CINEC
CAMPUS
Beyond A Graduate

BSc (Hons) in Software Engineering

Database Management Systems
Course Code: IT1203
Week 5

Logical Database Design (Relational Model)

Objectives

BE ABLE TO IDENTIFY:

- What is logical database design?.
- How to derive a logical model from the information represented in the ER model (conceptual model)
- We focus on one type of logical model which is relational model

Relational Model

- Simple & elegant mod:
  - Everything is a relation (= table)

    Every relation is a table with rows & columns

- There are standard ways to convert from the E-R model (**conceptual model**)

  TO

  Relational model (**logical model**)

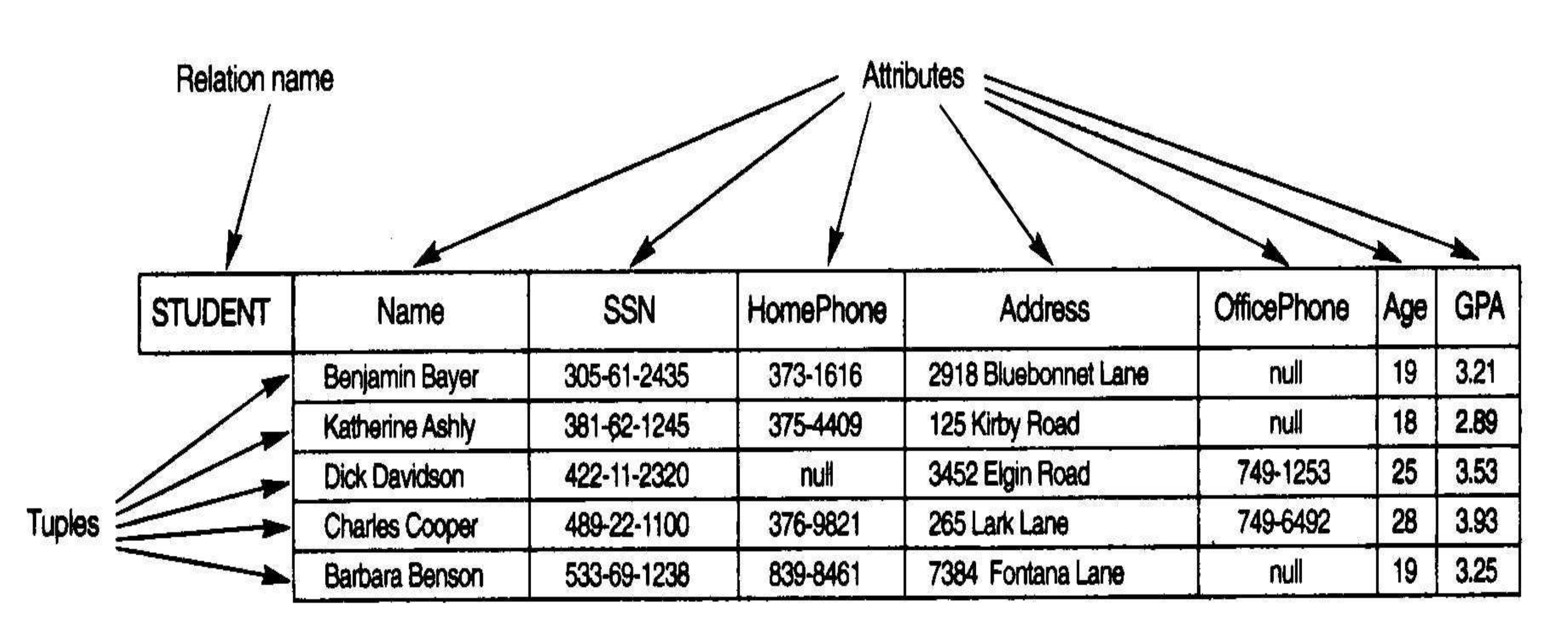
Relational Model (contd.)

- The relational model represents the database as a collection of **relations.**

- Relation consists of
  - Relation schema
  - Relation instance (table)

Relation

- RELATION:
  - Schema
  - Instance



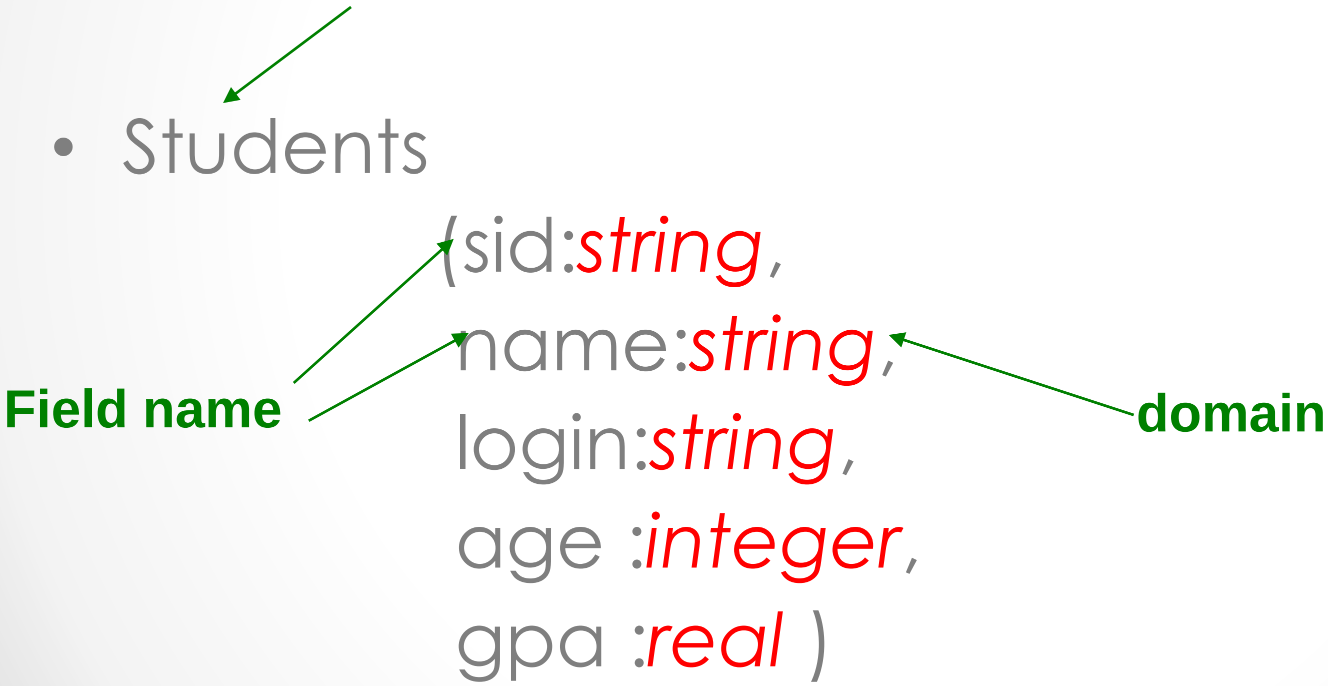
STUDENT	Name	SSN	HomePhone	Address	OfficePhone	Age	GPA
	Benjamin Bayer	305-61-2435	373-1616	2918 Bluebonnet Lane	null	19	3.21
	Katherine Ashly	381-62-1245	375-4409	125 Kirby Road	null	18	2.89
	Dick Davidson	422-11-2320	null	3452 Elgin Road	749-1253	25	3.53
	Charles Cooper	489-22-1100	376-9821	265 Lark Lane	749-6492	28	3.93
	Barbara Benson	533-69-1238	839-8461	7384 Fontana Lane	null	19	3.25

