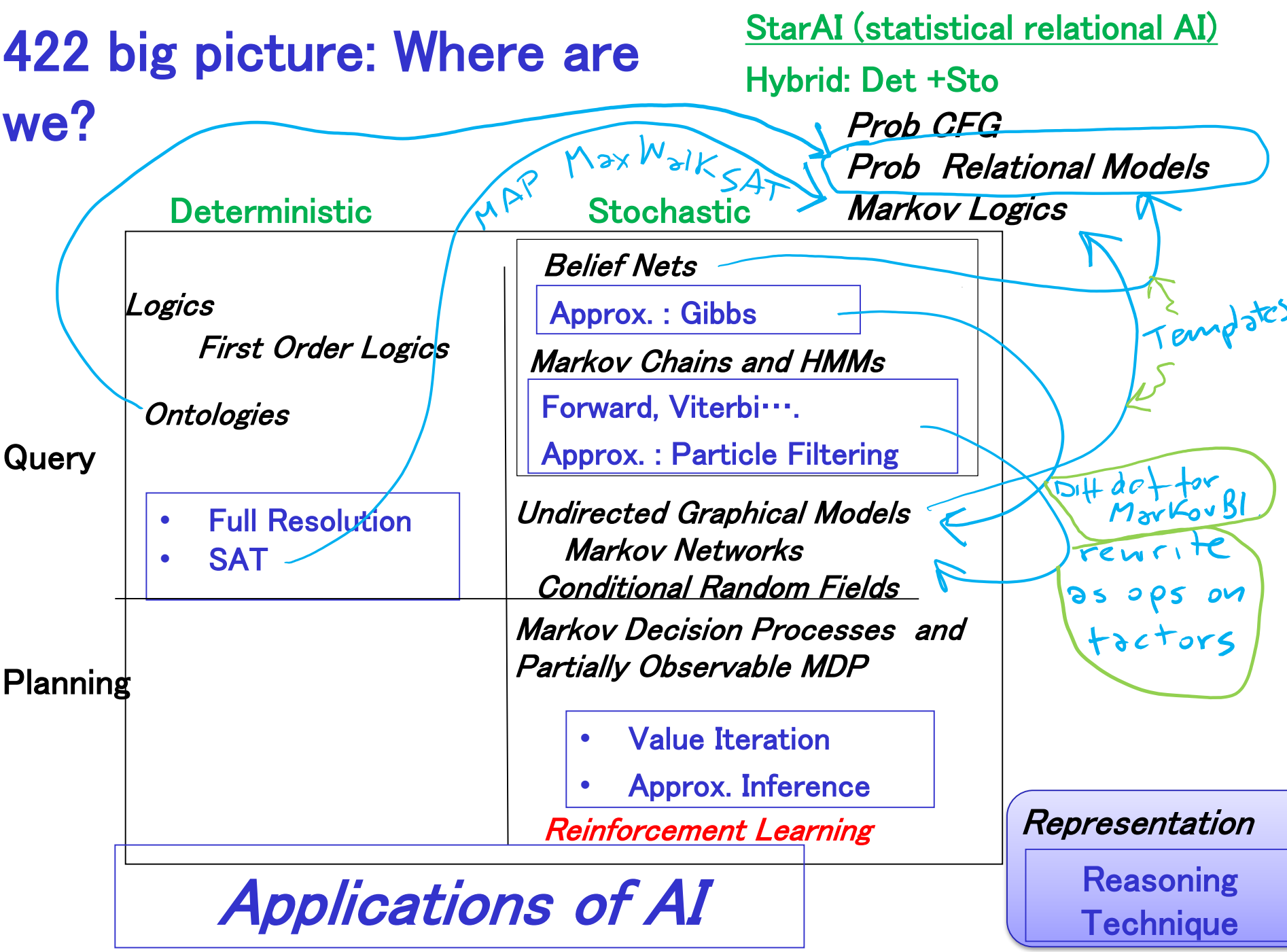
Intelligent Systems (AI-2)

Computer Science cpsc422, Lecture 32

Nov, 28, 2016

Slide source: from David Page (MIT) (which were from From Lise Getoor, Nir Friedman, Daphne Koller, and Avi Pfeffer) and from Lise Getoor

422 big picture: Where are we?



Combining Symbolic and Probabilistic R&R systems

(a) Probabilistic Context-Free Grammars

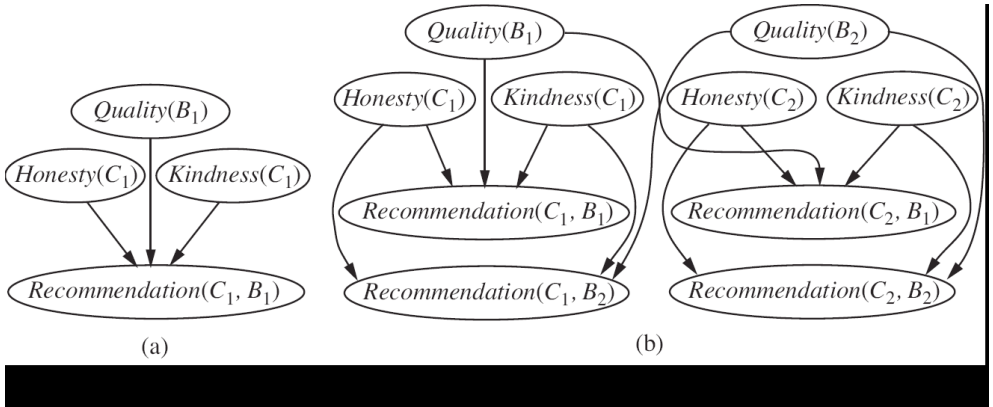
- Weights are conditional prob. on rewriting rules
- Applications: NLP parsing & Hierarchical Planning

(b) Markov Logics: weighted FOL

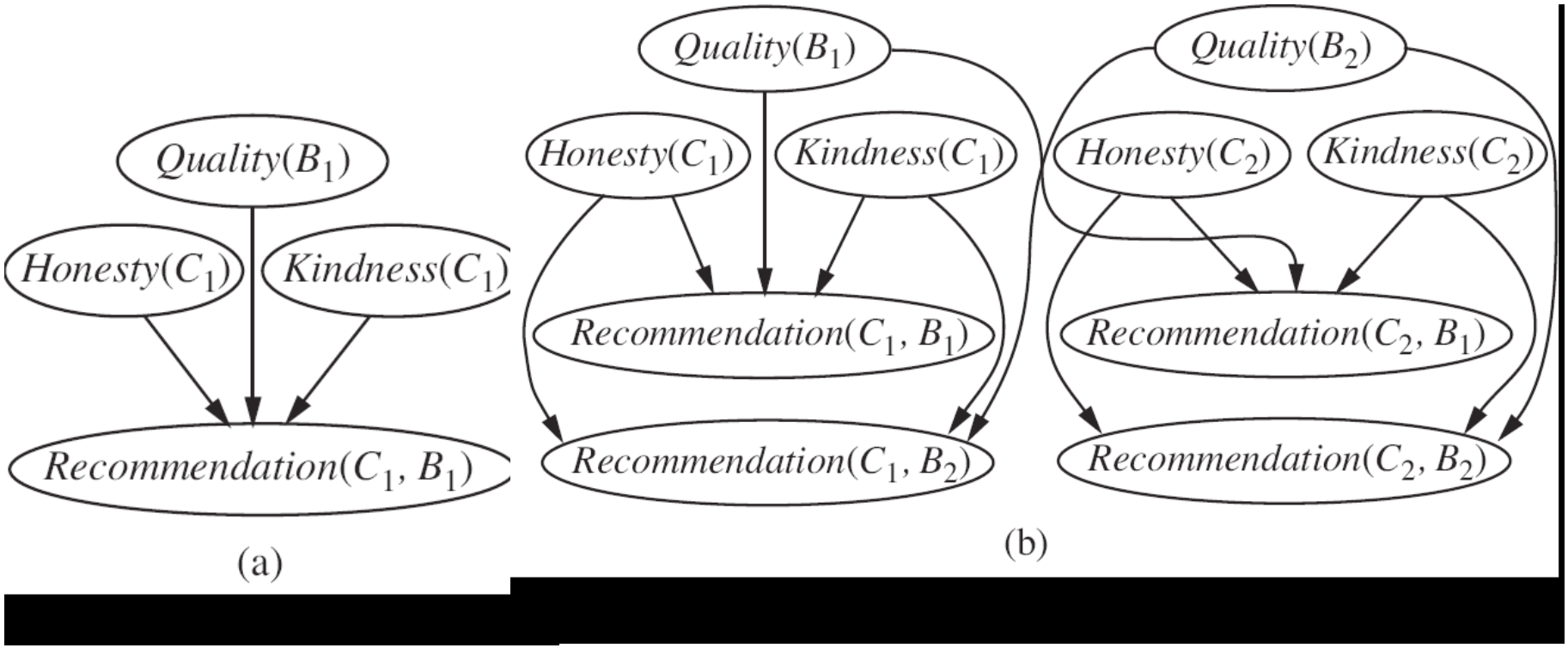
$$P(\text{world}) \propto \exp\left(\sum \text{weights of formulas it satisfies}\right)$$

(c) Probabilistic Relational models

- Probs specified on relations



Intuition for Prob. Relational models



A **customer** C1 will / will not *recommend* a **book** B1 depending on the book *quality*, and the customer *honesty* and *kindness*

