CSCC43H: Introduction to Databases

Lecture 2

Wael Aboulsaadat

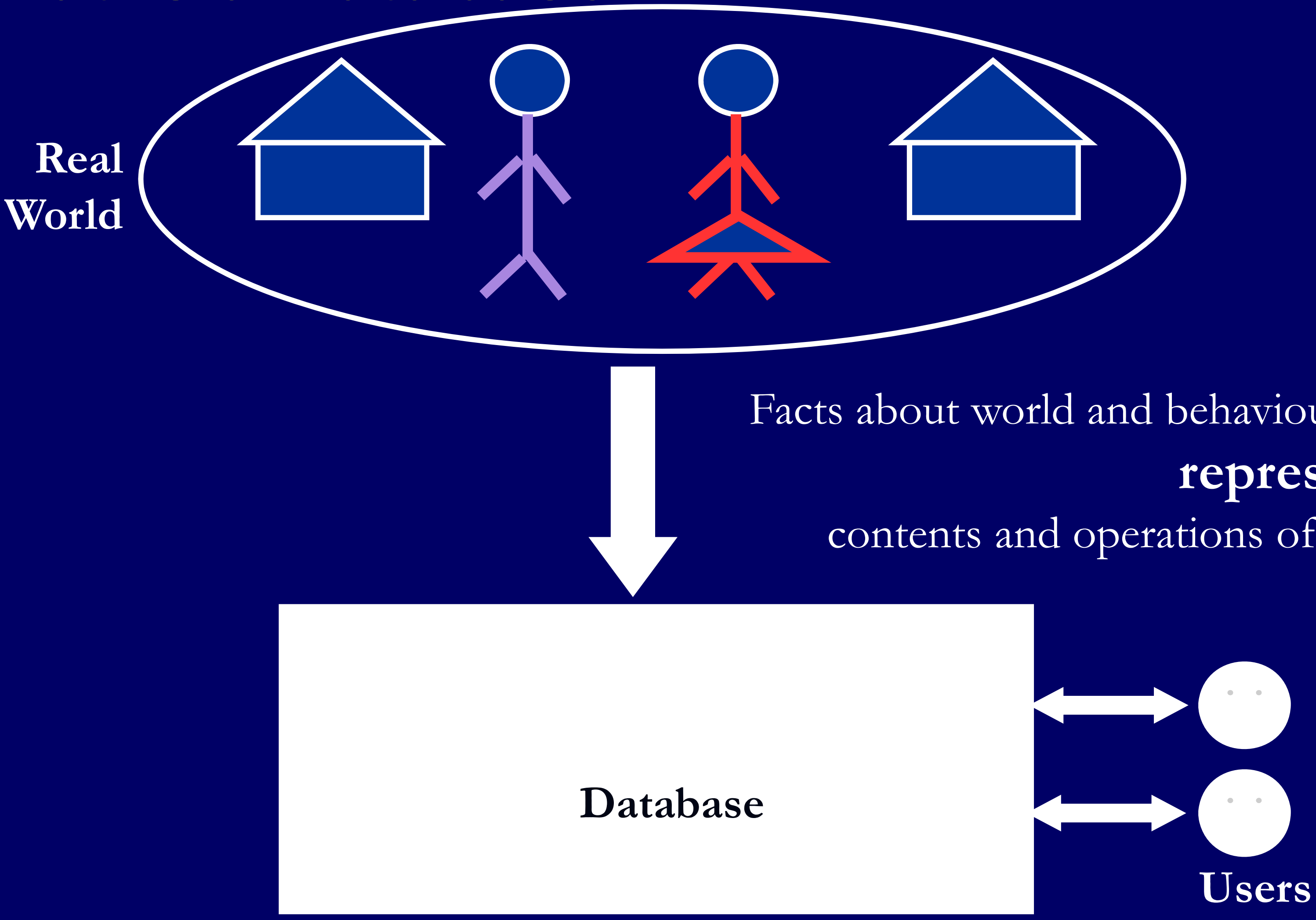
Acknowledgment: these slides are partially based on Prof. Garcia-Molina & Prof. Ullman slides accompanying the course’s textbook.

Database Management System (DBMS)

- A collection of programs that enable:
  - ➔Defining (describing the structure),
  - Populating by data (Constructing),
  - Manipulating (querying, updating),
  - Preserving consistency,
  - Protecting from misuse,
  - Recovering from failure, and
  - Concurrent using

  of a database.

What Is a Database?



Facts about world and behaviour of world **represented in** contents and operations of a database

Steps in Database Design

1. Requirements Analysis

2. Conceptual Design

3. Logical Design

4. Schema Refinement

5. Physical Design - indexes, disk layout

6. Security Design - who accesses what, and how

Steps in Database Design: conceptual design

A. Define ER Model

B. Translate ER Model to Relational Model

Entity Relation Model (ER)

- Entities
- Attributes
- Relations
- Roles

ER: entities

- A 'thing' is called an Entity
- An entity can be an actual physical object or a conceptual object
- And that's it!

ER: how to model entities?

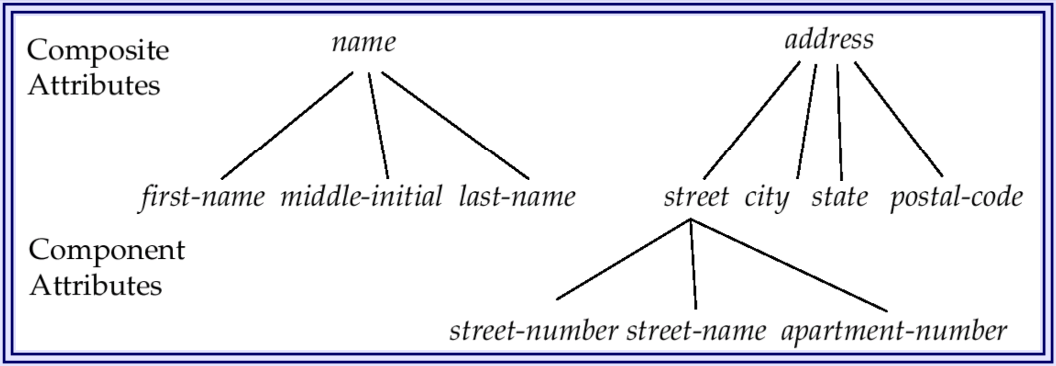
- An entity is an object that is distinguishable from other objects
  - E.g. a specific person, a course module, an event
- Note:
  - The fact that two people have the same name does not mean that they are indeed the same entity. They could just share the same attribute value

ER: attributes

- An entity is represented by a set of attributes, that is descriptive properties possessed by all members of an entity set.
  - Example:
    - *customer = (customer-id, customer-name, customer-street, customer-city)*
    - *loan = (loan-number, amount)*
- *Domain* – the set of permitted values for each attribute

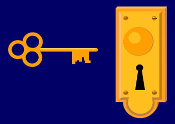
ER: attributes types

- Attribute types:
  - *Simple* and *composite* attributes (e.g., address).



  - *Single-valued* and *multi-valued* attributes
    - E.g. multi-valued attribute: *phone-numbers*
  - *Derived* attributes
    - Can be computed from other attributes
    - E.g. *age*, given the date of birth

ER: a special attribute – key



- How to distinguish between entities?
- A *key* of an entity is a set of one or more attributes whose values uniquely determine each entity.
- A Key can be *simple* (a single attribute) or *composite* (more than one field)

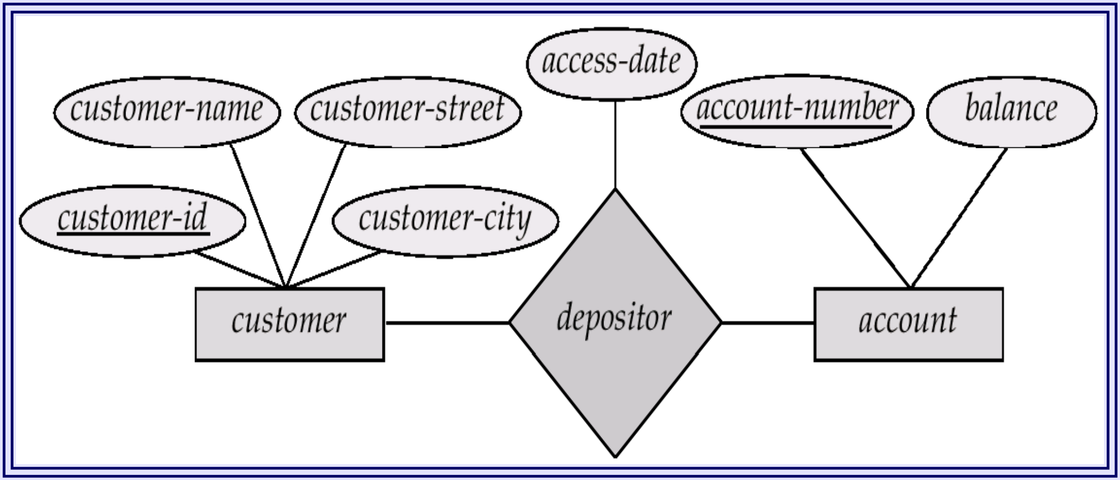
ER: relations

- Association among two or more entities.
  - E.g., John *works* in Pharmacy department.