Relation Schema

- Describes the column heads (attributes) of the relation
    - name of the relation,
    - name of each filed,
    - domain of each field
        - Domain : is described by domain name and set of associated values

Relation Schema

- Schema Name(relation name)



- Students

  (sid:*string*,
  name:*string*,
  login:*string*,
  age :*integer*,
  gpa :*real*)

Field name

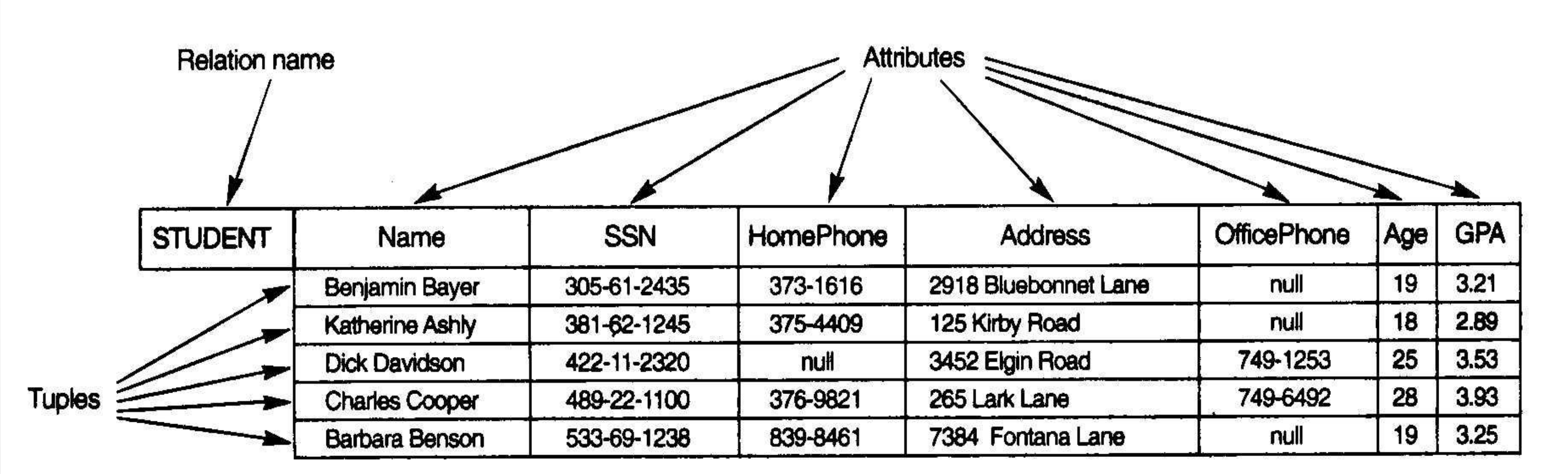
domain

- Domain GPA : real (0-4)

Relation Instance

- Set of tuples or records or rows :
- Each tuple has the same number of fields as the relation schema

Example : Relation Instance



STUDENT

Name	SSN	HomePhone	Address	OfficePhone	Age	GPA
Benjamin Bayer	305-61-2435	373-1616	2918 Bluebonnet Lane	null	19	3.21
Katherine Ashly	381-62-1245	375-4409	125 Kirby Road	null	18	2.89
Dick Davidson	422-11-2320	null	3452 Elgin Road	749-1253	25	3.53
Charles Cooper	489-22-1100	376-9821	265 Lark Lane	749-6492	28	3.93
Barbara Benson	533-69-1238	839-8461	7384 Fontana Lane	null	19	3.25

Degree of a relation

- The **degree** of R is the number of attributes in R
- (ID,Name,Address,Phone)=4

ID	Name	Address	Phone
100	Sampath	Moratuwa	01992883
110	Amali	Colombo -3	01983733
120	Sanath	Negombo	null

Integrity Constraints= IC

- Data base is only good as the information stored in it
- DBMS must prevent entry of incorrect information
- To prevent :

  Constraints / conditions are specified on a relational schema = ICs
- Database which satisfies all constraints specified on a database schema is a legal instance.
- DBMS enforces constraints - permits only legal instances to be stored
- When the application is run the DBMS checks for the violation and disallows the changes to the data that violates the specified IC

Integrity Constraints

- Specified and enforced at different times.

  - **Specified :** When the DBA /end user defines the data base schema
  - **Enforced :** When database application is run
    - DBMS checks for violations
    - Disallow violating entries

Integrity Constraints

Many kinds of ICs:

- Domain constraints
- Key constraints
- Entity integrity constraints
- Referential integrity constraints

Domain Constraints

- **Domain constraints**: value in the Column must be drawn from the domain associated with that column

- **Restricts the :**
  - Type
  - Values that can appear in the field

Eg.

Name	Char (25)
GPA	(real >=0, =<4)

Key constraints

- **Is a statement that;**
  - A certain minimal subset of the fields of a relation is a unique identifier for a tuple.