When you have two customers and two books…..

drawn on White Board

Lecture Overview

- **Motivation and Representation**
- Semantics of Probabilistic Relational Models (PRMs)
  - Classes and Relations
  - Attributes and Reference Slots
  - Full Relational Schema and its Instances
  - Fixed vs. Probabilistic Attributes
  - Relational Skeleton and its Completion Instance
  - Inverse Slot and Slot chain

Motivation for PRMs

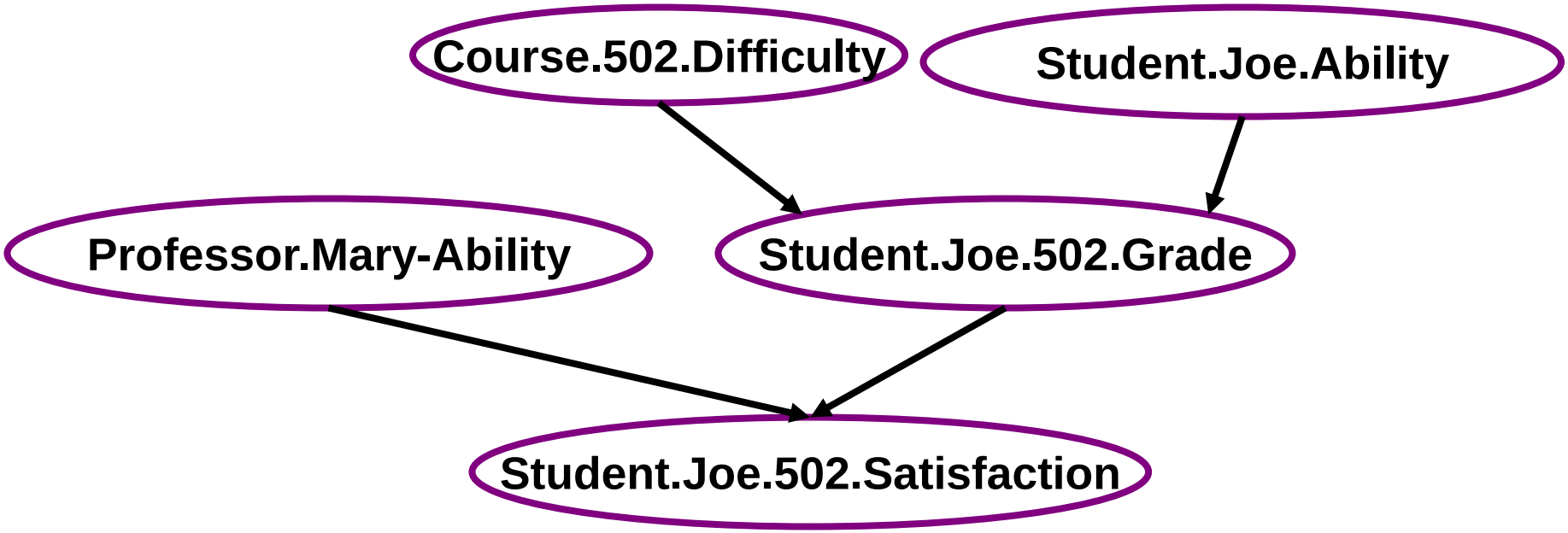
- Most real-world data are stored in relational DBMS

- Combine advantages of relational logic & Bayesian networks:
  - natural domain modeling: objects, properties, relations;
  - generalization over a variety of situations;
  - compact, natural probability models.

- Integrate uncertainty with relational model:
  - properties of domain entities can depend on properties of related entities;
  - uncertainty over relational structure of domain.

Limitations of Bayesian Networks

A Bayesian networks (BNs) represents a pre-specified set of attributes/variables whose relationship to each other is fixed in advance.



How PRMs extend BNs?

1. PRMs conceptually extend BNs to allow the specification of a probability model for *classes of objects* rather than a fixed set of simple attributes

2. PRMs also allow properties of an entity to depend probabilistically on properties of other *related entities*

Lecture Overview

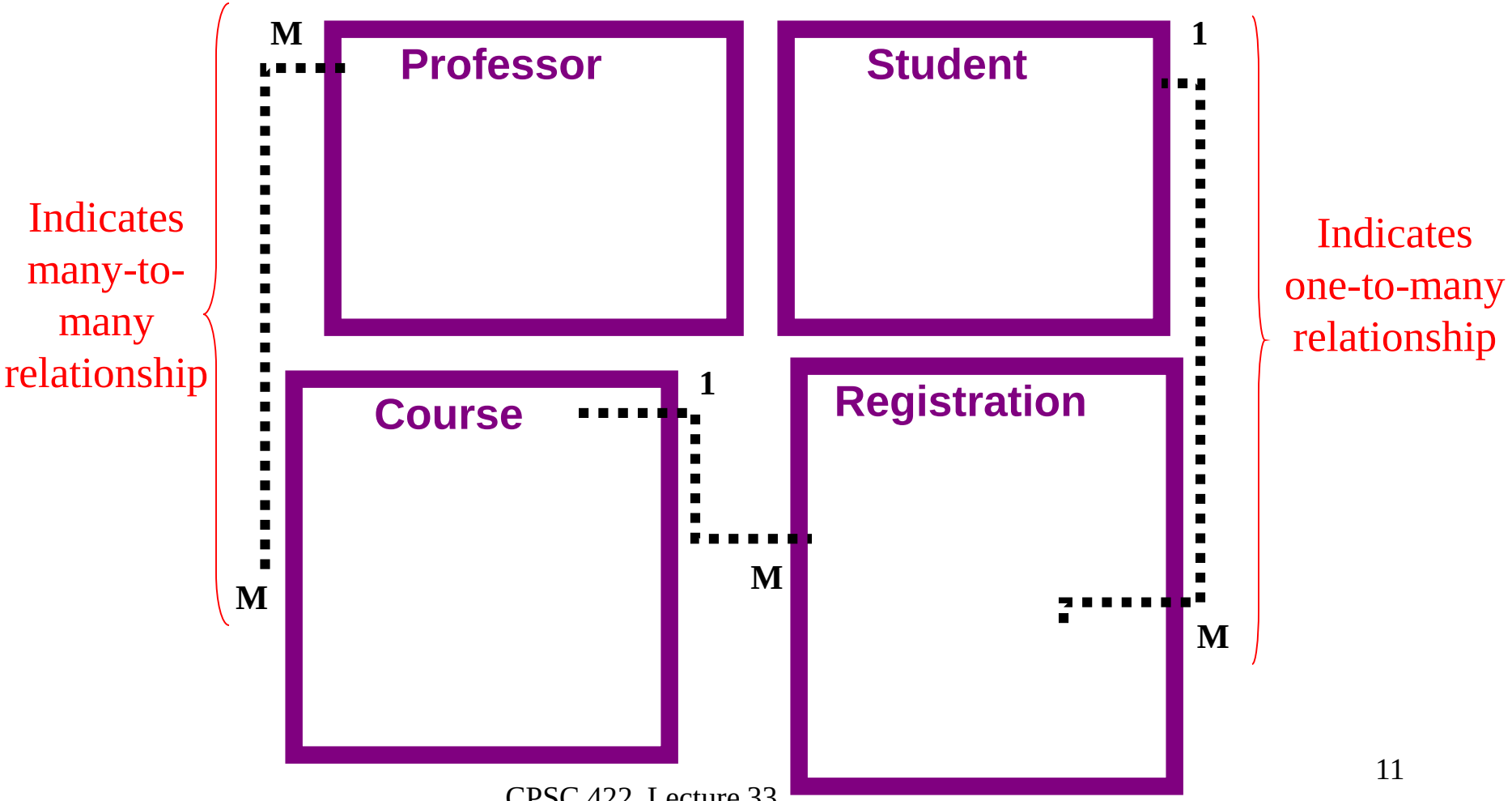
- Motivation and Representation
- **Semantics of Probabilistic Relational Models (PRMs)**
  - Classes and Relations
  - Attributes and Reference Slots
  - Full Relational Schema and its Instances
  - Fixed vs. Probabilistic Attributes
  - Relational Skeleton and its Completion Instance
  - Inverse Slot and Slot chain

Mapping PRMs from Relational Models

- **The representation of PRMs is a direct mapping from that of relational databases**

- ***A* relational model *consists of a set of classes* $X_1,\ldots,X_n$ *and a set of relations* $R_1,\ldots,R_m$, *where each relation* R_i *is typed***