- A relation can have it’s own attributes as well…

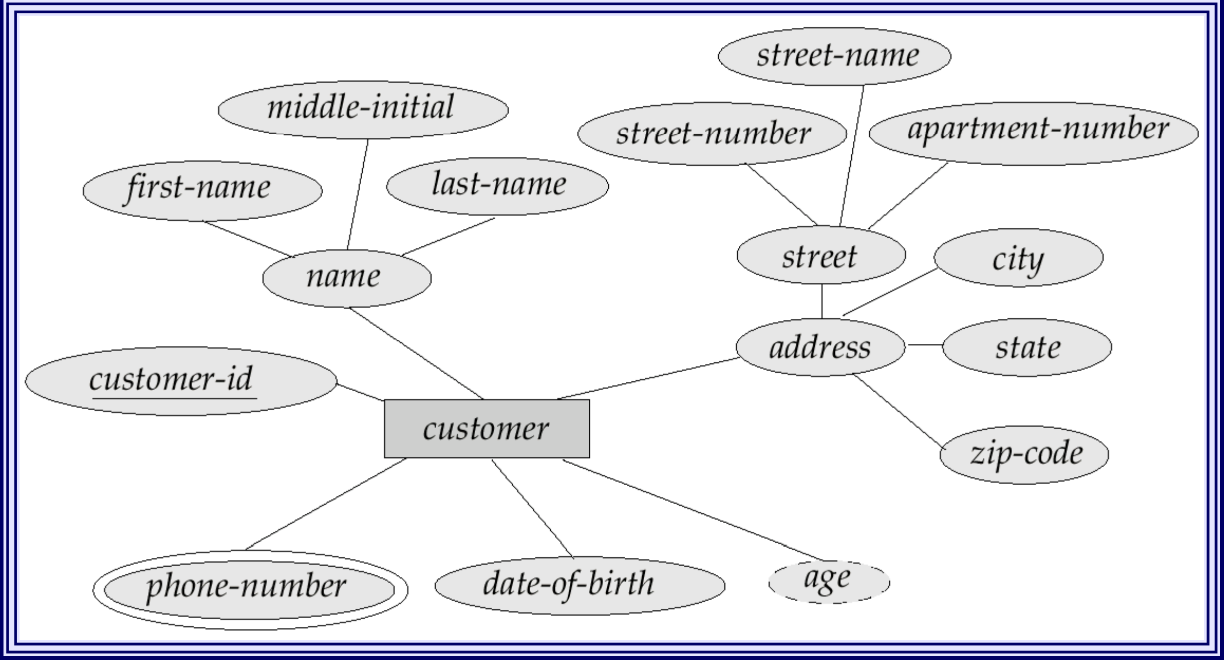
ER: visual notation

- Rectangles represent entity sets.
- Diamonds represent relationship sets.
- Lines link attributes to entity sets and entity sets to relationship sets.
- Ellipses represent attributes
- Underline for keys



ER: visual notation - cont’d

- Ellipses represent attributes
  - Double ellipses represent multi-valued attributes.
  - Dashed ellipses denote derived attributes.

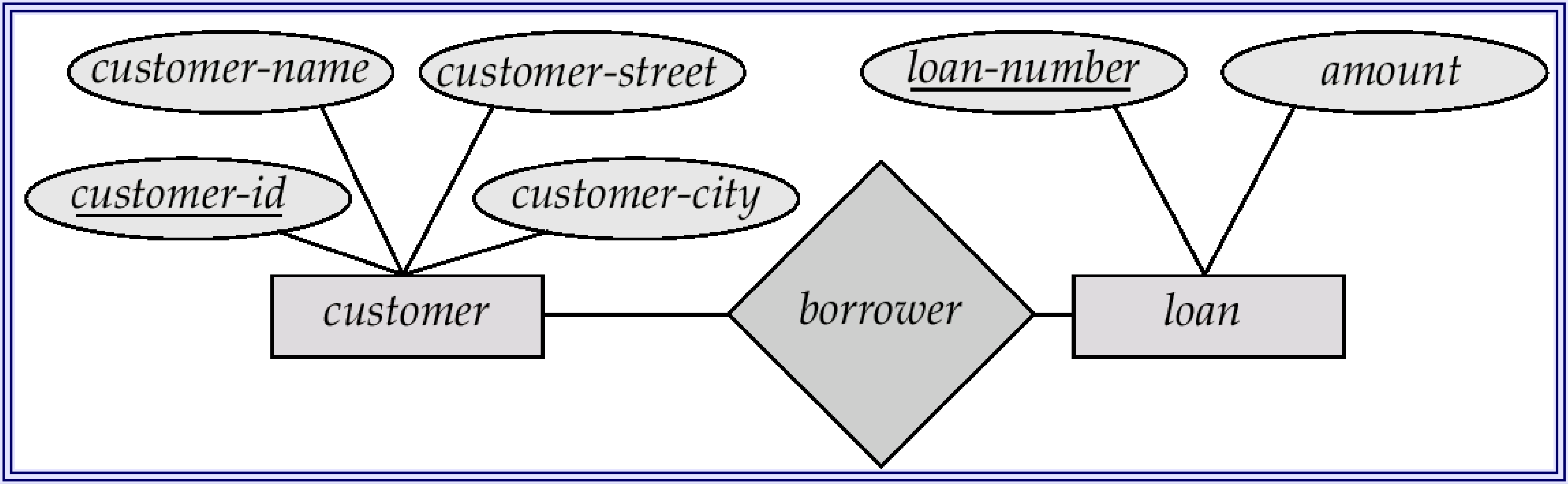


ER: cardinality constraints

- We express cardinality constraints by drawing either a directed line (→), signifying “one,” or an undirected line (—), signifying “many,” between the relationship and the entity.

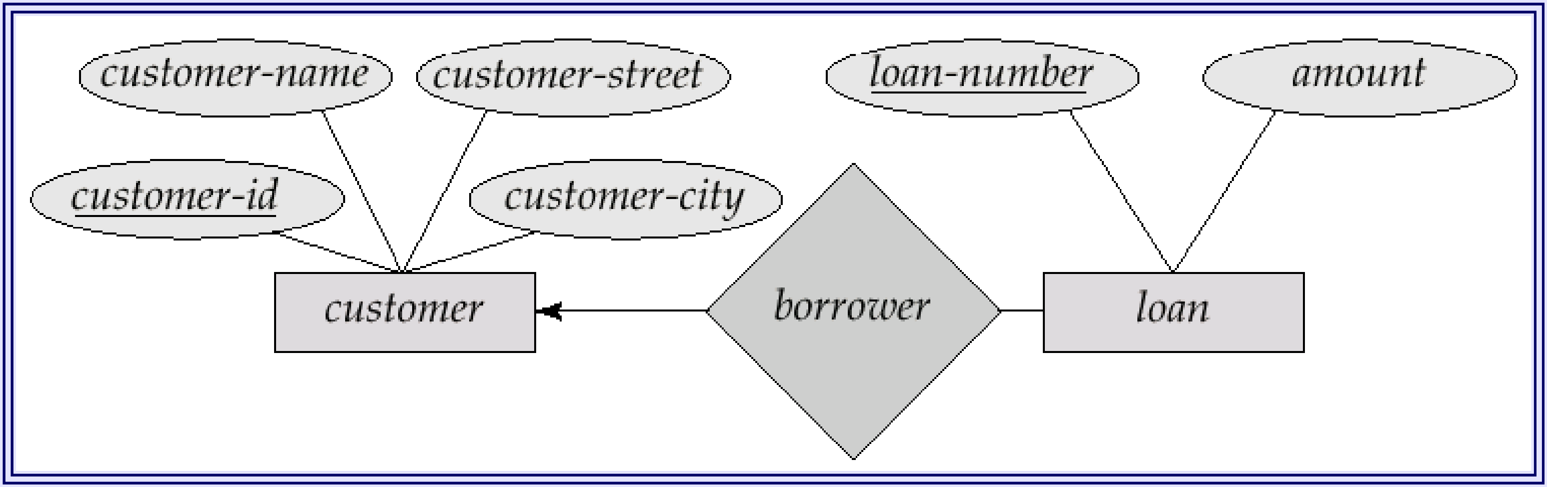
ER: cardinality constraints

- Many-to-many relationship
  - A customer is associated with several (possibly 0) loans via borrower
  - A loan is associated with several (possibly 0) customers via borrower



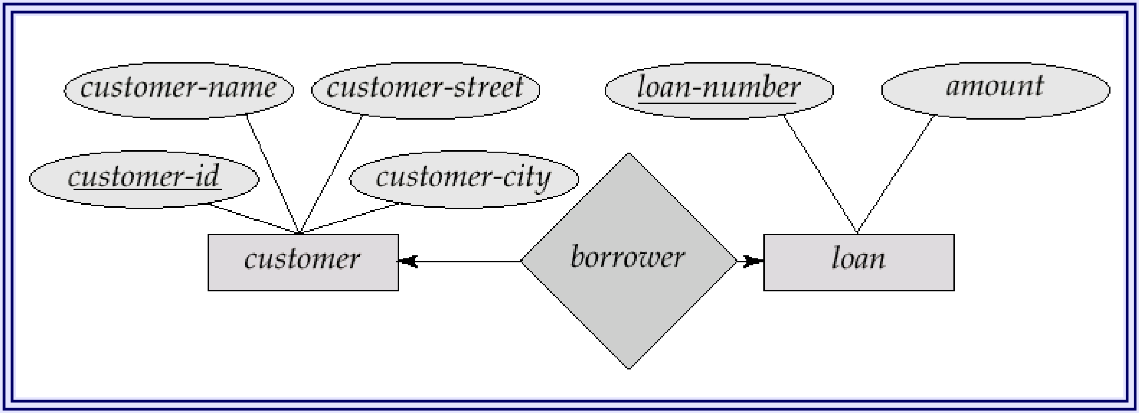
ER: cardinality constraints

- One-to-many relationship
  - a loan is associated with at most one customer via *borrower*, a customer is associated with several (including 0) loans via *borrower*



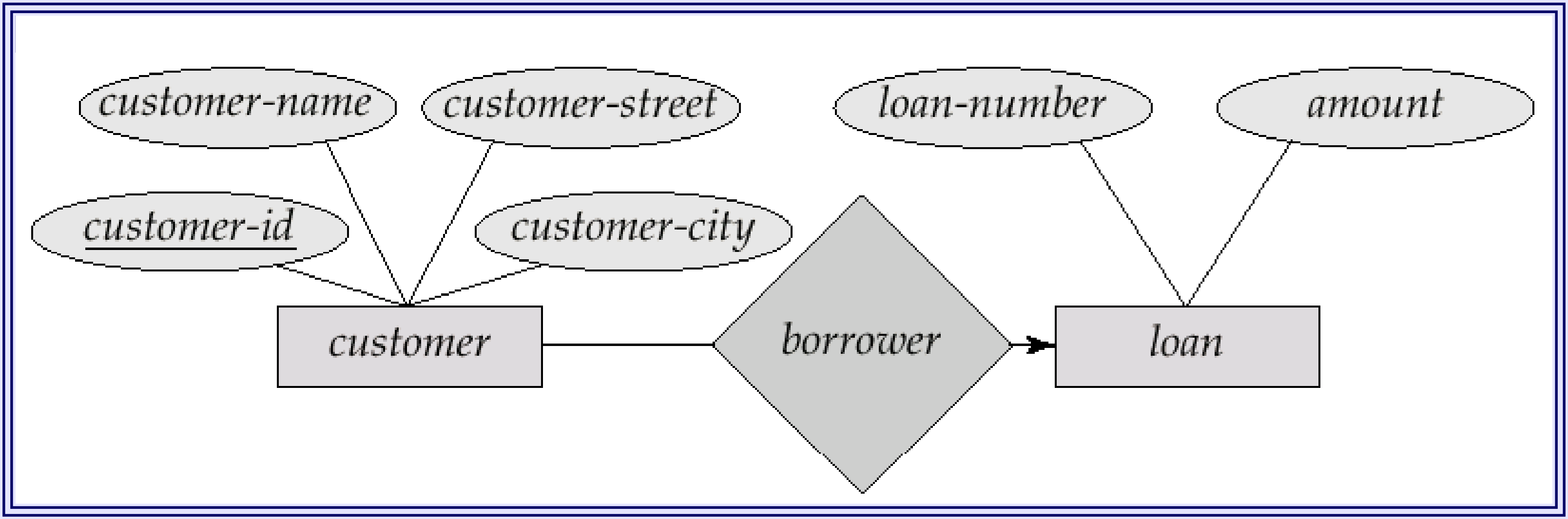
ER: cardinality constraints

- One-to-one relationship:
  - A customer is associated with at most one loan via the relationship *borrower*
  - A loan is associated with at most one customer via *borrower*



