

COSI 127b

Introduction to Database Systems

Lecture 10: Database Design (2)

What a DBMS Manages

1. Data Organization

- Logical: Relational Data Model, Database Design Techniques

2. Data Retrieval

- Logical: Query Languages: RA, TRC, SQL

3. Data Integrity

- Logical: Transactions, Integrity Constraints

Review: 1st Day on the Job

You Produce:

1. Data Requirements:

a) data to maintain	Logical DB Design
b) data relationships	
c) data constraints	Integrity Constraints

2. Functional Requirements

a) analysis	Queries
b) data modification	Xactions

3. Security Requirements

a) permissions to see	Views
b) permissions to modify	Authorization

4. Performance Requirements Physical DB Design

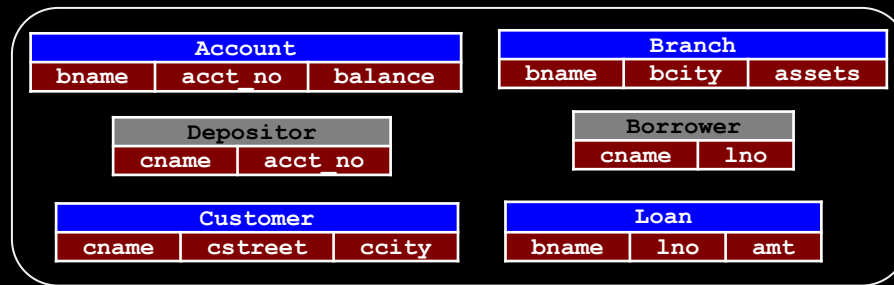
Review: Logical DB Designs for Bank

Advantage of Design #2:

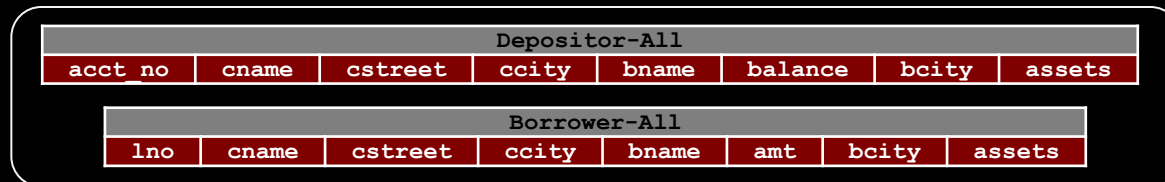
1. (Some) queries are faster (precomputed results)

Disadvantages of Design #2:

1. Redundancy (update anomalies more likely)
2. Missing Information



Design #1



Design #2

Review: Good DB Design

Three Approaches:

1. Ad hoc:

- use **Entity-Relationship Model** to model data requirements
- translate **ER** design into relational schema

Issue: How to tell if design is "good"?

2. Theoretical:

- construct **universal relations** (e.g., Borrower-All)
- **decompose** above using known functional dependencies

Issue: Time-Consuming and Complex

3. Practical:

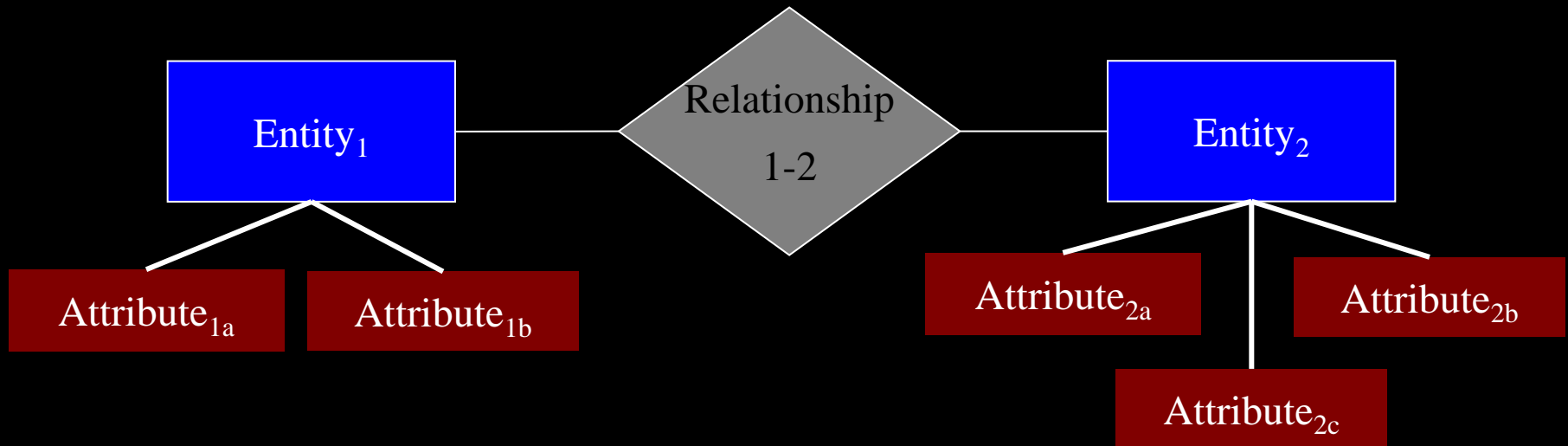
- use ER Model to produce 1st cut DB design
- use FDs to refine and verify

Review: Data Models

What is a Data Model?

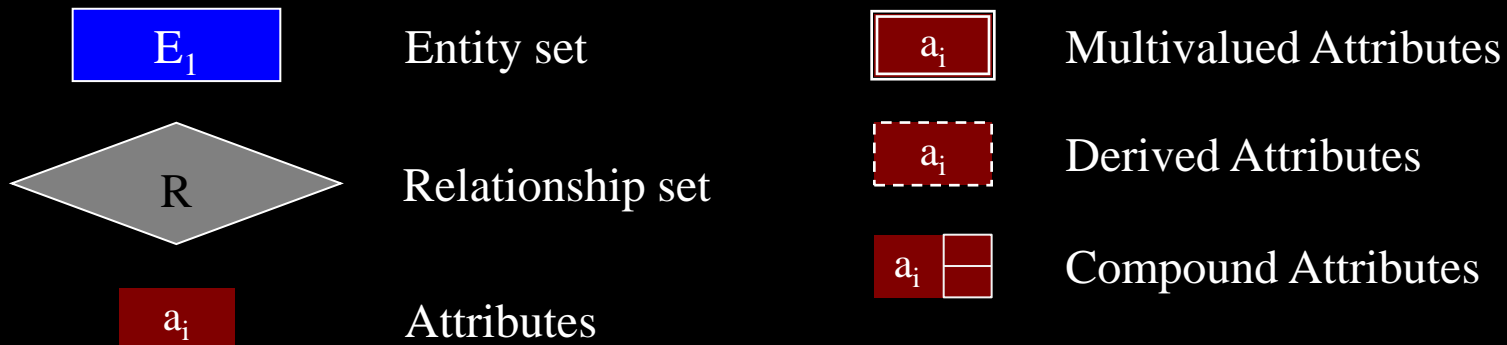
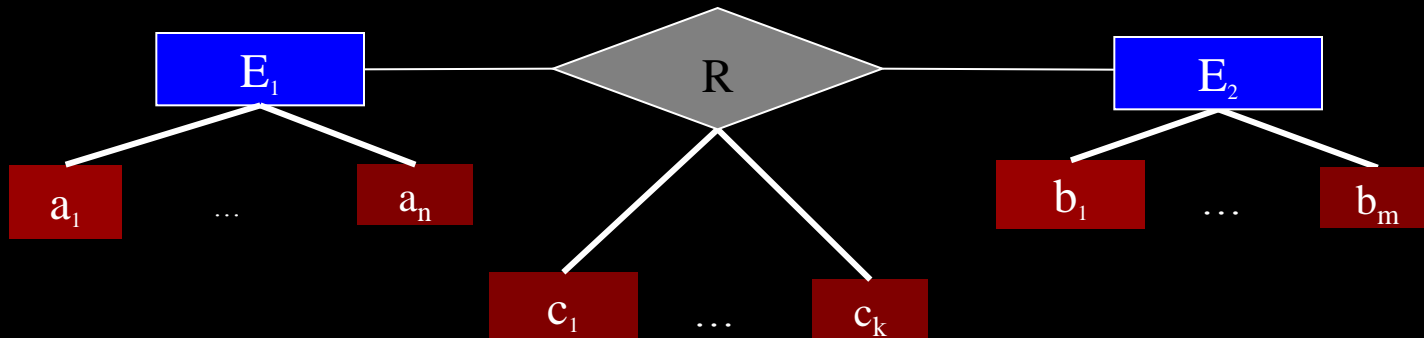
- framework for organizing and interpreting data

An Example: Entity-Relationship (ER) Data Model







Review: E/R Model

The Basics



Review: E/R Model

Relationship Cardinalities

Type	Illustrated
One-to-One (1:1)	
Many-to-one (n:1)	
One-to-many (1:n)	
Many-to-many (n:m)	

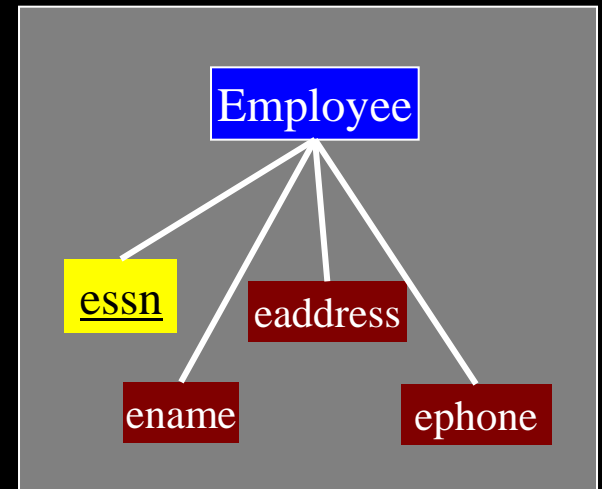
E/R: Keys

Keys:

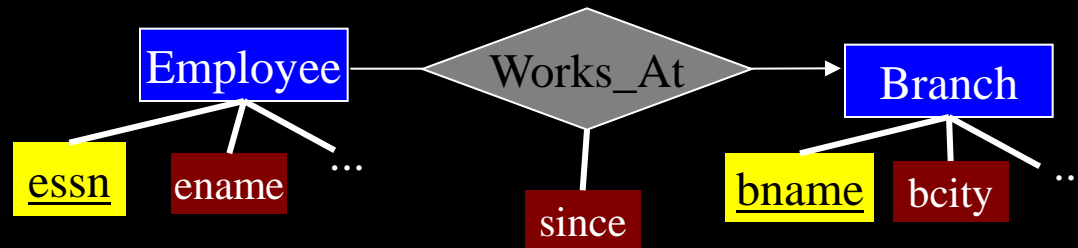
set of attributes that identifies individual entities or relationships

Same Kinds as Relational Data Model

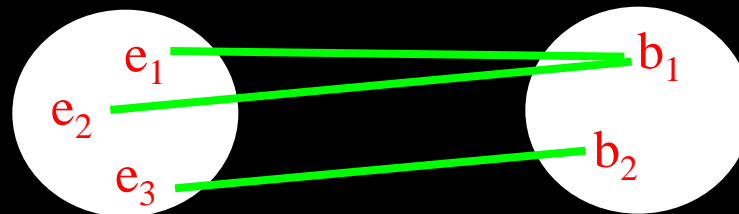
- **Superkey:**
 - any attribute set that distinguishes identities
 - e.g., {essn}, {essn, ename, eaddress}
- **Candidate Key:**
 - “minimal superkey”
(can’t remove attributes and still be a key)
 - e.g., {essn}, {ename, eaddress}
- **Primary Key:**
 - candidate key chosen as the key by a DBA
 - e.g., {essn} (denoted by underline)



E/R: Relationship Set Keys

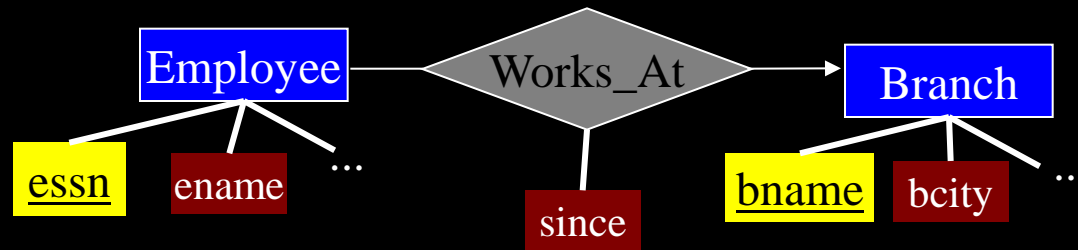


Q: What attributes can represent relationships in Works_At?

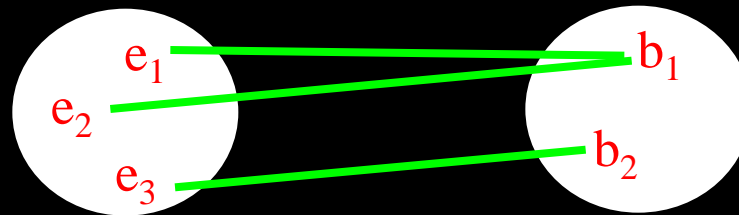


A: {essn, bname, since}

E/R: Relationship Set Keys

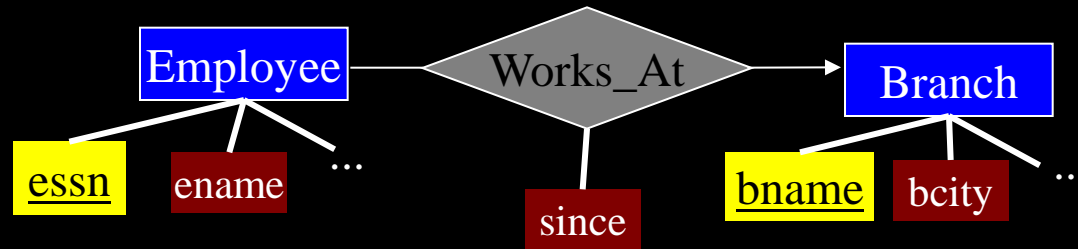


Q: What are the candidate keys of Works_At?



A: Just one: {essn}

E/R: Relationship Set Keys

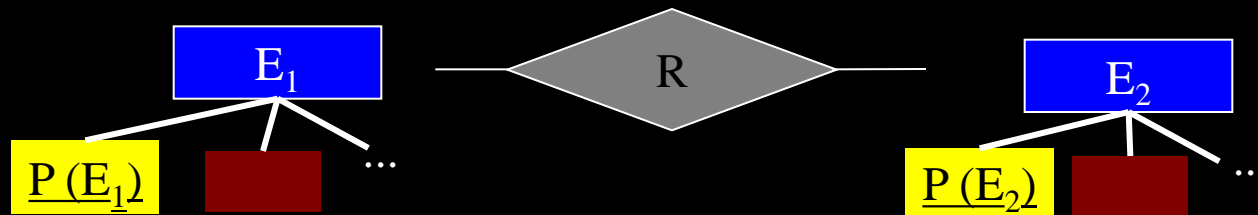


Q: What would be the candidate keys if Works_At were...?

- a. 1:n A: {bname}
- b. n:m A: {essn, bname}
- c. 1:1 A: {essn} or {bname}

E/R: Relationship Set Keys

General Rules for Relationship Set Keys



Depends on R:

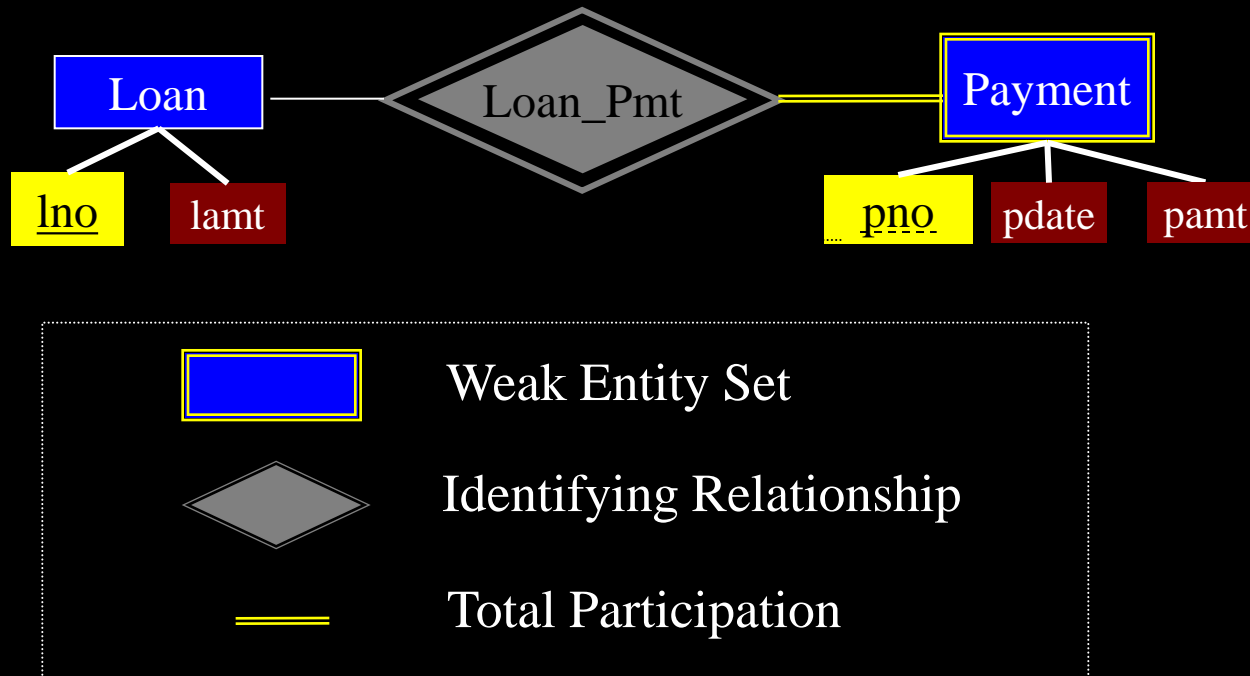
R	Candidate Keys
1:1	$P(E_1)$ or $P(E_2)$
1:n	$P(E_2)$
n:1	$P(E_1)$
n:m	$P(E_1) \cup P(E_2)$