University Domain Example – Classes and relations



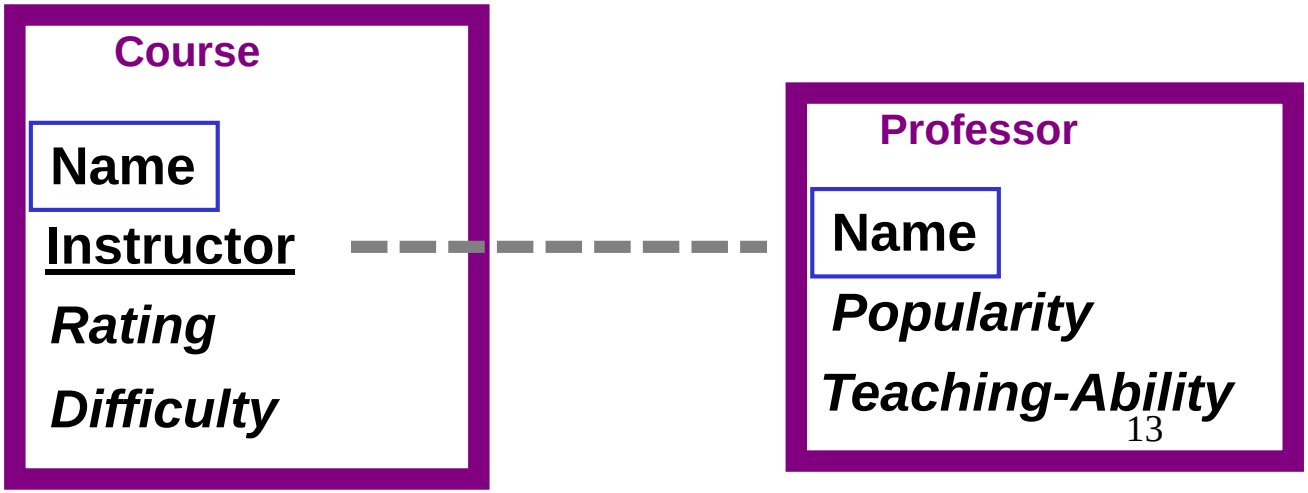
Mapping PRMs from Relational Models: attributes

- **Each class or entity type (corresponding to a single relational table) is associated with a set of *attributes* $\mathcal{A}(X_i)$ (at least one of which is a primary key)**

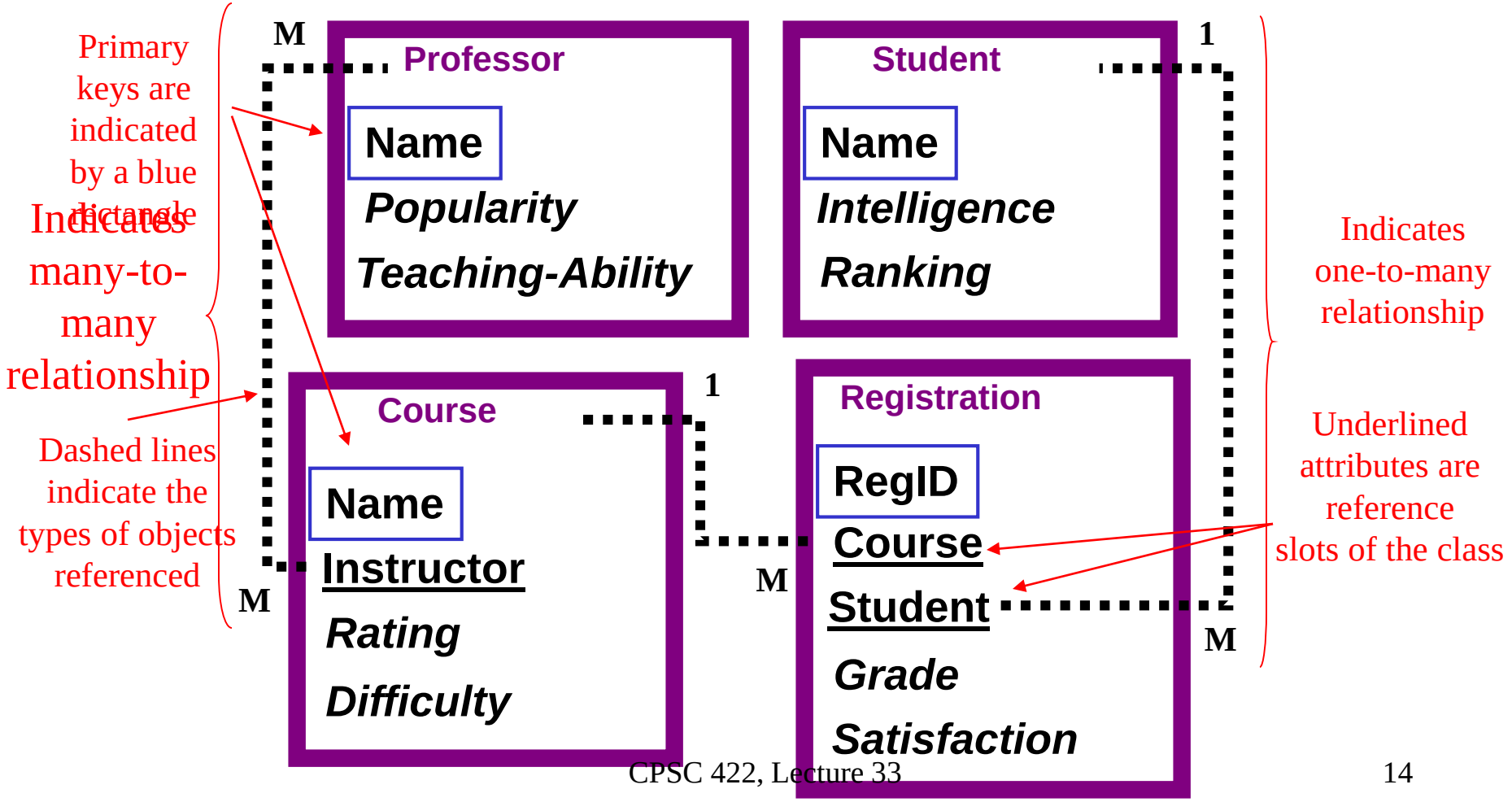
Mapping PRMs from Relational Models: reference slot

- Each class or entity type is also associated with a set of *reference slots* $\mathcal{R}(X)$
  - correspond to attributes that are *foreign keys* (key attributes of another table)
  - $X.\rho$, is used to denote reference slot ρ of X.