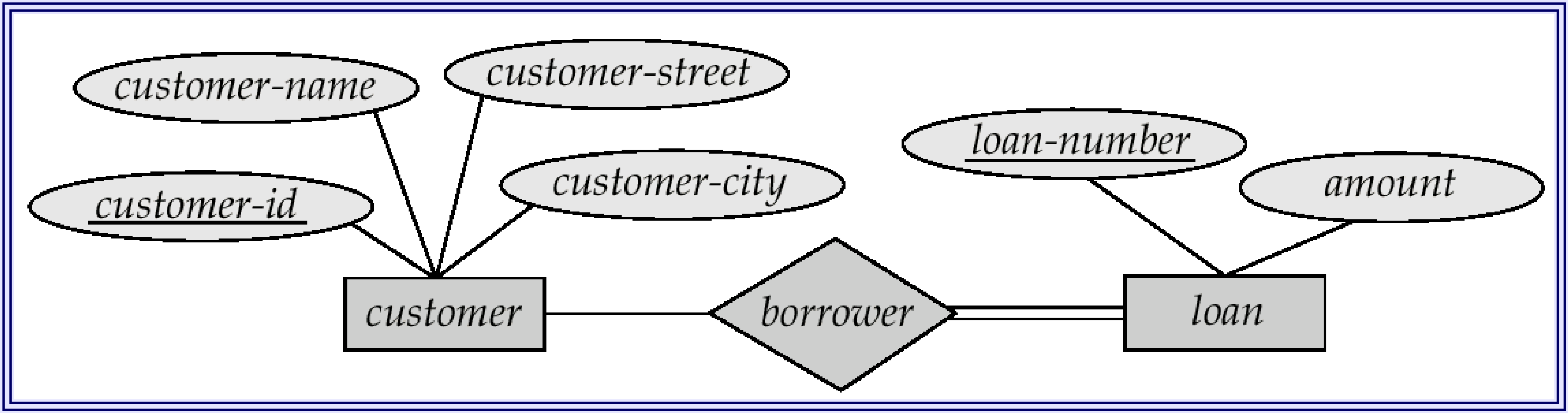
ER: cardinality constraints

- Many-to-one relationship
  - a loan is associated with several (including 0) customers via *borrower*, a customer is associated with at most one loan via *borrower*



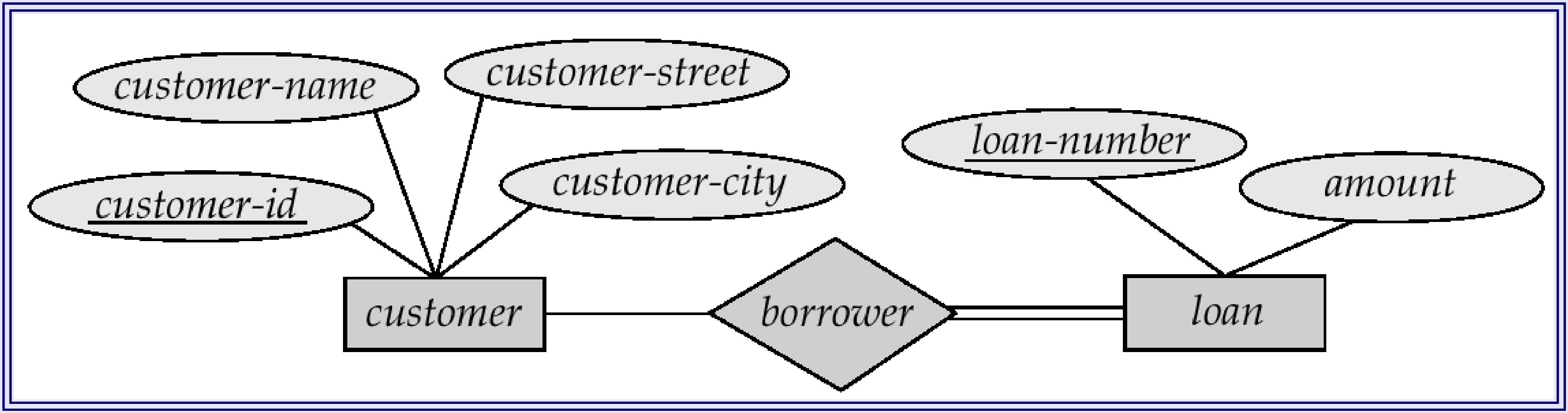
ER: Participation of an Entity Set in a Relationship Set

- Total participation (indicated by double line):
  - every entity in the entity set participates in at least one relationship in the relationship set
  - E.g. participation of *loan* in *borrower* is total
    - every loan must have a customer associated to it via borrower

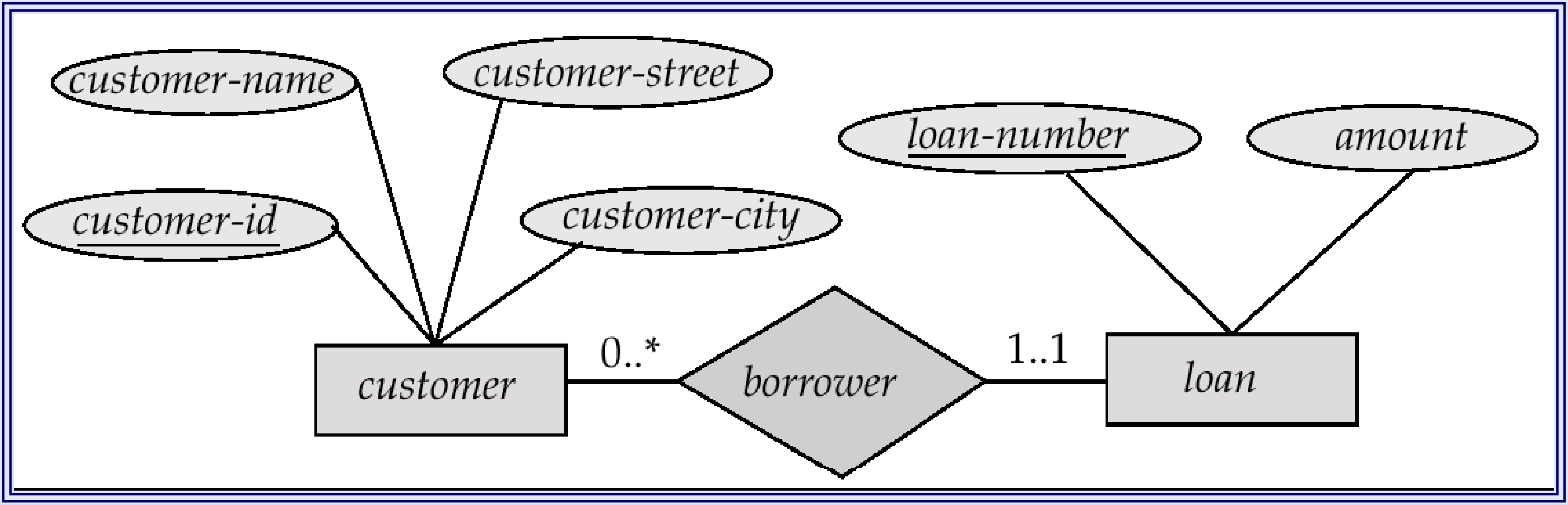


ER: Participation of an Entity Set in a Relationship Set

- Partial participation:
  - some entities may not participate in any relationship in the relationship set
  - E.g. participation of *customer* in *borrower* is partial

