"Those who cannot remember the past are doomed to repeat it"

DATA MODELS II

6.830 / 6.814 LECTURE 3

TIM KRASKA

OTHER DATA MODELS

- Object-Oriented Database Systems (OODB)
- Object-Relational Database Systems (ORDB)
- Document stores
- XML Database Systems / Xquery
- RDF
- Key/Value Stores

MODELING RELATIONS

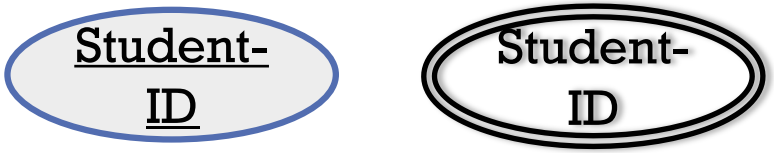
ENTITY/RELATIONSHIP (ER) MODEL

Entity

Relationship

Attribute

Key



Role

ENTITY/RELATIONSHIP (ER) MODEL

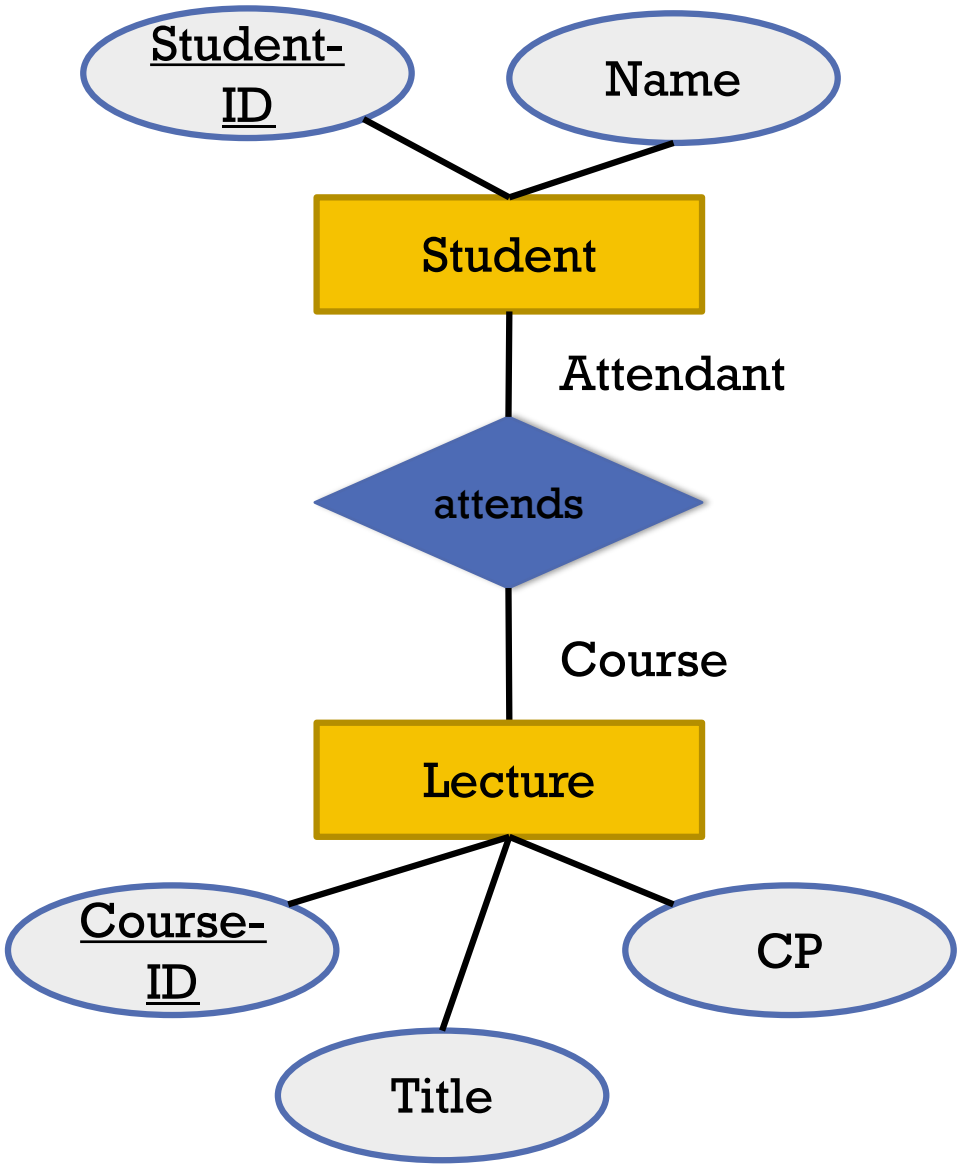
Entity

Relationship

Attribute

Key

Role



WHY ERM

Advantages

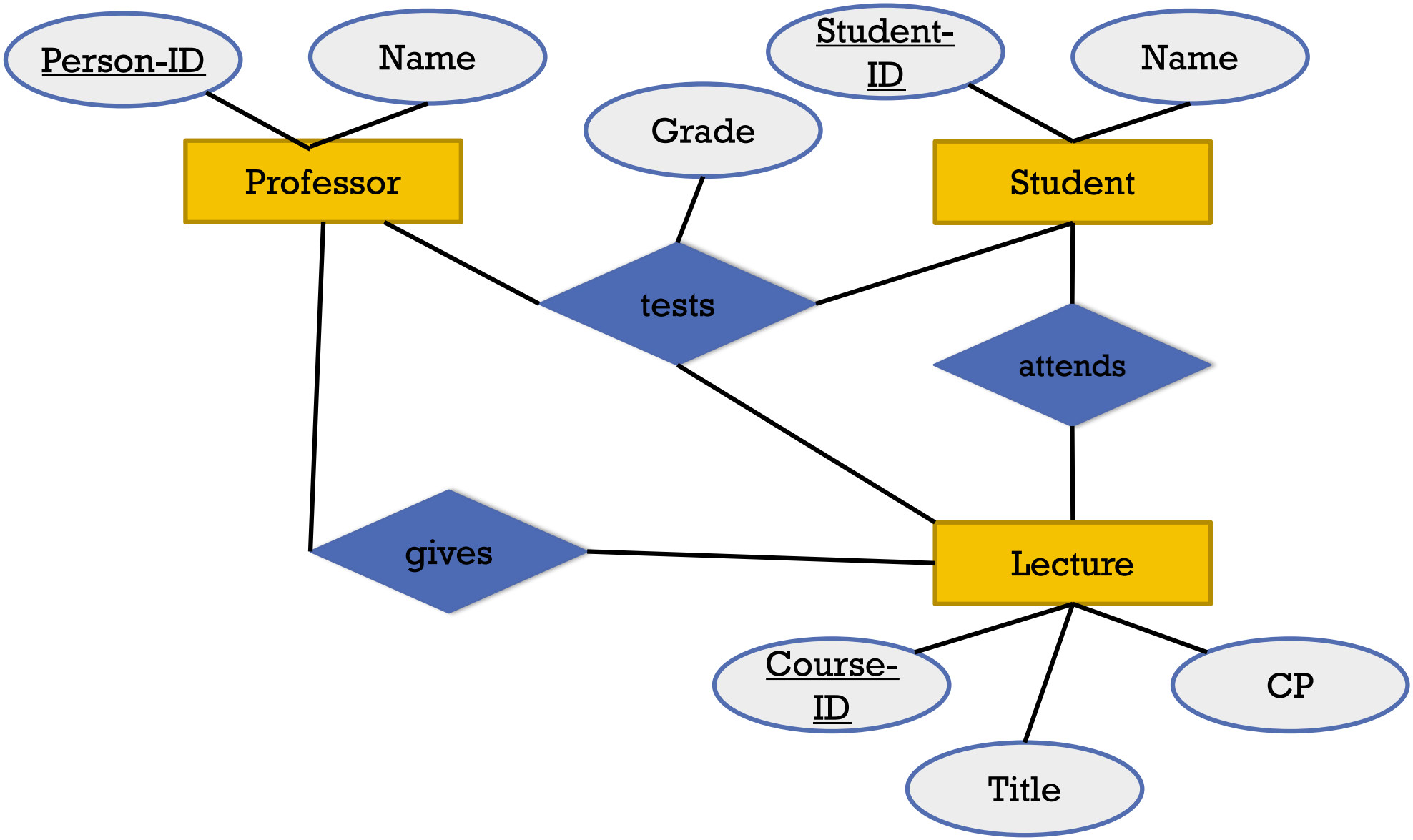
- ER diagrams are easy to create
- ER diagrams are easy to edit
- ER diagrams are easy to read (from the layman)
- ER diagrams express all information requirements

Other aspects

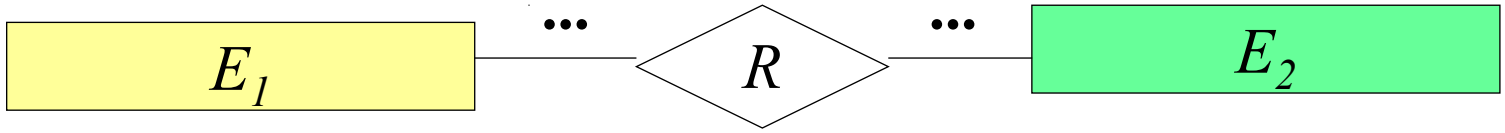
- Minimality
- Tools (e.g., Visio)
- Graphical representation