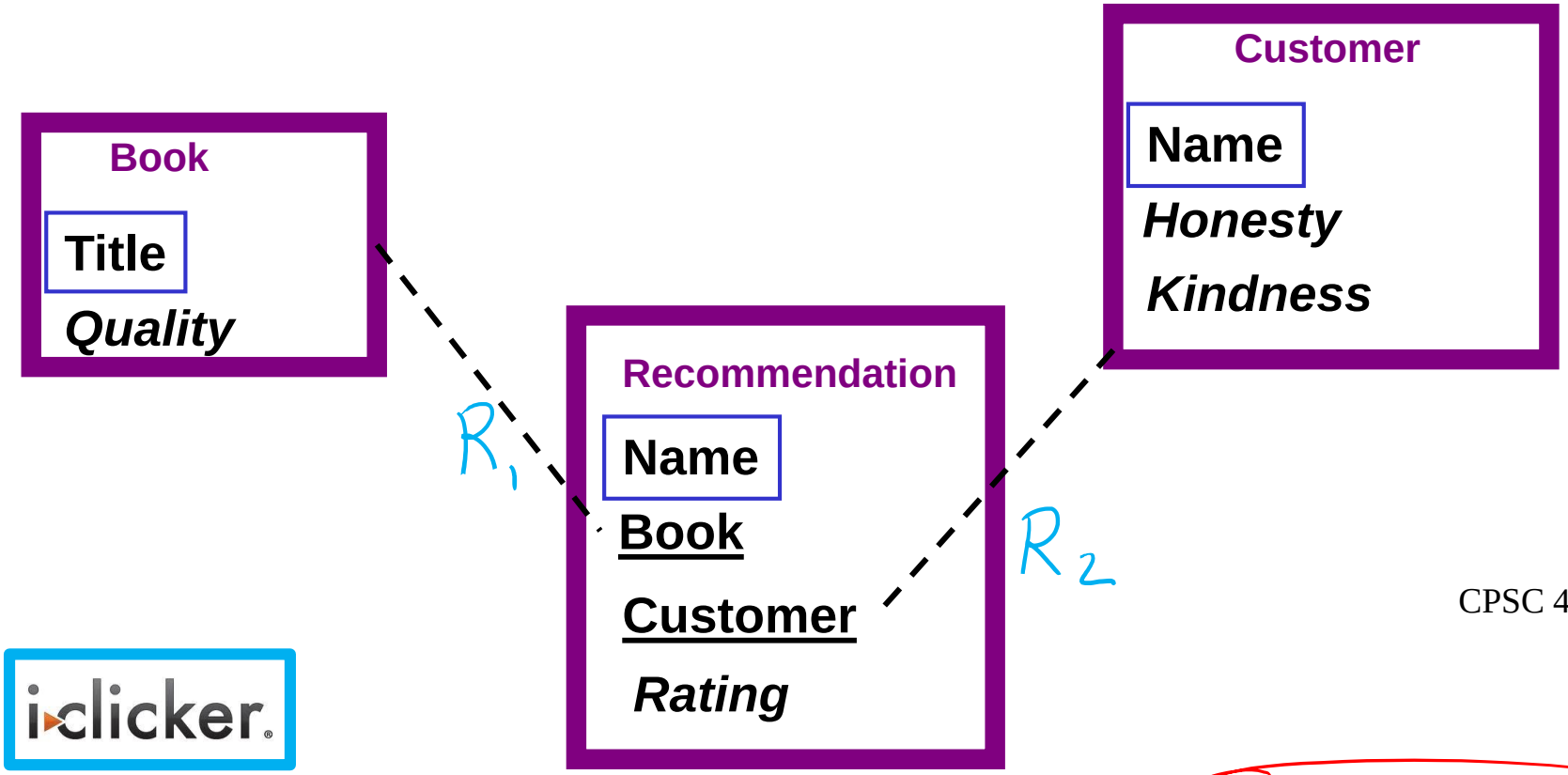
Course.Instructor



University Domain Example – Full Relational Schema



Book Recommendation Domain – Full Relational Schema



i-clicker

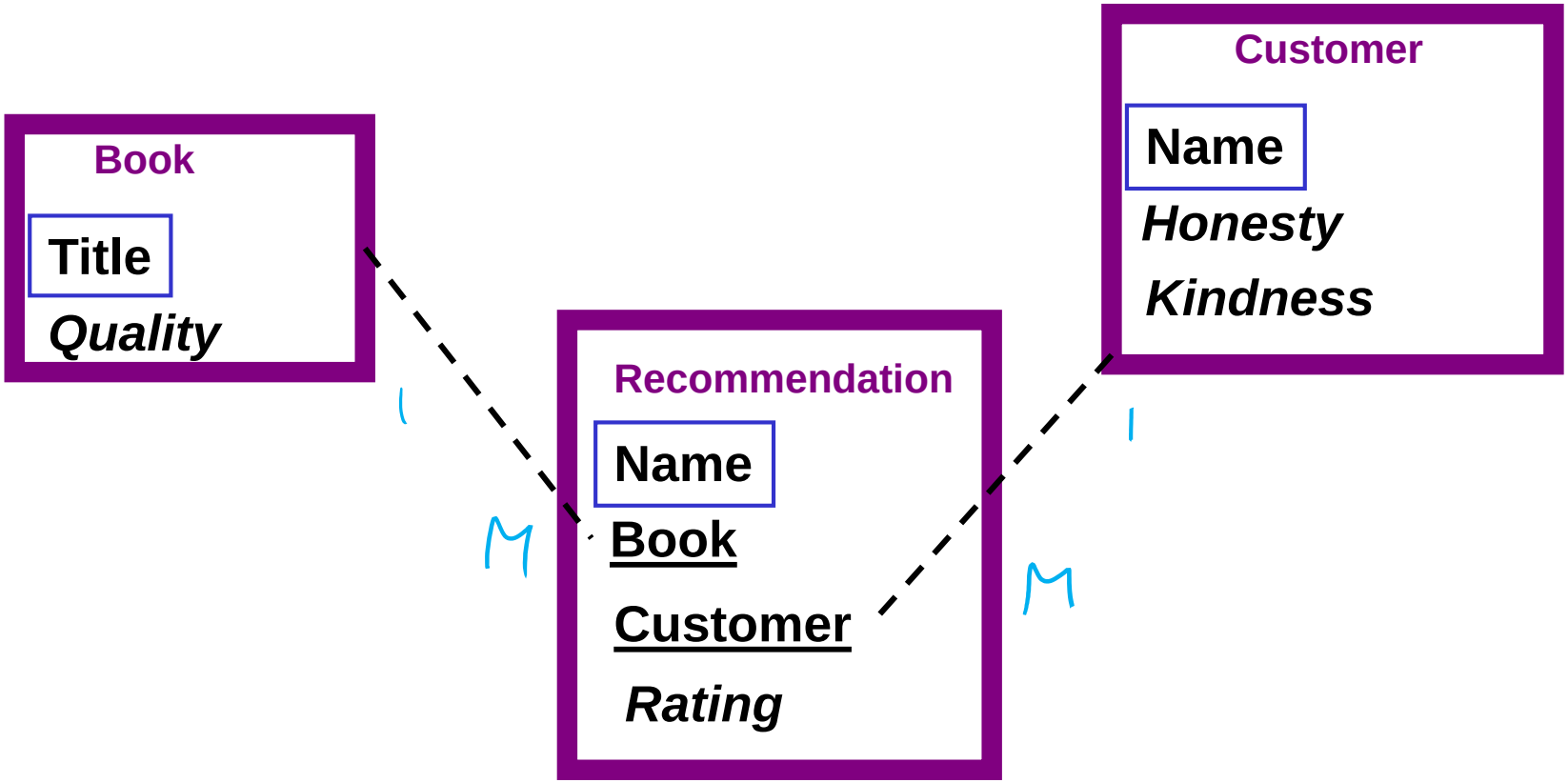
A. R_1 (1:M), R_2 (M:M)

B. R_1 (M:M), R_2 (1:M)

C. R_1 (1:M) R_2 (1:M)

D. R_1 (M:M) R_2 (M:M)

Book Recommendation Domain – Full Relational Schema



PRM Semantics: Attribute values

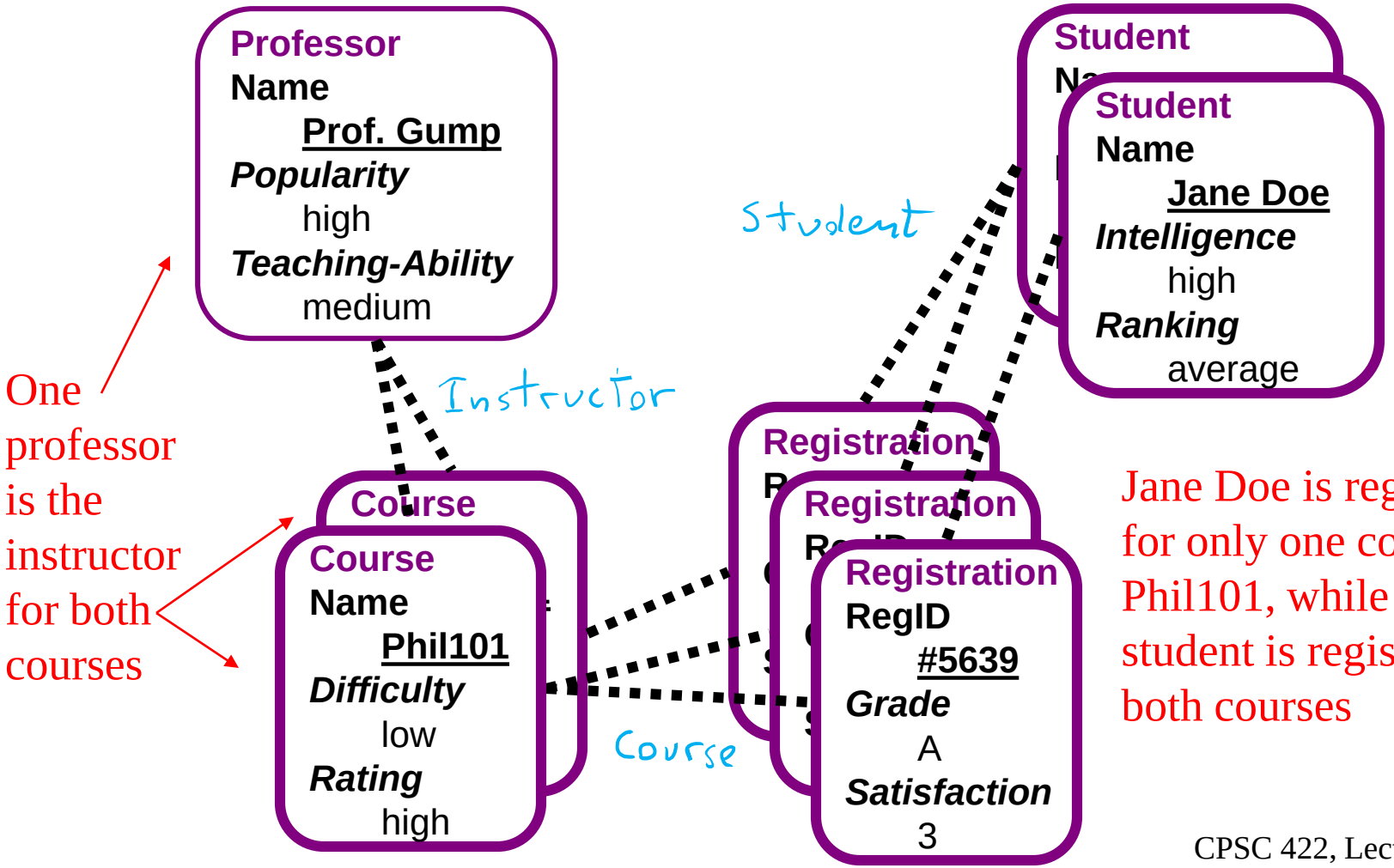
- Each attribute $A_j \in \mathcal{A}(X_i)$ takes on values in some fixed domain of possible values denoted $V(A_j)$. We assume that value spaces are finite
- Attribute *A* of class *X* is denoted *X.A*

