Database Management Systems

Shamim Ahmad

ERD to Tables

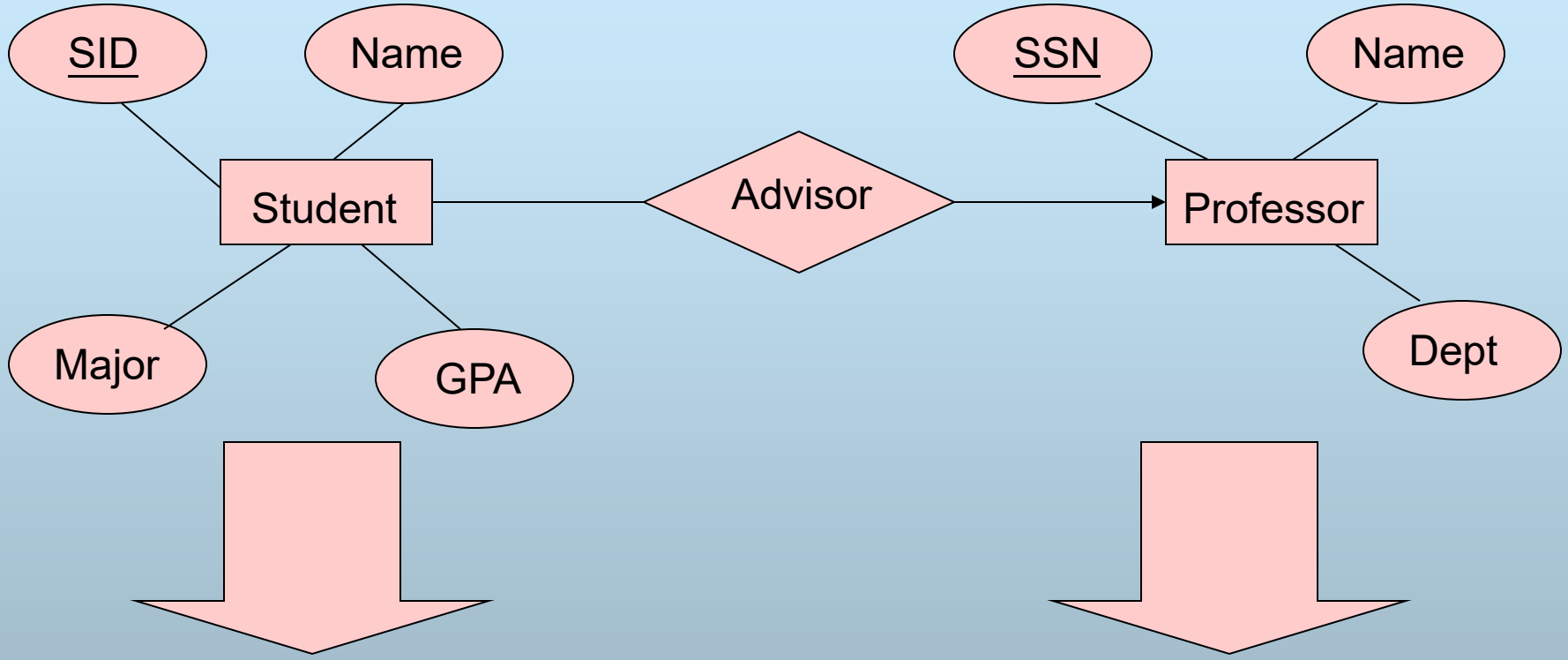
Reduction of an E-R Schema to Tables

- **Primary keys** allow **entity sets and relationship sets to be expressed uniformly as *tables*** which represent the contents of the database.
- A database which conforms to an E-R diagram can be represented by a **collection of tables**.
- For each entity set and relationship set there is **a unique table which is assigned th**e name of the corresponding ***entity set*** or relationship set.
- Each table has a number of columns **(generally corresponding to attributes**), which have unique names.
- **Converting an E-R diagram to a table format is the basis for deriving a relational database design from an E-R diagram**.

Representing Entity Sets as Tables

Representing strong Entity Sets as Tables

- **A strong entity** set reduces to a table with the same attributes.