General

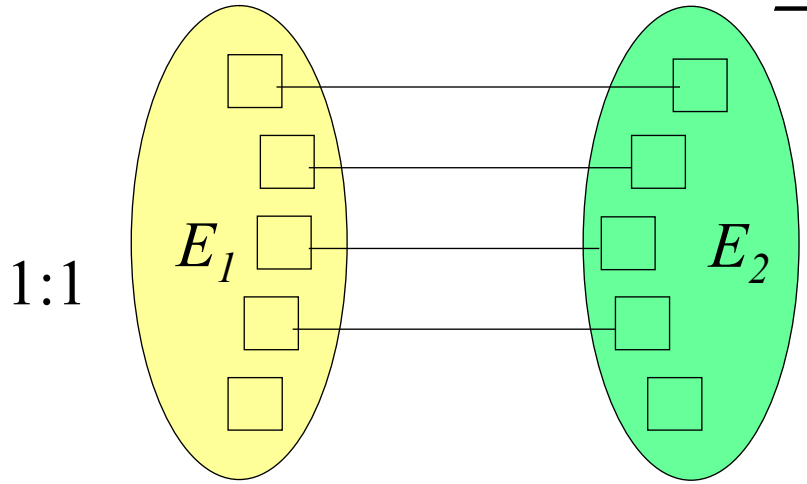
- Try to be concise, complete, comprehensible, and correct
- Controversy whether ER/UML is useful in practice
- No controversy that everybody needs to learn ER/UML

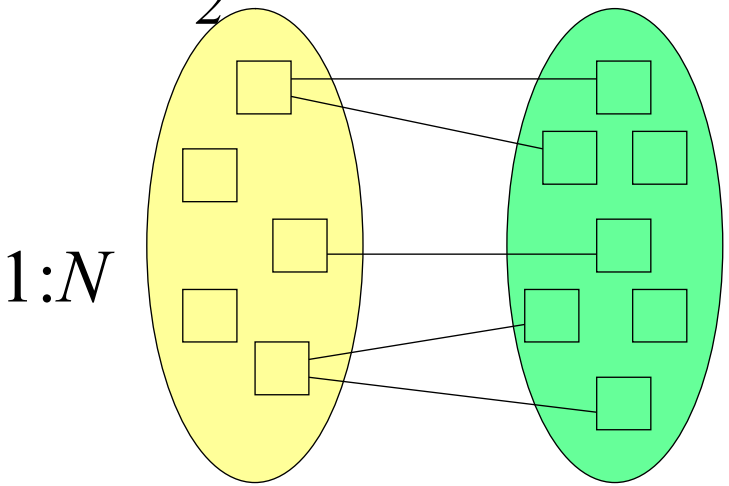

Person-ID
Name
Student-ID
Name
Grade
Professor
Student
tests
attends
gives
Lecture
Course-ID
CP
Title