- E.g., V(Student.*Intelligence*) might be { *high, low* }

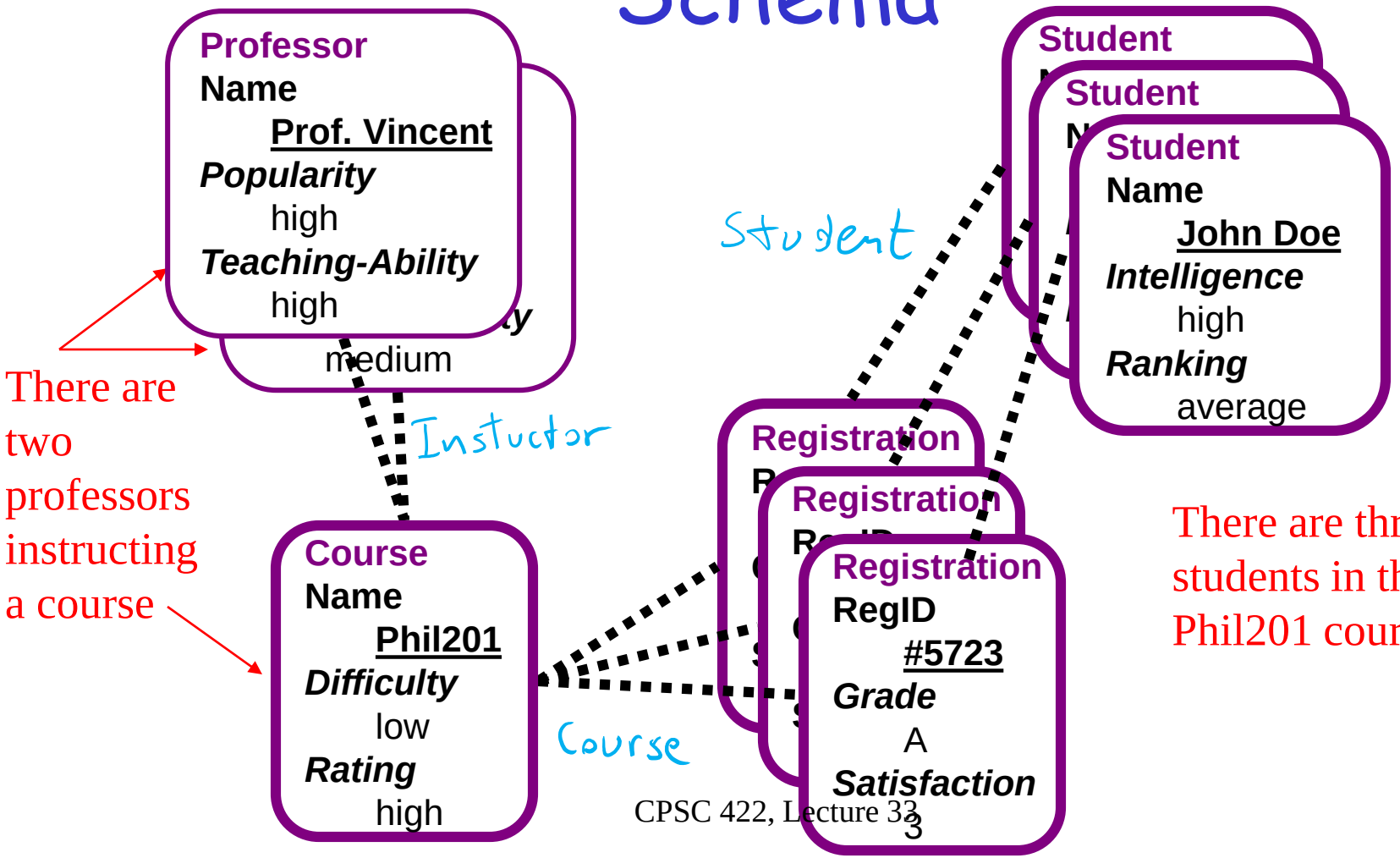
PRM Semantics: Instance of Schema

- An *instance I* of a schema/model specifies a set of objects *x*, partitioned into classes; such that there is
  - a value for each attribute *x.A*
  - and a value for each reference slot *x.ρ*

University Domain Example – An Instance of the Schema



University Domain Example – Another Instance of the Schema

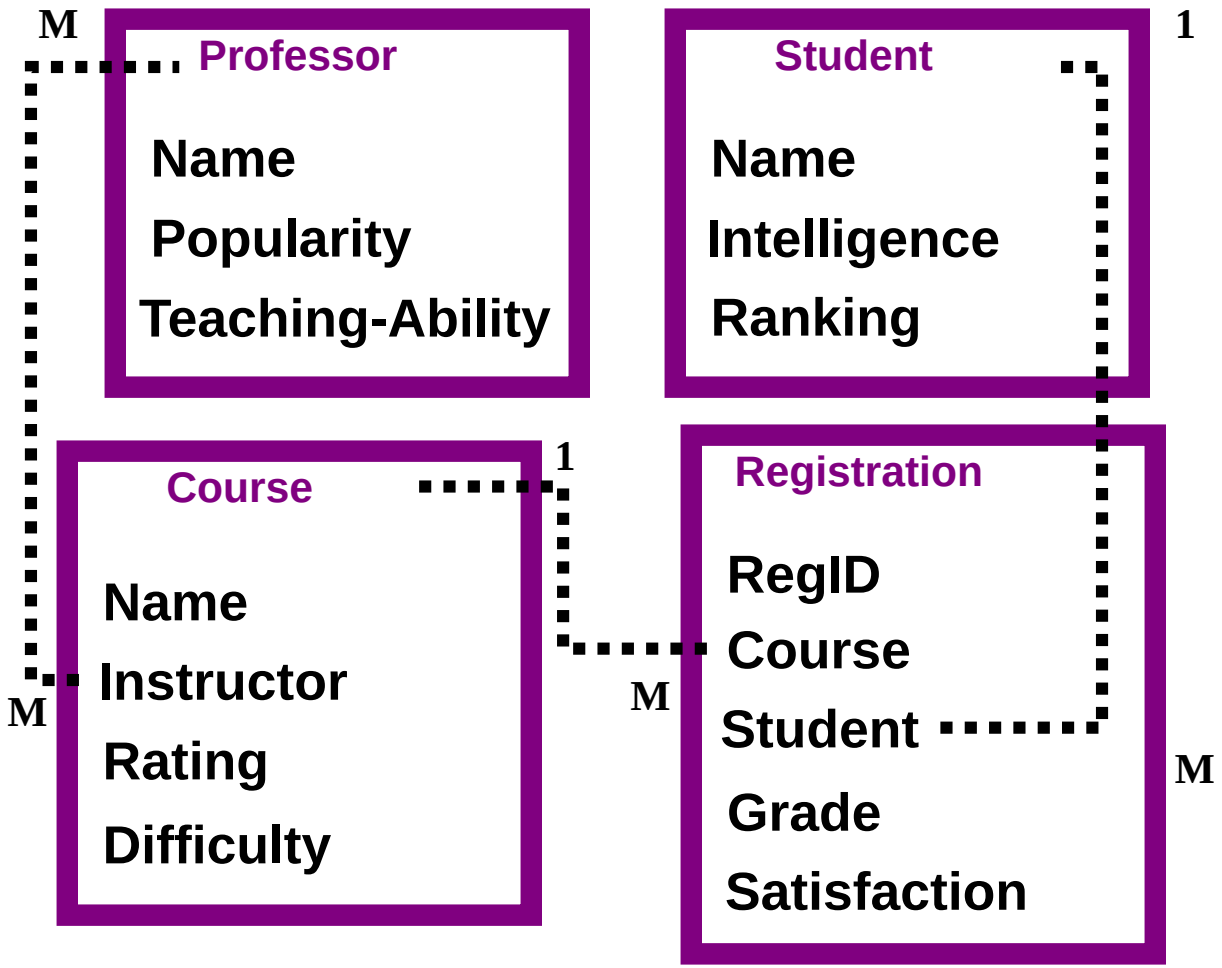


PRM Semantics: fixed vs. prob. attributes

- Some attributes, such as *Name* or *Social Security Number*, are fully determined. Such attributes are labeled as **fixed**. Assume that they are known in any instantiation of the schema

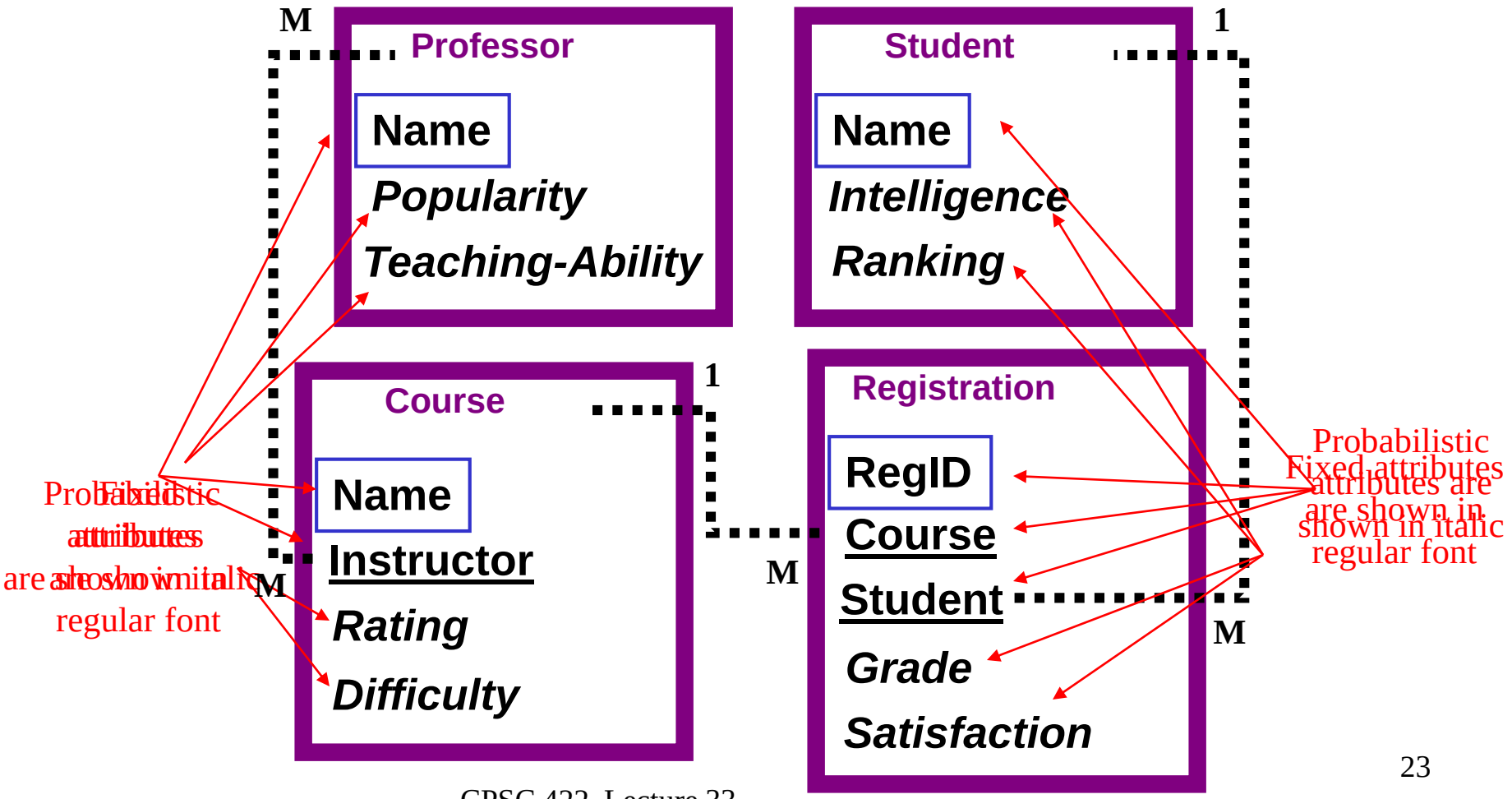
- The other attributes are called **probabilistic**. We may be uncertain about their value