  A set of fields that uniquely identifies a tuple according to a key constraint is called **Candidate Key**
- **Which Means**
  - No two in a legal instance cannot have identical values in all the fields of a key.
  - No subset of the set of fields in a key is a unique identifier for a tuple. =(minimal)

Super Key

- Any other set of attributes that uniquely identify a tuple is called the **superkey** of a relation
  Student (SID, Name, Address, Contact, GPA)

- What is the minimal set of attributes that uniquely identify the relation ?
  - SID =Referred to as Key

- (SID + Name) Is this unique?
  - Yes, but NOT the minimal set
- Referred to as Super Key
-

Key Constraints

- **Example**: The CAR relation schema:
  CAR(State, Reg#, SerialNo, Make, Model, Year)

  Can you identify the possible keys?

  Key1 = {State, Reg#},
  Key2 = {SerialNo},

- If a relation has *several* **candidate keys**, one is chosen arbitrarily to be the **primary key**. The primary key attributes are *underlined*.

-
  - {SerialNo, Make} is a superkey but *not* a key.

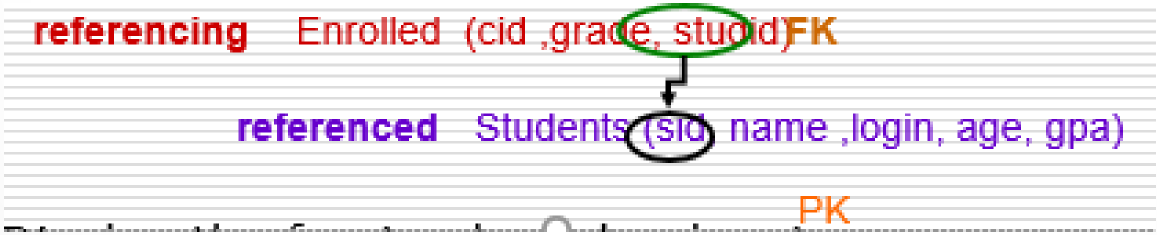
-

Constraints…

- **Entity Integrity Constraints:** states that **primary key** values cannot be null
  - This is because primary key values are used to *identify* the individual tuples.
- **Referential Integrity Constraints**
  - Some times information stored in one relation is linked to information stored in another relation.
  - If one is modified the other must be modified to keep the data consistent.
  - An IC involving both relations must be specified
  - IC involving 2 relations is a **foreign key constraint.**
  - Foreign keys enforce referential integrity constraints

Referential Integrity

- A constraint involving *two* relations
  - **referencing relation**
  - **referenced relation.**
- Tuples in the *referencing relation* have attributes **FK** (called **foreign key** attributes) that reference the primary key attributes **PK** of the *referenced relation*



referencing Enrolled (cid ,grade, studid) FK

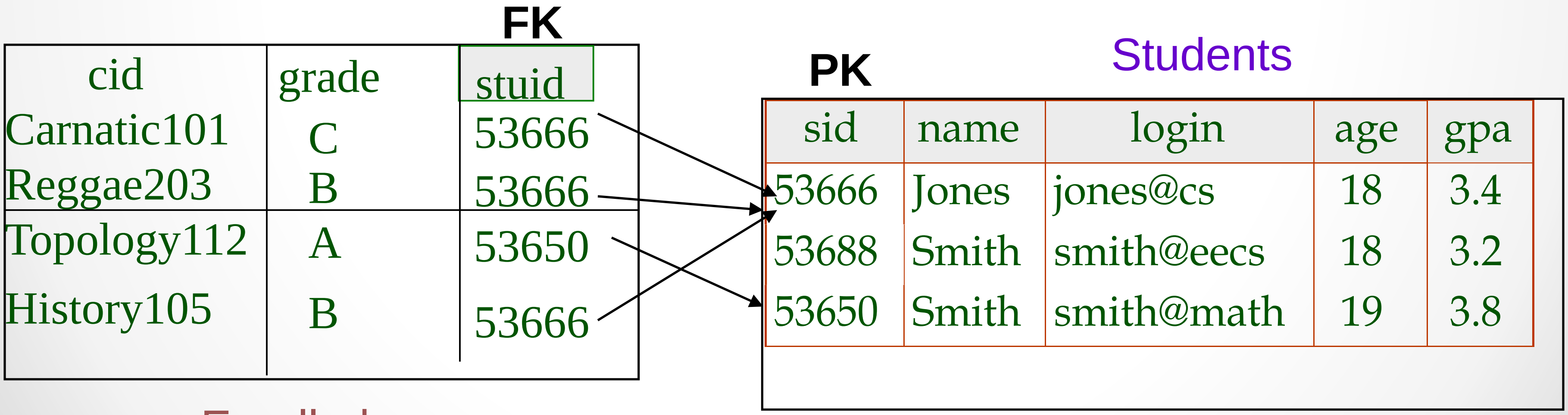
referenced Students (sid, name ,login, age, gpa)

PK

- Display the foreign keys by drawing an arrow from the **foreign key** to the **primary key**

Referential Integrity

- The value in the foreign key column (can be either:
  - a value of an existing primary key in the **referenced relation** or a **null**



FK

cid	grade	stuid
Carnatic101	C	53666
Reggae203	B	53666
Topology112	A	53650
History105	B	53666

Enrolled

PK

Students

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8

DB operations & constraints

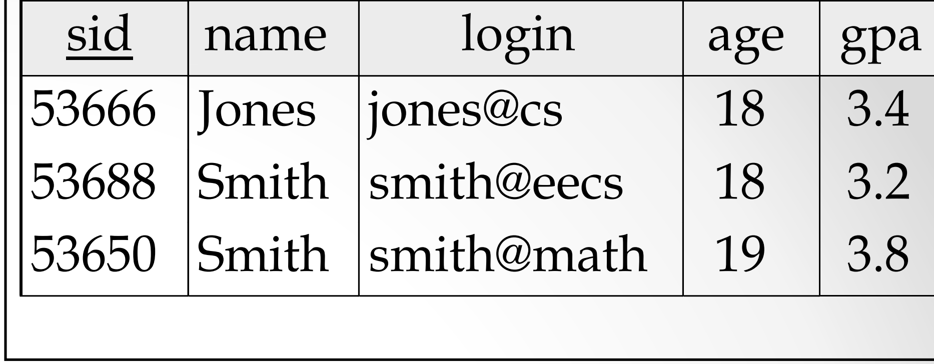
- IC are specified when a relation is created and enforced (checked) when a relation is modified
- 3 types of modifications to the relation :
  - Insert : inserts a new tuple(s) into a relation.
  - Delete : delete tuple(s) in a relation.
  - Update : changes the values of some attributes in existing tuples .