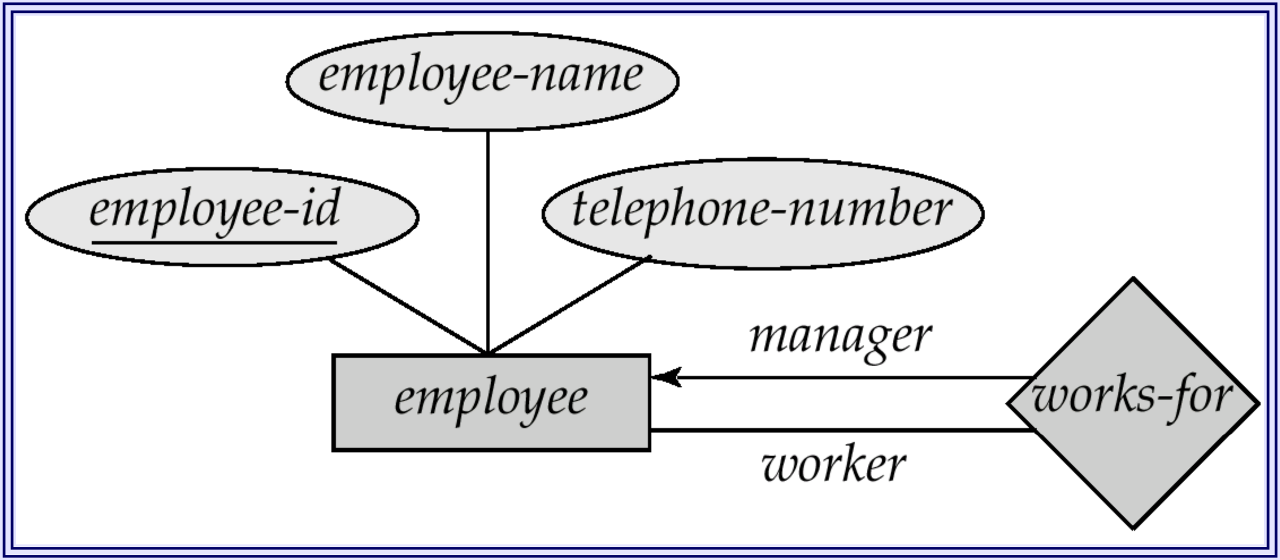


ER: alternative notation for cardinality limits



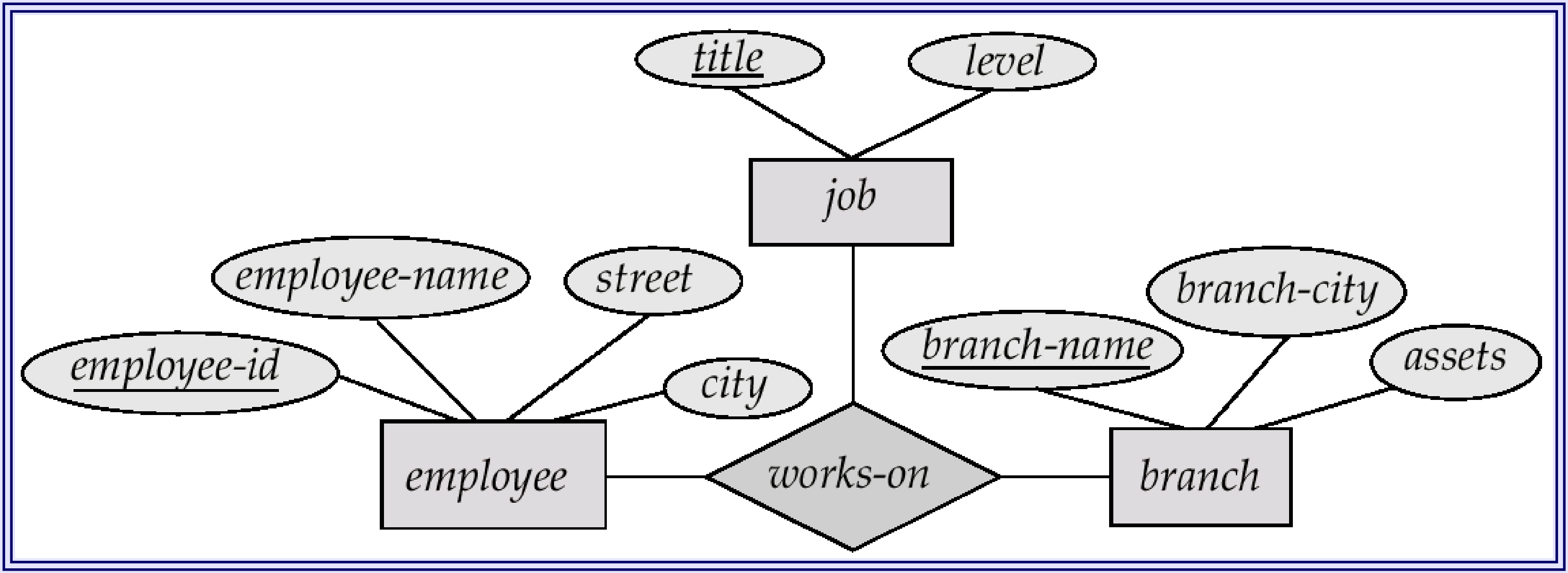
ER: roles

- Entity sets of a relationship need not be distinct
  - The labels “manager” and “worker” are called roles; they specify how employee entities interact via the works-for relationship set.
  - Roles are indicated in E-R diagrams by labeling the lines that connect diamonds to rectangles.
  - Role labels are optional, and are used to clarify semantics of the relationship



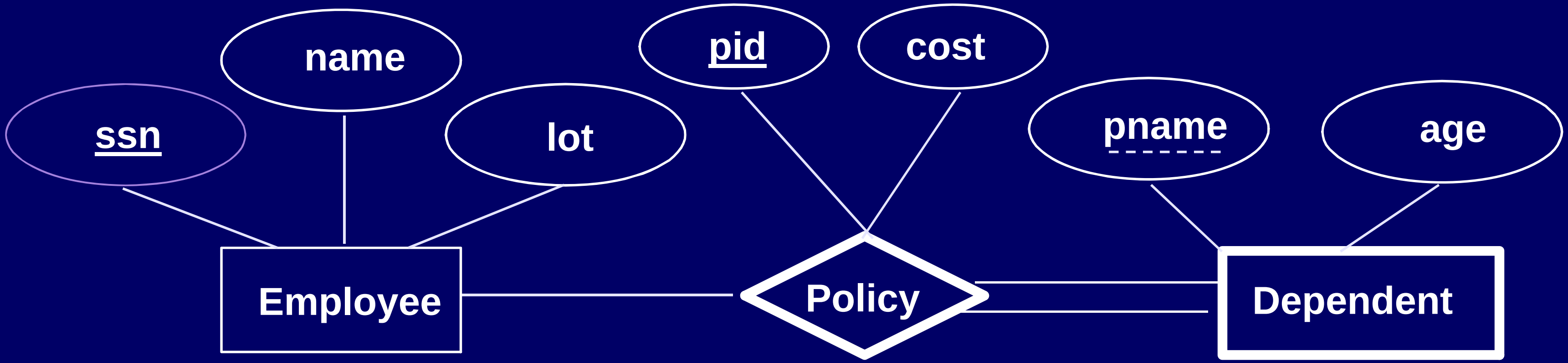
E-R: ternary Relationship

- Suppose employees of a bank may have jobs (responsibilities) at multiple branches, with different jobs at different branches. Then there is a ternary relationship set between entity sets *employee, job and branch*