University Domain Example – fixed vs. probabilistic attributes



Which ones are fixed? Which are probabilistic?

University Domain Example – fixed vs. probabilistic attributes



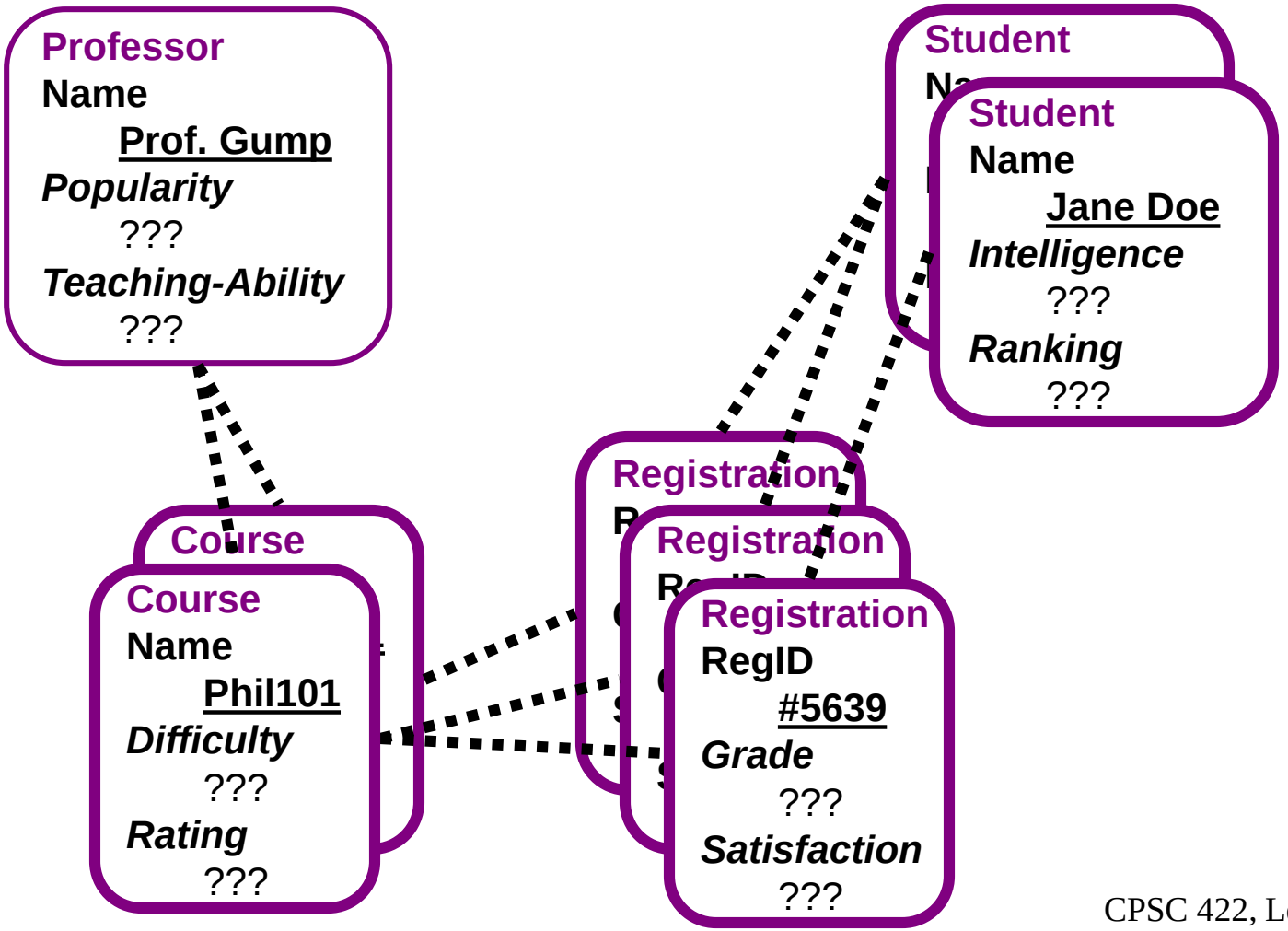
PRM Semantics: Skeleton Structure

- A *skeleton structure σ* of a relational schema is a partial specification of an instance of the schema. It specifies
  - set of objects for each class,
  - values of the fixed attributes of these objects,
  - relations that hold between the objects
- The values of probabilistic attributes are left unspecified

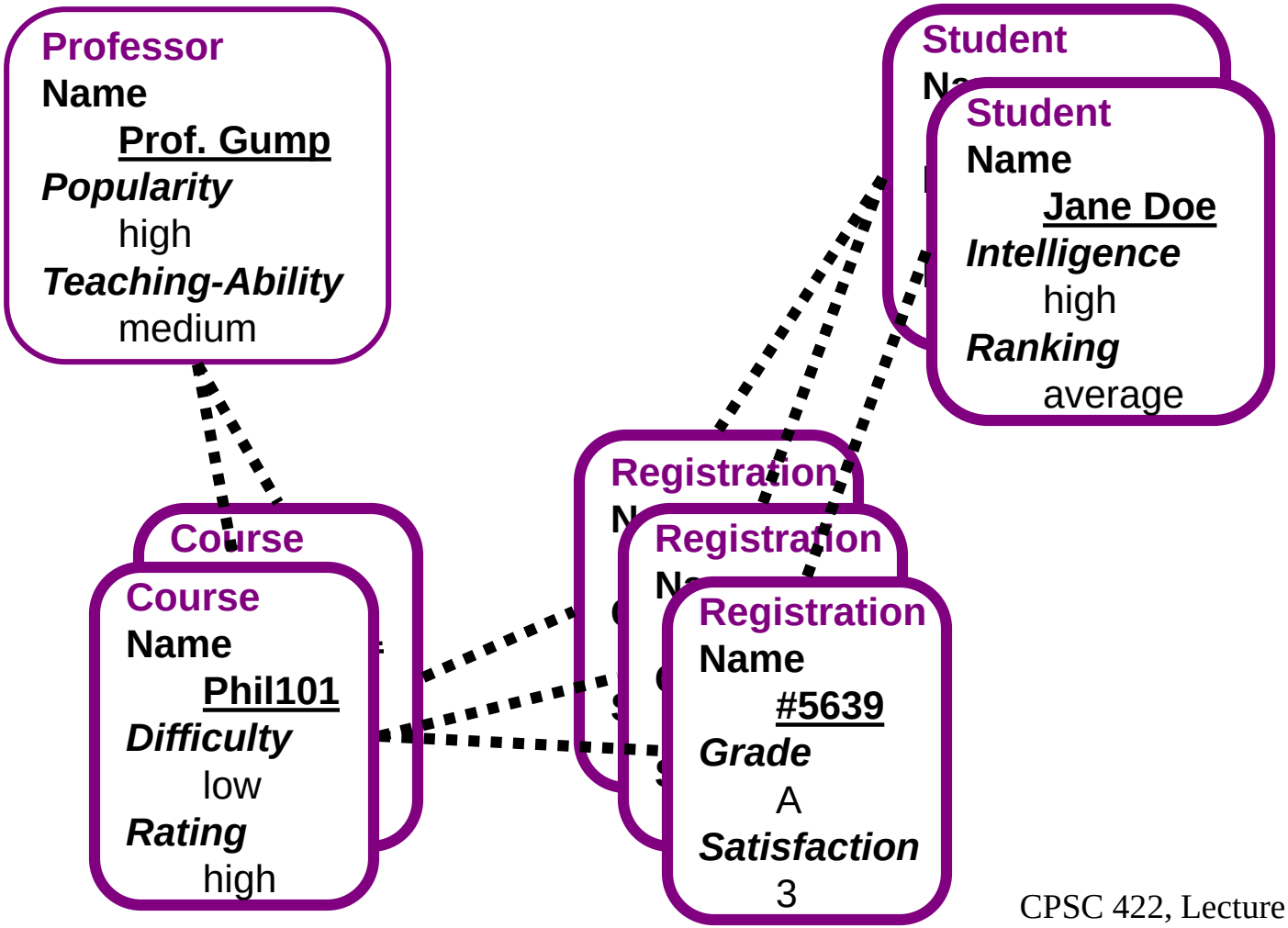
- A *completion I* of the skeleton structure σ extends the skeleton by also specifying the values of the probabilistic attributes

a possible world...