E/R: Existence Dependence

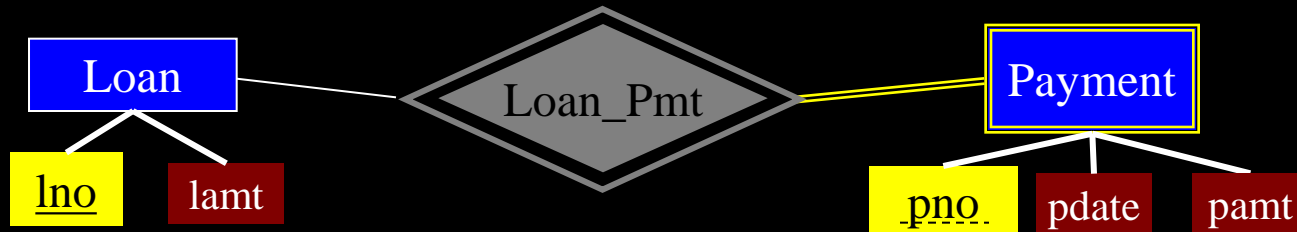
Idea:

existence of one entity depends on another

Example: Loans and Loan Payments



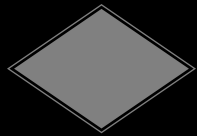
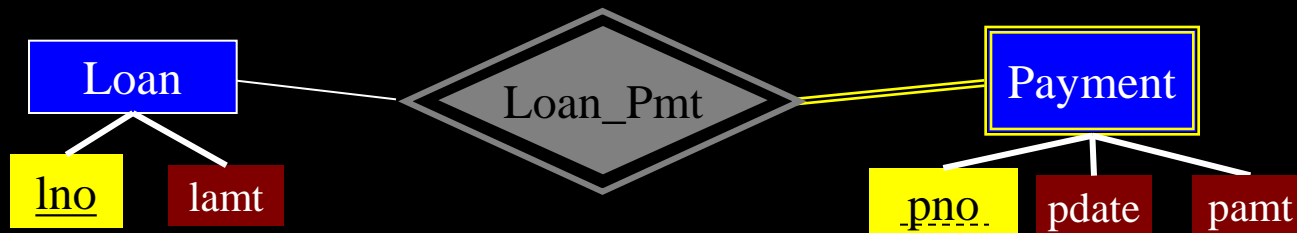
E/R: Existence Dependence



Weak Entity Sets

- existence of payments depends upon loans
- have no superkeys: different payment records (for different loans) can be identical
- instead of keys, discriminators: discriminate between payments for given loan (e.g., pno)

E/R: Existence Dependence

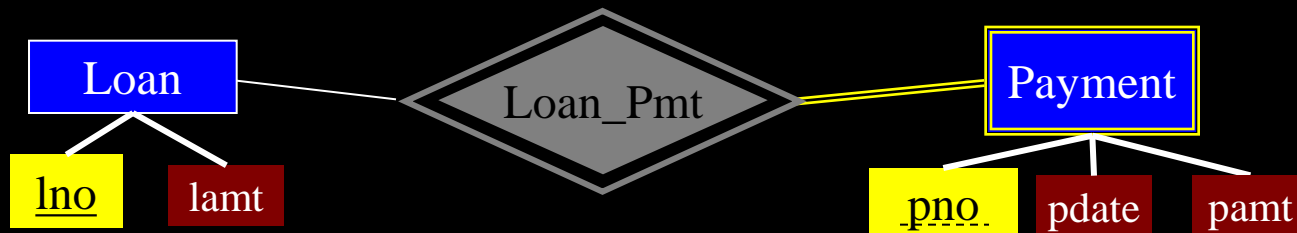


Identifying Relationships

We can say any of the following:

- Loan is **dominant** in Loan_Pmt
- Payment is **subordinate** in Loan_Pmt
- Payment is **existence dependent** on Loan