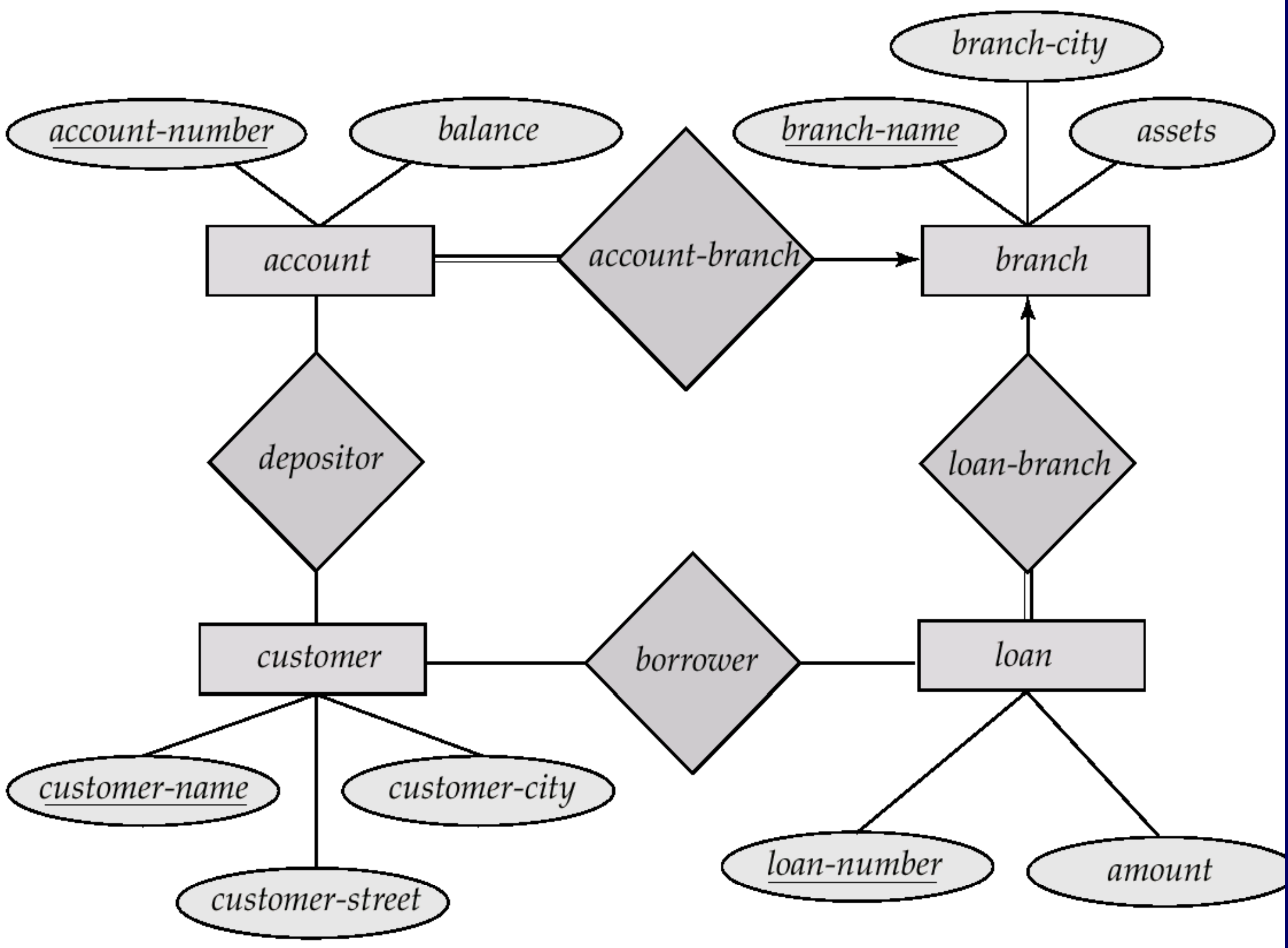


ER: weak entities

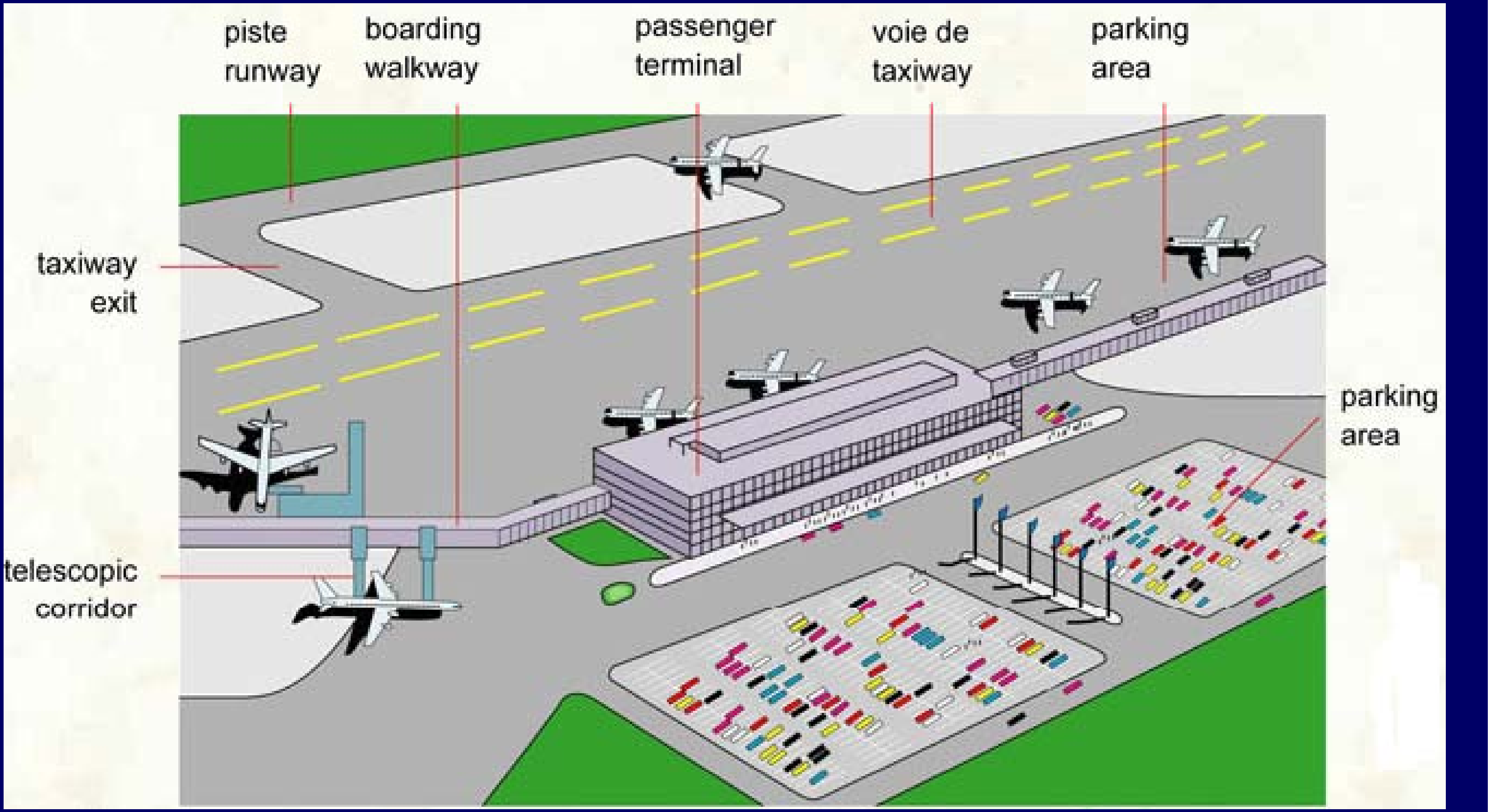
- A *weak entity* can be identified uniquely only by considering the primary key of another (*owner*) entity.
  - Owner entity set and weak entity set must participate in a one-to-many relationship set (one owner, many weak entities).
  - Weak entity set must have total participation in this *identifying* relationship set.
  - Weak entities have only a "partial key" (dashed underline)



Example


branch-city
account-number
balance
branch-name
assets
account
account-branch
branch
depositor
loan-branch
customer
borrower
loan
customer-name
customer-city
customer-street
loan-number
amount

Example


piste
runway
boarding
walkway
passenger
terminal
voie de
taxiway
parking
area
taxiway
exit
telescopic
corridor
parking
area

Pause

Steps in Database Design: conceptual design

A. Define ER Model

B. ➔Translate ER Model to Relational Model

The Relational Data Model (1970)

Lessons from the Codd paper

- Let’s separate physical implementation from logical
- Model the data independently from how it will be used (accessed, printed, etc.)
  - Describe the data minimally and mathematically
    - A relation describes an association between data items – tuples with attributes
    - We generally think of tables and rows, but that’s somewhat imprecise
  - Use standard mathematical (logical) operations over the data – these are the relational algebra or relational calculus

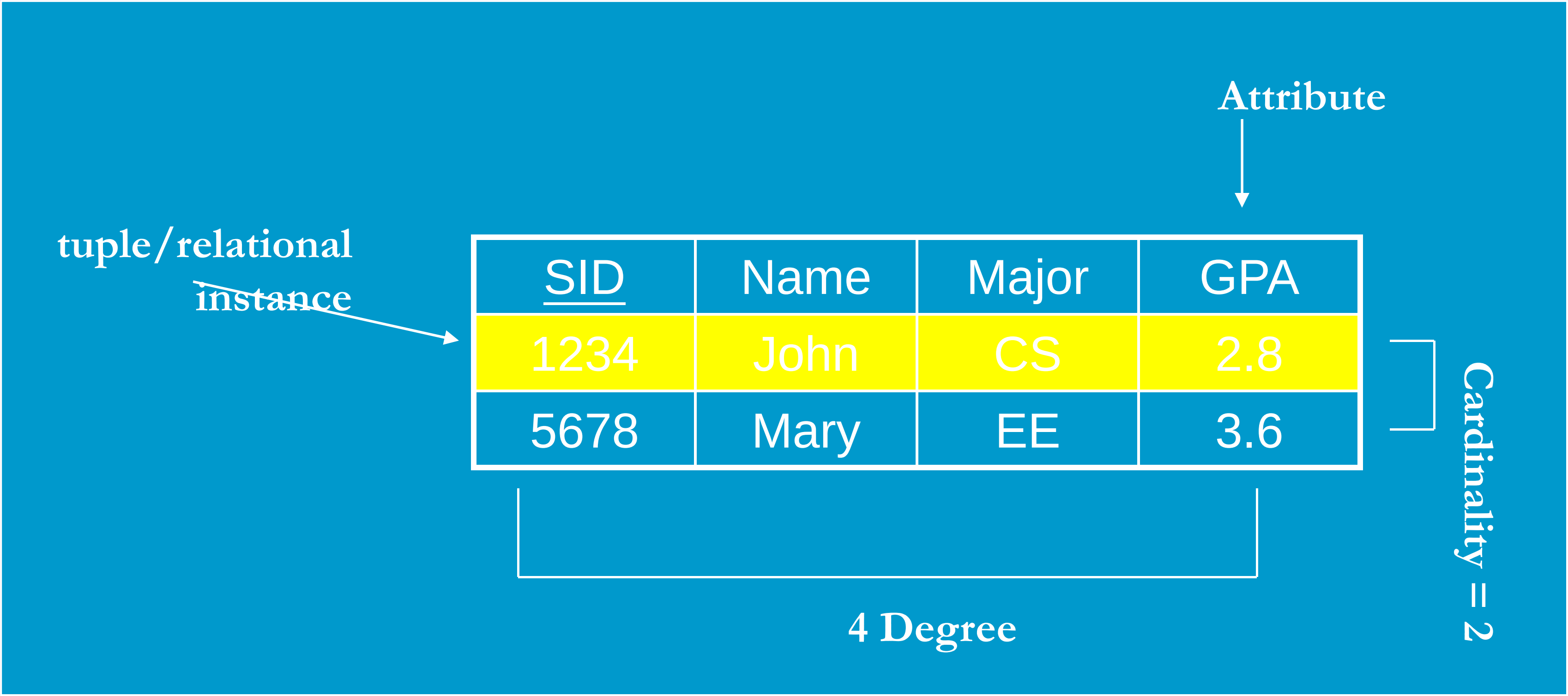
How does this model relate to objects, properties? What are its abilities and limitations?

Relational Model Concepts

Relational Model is made up of tables

A table	= an entity or relation
A row of table	= a relational instance/tuple
A column of table	= an attribute
Cardinality	= number of rows
Degree	= number of columns

Review - Example



Attribute

tuple/relational instance

SID	Name	Major	GPA
1234	John	CS	2.8
5678	Mary	EE	3.6

Cardinality = 2

4 Degree

A Relation

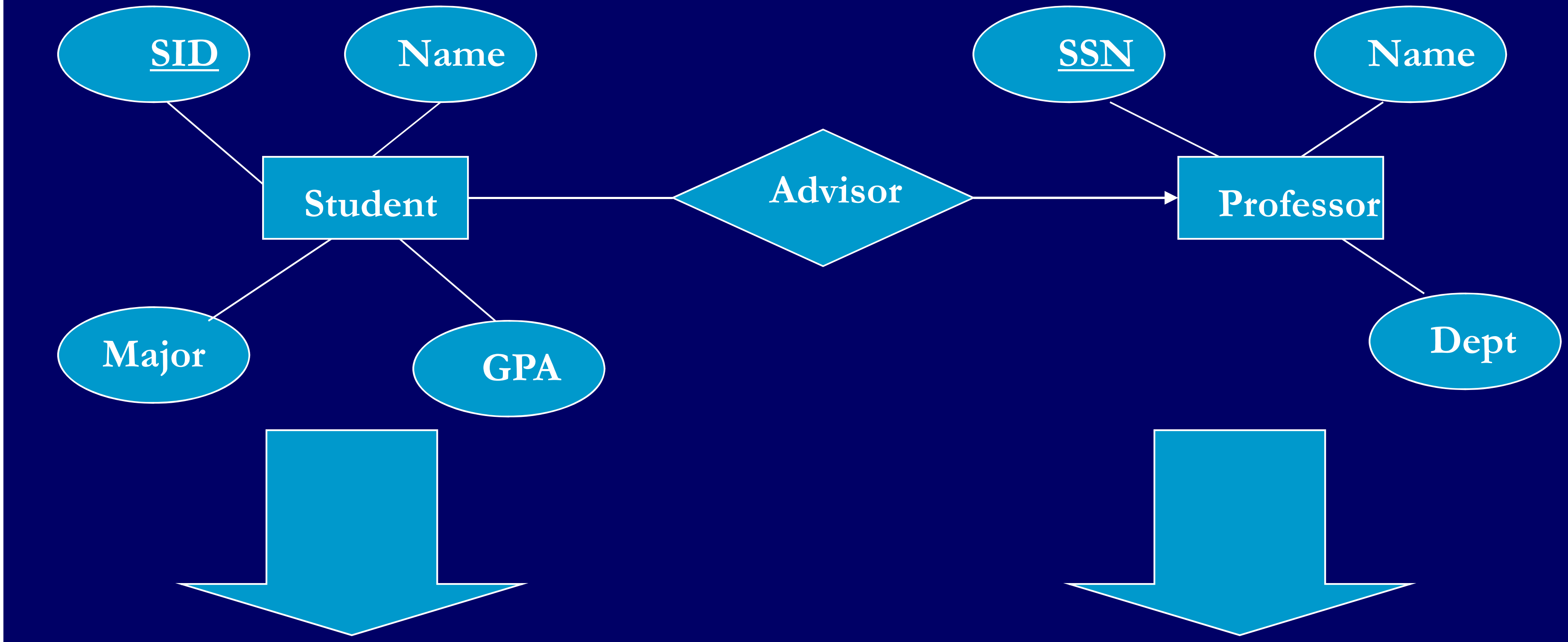
From ER Model to Relational Model

So… how do we convert an ER diagram into a table?? Simple!!