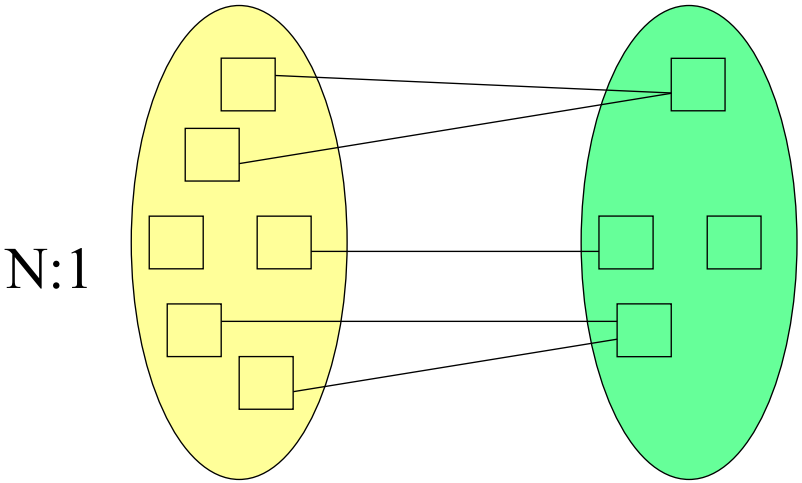
FUNCTIONALITIES

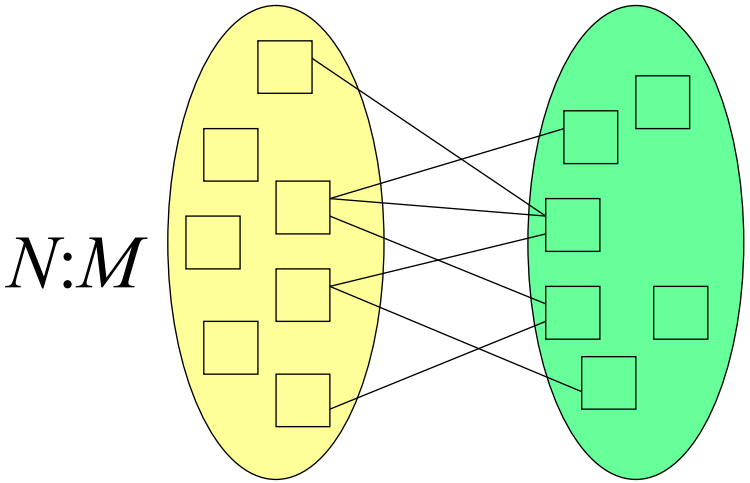


$$R \subseteq E_1 \times E_2$$

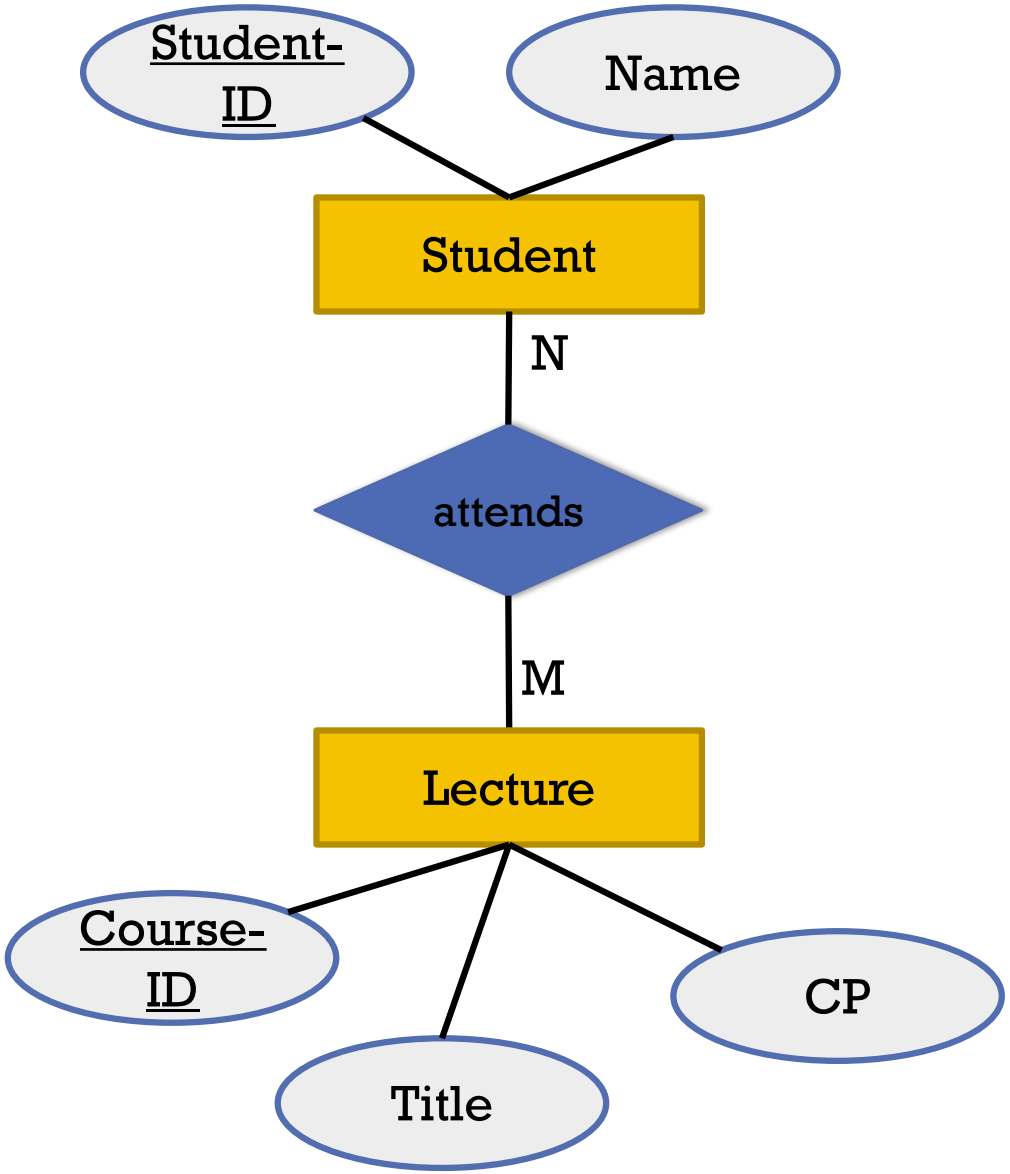




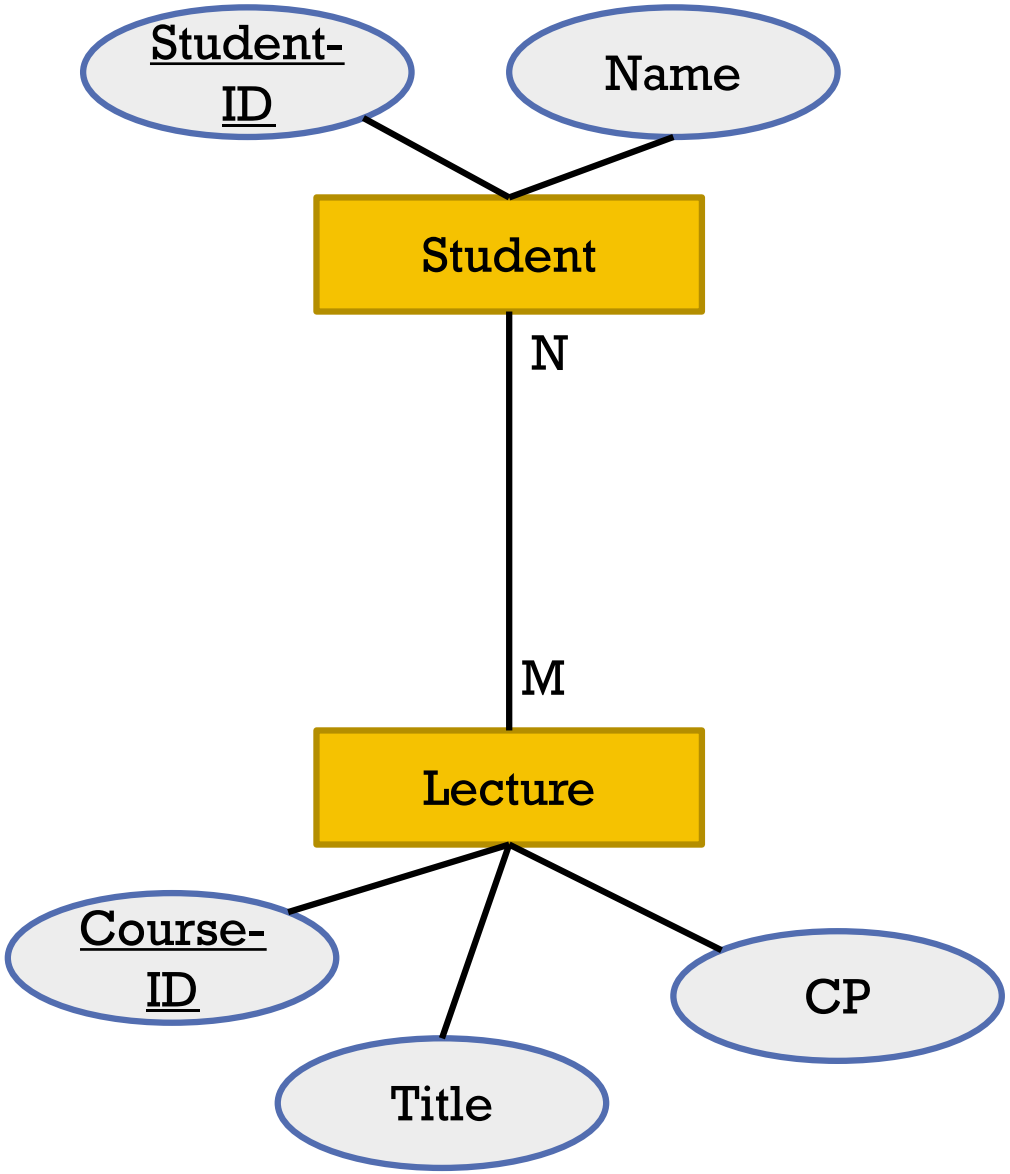




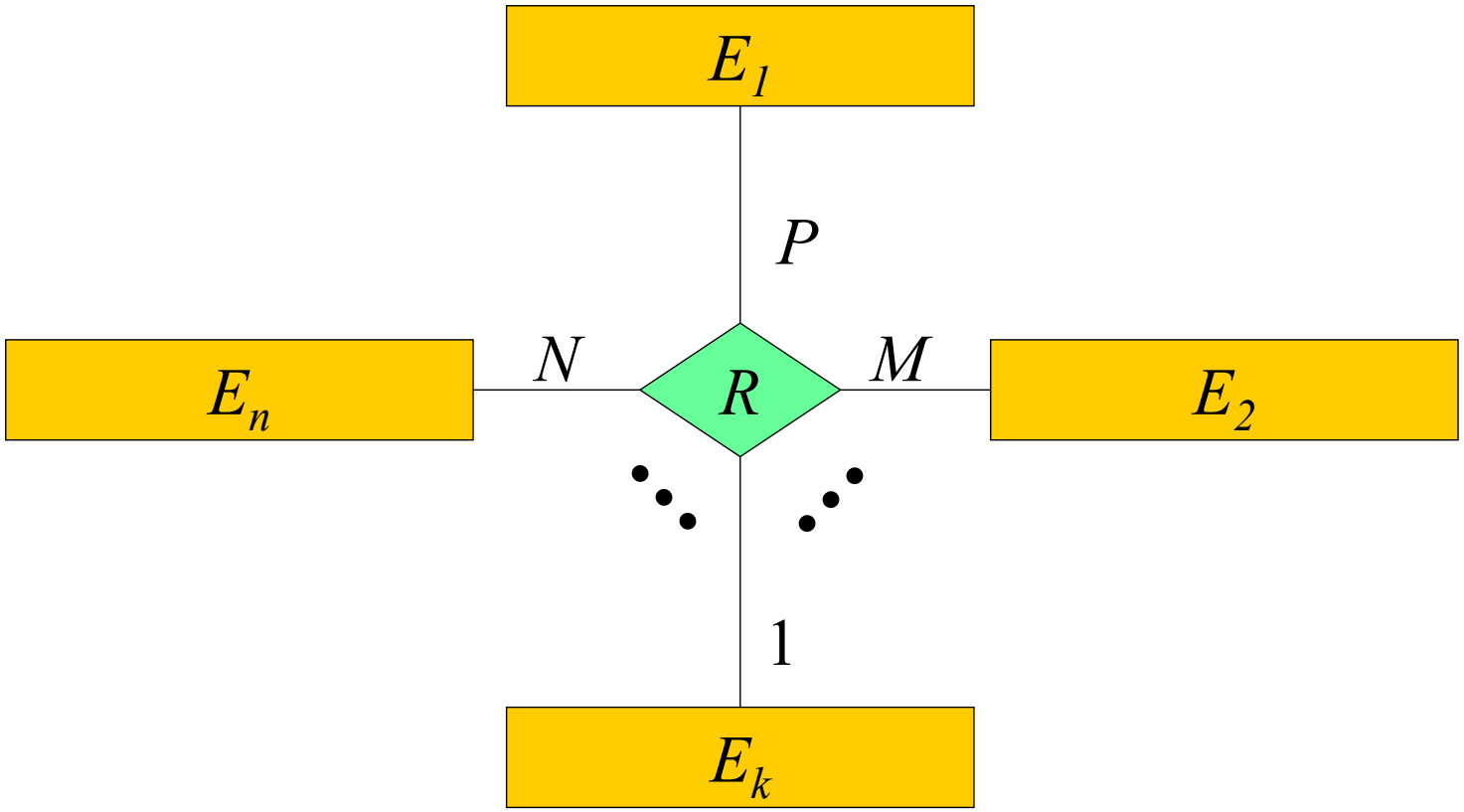
EXAMPLE: PROFESSOR <-> LECTURE



SOMETIMES ALSO SHOWN AS

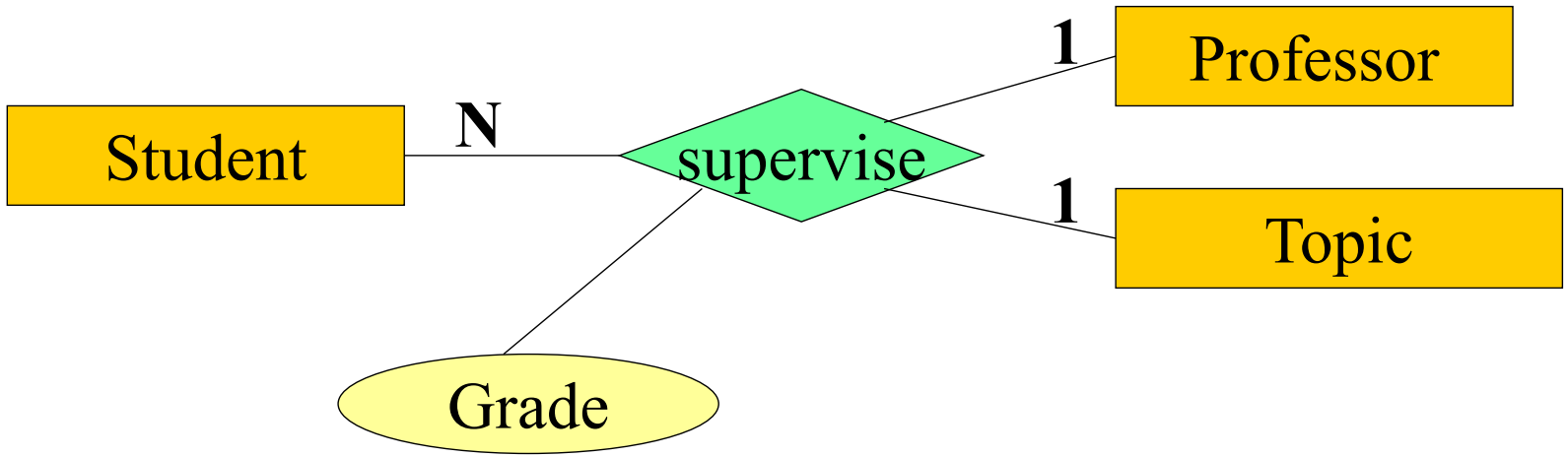


FUNCTIONALITIES OF N-ARY RELATIONSHIPS



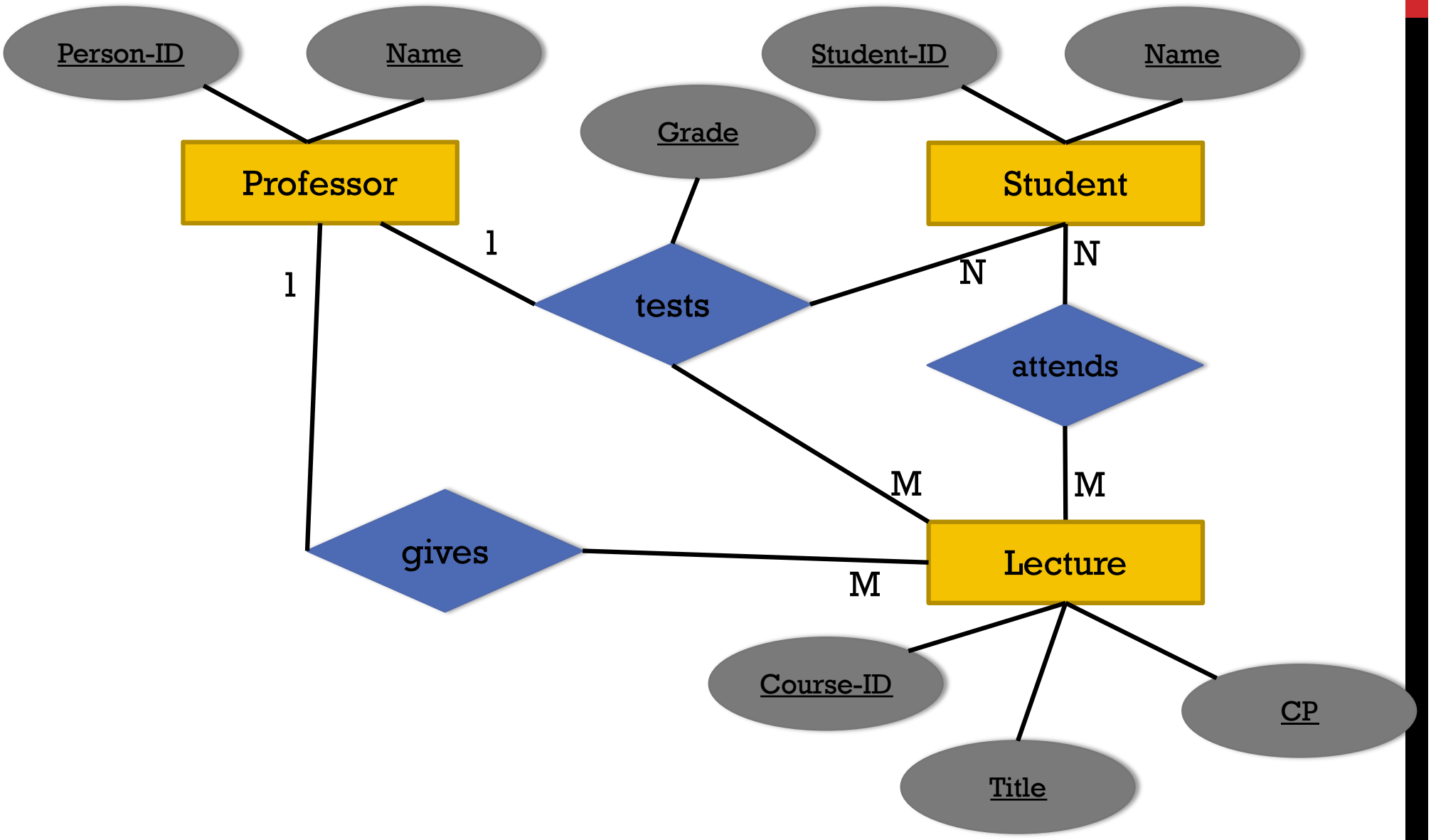
$$R : E_1 \text{ x } \ldots \text{ x } E_{k\text{-}1} \text{ x } E_{k+1} \text{ x } \ldots \text{ x } E_n \rightarrow E_k$$