Insert operation

- The insert operation can violate the following constraints:
  - Domain constraints (invalid value)
  - Key constraints (duplicate key values)
  - Entity integrity constraints (null primary key value)
  - Referential integrity constraint (non-existing primary key value)

examples

Students



sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8

- Domain constraints (invalid value)

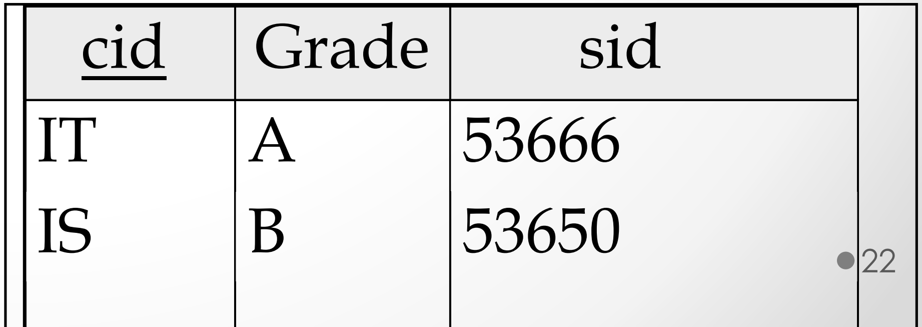
  Insert <‘abc’,’Tom’,’tom@lk’,’17’,3.2>into Students
  - sid value is not in the domain
- Key constraints (duplicate key values)

  Insert <‘53666’,’Tom’,’tom@lk’,’17’,3.2>into Students
- Entity integrity constraints (null primary key value)

  Insert <null,’Tom’,’tom@lk’,’17’,3.2>into Students
- Referential integrity constraint (non-existing primary key value)

  Insert <‘IT’,’A’,’53900’>into Enrolled

Enrolled



cid	Grade	sid
IT	A	53666
IS	B	53650

Delete operation

- Delete operation can violate referential integrity.

Enrolled

cid	grade	stuid
Carnatic101	C	53666
Reggae203	B	53666
Topology112	A	53650
History105	B	53666

Students

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8

- Two options:
    - Reject the deletion
    - Cascade the delete

Update operation

- Update operation can be considered as a deleting a tuple and re-inserting the tuple with new values
- All constraints discussed in Insert & Delete need to be considered
  - Domain constraints (invalid value)
  - Key constraints (duplicate key values)
  - Entity integrity constraints (null primary key value)
  - Referential integrity constraint (non-existing primary key value)

In-Class Exercise

- Consider the following relations for a database that keeps track of student enrollment in courses and the books adopted for each course:
- STUDENT(SSN, Name, Major, Bdate)
- COURSE(Course#, Cname, Dept)
- ENROLL(SSN, Course#, Quarter, Grade)
- BOOK_ADOPTION(Course#, Quarter, Book_ISBN)
- TEXT(Book_ISBN, Book_Title, Publisher, Author)
- Draw a relational schema diagram specifying the foreign keys for this schema.

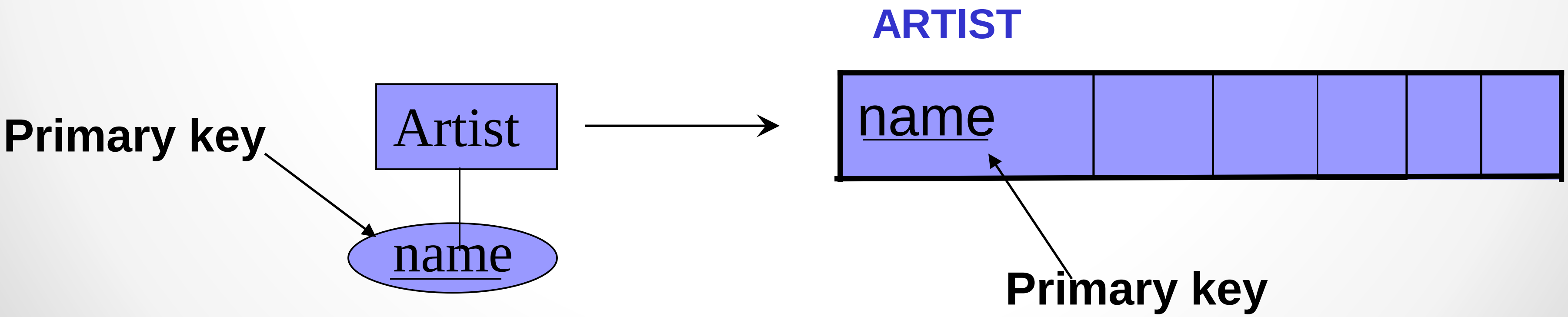
ER to Relational Mapping…

- In the Database Design process, we firstly derive a **conceptual model** (ER Diagram)
- This model needs to be mapped to the **relational model** in order to be implemented using a relational DBMS (RDBMS). Moving from Conceptual (ER) to lower level Logical Model (Relational)
- ER is independent of the details of the implementation (relational, network or OO)_
- This section discusses the rules that can be used for this process…