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

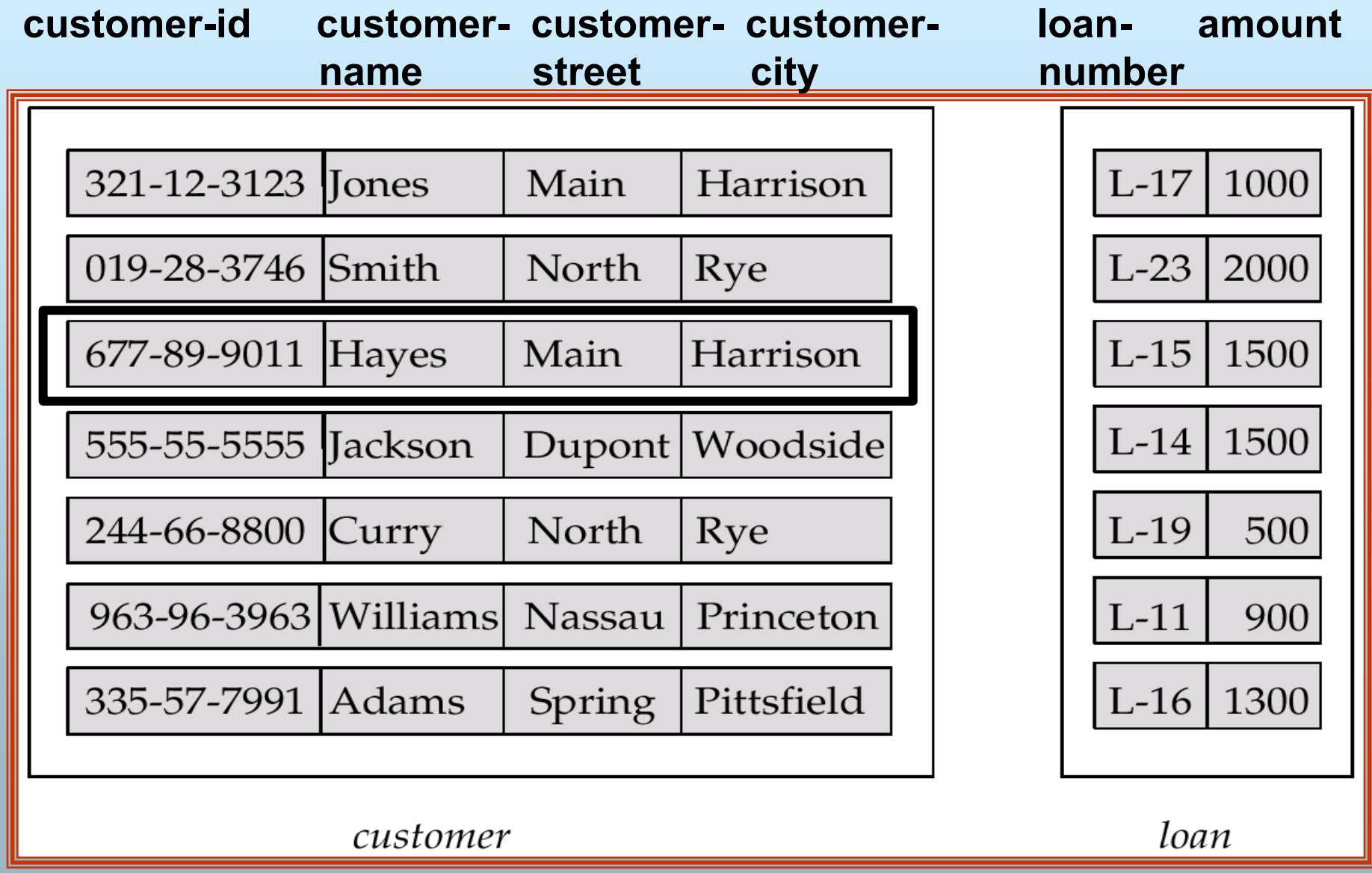
SSN	Name	Dept
9999	Smith	Math
8888	Lee	CS

Representing Relationship Sets as Tables

Many-to-many Relationship Sets

- 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.