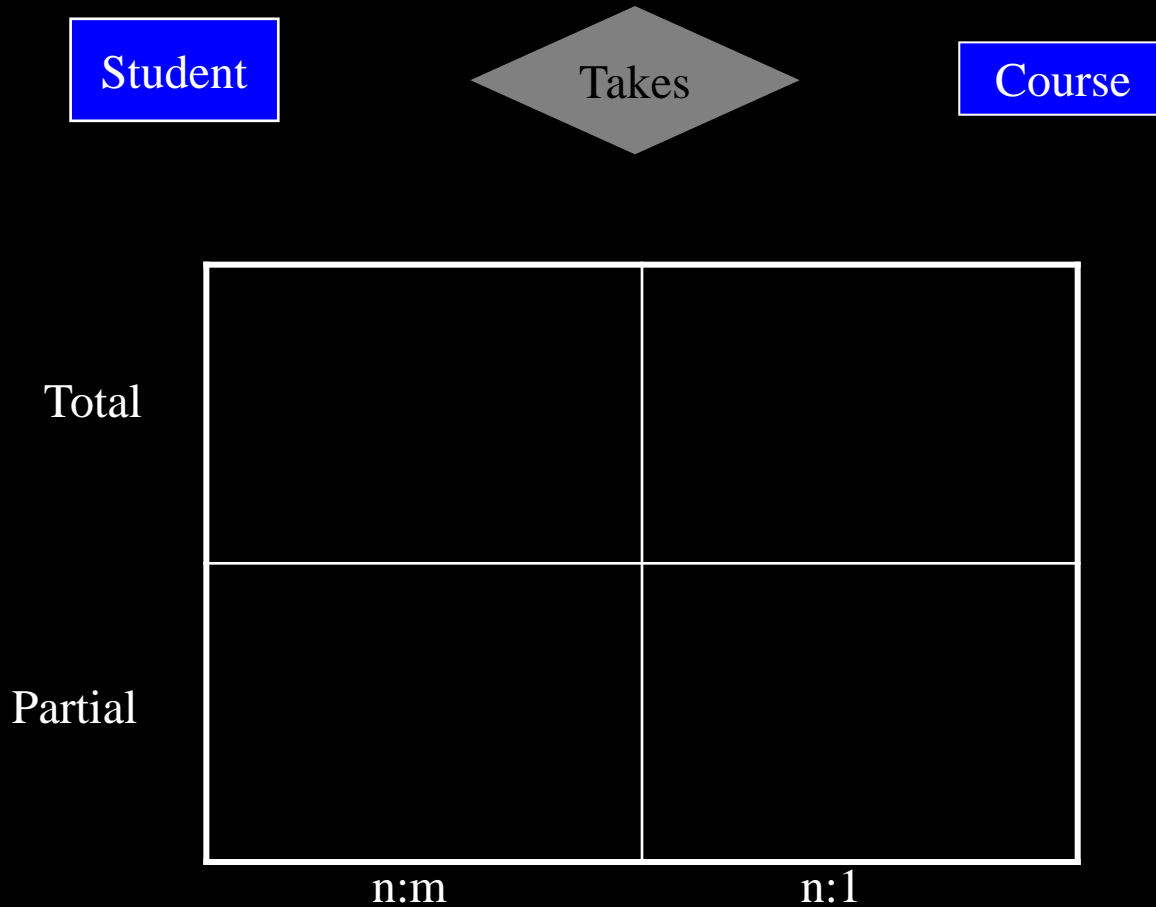
E/R: Existence Dependence



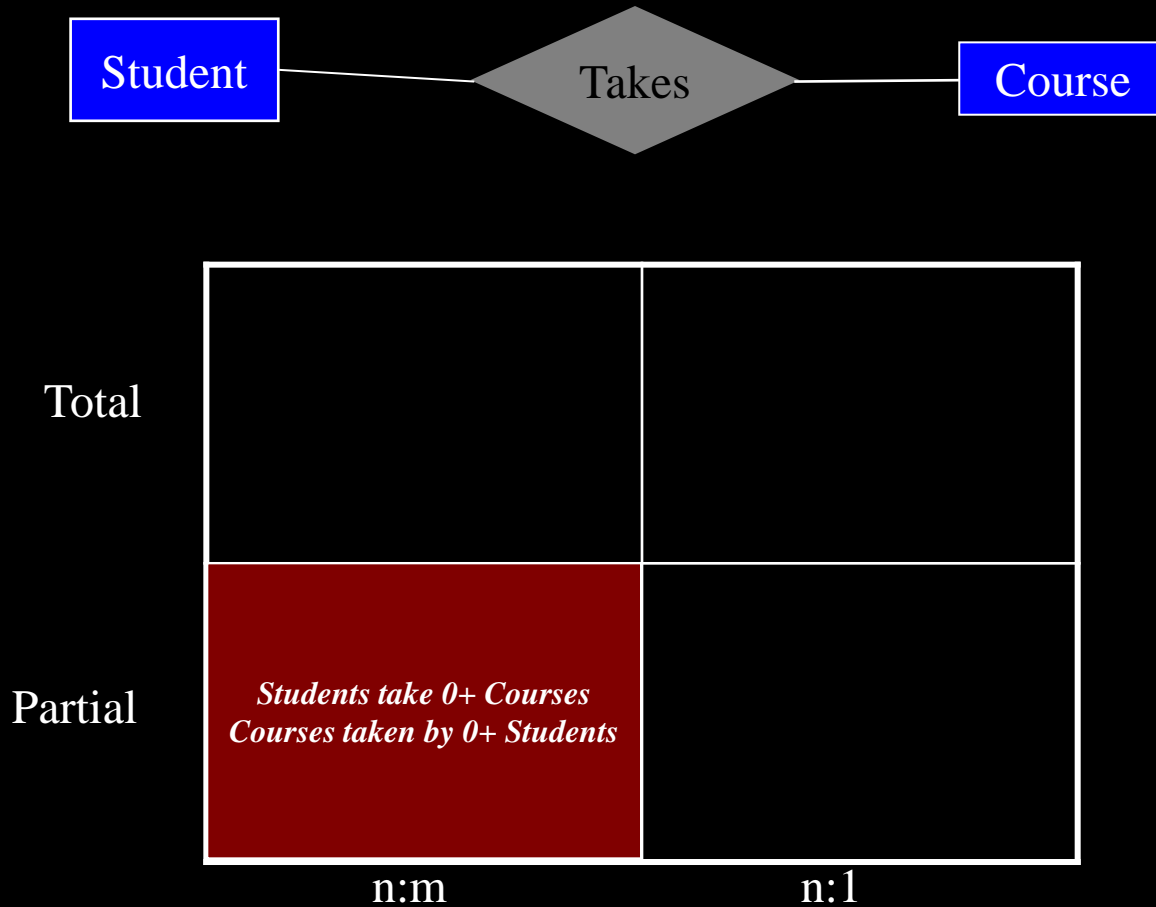
== Total Participation

All elements of Payment appear in Loan_Pmt

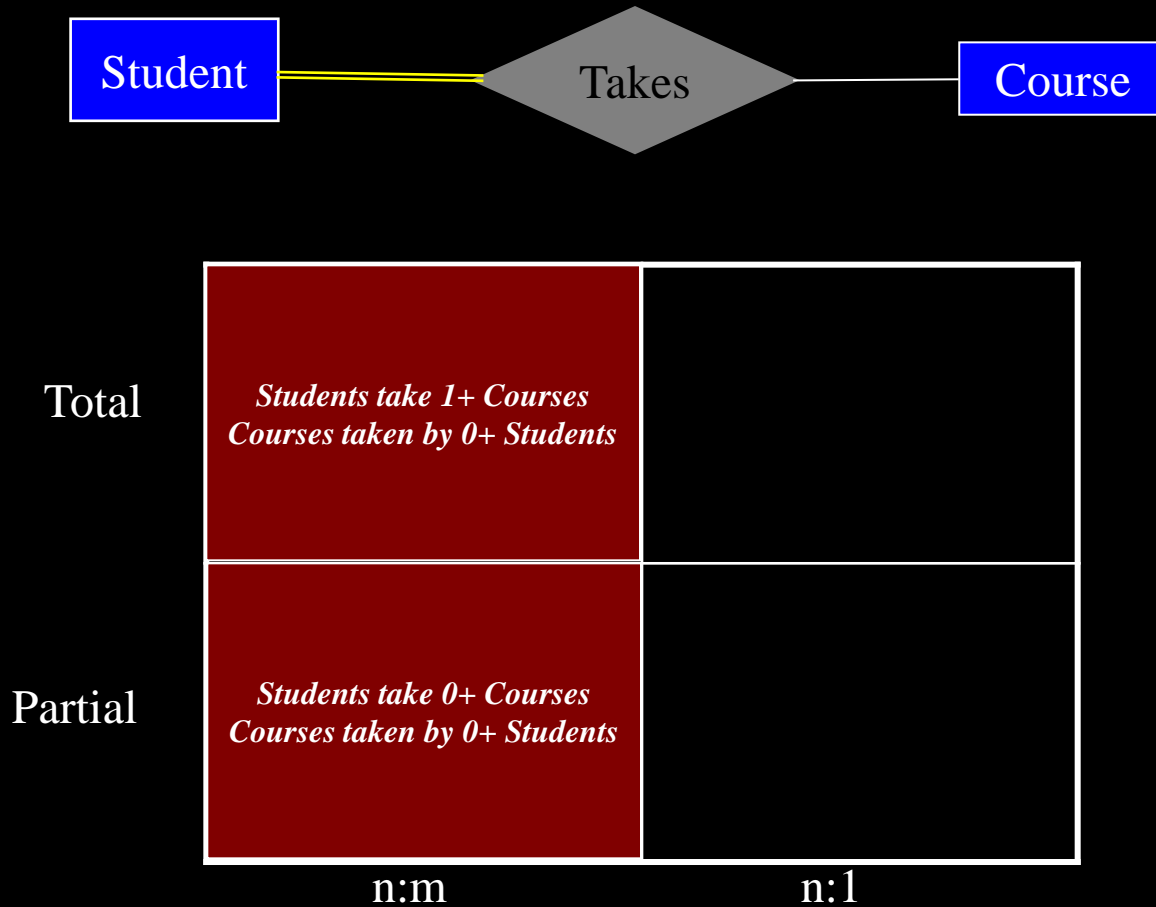
E/R: Total/Partial vs $n:m/n:1$



E/R: Total/Partial vs n:m/n:1



E/R: Total/Partial vs n::m/n::1



E/R: Total/Partial vs n::m/n::1



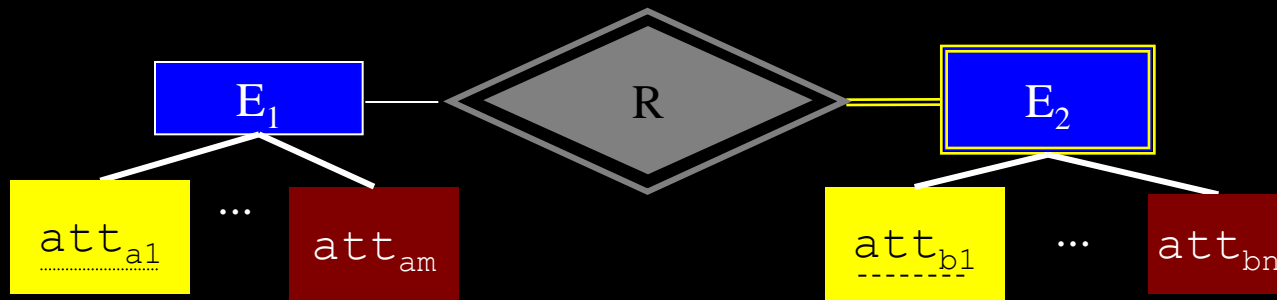
Total	<i>Students take 1+ Courses</i> <i>Courses taken by 0+ Students</i>	
Partial	<i>Students take 0+ Courses</i> <i>Courses taken by 0+ Students</i>	<i>Students take 0 or 1 Course</i> <i>Courses taken by 0+ Students</i>
	n:m	n:1

E/R: Total/Partial vs n::m/n::1



Total	<i>Students take 1+ Courses</i> <i>Courses taken by 0+ Students</i>	<i>Students take 1 Course</i> <i>Courses taken by 0+ students</i>
Partial	<i>Students take 0+ Courses</i> <i>Courses taken by 0+ Students</i>	<i>Students take 0 or 1 Course</i> <i>Courses taken by 0+ Students</i>
	n:m	n:1

E/R: Existence Dependence



Q. Is $\{att_{b1}, \dots, att_{bn}\}$ a superkey of E_2 ?

A: No

Q. Name a candidate key of E_2

A: $\{att_{a1}, att_{b1}\}$

Q. Does total participation of E_2 in $R \Rightarrow E_2$ is existence-dependent?

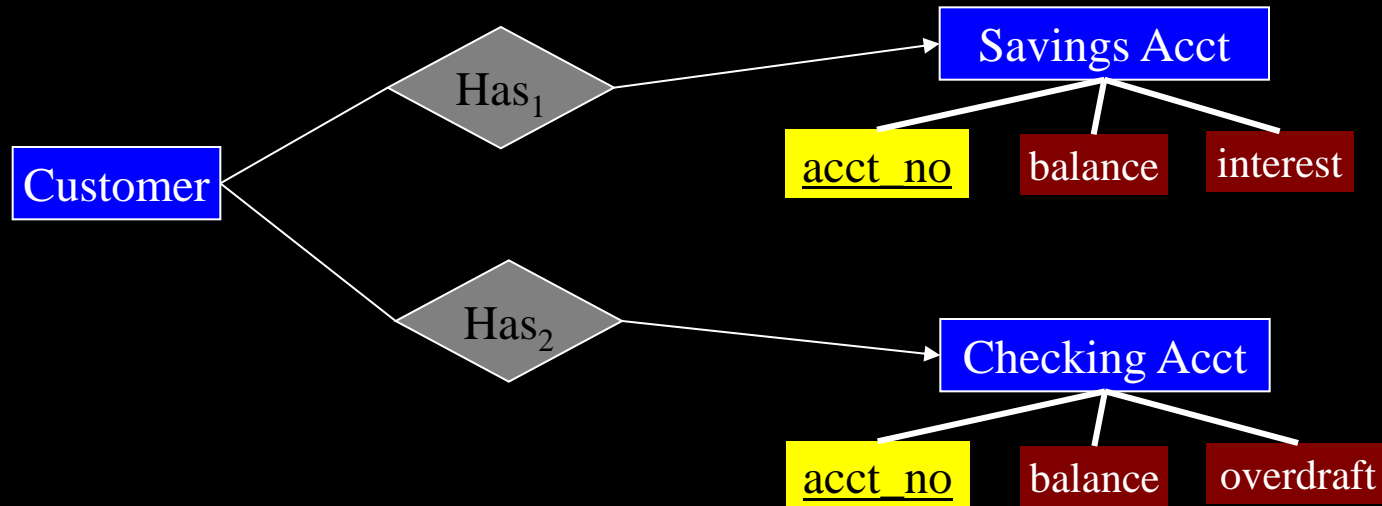
A: No. (e.g., student totally participates in course)

E/R: Specialization and Generalization

An Example:

- Customers can have checking or savings acct
- Checking ~ Savings (many of the same attributes)

Old Way:

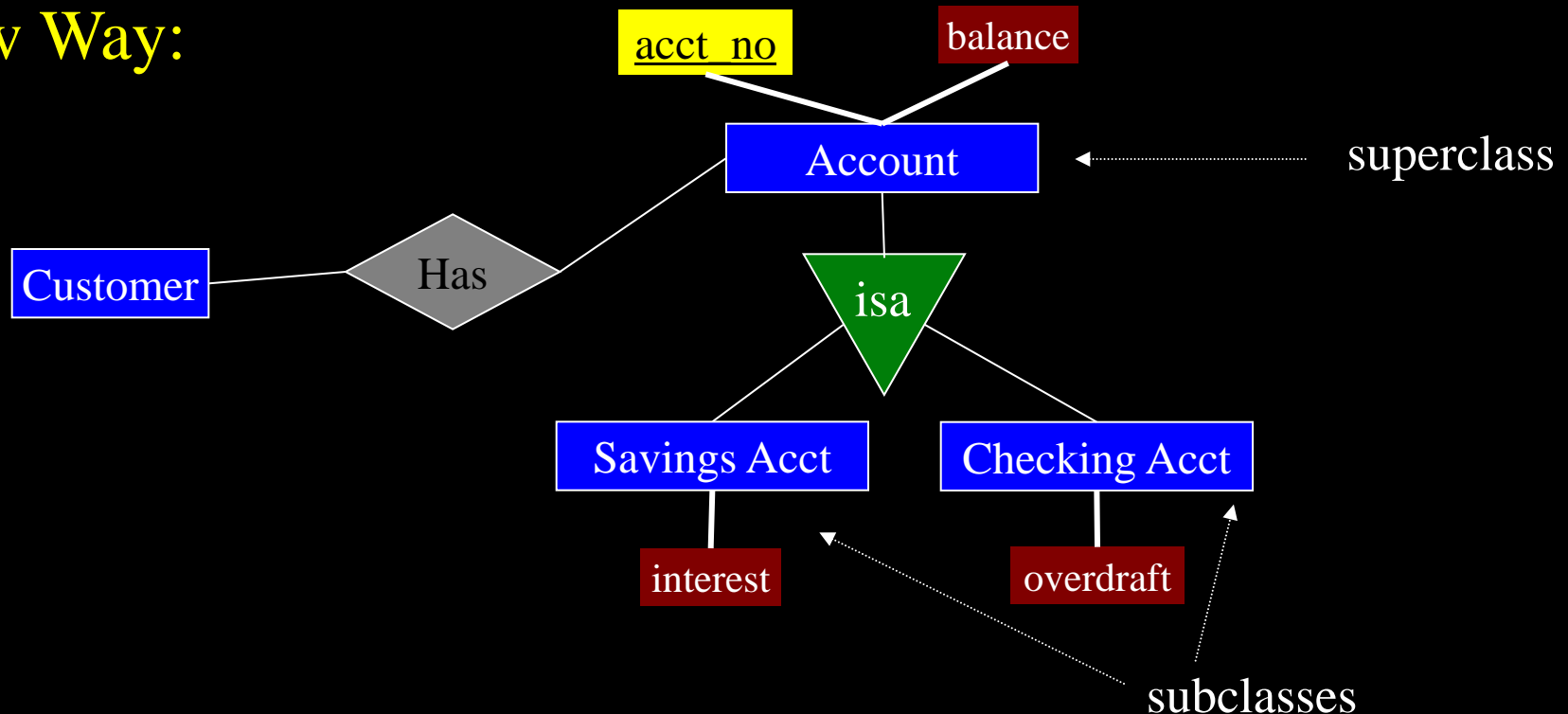


E/R: Specialization and Generalization

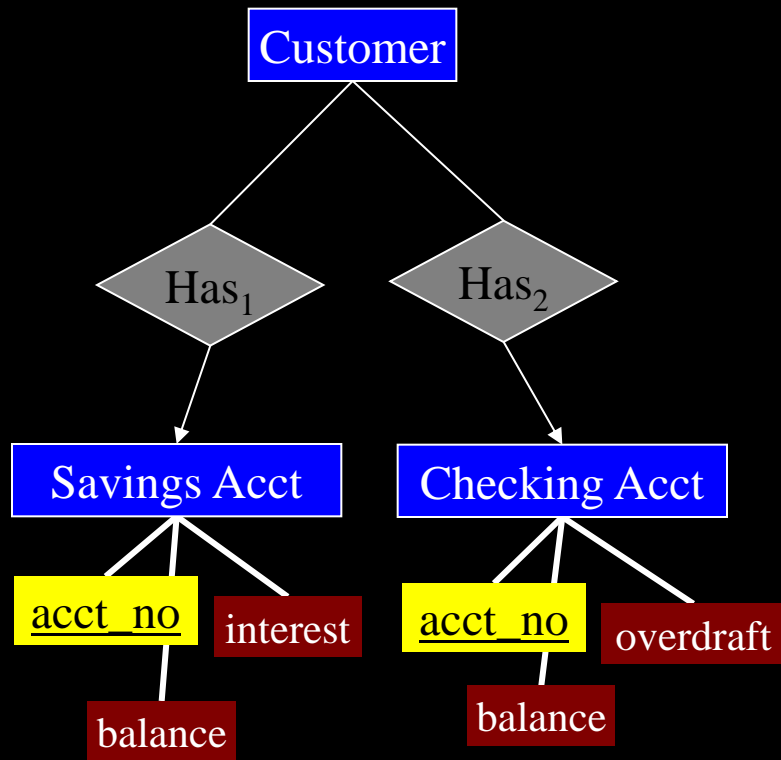
An Example:

- Customers can have checking or savings acct
- Checking ~ Savings (many of the same attributes)

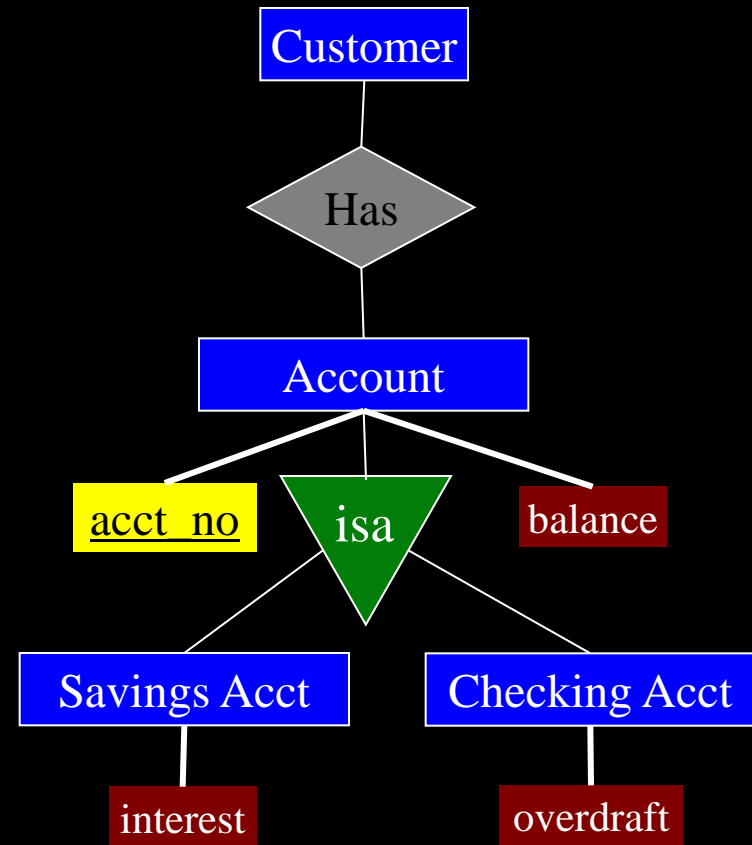
New Way:



E/R: Specialization and Generalization



Old



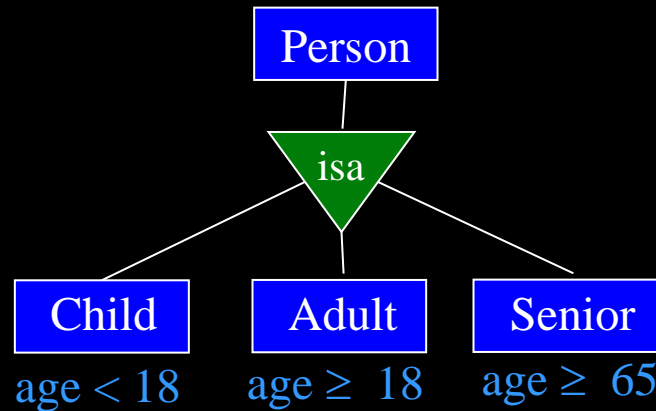
New

E/R: Specialization and Generalization

Subclass Distinctions:

1. User-Defined vs Condition-Defined

- User: Membership in subclasses explicitly determined (e.g., Employee, Manager < Person)
- Condition: Membership predicate associated with subclasses (e.g:

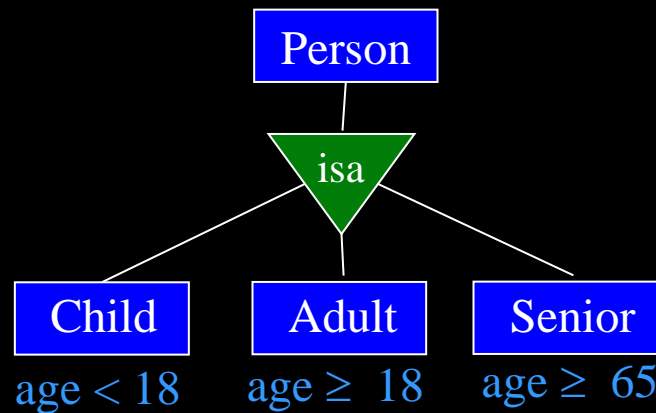


E/R: Specialization and Generalization

Subclass Distinctions:

2. Overlapping vs Disjoint

- Overlapping: Entities can belong to >1 entity set (e.g., Adult, Senior)
- Disjoint: Entities belong to exactly 1 entity set (e.g., Child)

