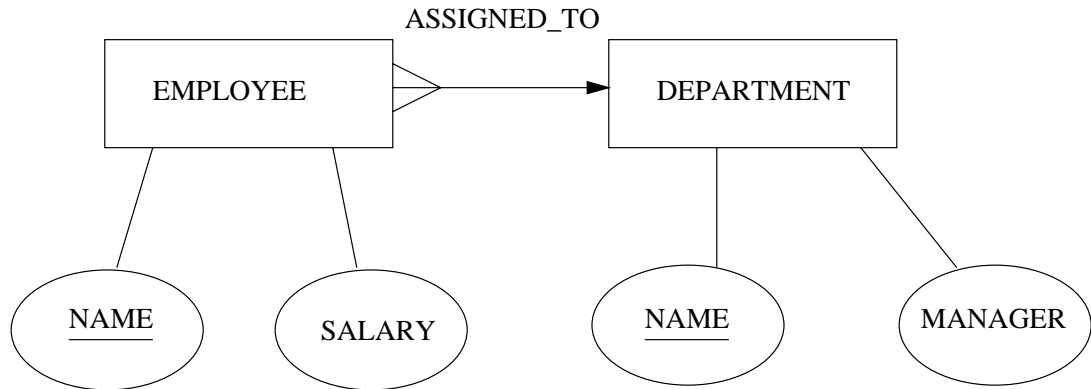
Definition. Each attribute of an entity type plays a unique role in the ERD.

I.e. all occurrences of attributes in an ERD have the same meaning.

E.g. Name can be department name or Person name and therefore needs to be modified to Dname and Pname.

⇒ The URSA is important in relational database design.





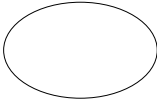

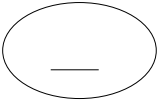
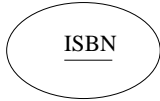
Example Illustrating the UR Problem



An Informal Algorithm for Constructing an ERD

1. Identify the entity types (including weak entity types) of the enterprise.
2. Draw some instances of the identified entity types.
3. Identify the relationships (including ISA and ID relationships) of the enterprise.
4. Classify each relationship identified in step 3 according to its functionality, i. e. if it is a one-to-one, many-to-one or many-to-many.
5. Draw some instances of the identified relationships.
6. Draw an ERD with the entity types and the relationships between them.
7. Identify the attributes of entity types and their underlying domain; if you are familiar with DD notation then give the DD definitions of the domains.
8. Identify a primary key for each entity type.
9. Draw some instances of attribute values of entities.
10. Add the attributes and keys to the ERD drawn in step 6.

Basic ERD Constructs

Basic ERD Constructs		
Concept	Representation	Example
Entity type		
Relationship type		
Attribute		
Primary key attribute		

[Home Page](#)

[Title Page](#)



Page 39 of 43

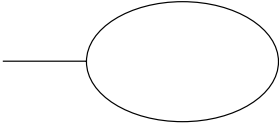
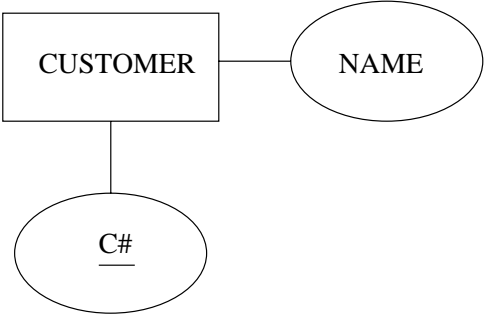
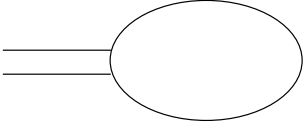
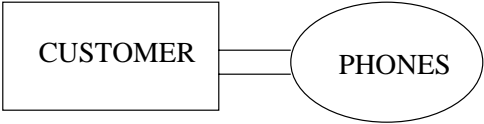
[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

Attribute Constructs

Attribute Constructs		
Concept	Representation	Example
Single-valued attribute		
Multi-valued attribute		

[Home Page](#)

[Title Page](#)

[◀](#) [▶](#)

[◀](#) [▶](#)

Page 40 of 43

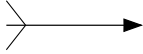
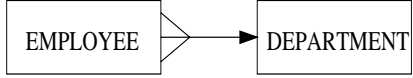
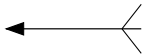
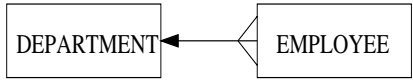
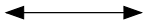
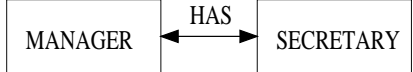
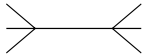
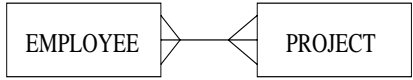
[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

Relationships Constructs

Basic ERD Constructs		
Concept	Representation	Example
Many-to-one		<p>ASSIGNED_TO</p> 
One-to-many		<p>EMPLOYS</p> 
One-to-one		<p>HAS</p> 
Many-to-many		<p>WORKS_FOR</p> 


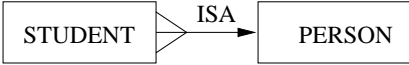
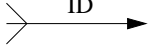
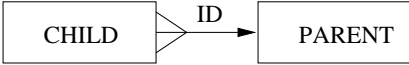
[Home Page](#)
[Title Page](#)



Page 41 of 43

[Go Back](#)
[Full Screen](#)
[Close](#)
[Quit](#)

Built-in Relationship Constructs

Built-in Relationship Types		
Concept	Representation	Example
ISA	 ISA →	
ID	 ID →	

References

- [Che76] P.P.-S. Chen. The entity-relationship model - toward a unified view of data. *ACM Transactions on Database Systems*, 1:9–36, 1976.

[Home Page](#)

[Title Page](#)



Page 43 of 43

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)