Basic Ideas:

- Build a table for each entity set
- Build a table for each relationship set if necessary (more on this later)
- Make a column in the table for each attribute in the entity set
- Underline Key

Example – Strong Entity Set



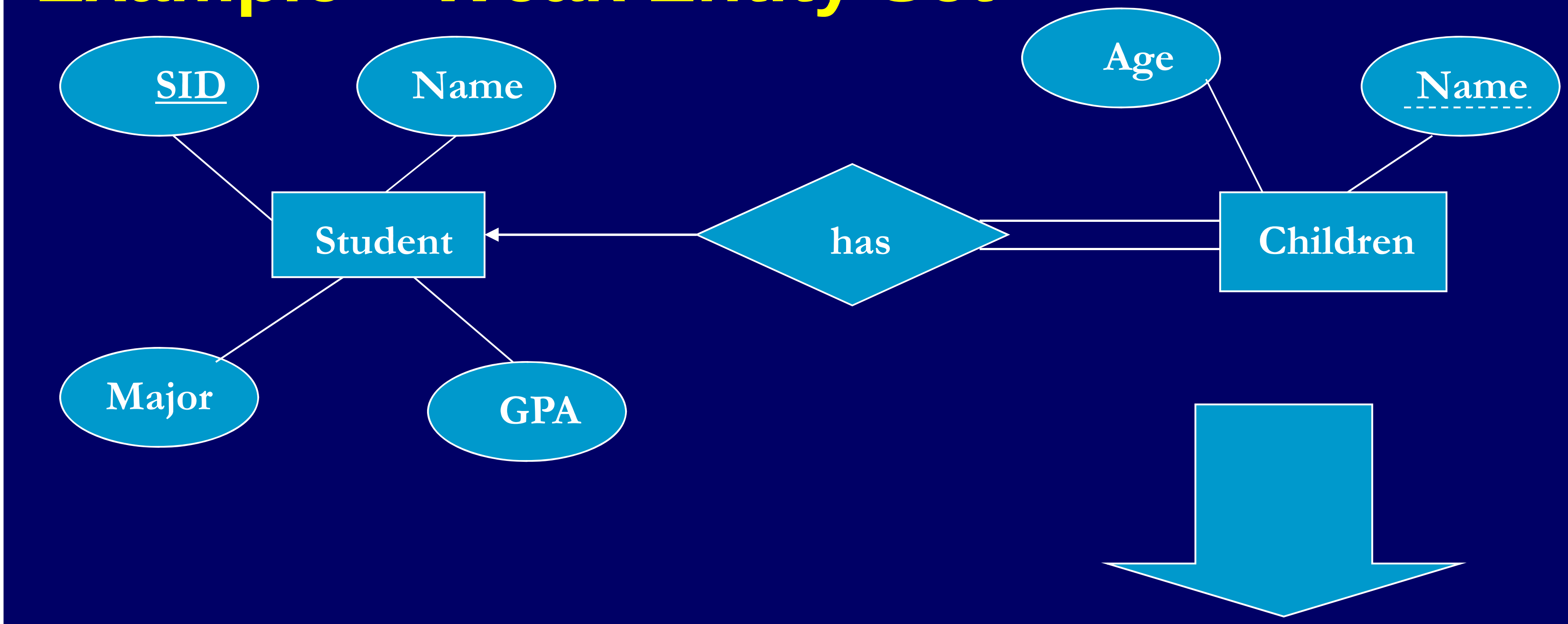
SID	Name	Major	GPA
1234	John	CS	2.8
5678	Mary	EE	3.6

SSN	Name	Dept
9999	Smith	Math
8888	Lee	CS

Representation of Weak Entity Set

- Weak Entity Set Cannot exists alone
- To build a table/schema for weak entity set
  - Construct a table with one column for each attribute in the weak entity set
  - Remember to include discriminator
  - Augment one extra column on the right side of the table, put in there the primary key of the Strong Entity Set (the entity set that the weak entity set is depending on)
  - Primary Key of the weak entity set = Discriminator + foreign key

Example – Weak Entity Set



Age	Name	SID
10	Bart	1234
8	Lisa	5678

* key of *Children* is *Parent_SID* + *Name*

Representation of Relationship Set

--This is a little more complicated--

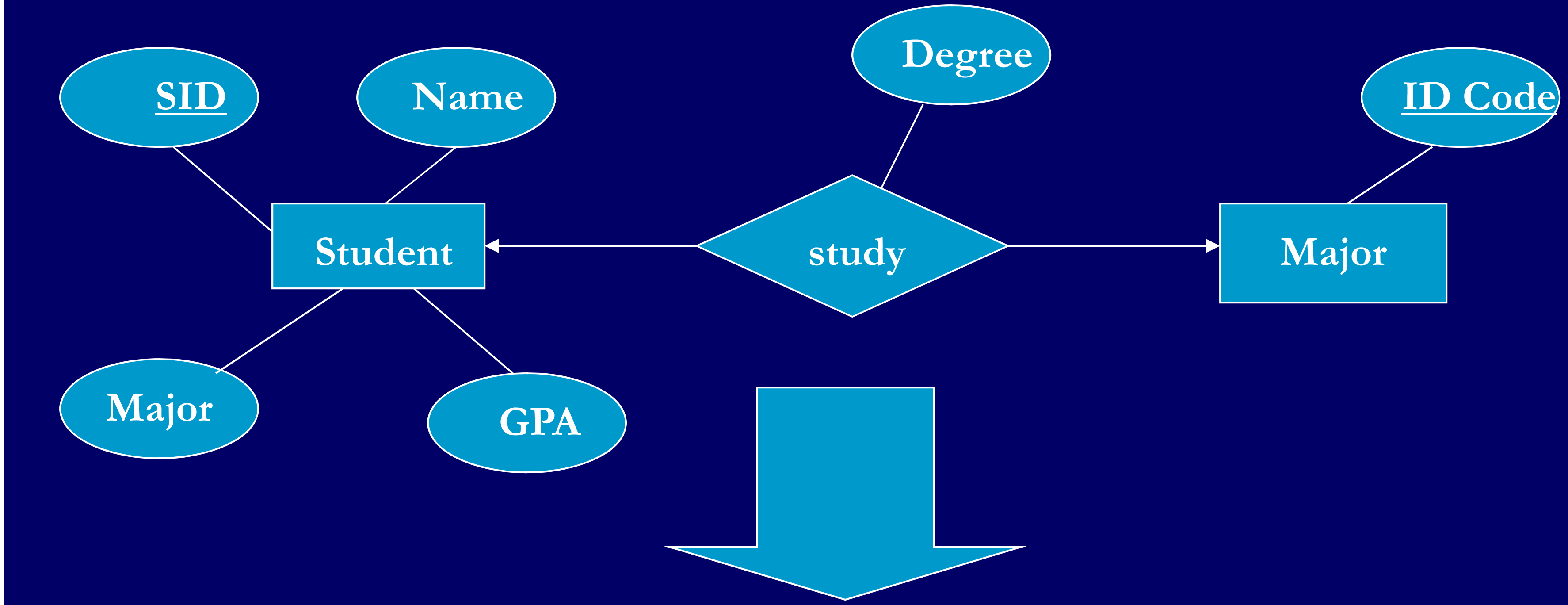
- ✓ Unary/Binary Relationship set
  - ➢ Depends on the cardinality and participation of the relationship
  - ➢ Two possible approaches
- ✓ N-ary (multiple) Relationship set
  - ➢ Primary Key Issue
- ✓ Identifying Relationship
  - ➢ No relational model representation necessary

Representing Relationship Set

Unary/Binary Relationship

- For one-to-one relationship w/out total participation
  - Build a table with two columns, one column for each participating entity set's primary key. Add successive columns, one for each descriptive attributes of the relationship set (if any).
- For one-to-one relationship with one entity set having total participation
  - Augment one extra column on the right side of the table of the entity set with total participation, put in there the primary key of the entity set without complete participation as per to the relationship.