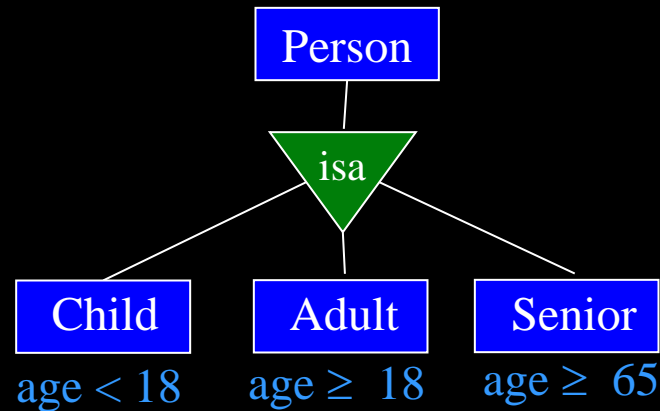


E/R: Specialization and Generalization

Subclass Distinctions:

3. Total vs Partial

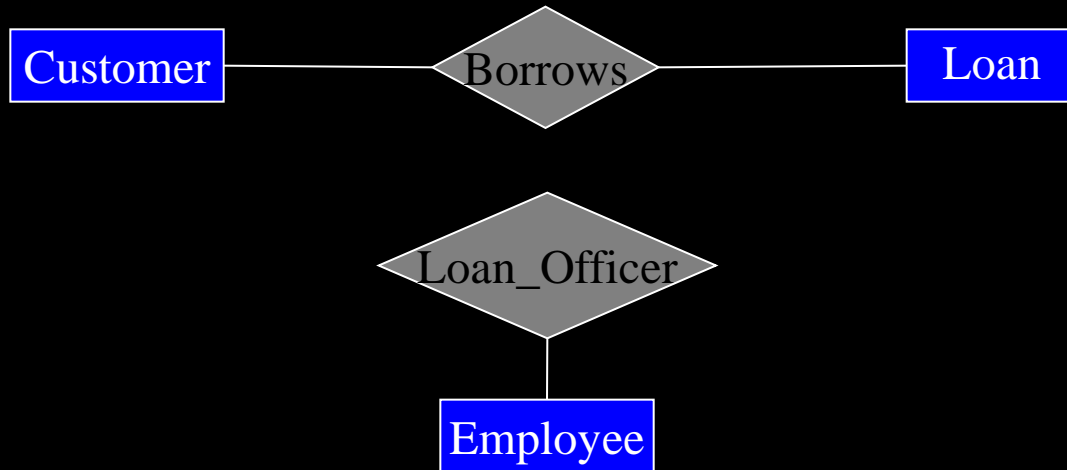
- Total: Every entity of superclass belongs to a subclass
e.g.,



- Partial: Some entities of superclass do not belong to any subclass (e.g., if Adults condition is age ≥ 21)

E/R: Aggregation

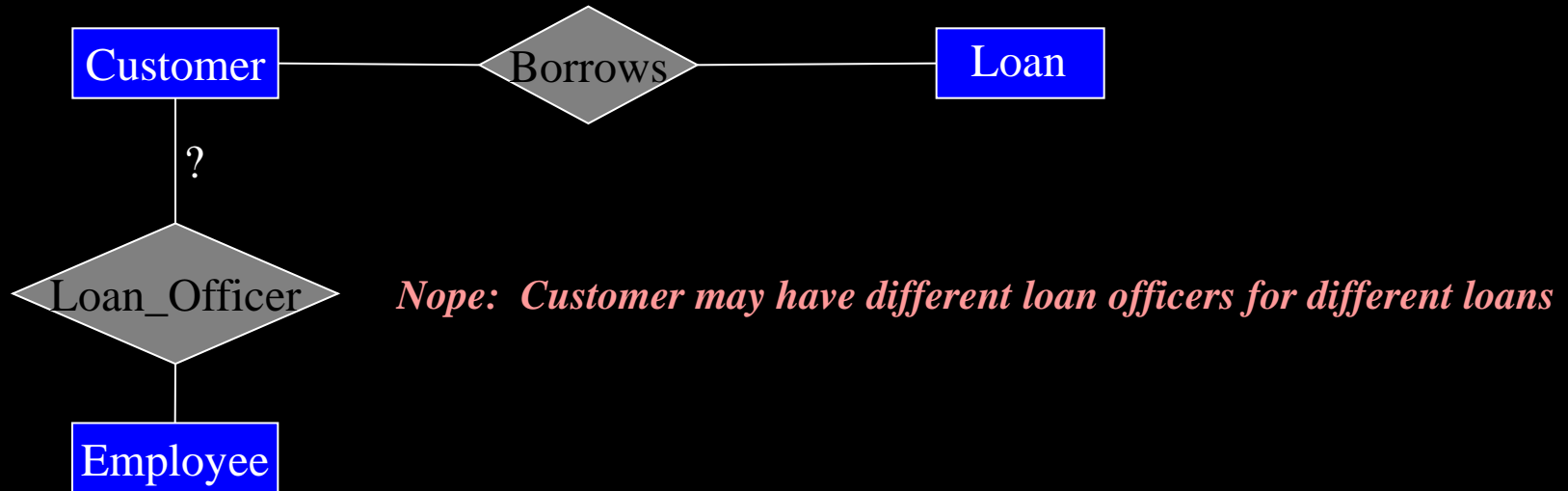
E/R: No relationships between relationships



e.g., may want to assign a loan officer for every customer, loan

E/R: Aggregation

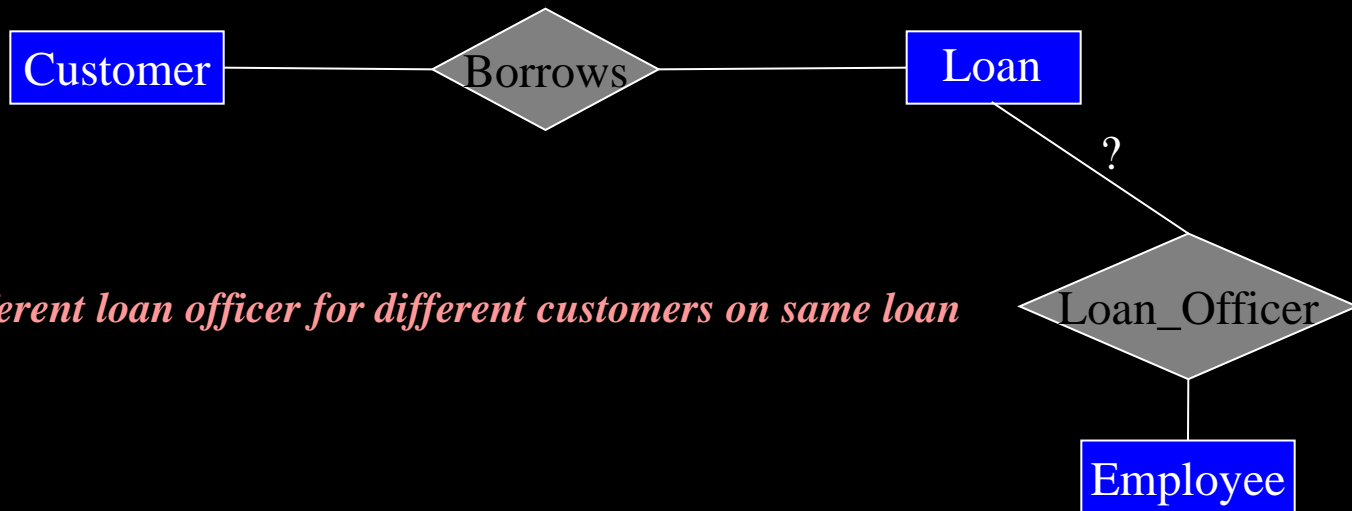
E/R: No relationships between relationships