Entity Sets *customer* and *loan*

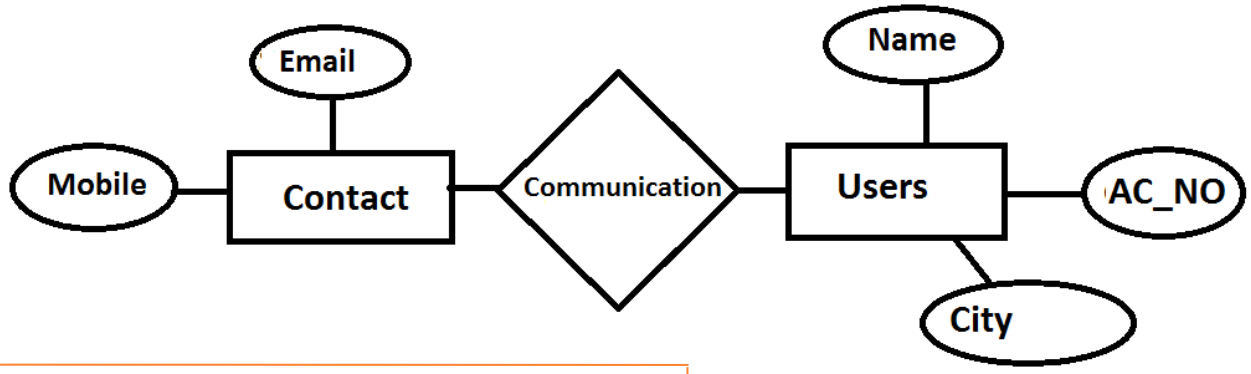


Representing Relationship Sets as Tables

- A **many-to-many relationship** set is represented as a table with columns for the **primary keys of the two participating entity sets**, and any descriptive attributes of the relationship set.
- E.g.: table for relationship set *borrower*

customer-id	*loan-number*
019-28-3746	L-11
019-28-3746	L-23
244-66-8800	L-93
321-12-3123	L-17
335-57-7991	L-16
555-55-5555	L-14
677-89-9011	L-15
963-96-3963	L-17

Many To Many Relationship(Cont.)



PK

Email	Mobile
ononnaontora@gmail.com	01700000000
sumaia@vu.edu.bd	01700000000
sumaiavu381332@gmail.com	01700000000
raisha12@gmail.com	01700000011
raisha_afroza@gmail.com	01700000011

FK

FK

Email	Name
ononnaontora@gmail.com	ononna
sumaia@vu.edu.bd	ononna
sumaiavu381332@gmail.com	ononna
raisha12@gmail.com	Raisha
raisha_afroza@gmail.com	Raisha
mahbub@gmail.com	Mahbub

PK

Name	AC_NO.	City
Ononna	AC11	Raj.
Raisha	AC12	Khul.
Mahbub	AC25	Dhaka

Foreign Key:

- A **FOREIGN KEY** is a key used to link two tables together.
- A **FOREIGN KEY** is a field (or collection of fields) in one table that refers to the PRIMARY KEY in another table.
- The table containing the foreign key is called the child table, and the table containing the candidate key is called the referenced or parent table.

S_id	S_Name	S_CGPA
13231003	Tasnim	3.59
13231001	Jui	3.66
13231111	Dipty	3.78

TableName: Student_Info

S_id	Session	Dept
13231003	Summer13	CSE
13231001	Summer13	CSE
13231117	Fall14	EEE