EXAMPLE CS MASTER THESIS



supervise : Professor x Student → Topic

supervise : Topic x Student → Professor


Person-ID
Name
Student-ID
Name
Grade
Professor
Student
1
1
N
N
tests
attends
M
M
gives
Lecture
M
Course-ID
CP
Title

QUESTION

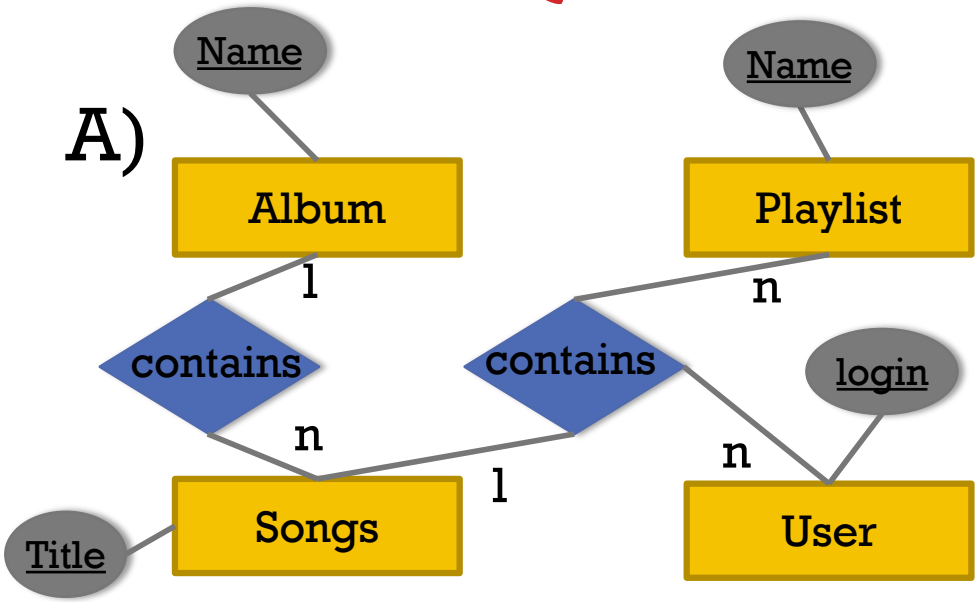
Model a music record database

- An album has a unique name and songs have unique titles
- An album contains several songs
- A playlist has a unique name and is created by one user with a unique login
- A playlist contains several songs from potential different albums