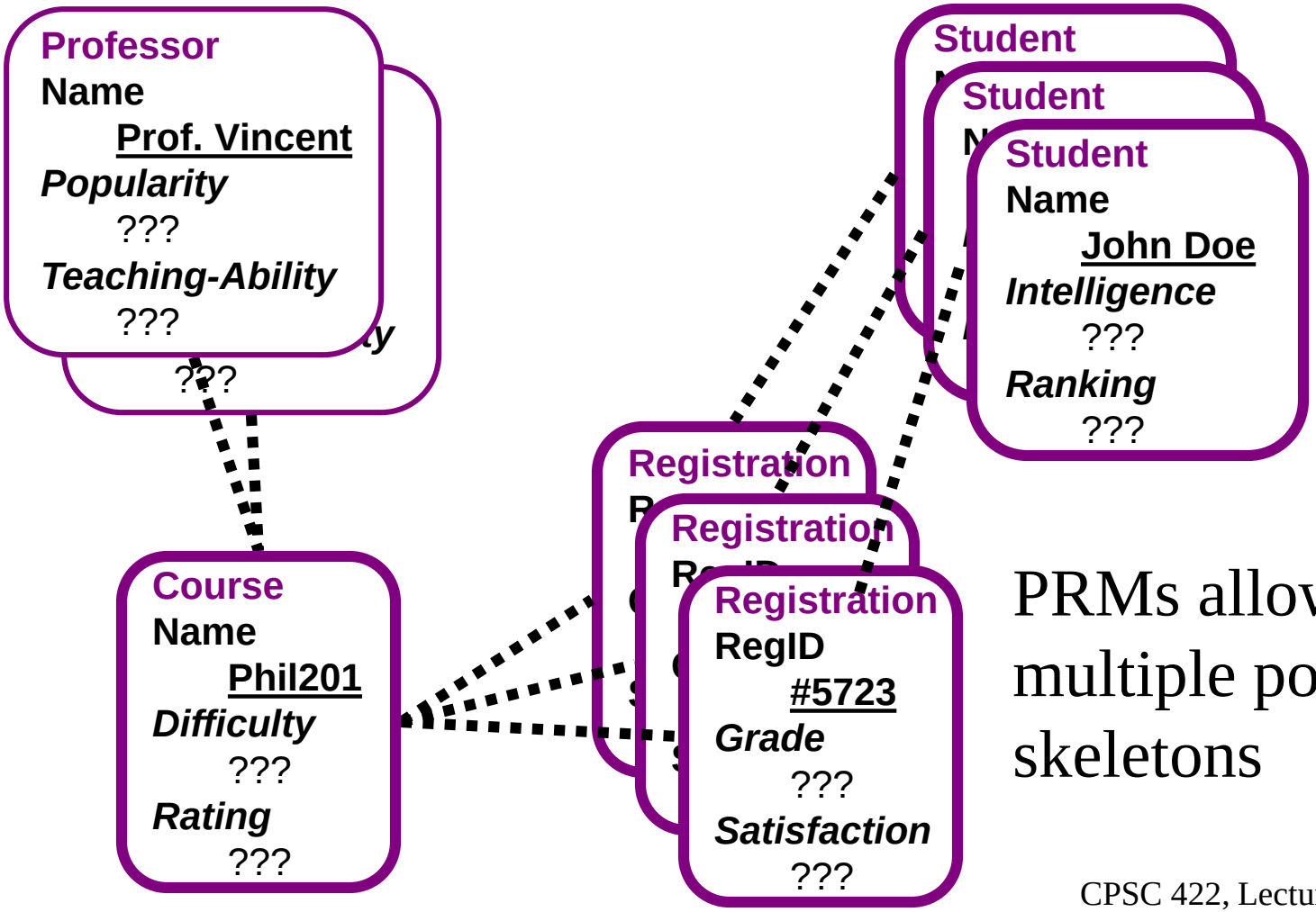
University Domain Example – Relational Skeleton