TableName: Dept_Info

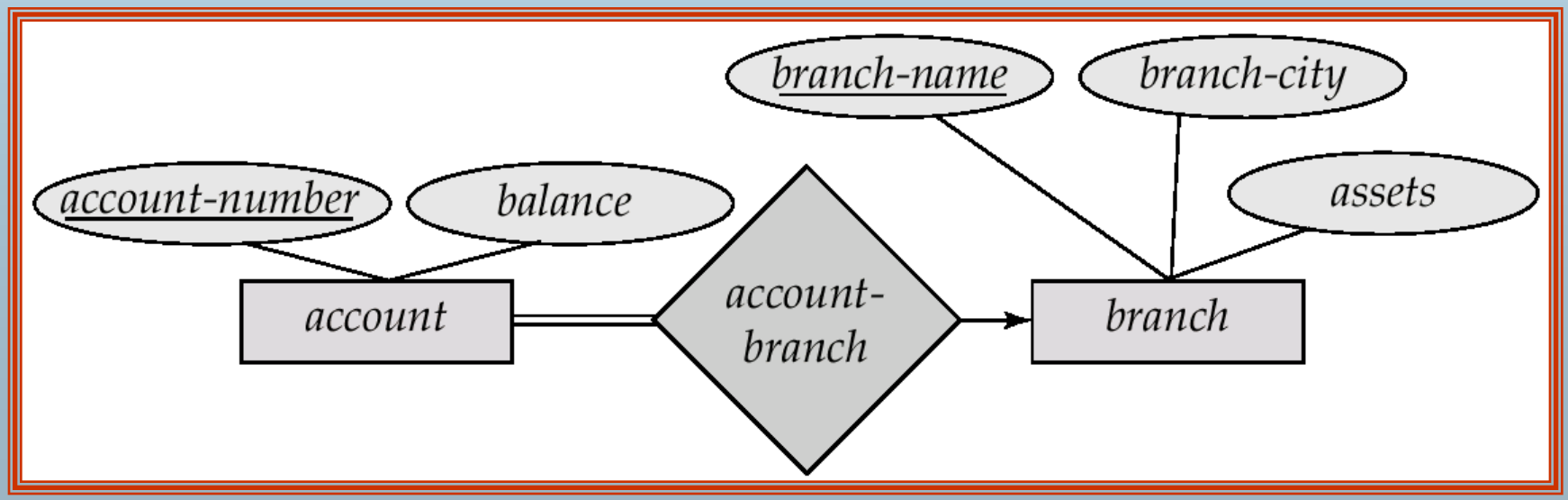
Representing Relationship sets without a separate Tables

Many-to-one and One-to-many

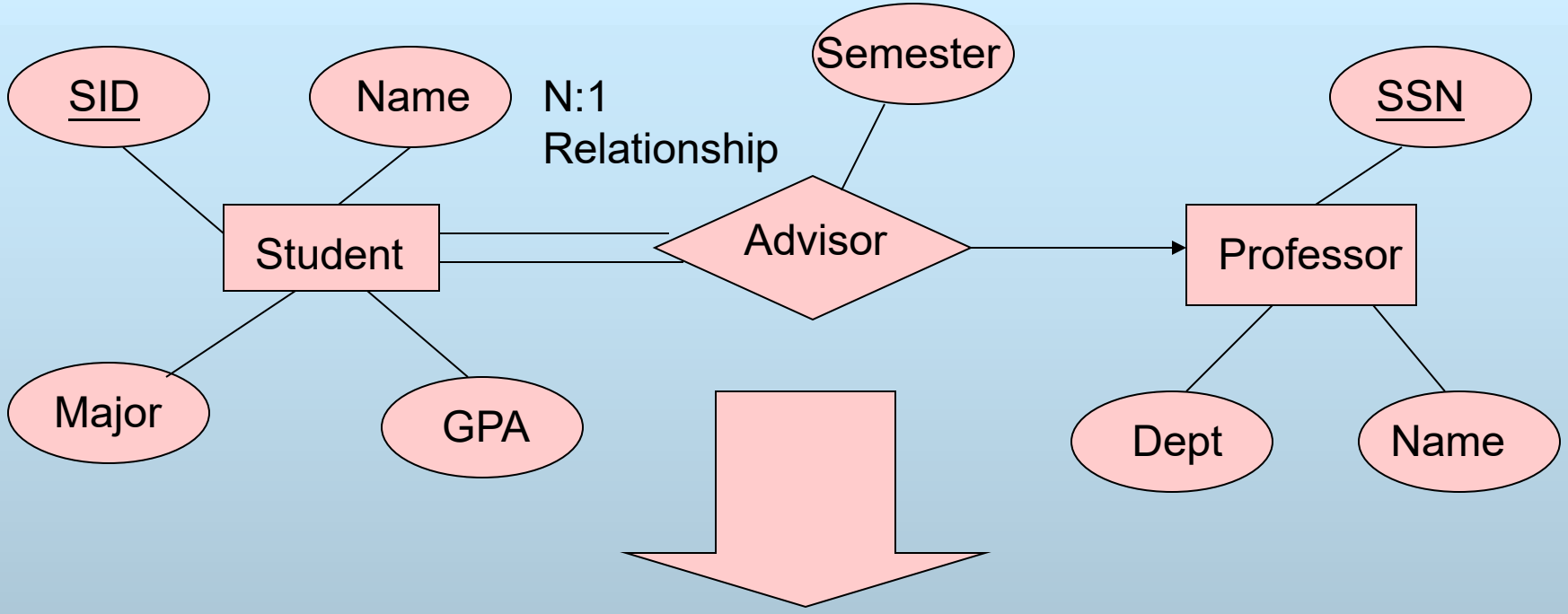
- **For one-to-many** relationship **w/out total** participation
  - Same thing as one-to-one