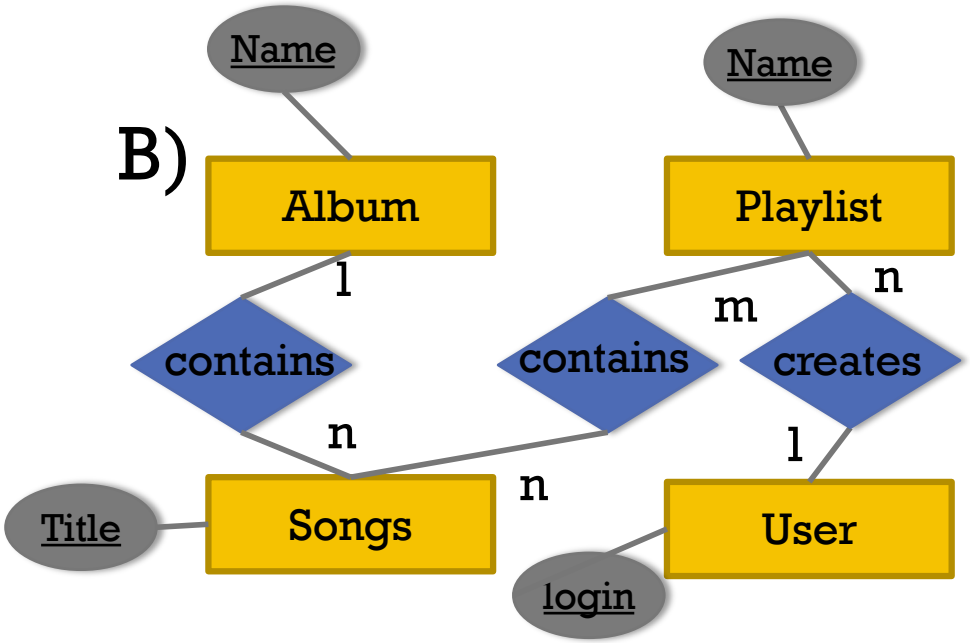
CLICKER QUESTION

http://clicker.csail.mit.edu/6.814/

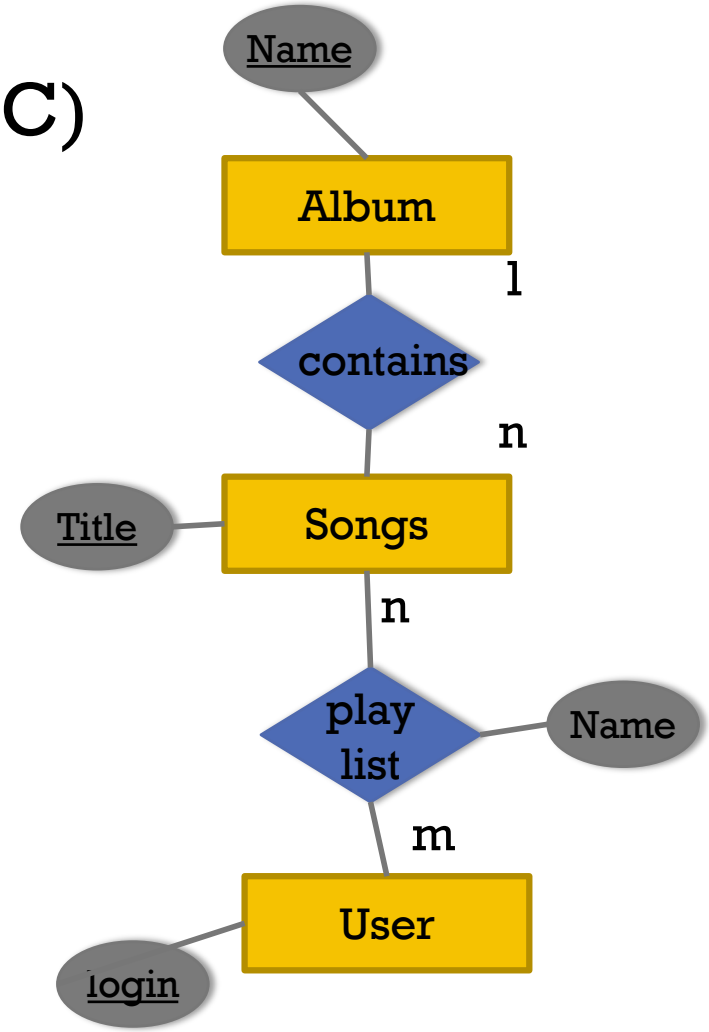
A)



B)



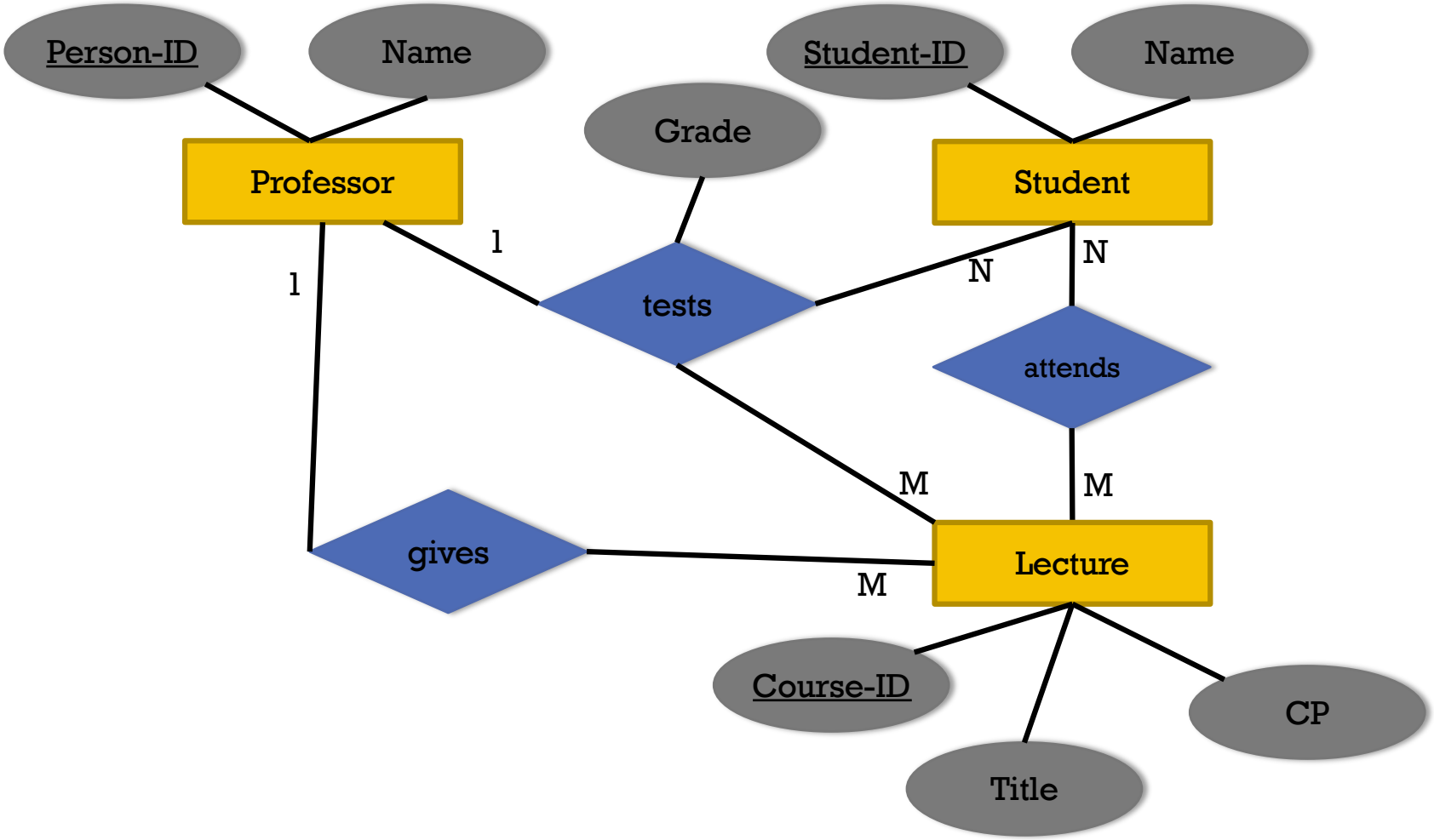
C)



ATTRIBUTE VS ENTITY

Should the *grade* be an entity or attribute?

Should test be an entity or relationship?



RULES OF THUMB

Attribute vs. Entity

- Entity if the concept has more than one relationship
- Attribute if the concept has only one 1:1 relationship

Partitioning of ER Models

- Most realistic models are larger than a page
- Partition by domains (library, research, finances, ...)

Good vs. Bad models

- Do not model redundancy or tricks to improve performance
- Less entities is better (the fewer, the better!)
- Remember the 5 C's (clear, concise, correct, complete, compliant)

LIMITATIONS OF ERM

LIMITATIONS OF ERM

ER has no formal semantics

- unclear whether this is a bug or a feature
- (natural language has no formal semantics either)

No way to express relationships between sets of entities

- e.g., existence of person depends on a set of organs
- sets of sets are notoriously hard to model
- (more on that when we talk about 4 NF)

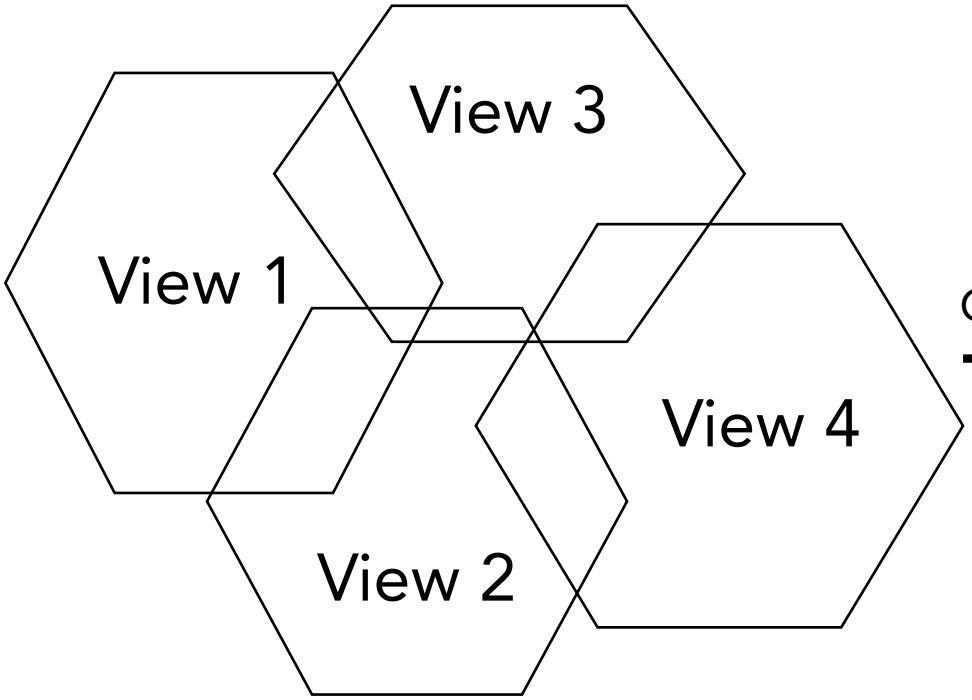
No way to express negative rules

- e.g., same entity cannot be an Assistant and Professor
- again, negation notoriously hard (e.g., 2nd-order logic)

ER has been around for 30+ years

- maybe, ER hit sweet spot of expressivity vs. simplicity
- (UML class diagrams inherit same weaknesses)

WHY IS ER MODELLING SO DIFFICULT?