University Domain Example – The Completion Instance *I*

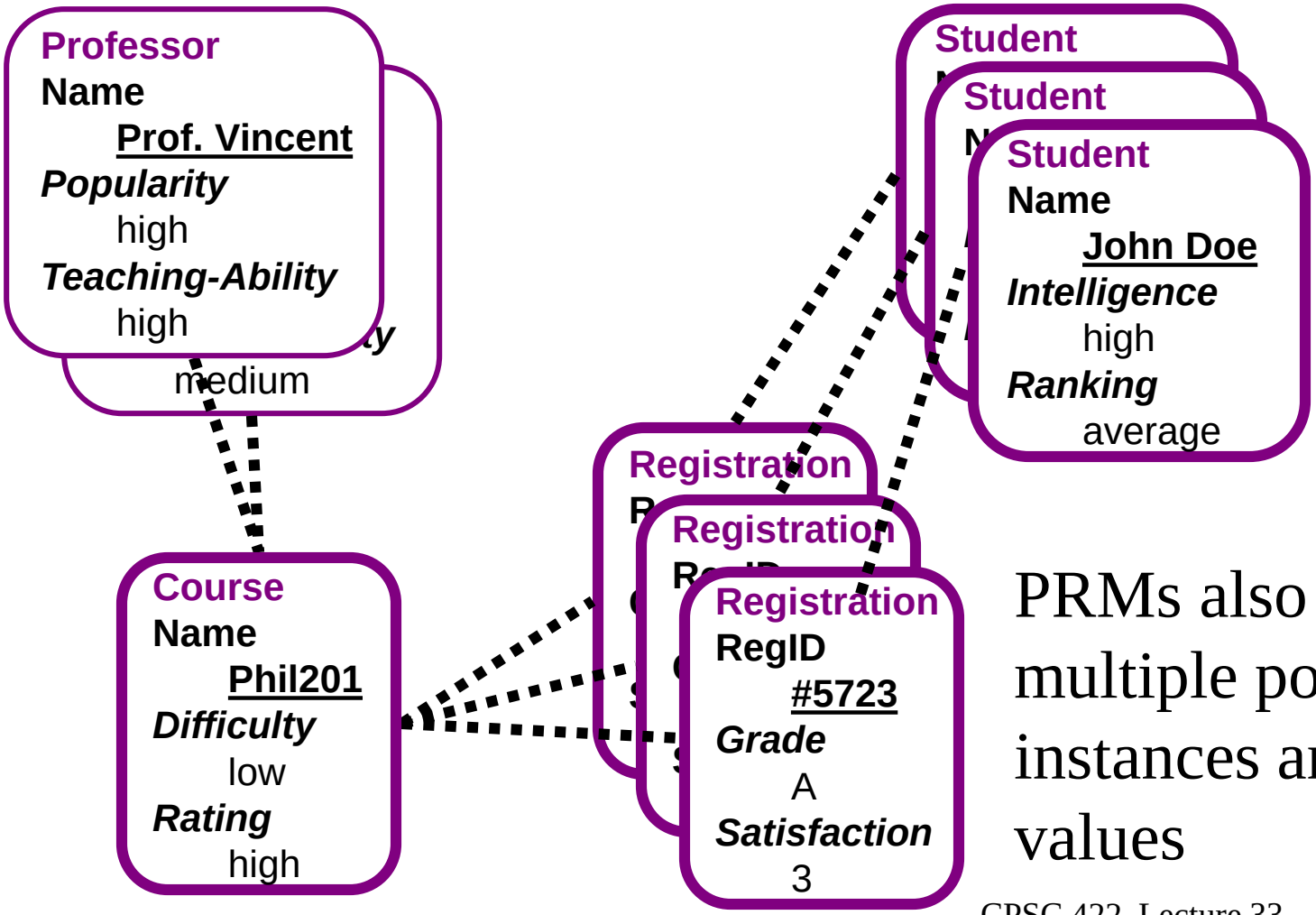


University Domain Example – Another Relational Skeleton



PRMs allow multiple possible skeletons

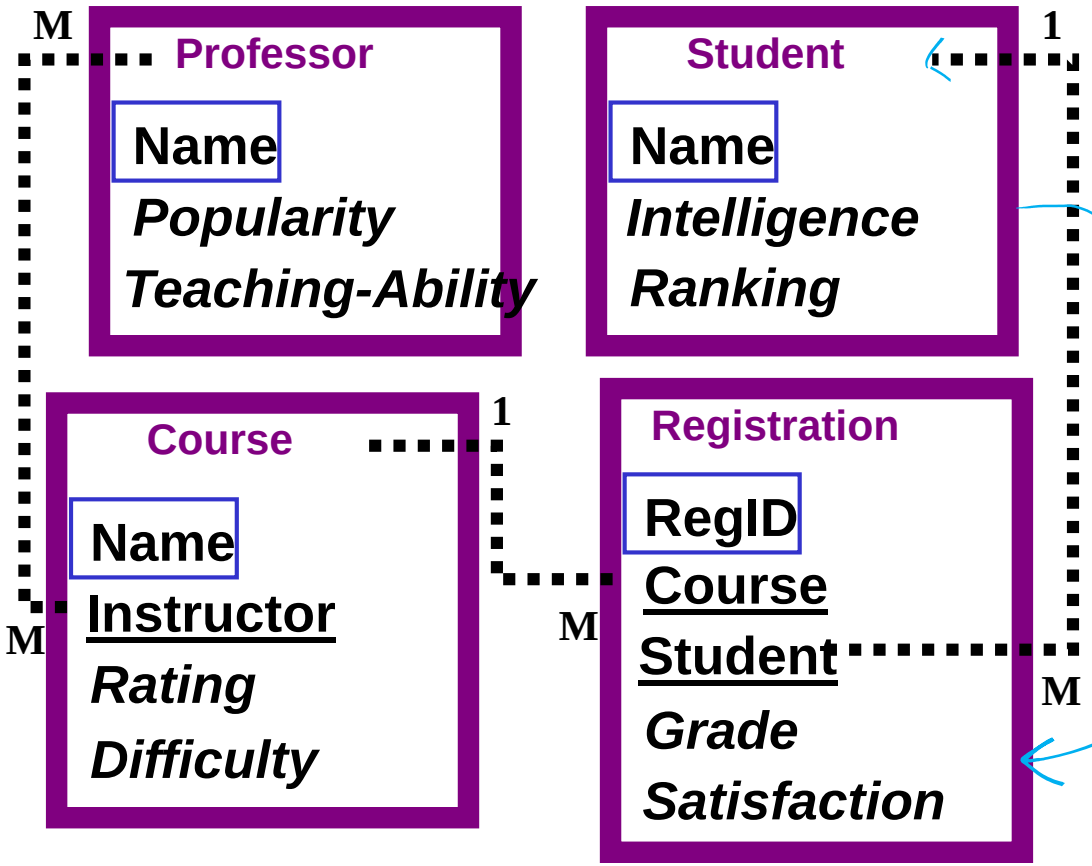
University Domain Example – The Completion Instance *I*



PRMs also allow multiple possible instances and values

PRM Semantics: inverse slot

- For each reference slot ρ, we define an ***inverse slot***, ρ^{-1}, which is the inverse function of ρ



Registered-in

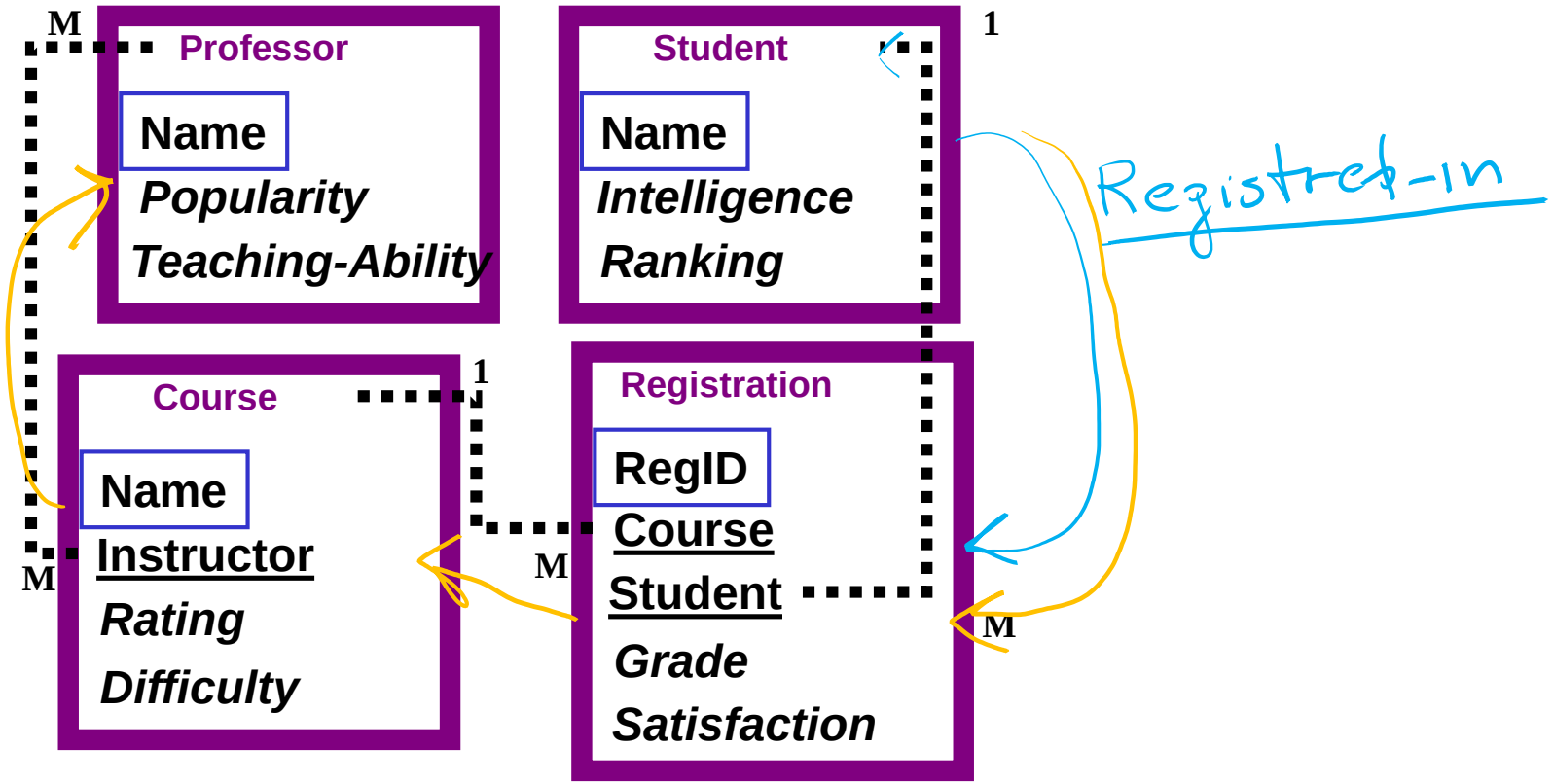
ρ = Student

ρ^{-1} = Registered-in

returns a set of Registration objects

PRM Semantics: slot chain

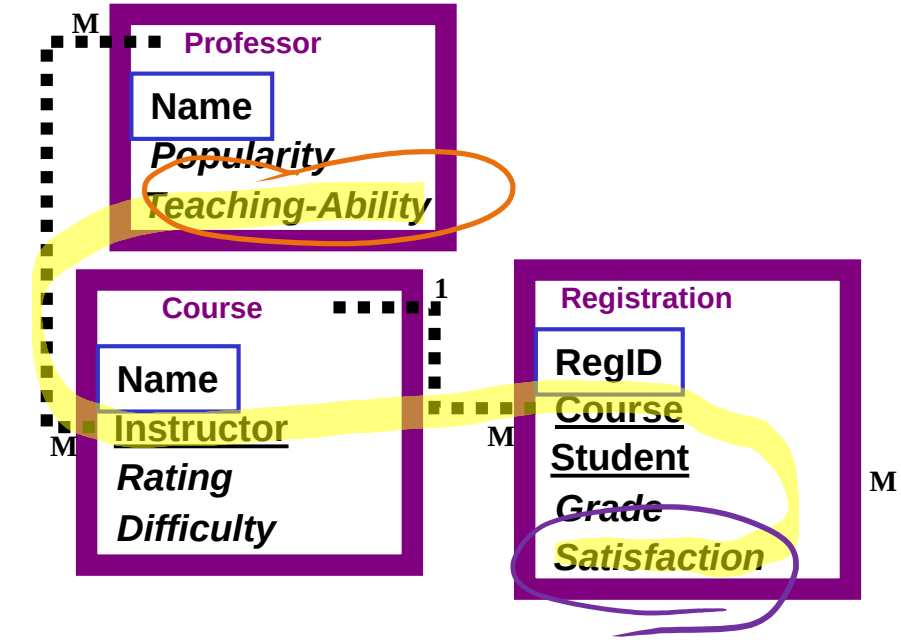
A *slot chain* $\tau=\rho_1..\rho_m$ is a sequence of reference slots that defines functions from objects to other objects to which they are indirectly related.



Student.*Registered-In*.Course.Instructor can be used to denote…… a student's set of instructors

Slot chains will allow us…

To specify probabilistic dependencies between attributes of related entities



Learning Goals for today's class

You can:

- **Explain the need for Probabilistic relational model**
- **Explain how PRMs generalize BNs**
- **Define a Full Relational Schema and its instances**
- **Define a Relational Skeleton and its completion Instances**
- **Define an inverse slot and an slot chain**

Next class on Wed

Finish Probabilistic Relational Models

- Probabilistic Model
- Dependency Structure
- Aggregation
- Parameters
- Class dependency Graph
- Inference

Keep working on **Assignment-4**

Due **Dec 2**