Mapping: Regular Entity

ER Model **Relational Model**

- Entity (strong) → Relation
- For example,



Primary key

Artist

name

ARTIST

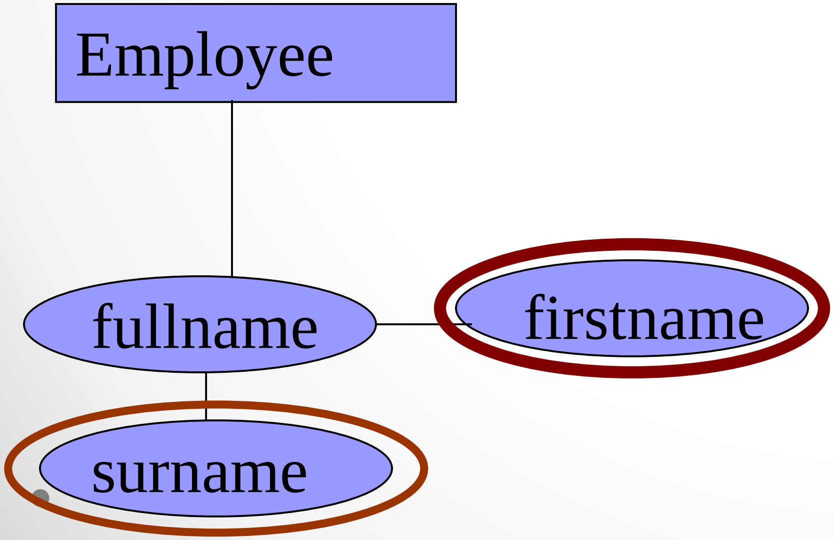
name

Primary key

Mapping: Composite Attribute

ER Model **Relational Model**

- Composite attributes → Set of simple atomic attributes



EMPLOYEE

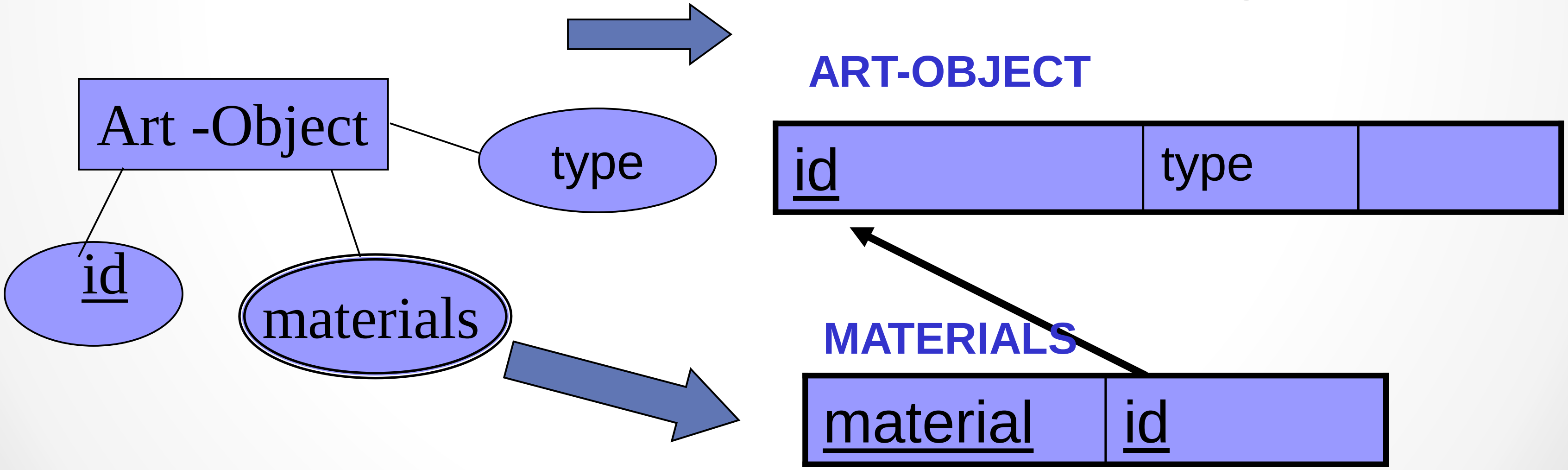
surname	firstname

Mapping: Multivalued Attributes

ER Model Model

Relational

- Multivalued attribute → Relation & Foreign Key



ART-OBJECT

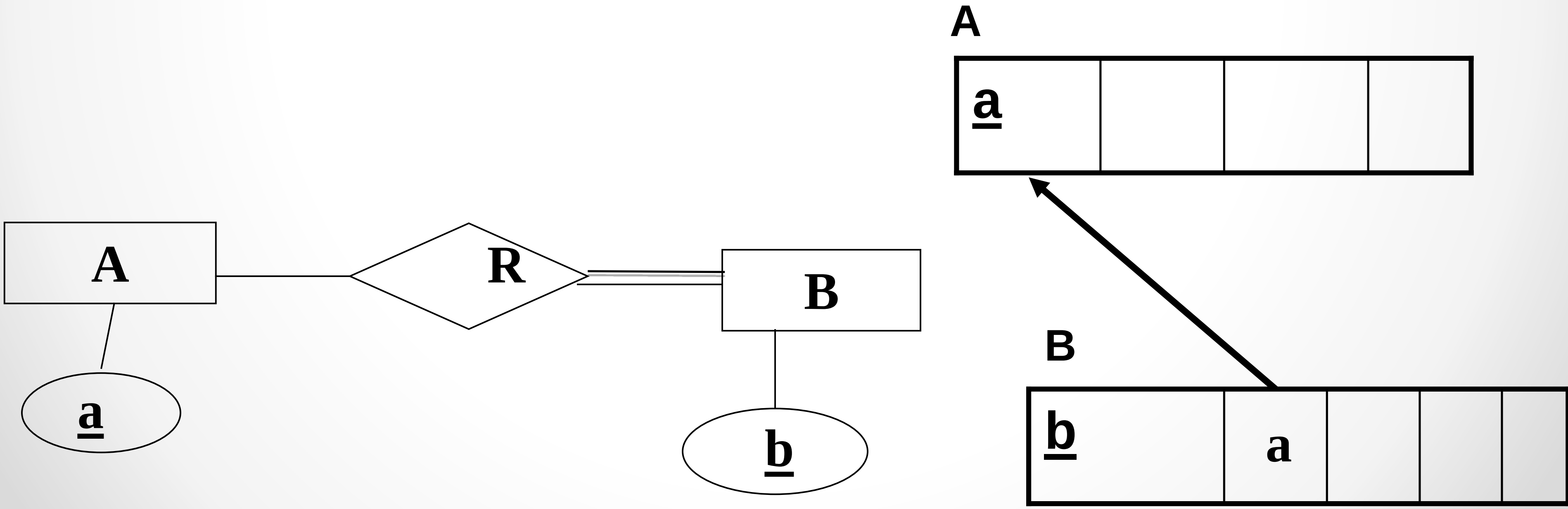
id	type	

MATERIALS

material	id

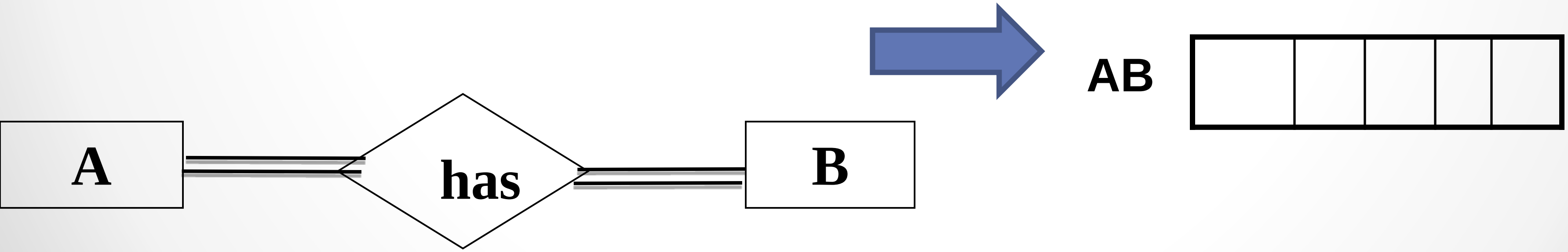
Mapping:1:1 conversion with total participation