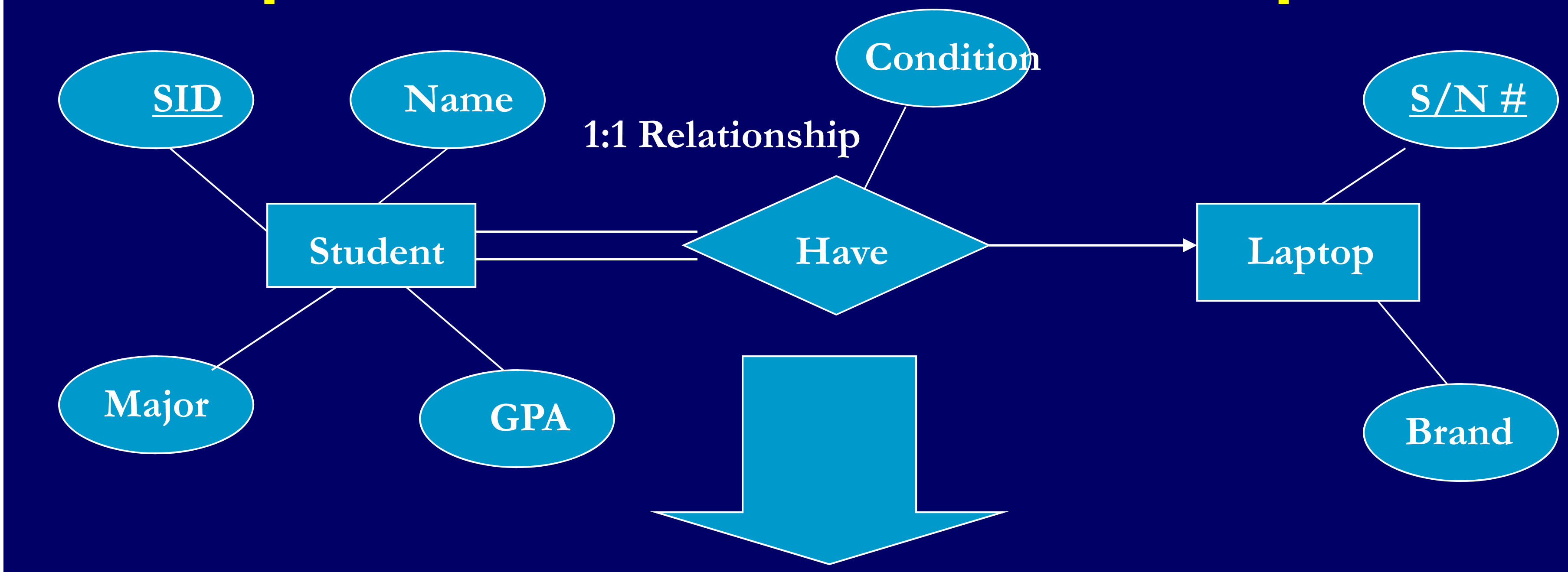
Example – One-to-One Relationship Set



SID	Maj_ID Co	S_Degree
9999	07	1234
8888	05	5678

* key can be either *SID* or Maj_*ID_ Co*

Example – One-to-One Relationship Set



<u>SID</u>	Name	Major	GPA	LP_S/N	Hav_Cond
9999	Bart	Economy	-4.0	123-456	Own
8888	Lisa	Physics	4.0	567-890	Loan

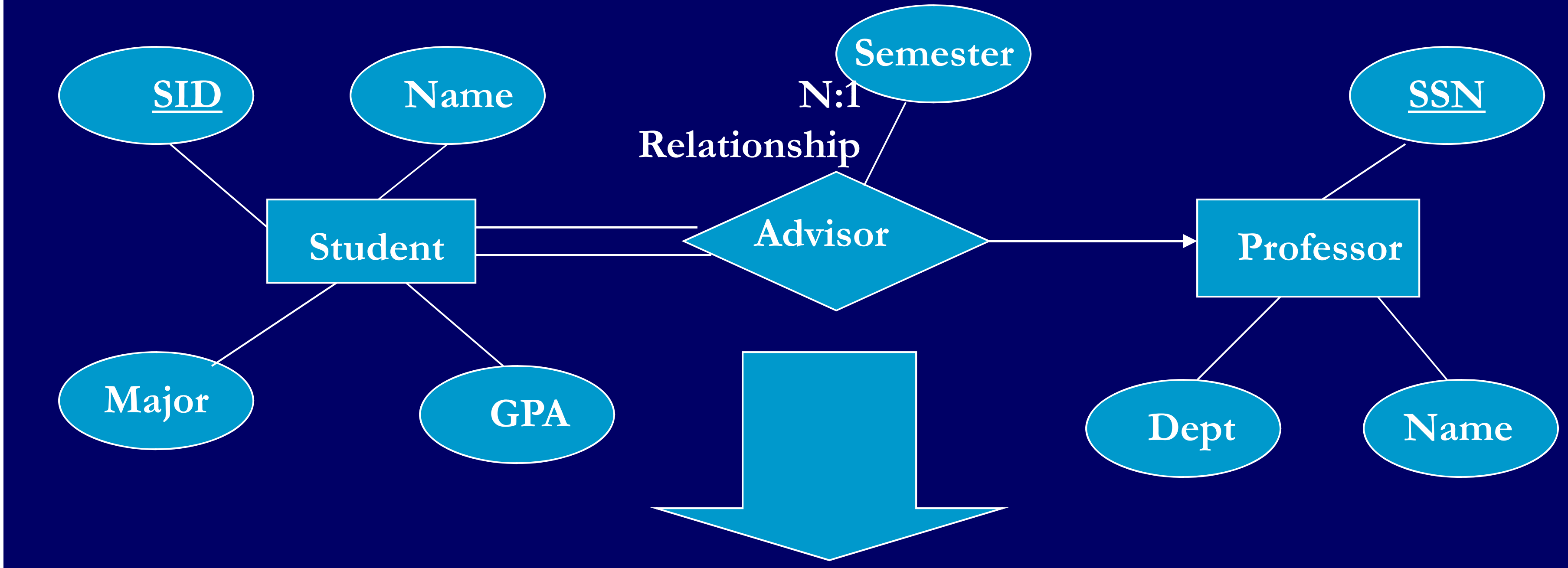
* key can be either *SID* or *LP_S/N*

Representing Relationship Set Unary/Binary Relationship

- <u>For one-to-many relationship w/out total participation</u>
  - Same thing as one-to-one

- <u>For one-to-many/many-to-one relationship with one entity set having total participation on “many” side</u>
  - Augment one extra column on the right side of the table of the entity set <u>on the “many” side</u>, put in there the primary key of the entity set <u>on the “one” side</u> as per to the relationship.

Example – Many-to-One Relationship Set



SID	Name	Major	GPA	Pro_SSN	Ad_Sem
9999	Bart	Economy	-4.0	123-456	Fall 2006
8888	Lisa	Physics	4.0	567-890	Fall 2005

* Primary key of this table is *SID*

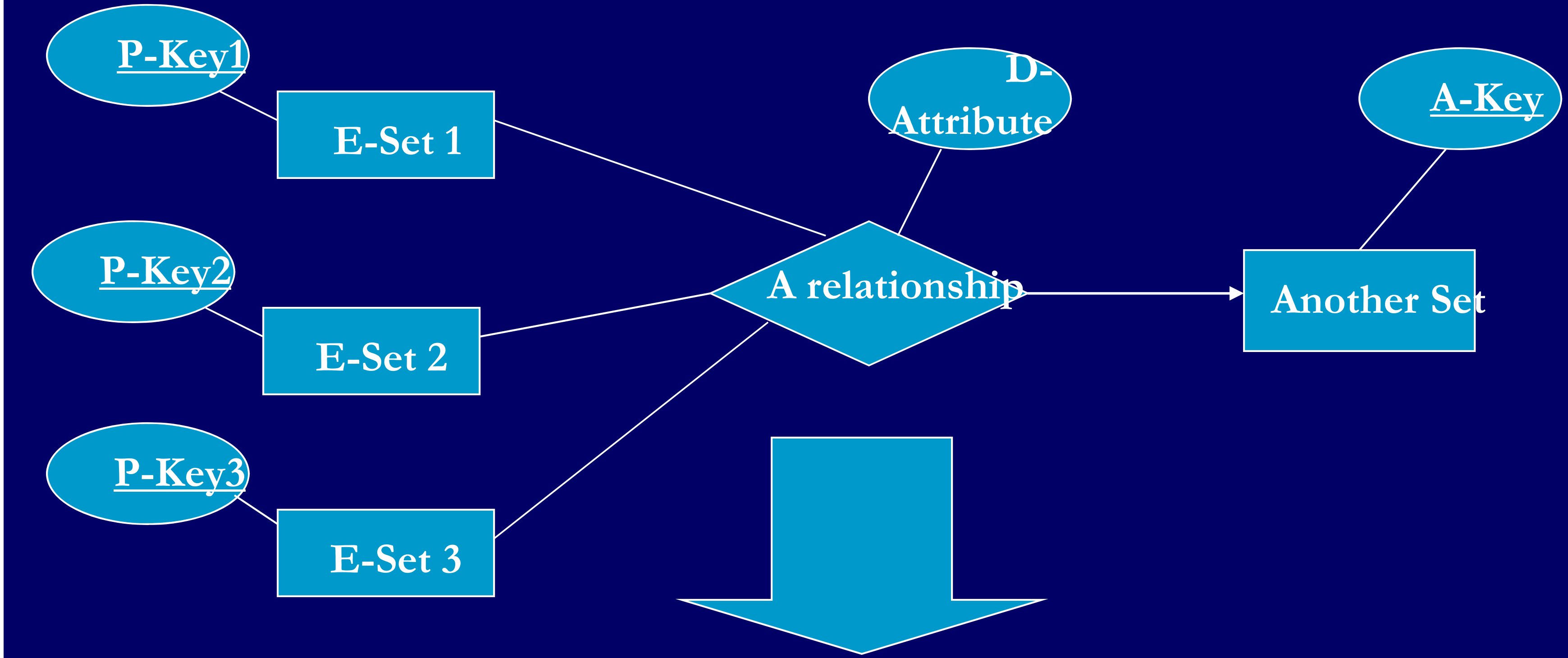
Representing Relationship Set Unary/Binary Relationship

- For many-to-many relationship
  - Same thing as one-to-one relationship without total participation.
  - Primary key of this new schema is the union of the foreign keys of both entity sets.
  - No augmentation approach possible…

Representing Relationship Set N-ary Relationship

- Intuitively Simple
  - Build a new table with as many columns as there are attributes for the union of the primary keys of all participating entity sets.
  - Augment additional columns for descriptive attributes of the relationship set (if necessary)
  - The primary key of this table is the union of all primary keys of entity sets that are on “many” side
  - That is it, we are done.