Redundancy of Tables

- **Many-to-one** and **one-to-many** **relationship** sets that **are total** on the **many-side** can be represented by adding an **extra attribute** to the many side, containing the **primary key of the one side**
- E.g.: Instead of creating a table for relationship *account-branch*, add an attribute *branch* name to the entity set *account*



Example – Many-to-One Relationship Set



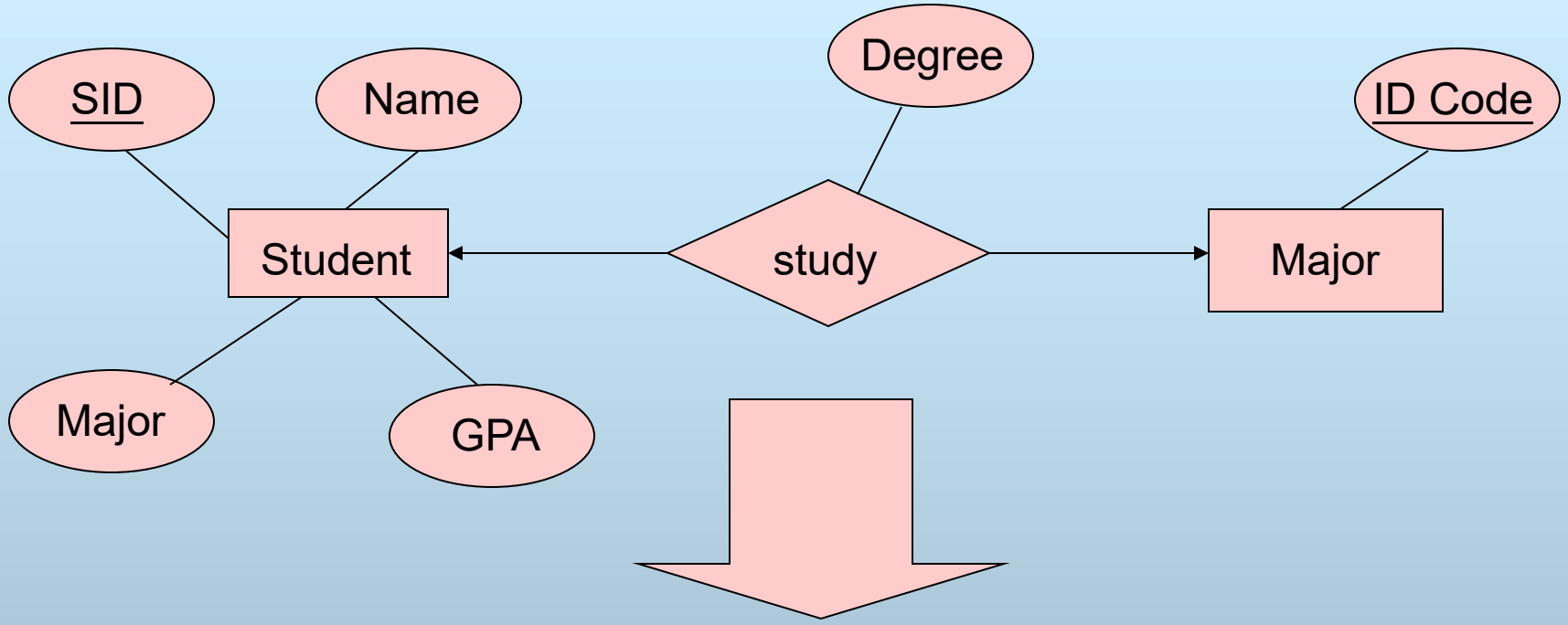
SID	Name	Major	GPA	Pro_SSN	Ad_Sem
9999	Bart	Economy	-4.0	123-456	Fall 2006
8888	Lisa	Physics	4.0	567-890	Fall 2005