If there is a 1:1 relationship R from entity A to B and if B is in total participation with A on R then the foreign key is placed in B



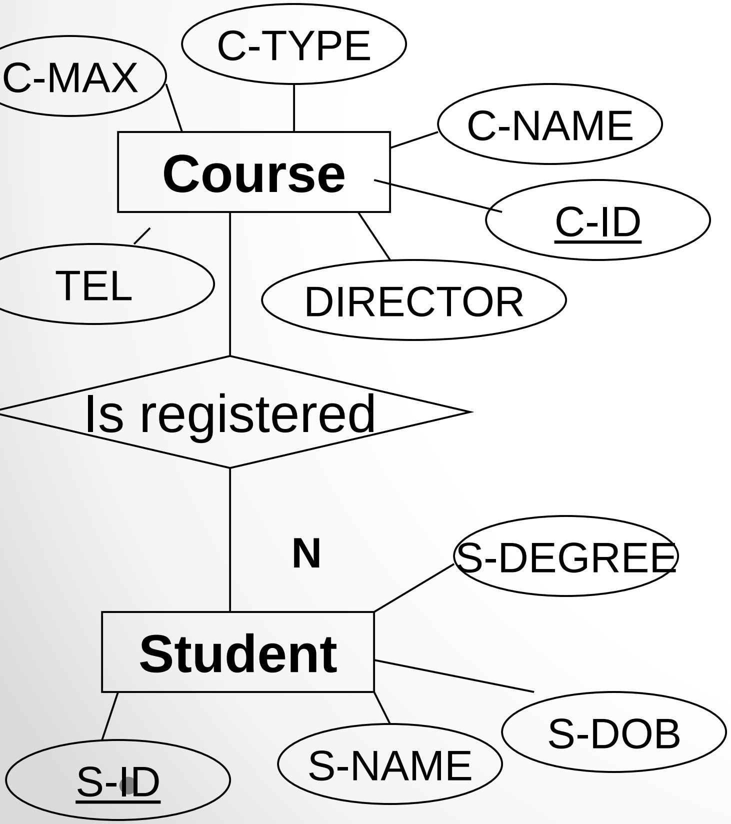
Mapping:1:1 conversion with total participation

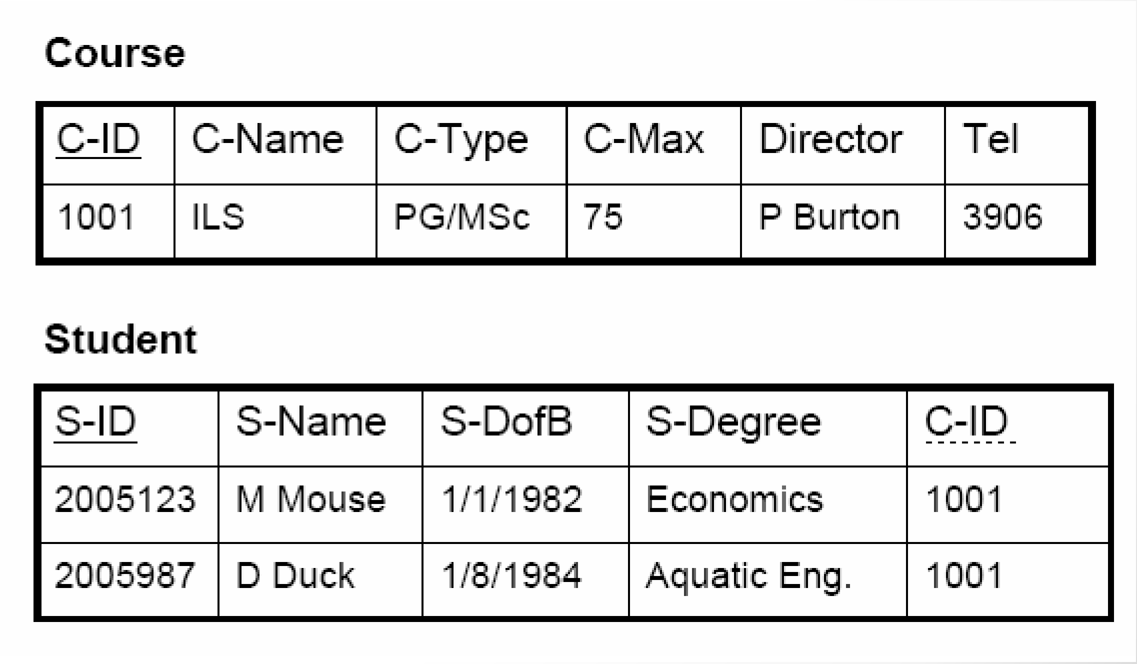
If there is a 1:1 relationship R from entity A to B and if A and B are both in total participation with R then A & B can be collapsed as 1 table



Mapping :1:N Conversion

1:N relationships, post the identifier (Primary Key) from the ‘one’ side as an attribute into the ‘many’ side





Course

C-ID	C-Name	C-Type	C-Max	Director	Tel
1001	ILS	PG/MSc	75	P Burton	3906

Student

S-ID	S-Name	S-DofB	S-Degree	C-ID
2005123	M Mouse	1/1/1982	Economics	1001
2005987	D Duck	1/8/1984	Aquatic Eng.	1001

Mapping : M:N Conversion

- For N:M relationships, create a new table and post the identifiers from each of the linked entities as attributes in this table