Consolidate

Global Schema

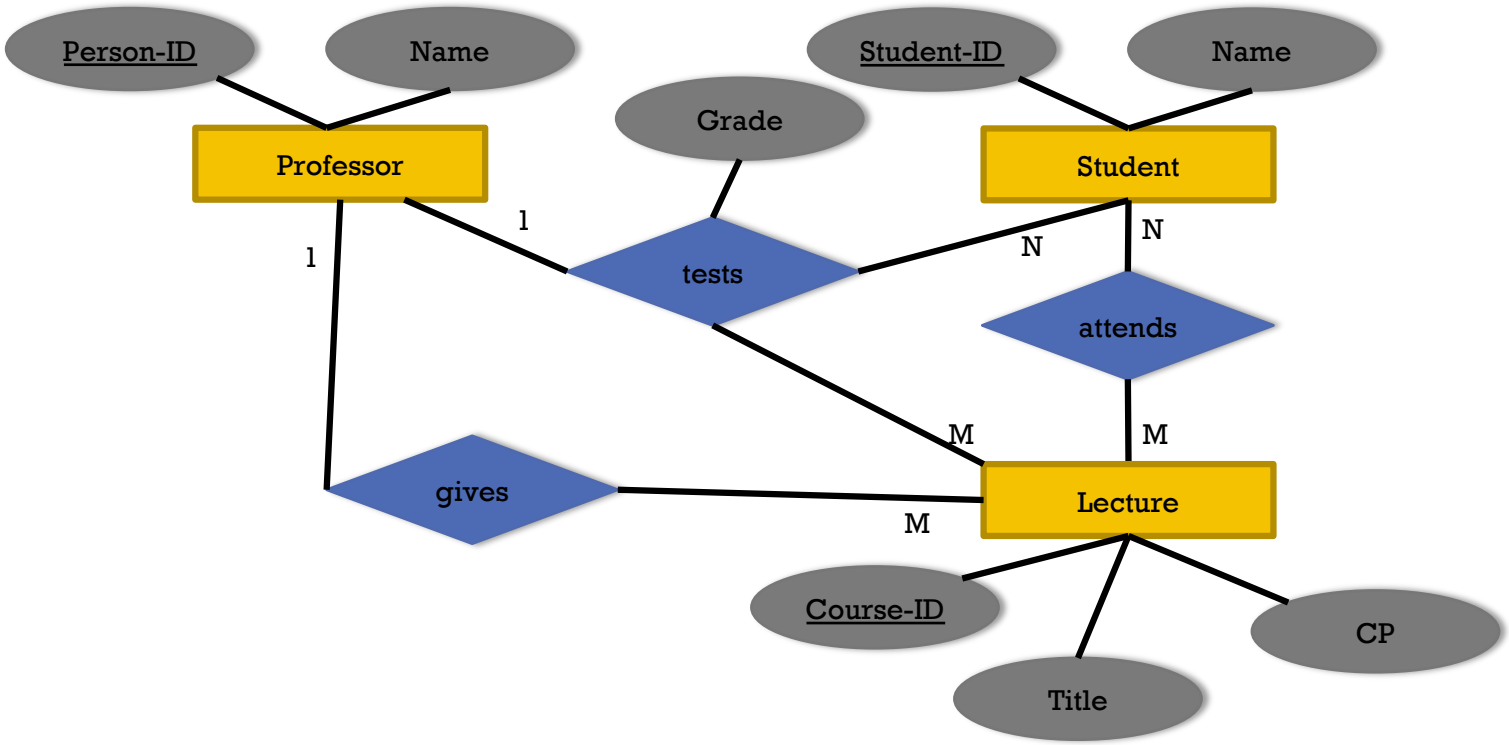
- **No redundancy**
- **No conflicts**
- **Avoid synonyms**
- **Avoid homonyms**

Where is the current research going?

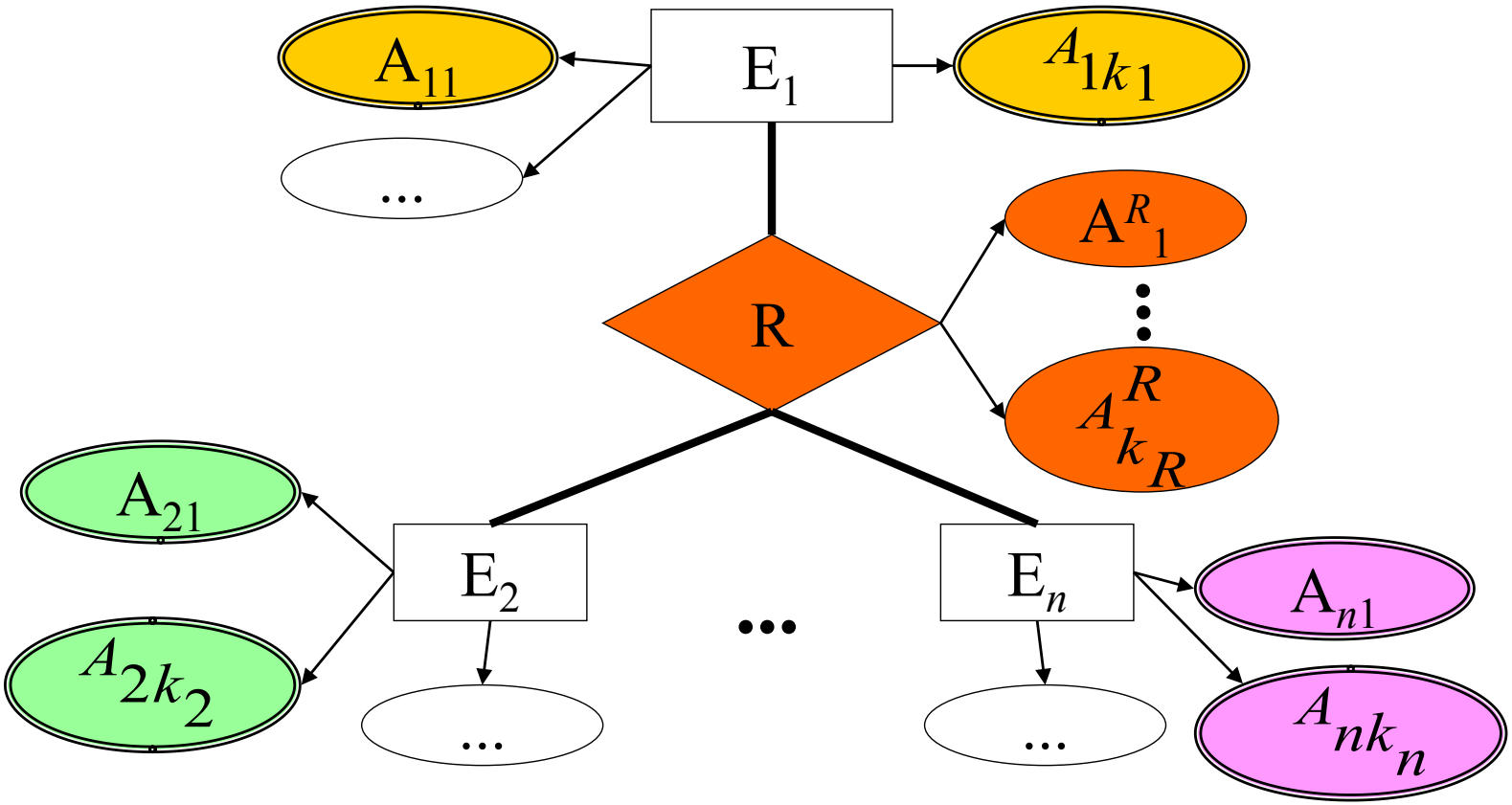
ERM TO RELATIONS

RULE #1: ENTITIES

Professor(Person-ID:integer, Name:string)
Student(Student-ID:integer, Name:string)
Lecture(Course-ID:string, Title:string, CP:float)