* Primary key of this table is *SID*

One-to-one Relationship Set

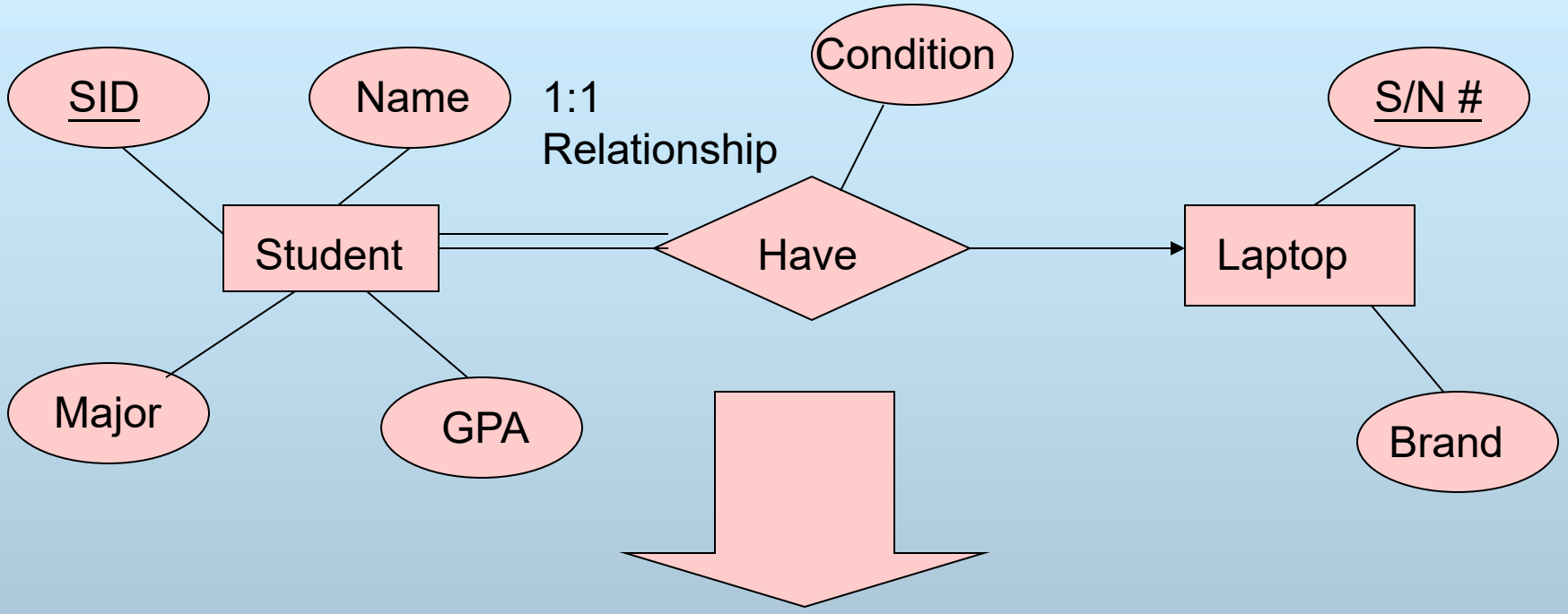
- For **one-to-one r**elationship **w/out total** participation
  - Build a table with **two columns**, one column for each participating **entity set's primary** key. Add successive columns, one for each descriptive attributes of the relationship set (if any).
- For **one-to-one** relationship with **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

One-to-One (W/out Total)



SID	Maj_ID Co	S_Degree
9999	07	1234
8888	05	5678

One-to-One (Total Student side)