e.g., may want to assign a loan officer for every customer, loan

E/R: Aggregation

E/R: No relationships between relationships

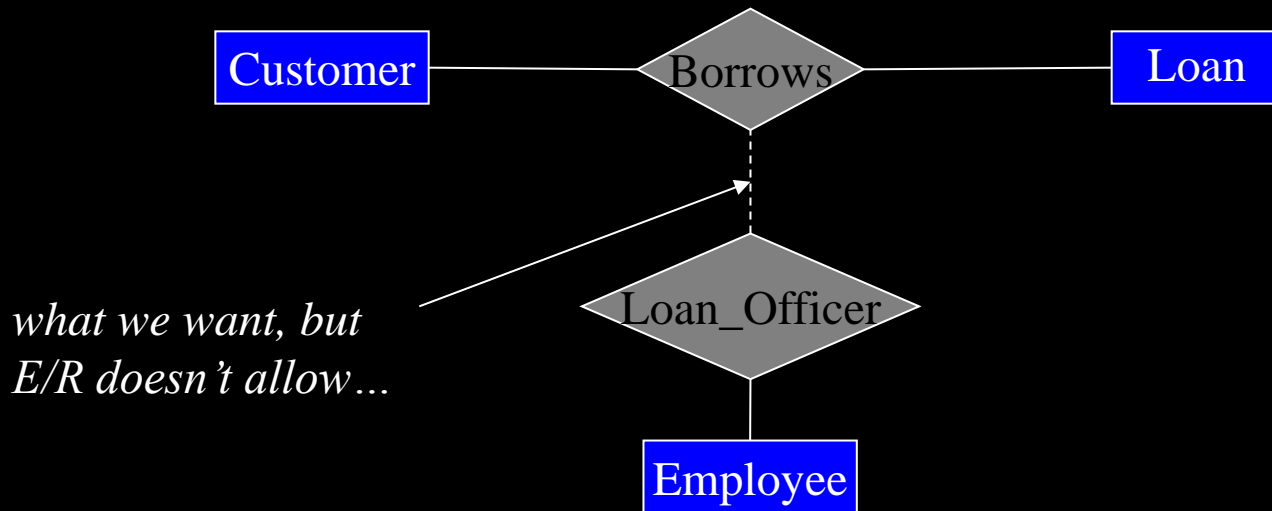


Nope: Different loan officer for different customers on same loan

e.g., may want to assign a loan officer for every customer, loan

E/R: Aggregation

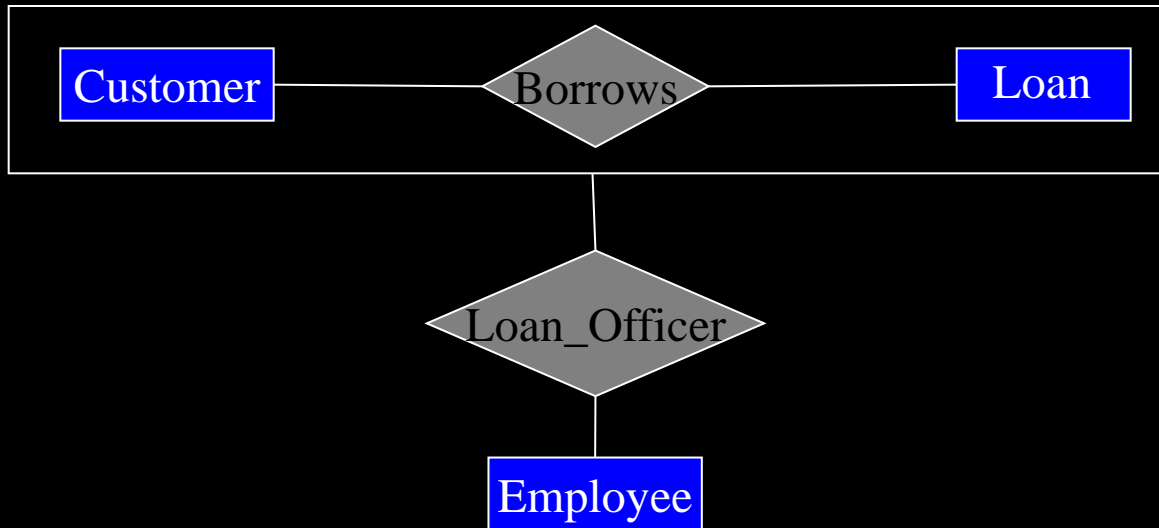
E/R: No relationships between relationships



e.g., may want to assign a loan officer for every customer, loan

E/R: Aggregation

E/R: No relationships between relationships



To associate Loan Officer with Borrows...

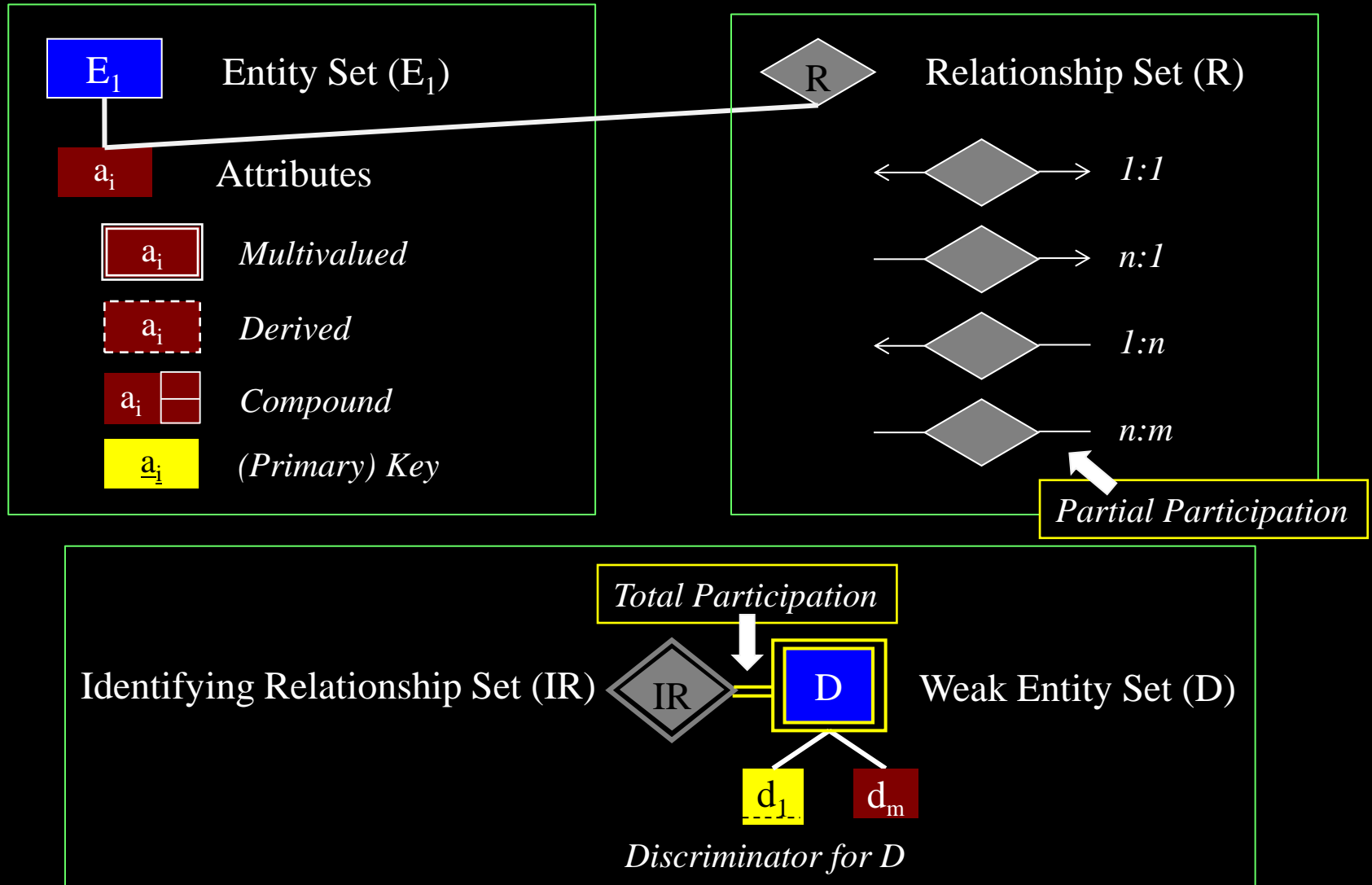
- must first **aggregate**
- makes an entity set out of a relationship set

E/R Data Model

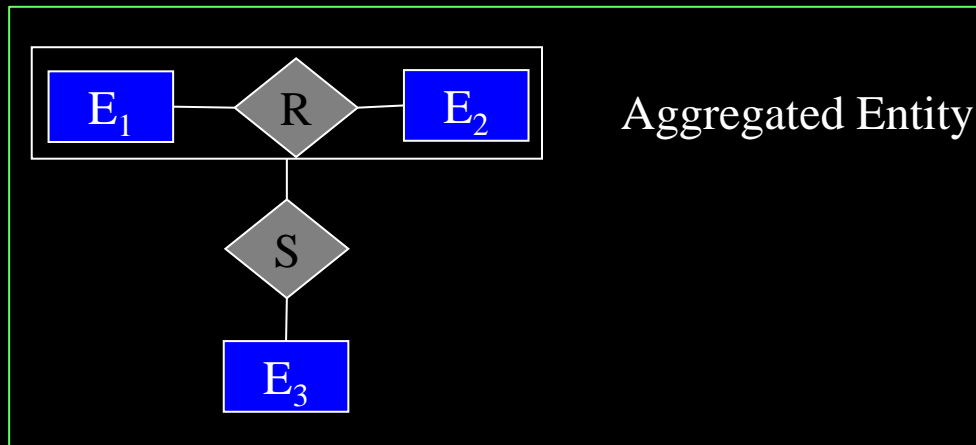
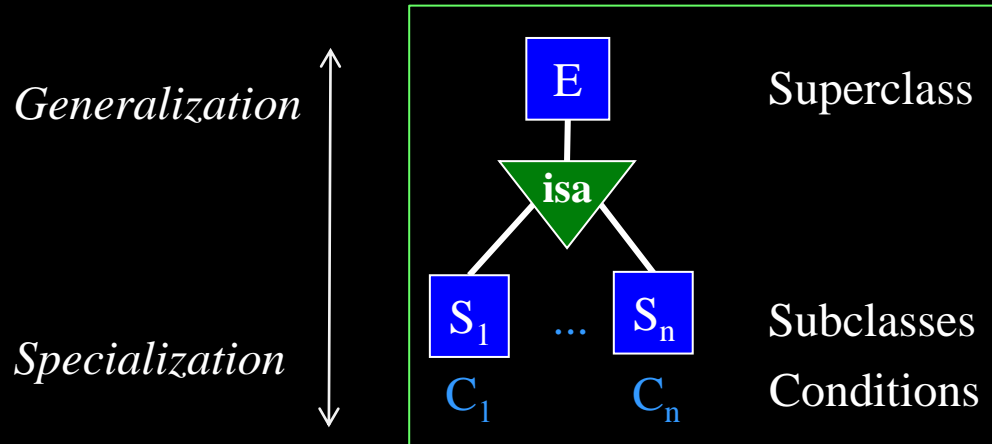
Summary

- Entities, Relationships (sets)
- Both can have attributes (simple, multivalued, derived, composite)
- Cardinality or relationship sets (1:1, n:1, n:m)
- Keys: superkeys, candidate keys, primary key
 - DBA chooses primary key for entity sets
 - Automatically determined for relationship sets
- Weak Entity Sets, Existence Dependence, Total/Partial Participation
- Specialization and Generalization (E/R + inheritance)
- Aggregation (E/R + higher-order relationships)

E/R Cheat Sheet



E/R Cheat Sheet



ER: Exercise

Model the Data Requirements for a University

1. Professors have an SSN, rank, name, research area
2. Projects have a project #, sponsor name, start/end dates and budget
3. Grad students have an ssn, name, age and degree program (M.S or Ph.D)
4. Each project is managed by a professor, who can manage more than 1 project (PI)
5. Any project can be worked on by professors who aren't managers (co-PI's)
6. Each project is worked on by 1+ grad students (RA's). Each grad student works on 1+ project.
7. Every RA appointment must be supervised by a professor.
8. Depts have a dept number, name and office.
9. Depts have a professor who is chairman.
10. Professors can be cross-appointed to 1+ depts. Each appointment has an associated time %.
11. Grad students have 1 major department. Every department has at least 1 major.
12. Each grad student is advised by another (more senior) grad student.