RULE #2: RELATIONSHIPS



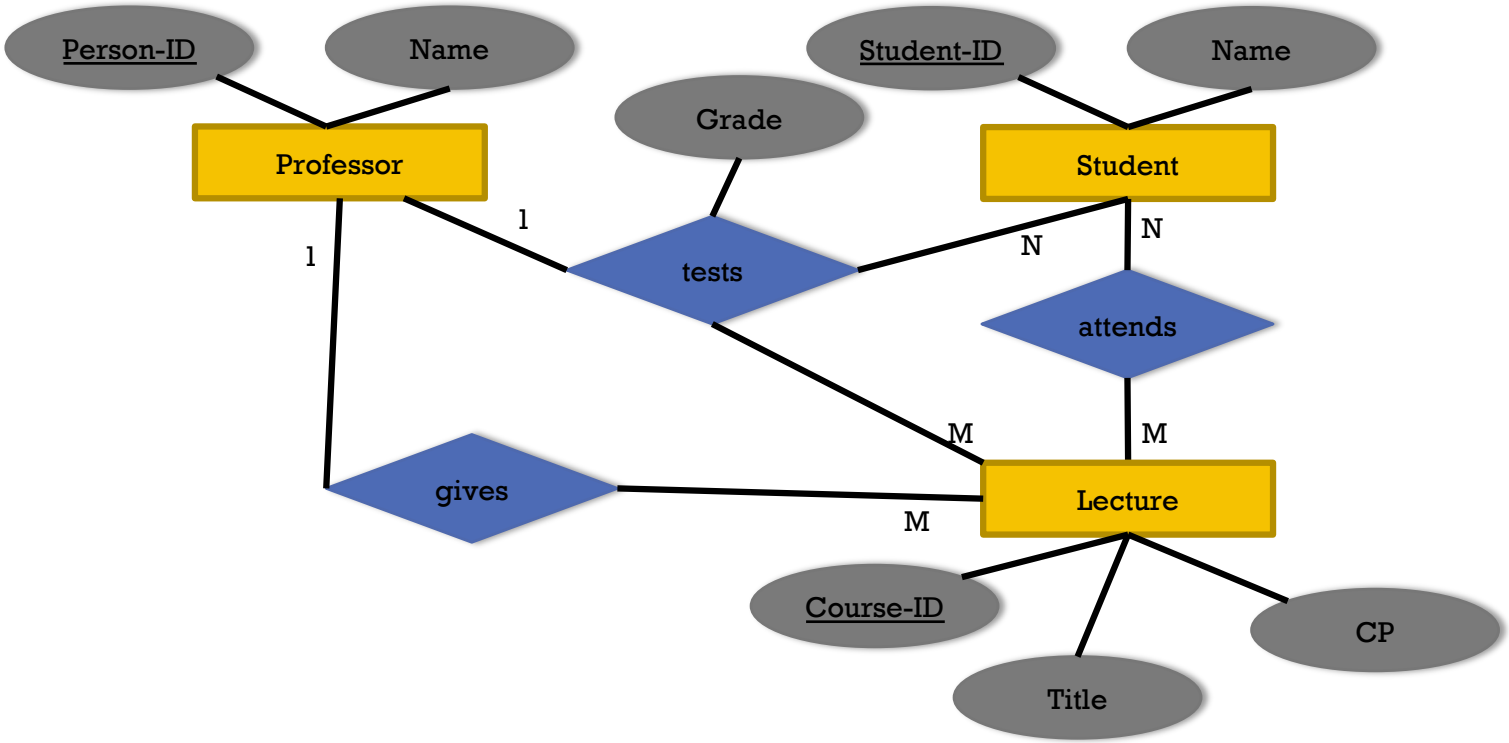
$$R: \{[A_{11},\ldots,A_{1k_1}, A_{21},\ldots,A_{2k_2}, \ldots, A_{n1},\ldots,A_{nk_n}, A_1^R,\ldots,A_{k_R}^R]\}$$

Key of E_1 | Key of E_2 | Key of E_n | Attributes of R

RULE #2: RELATIONSHIPS

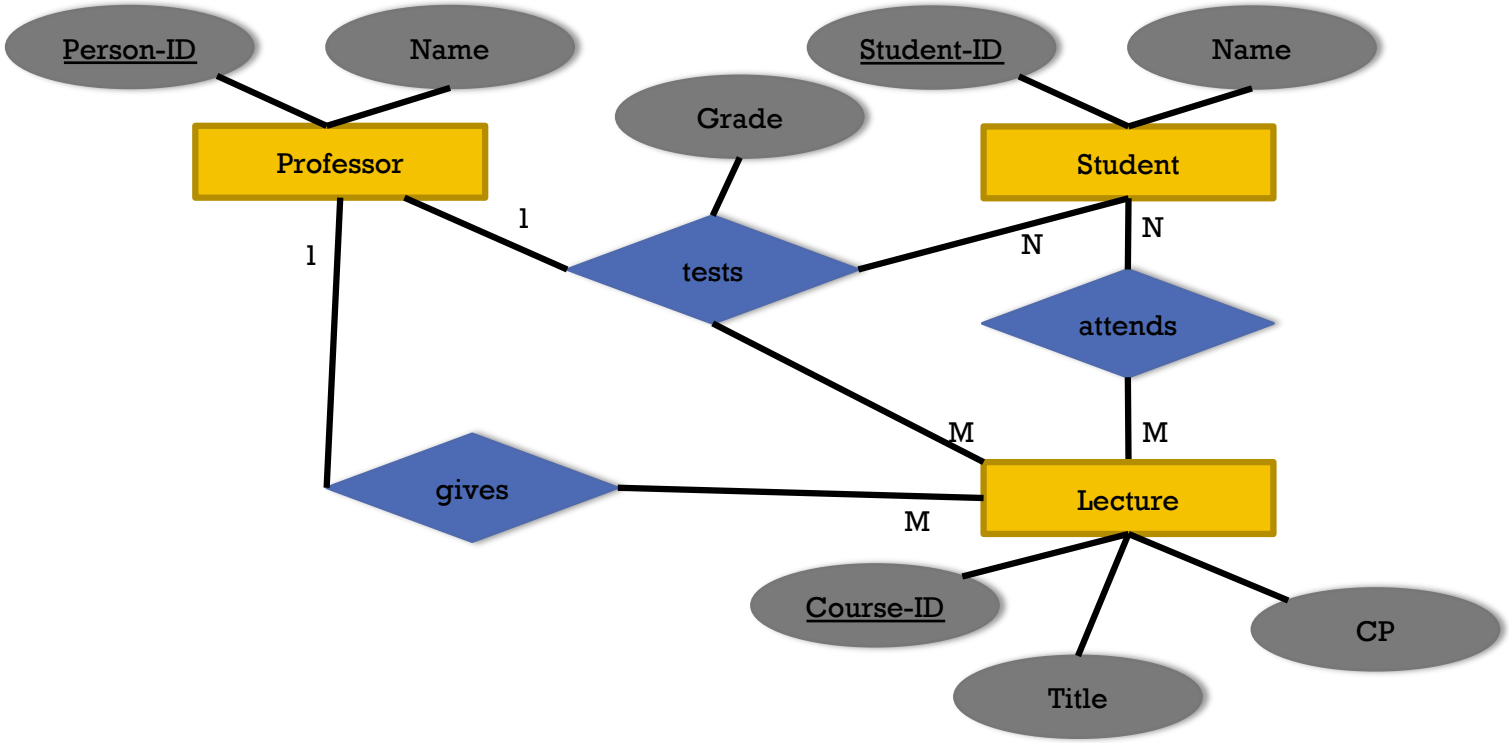
Professor(Person-ID:integer, Name:string)
Student(Student-ID:integer, Name:string)
Lecture(Course-ID:string, Title:string, CP:float)
Gives(Person-ID:integer, Course-ID:string)
Attends(Student-ID:integer, Course-ID:string)
Tests(Student-ID:integer, Course-ID:string, Person-ID:integer, Grade:String)

What about keys?



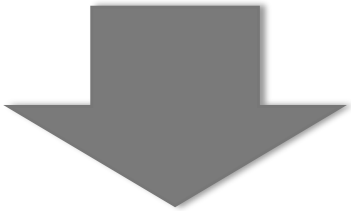
RULE #2: RELATIONSHIPS

Professor(Person-ID:integer, Name:string)
Student(Student-ID:integer, Name:string)
Lecture(Course-ID:string, Title:string, CP:float)
Gives(Person-ID:integer, Course-ID:string)
Attends(Student-ID:integer, Course-ID:string)
Tests(Student-ID:integer, Course-ID:string, Person-ID:integer, Grade:string)



RULE #3: MERGE RELATIONS WITH THE SAME KEY

Professor(Person-ID:integer, Name:string)
Lecture(**Course-ID:string**, Title:string, CP:float)
Gives(Person-ID:integer, **Course-ID:string**)



Professor(Person-ID:integer, Name:string)
Lecture(Course-ID:string, Title:string, CP:float, Person-ID:integer)

FINAL

Professor(Person-ID:integer, Name:string)

Student(Student-ID:integer, Name:string)

Lecture(Course-ID:string, Title:string, CP:float,
Person-ID:integer)

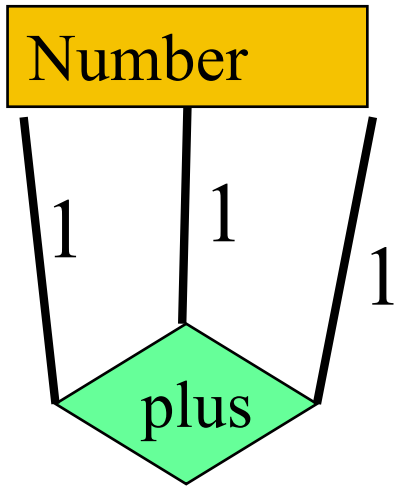
Attends(Student-ID:integer, Course-ID:string)

Tests(Student-ID:integer, Course-ID:string,
Person-ID:integer, Grade:string)

Why didn't we merge **Attends** and **Tests**?