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

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

Composite and Multivalued Attributes

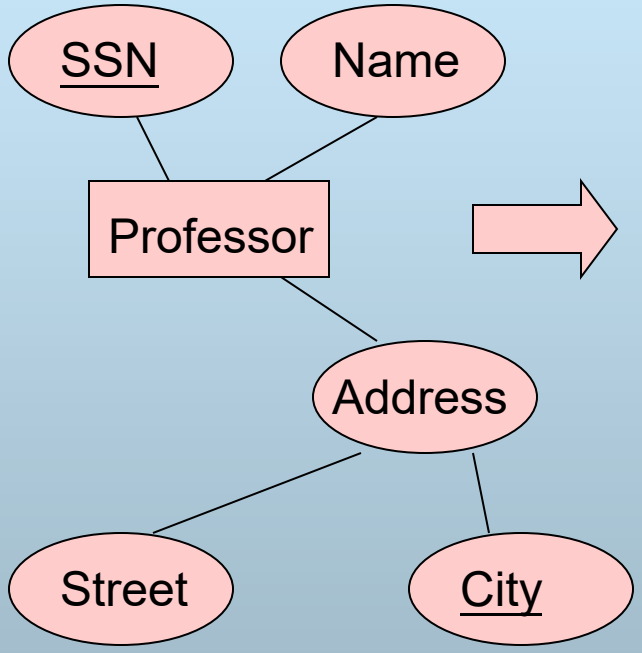
- **Composite** attributes are **flattened** out by creating a separate attribute for **each component attribute**
  - E.g. given entity set *customer* with composite attribute *name* with component attributes *first-name* and *last-name* the table corresponding to the entity set has two attributes

    name.first-name and *name.last-name*

Composite and Multivalued Attributes

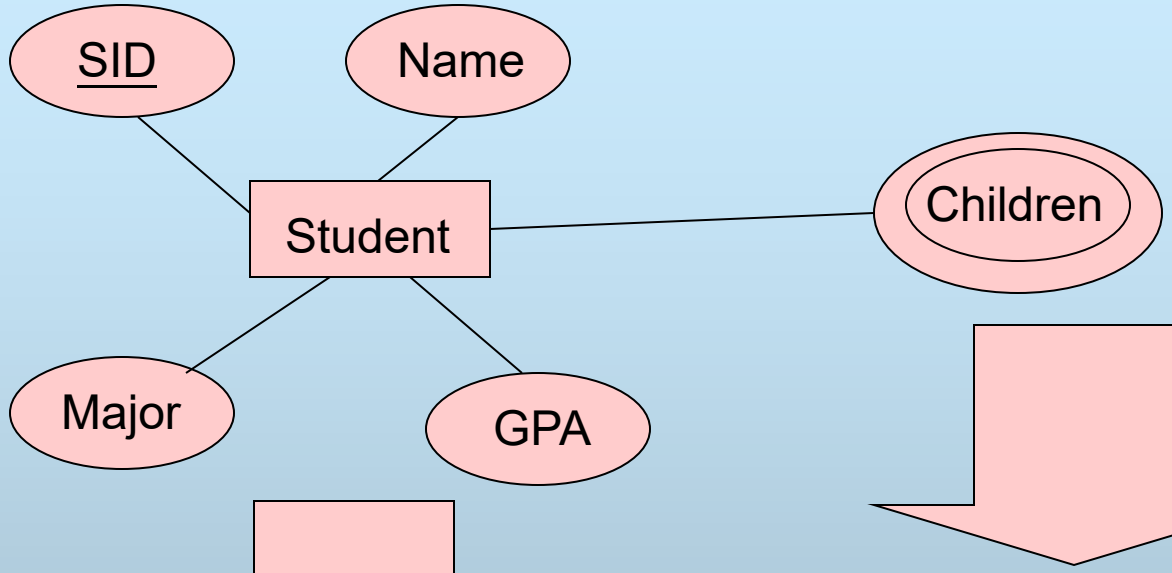
- A multivalued attribute M of an entity E is represented by a separate table EM
  - Table EM has attributes corresponding to the primary key of E and an attribute corresponding to multivalued attribute M
  - Each value of the multivalued attribute maps to a separate row of the table EM

Representing Composite Attribute



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

Representing Multivalue Attribute



The primary 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

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.