FK

Class

Cl-ID	Cl-Name	Cl-Type
CS504	IR	Lectures
CS520	DBMS	Lectures

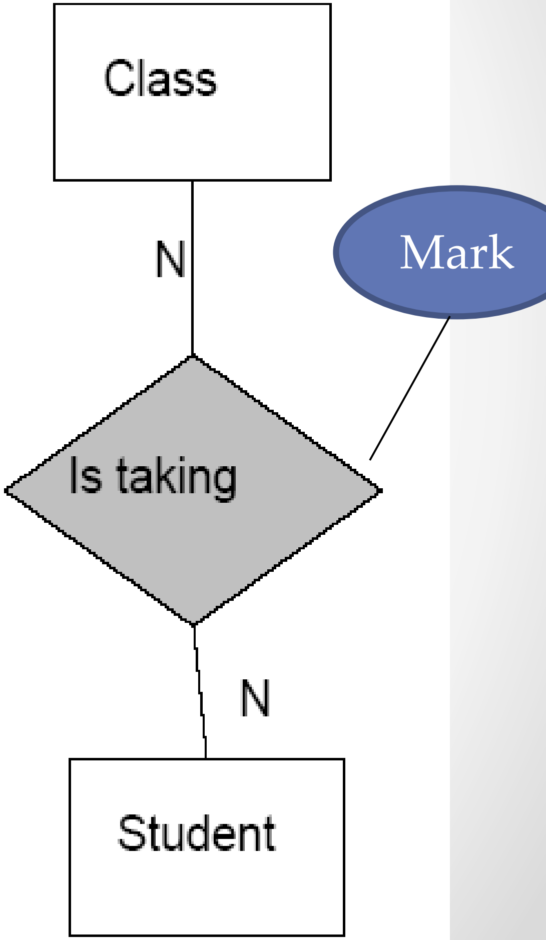
Class-Student

Cl-ID	S-ID	Mark
CS504	2005123	67%
CS504	2005987	59%

FK

Student

S-ID	S-Name	S-DofB	S-Degree	~~C-ID~~
2005123	M Mouse	1/1/1982	Economics	~~1001~~
2005987	D Duck	1/8/1984	Aquatic Eng.	1001



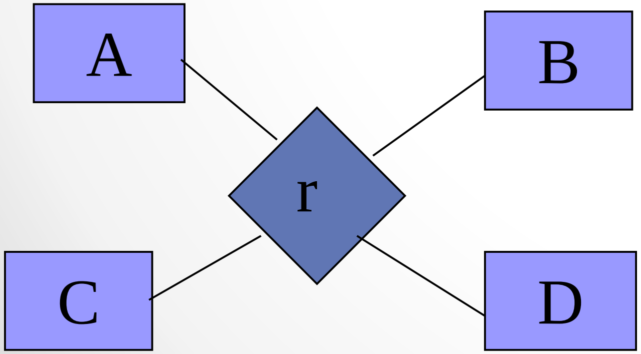
Mapping ..N-ary Relationships

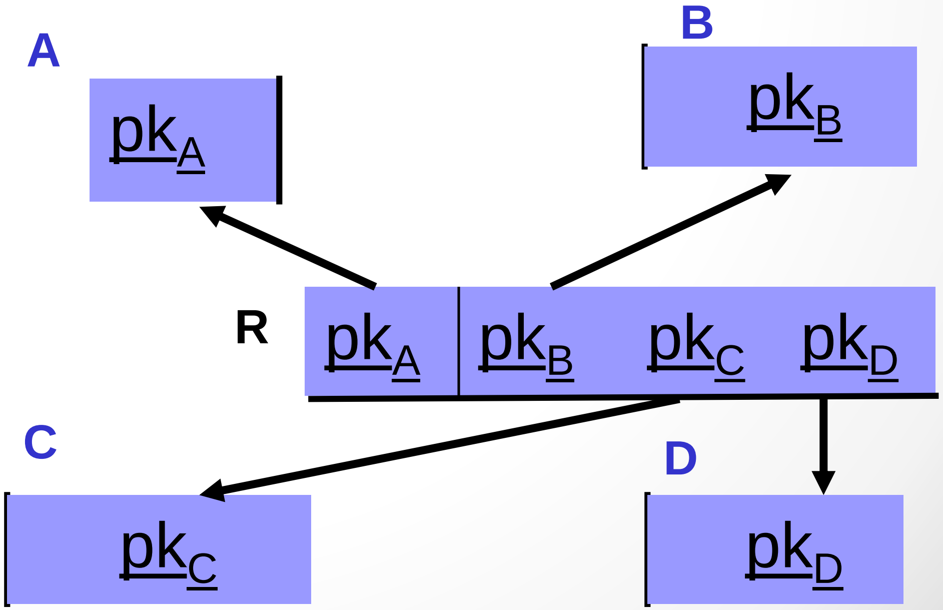
ER Model

- N-ary relationship relation and n foreign keys

Relational Model

“Relationship”





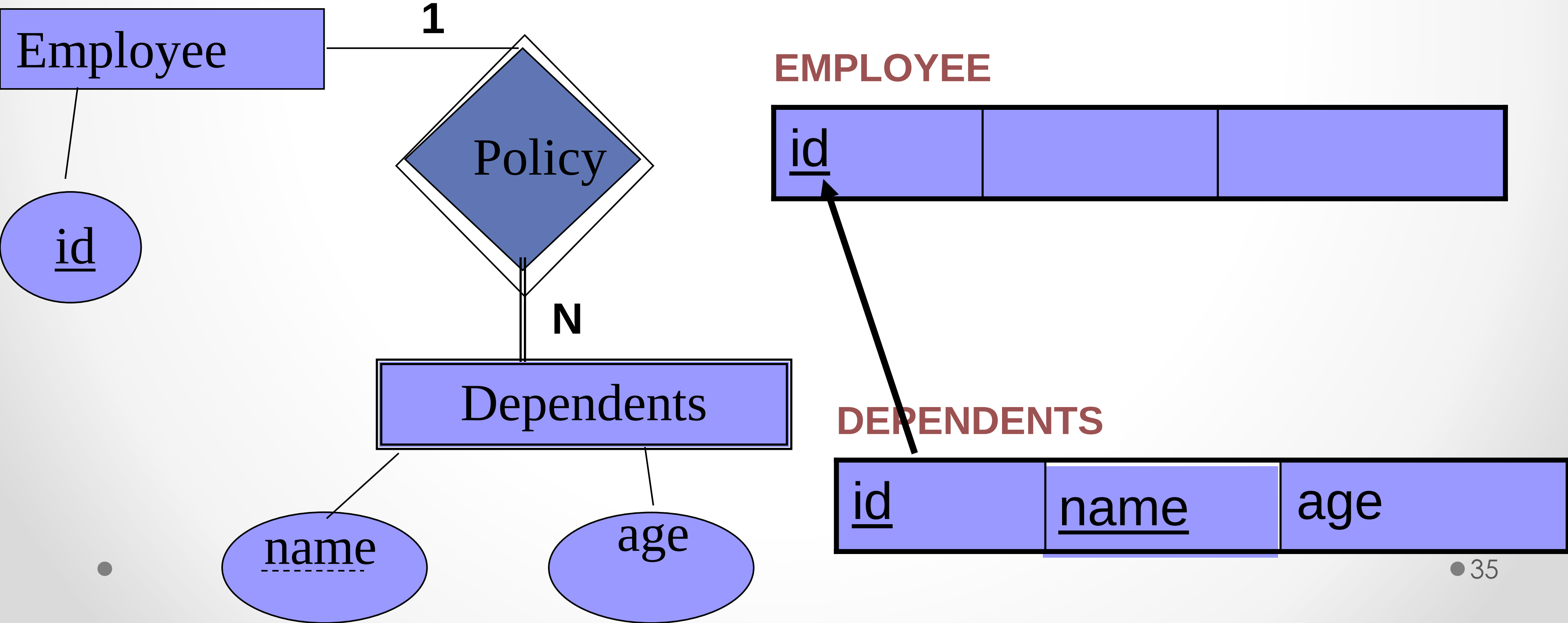
Mapping Weak Entities … (contd.)