Example – N-ary Relationship Set

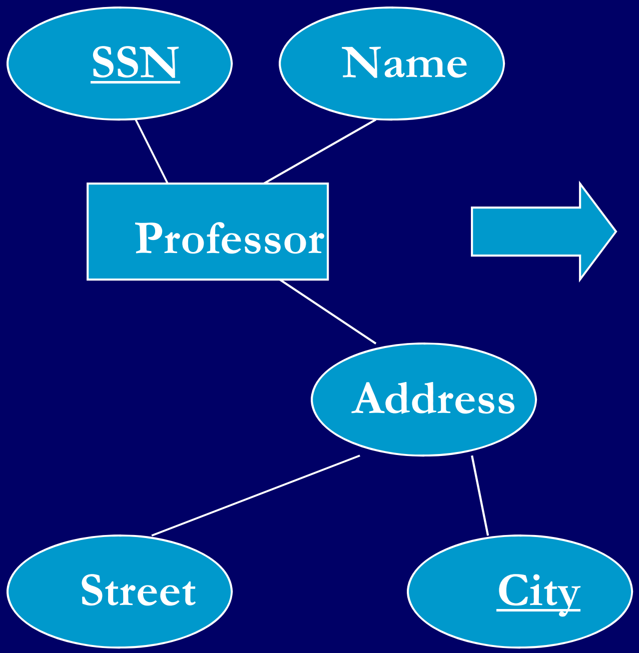


P-Key1	P-Key2	P-Key3	A-Key	D-Attribute
9999	8888	7777	6666	Yes
1234	5678	9012	3456	No

* key of this table is *P-Key1 + P-Key2 + P-Key3*

Representing Composite Attribute

- One column for each component attribute
- NO column for the composite attribute itself

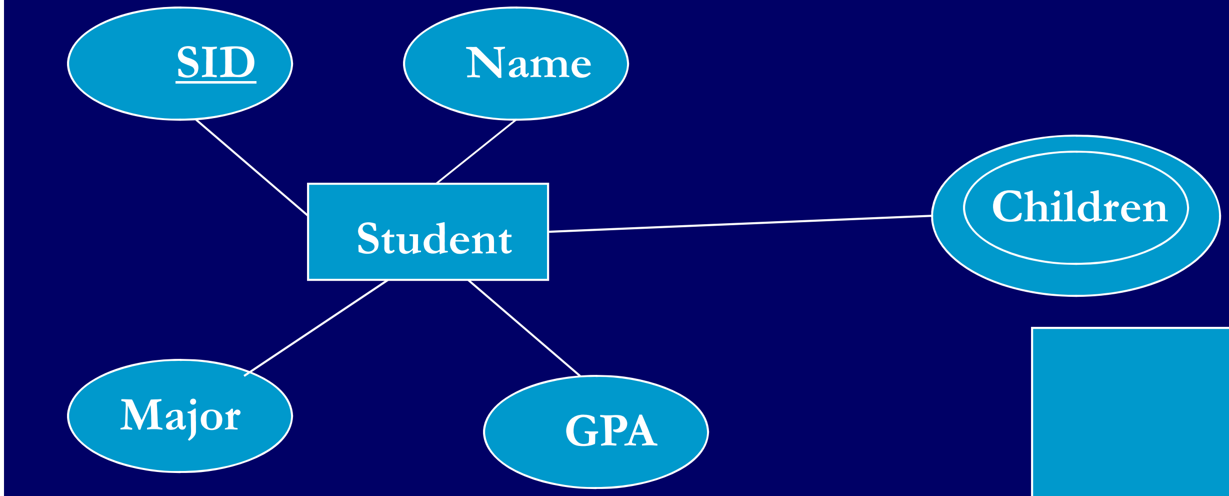


SSN	Name	Street	City
9999	Dr. Smith	50 1st St.	Fake City
8888	Dr. Lee	1 B St.	San Jose

Representing Multivalue Attribute

- For each multivalue attribute in an entity set/relationship set
  - Build a new relation schema with two columns
  - One column for the primary keys of the entity set/relationship set that has the multivalue attribute
  - Another column for the multivalue attributes. Each cell of this column holds only one value. So each value is represented as an unique tuple
  - Primary key for this schema is the union of all attributes

Example – Multivalue attribute

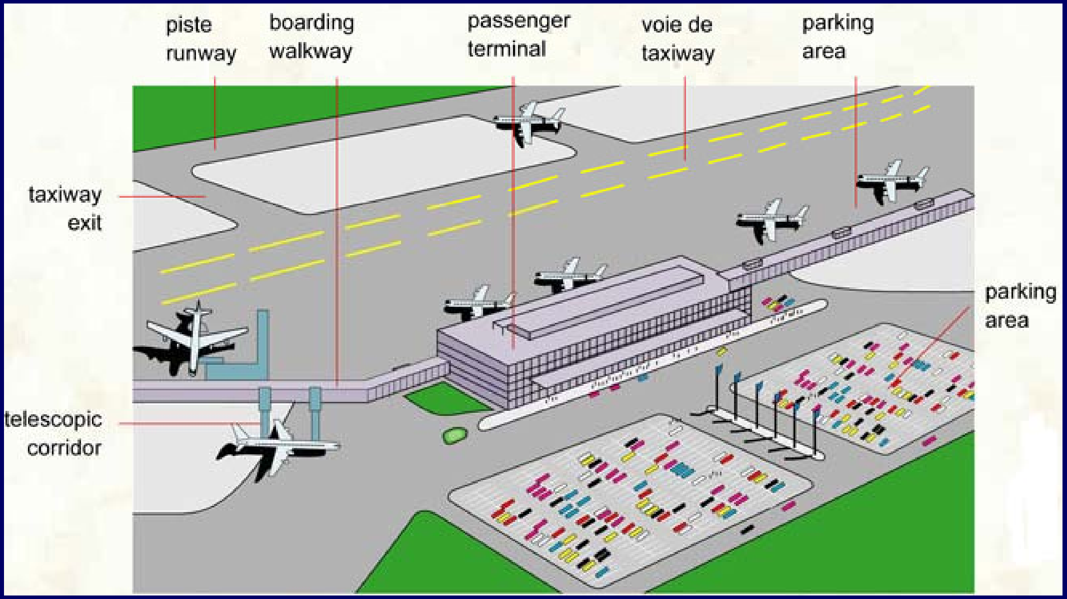


The key for this table is Student_SID + Children, the union of all attributes

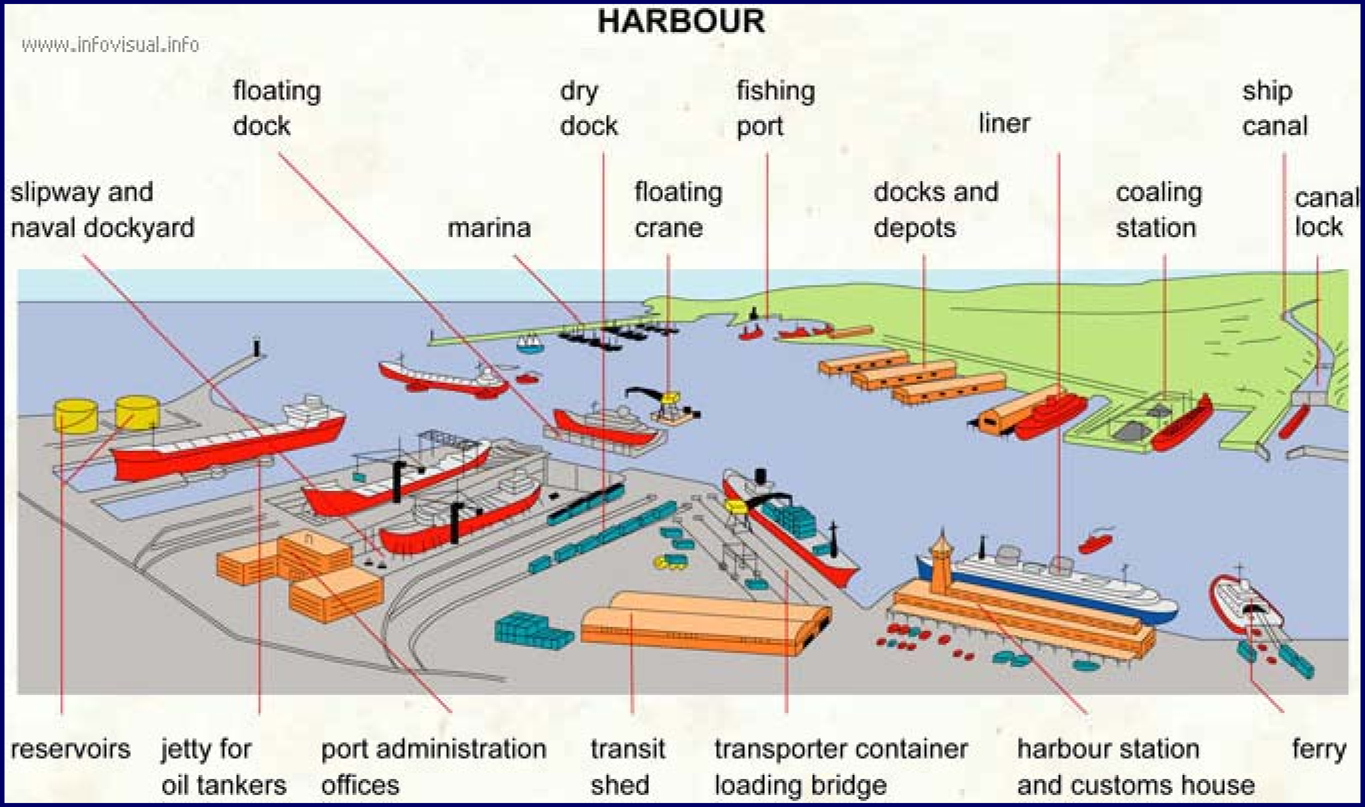
SID	Name	Major	GPA
1234	John	CS	2.8
5678	Homer	EE	3.6

Stud_SID	Children
1234	Johnson
1234	Mary
5678	Bart
5678	Lisa
5678	Maggie

Example


piste
runway
boarding
walkway
passenger
terminal
voie de
taxiway
parking
area
taxiway
exit
parking
area
telescopic
corridor

Example


HARBOUR
www.infovisual.info
floating dock
dry dock
fishing port
liner
ship canal
slipway and naval dockyard
marina
floating crane
docks and depots
coaling station
canal lock
reservoirs
jetty for oil tankers
port administration offices
transit shed
transporter container loading bridge
harbour station and customs house
ferry

Database Management System (DBMS)

- A collection of programs that enable:
  - ➔Defining (describing the structure),
  - – Populating by data (Constructing),
  - – Manipulating (querying, updating),
  - – Preserving consistency,
  - – Protecting from misuse,
  - – Recovering from failure, and
  - – Concurrent using

  of a database.