EXERCISE

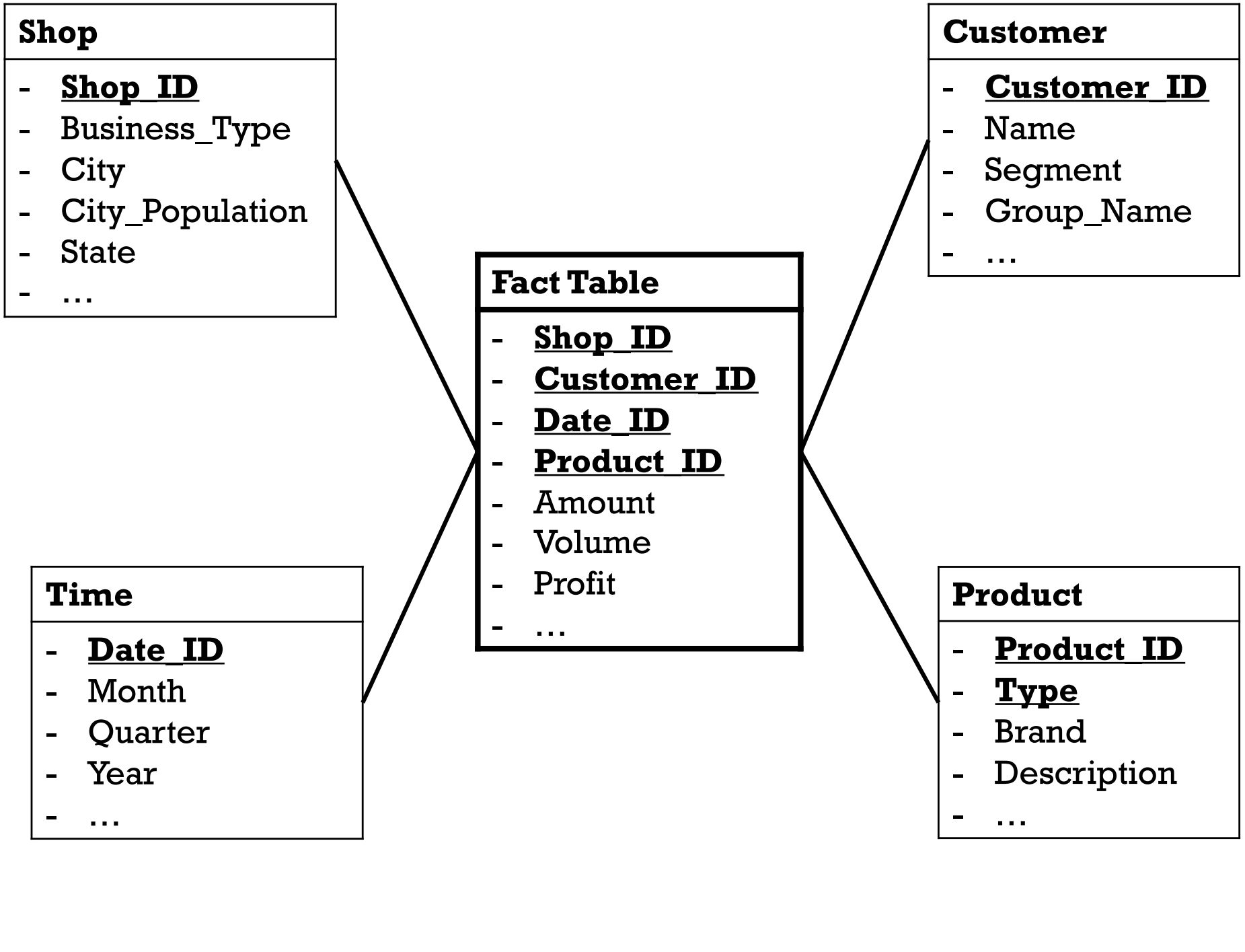
Implement the following ER diagram using the rel. data model

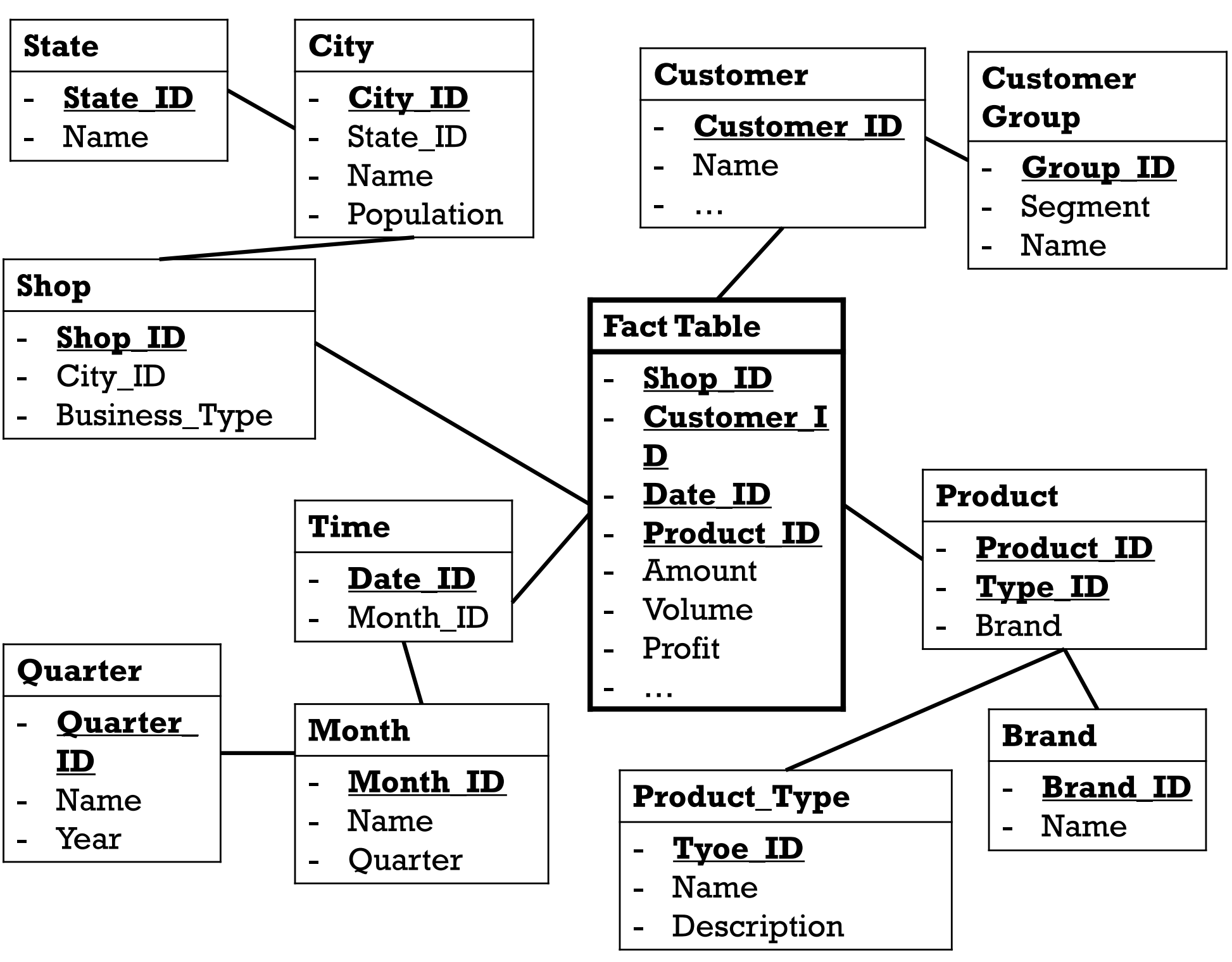


PROBLEM

- **You are the new Data Scientist at Evil Market**
- Evil Market is tracking all customer purchases with their membership card or credit card
- They also have data about their customers (estimated income, family status,…)
- Recently, they are trying to improve their image for young mothers
- As a start they want to know the following information for mothers under 30 for 2013:
  - How much do they spend?
  - How much do they spend per state?
  - How does this compare to all customers under 30?
  - What are their favorite products?
  - How much do they spend per year?

Your first project: Design the schema for Evil Market!


Shop
- Shop_ID
- Business_Type
- City
- City_Population
- State
- …
Customer
- Customer_ID
- Name
- Segment
- Group_Name
- …
Fact Table
- Shop_ID
- Customer_ID
- Date_ID
- Product_ID
- Amount
- Volume
- Profit
- …
Time
- Date_ID
- Month
- Quarter
- Year
- …
Product
- Product_ID
- Type
- Brand
- Description
- …


State
- State_ID
- Name
City
- City_ID
- State_ID
- Name
- Population
Customer
- Customer_ID
- Name
- …
Customer Group
- Group_ID
- Segment
- Name
Shop
- Shop_ID
- City_ID
- Business_Type
Fact Table
- Shop_ID
- Customer_ID
- Date_ID
- Product_ID
- Amount
- Volume
- Profit
- …
Time
- Date_ID
- Month_ID
Product
- Product_ID
- Type_ID
- Brand
Quarter
- Quarter_ID
- Name
- Year
Month
- Month_ID
- Name
- Quarter
Product_Type
- Tyoe_ID
- Name
- Description
Brand
- Brand_ID
- Name