ER Model

Weak Entity

Relational Model

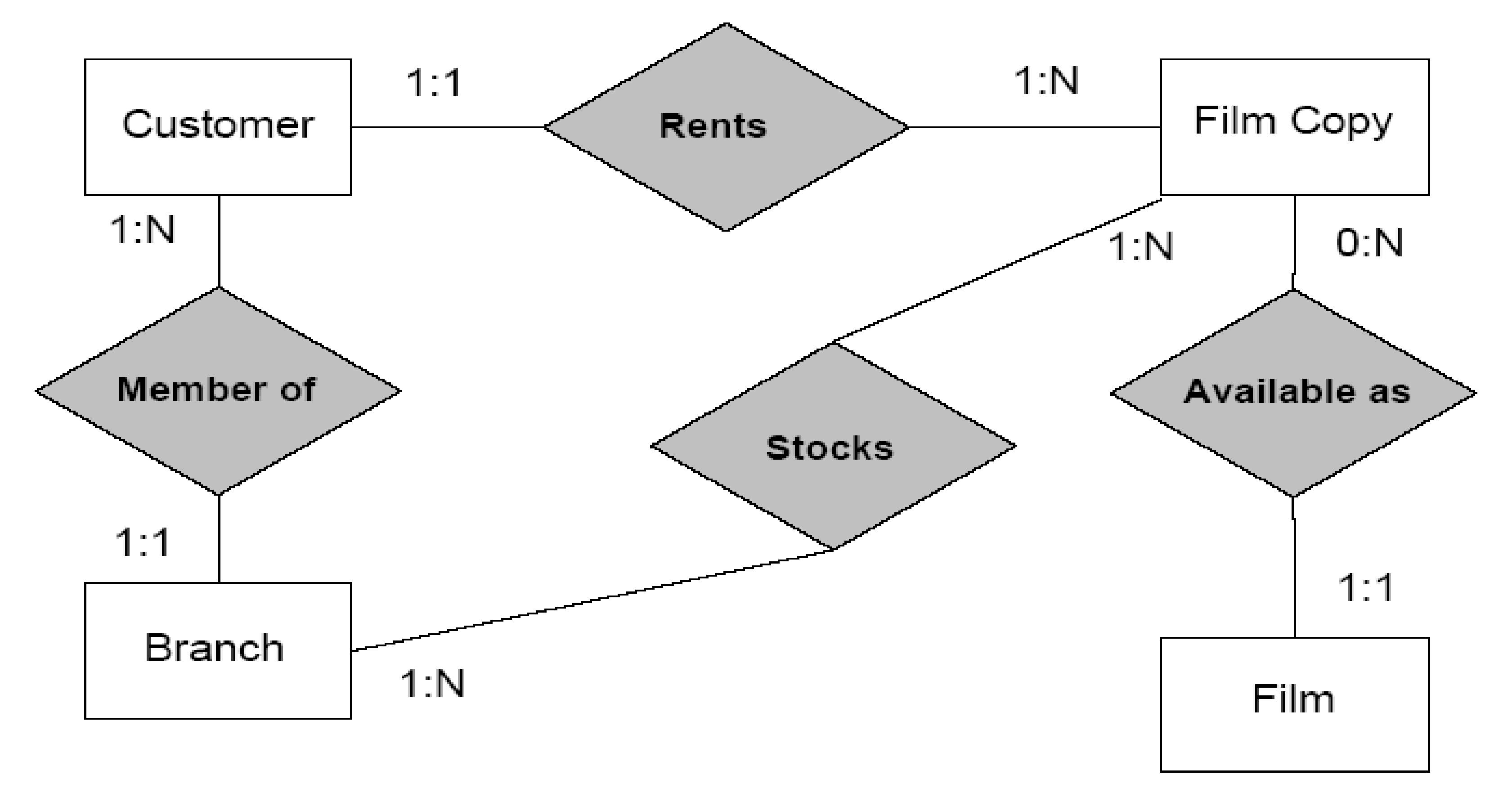
Relations and combination of partial and primary keys



Mapping - Summary

ER Model		**Relational Model**
Entity (strong)	->	Relation
Simple Attributes	->	Attributes
Primary Key	->	Primary Key
Composite attributes	->	Set of simple attributes
1:1 or 1:N relationship	->	Foreign keys
M:N relationship	->	Relation and foreign keys
Multivalued attribute	->	Relation and foreign key
N-ary relationship	->	Relation and n foreign keys
Weak Entity	>	Relation and combination of primary and partial keys

Your Turn


Customer
1:1
Rents
1:N
Film Copy
1:N
1:N
0:N
Member of
Stocks
Available as
1:1
1:1
Branch
1:N
Film

Summary

- What is logical database design?.

- How to derive a logical model from the information represented in the ER model (conceptual model)

- We focus on one type of logical model which is relational model

Next Lecture

Creating Tables

Q & A

Thank You.