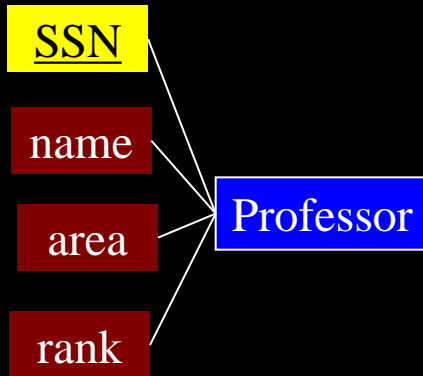
ER: Exercise

Model the Data Requirements for a University

1. Professors have an SSN, rank, name, research area

ER: Exercise

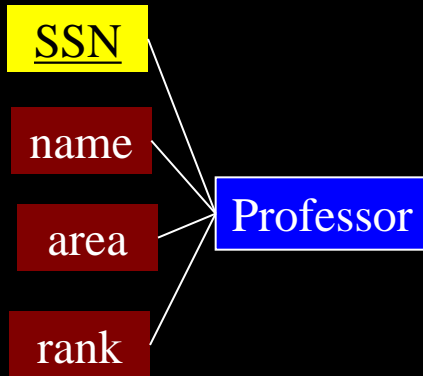
Model the Data Requirements for a University



1. Professors have an SSN, rank, name, research area

ER: Exercise

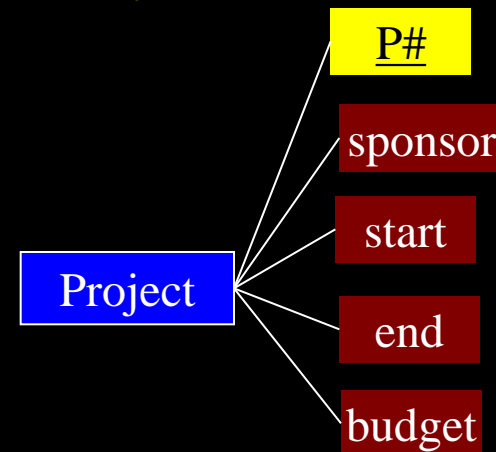
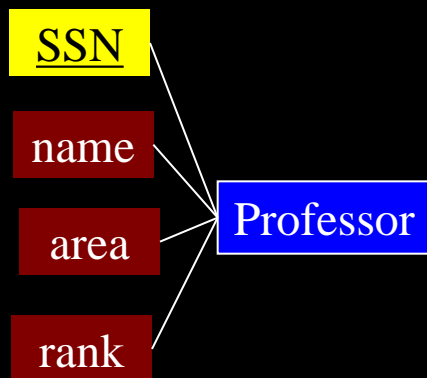
Model the Data Requirements for a University



2. Projects have a project #, sponsor name, start/end dates and budget

ER: Exercise

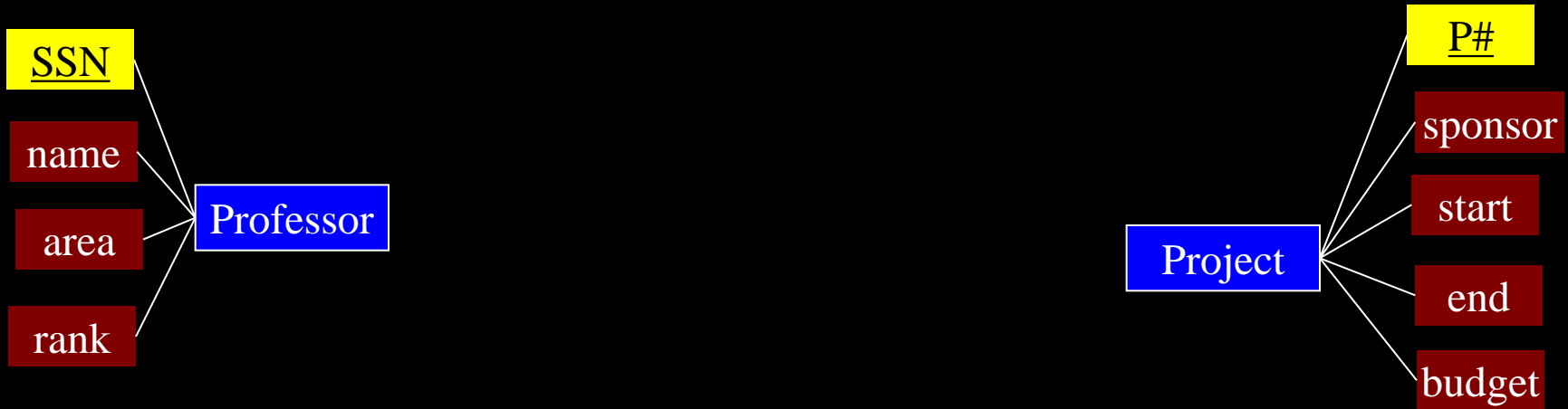
Model the Data Requirements for a University



2. Projects have a project #, sponsor name, start/end dates and budget

ER: Exercise

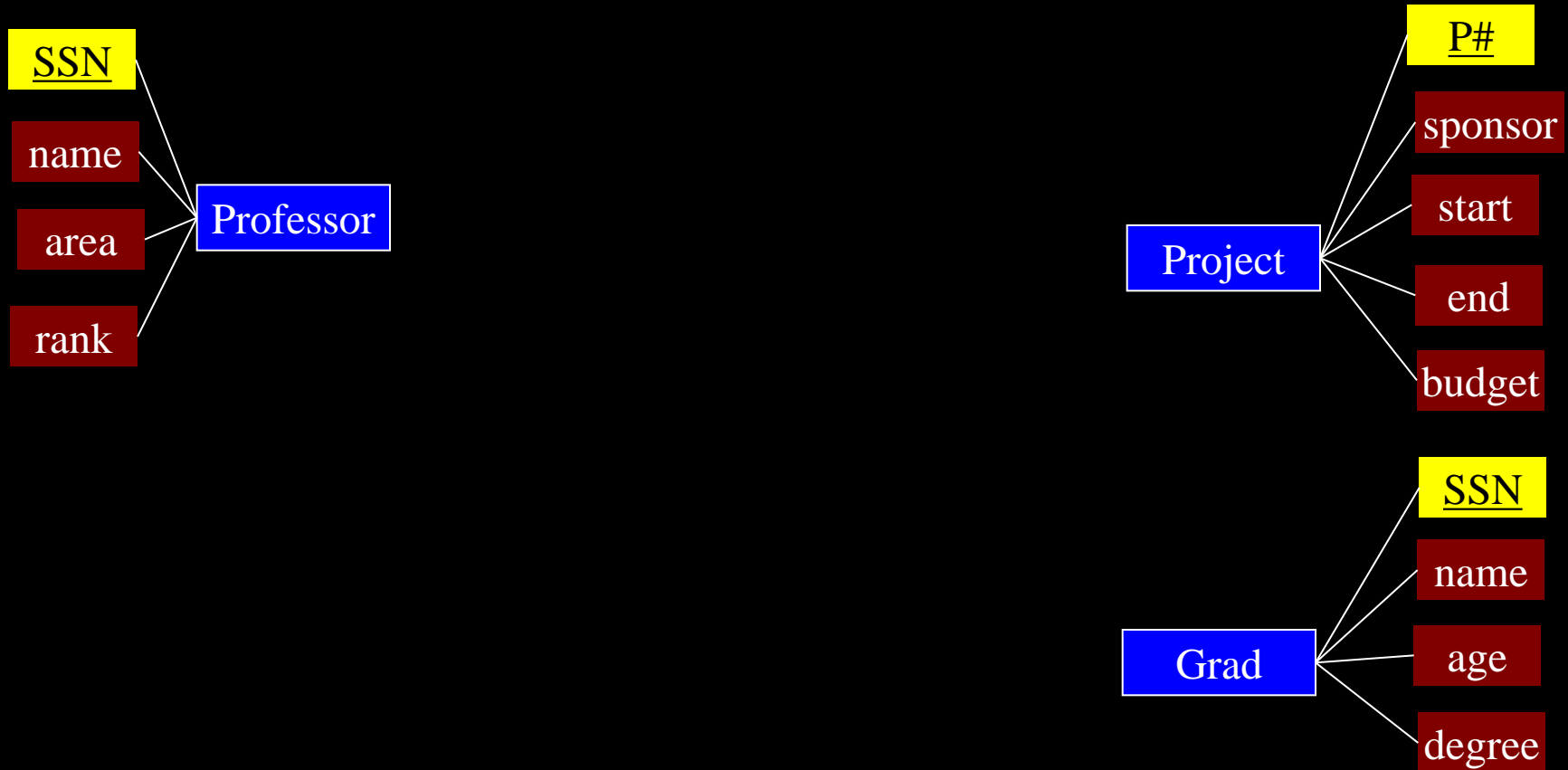
Model the Data Requirements for a University



3. Grad students have an ssn, name, age and degree program (M.S or Ph.D)

ER: Exercise

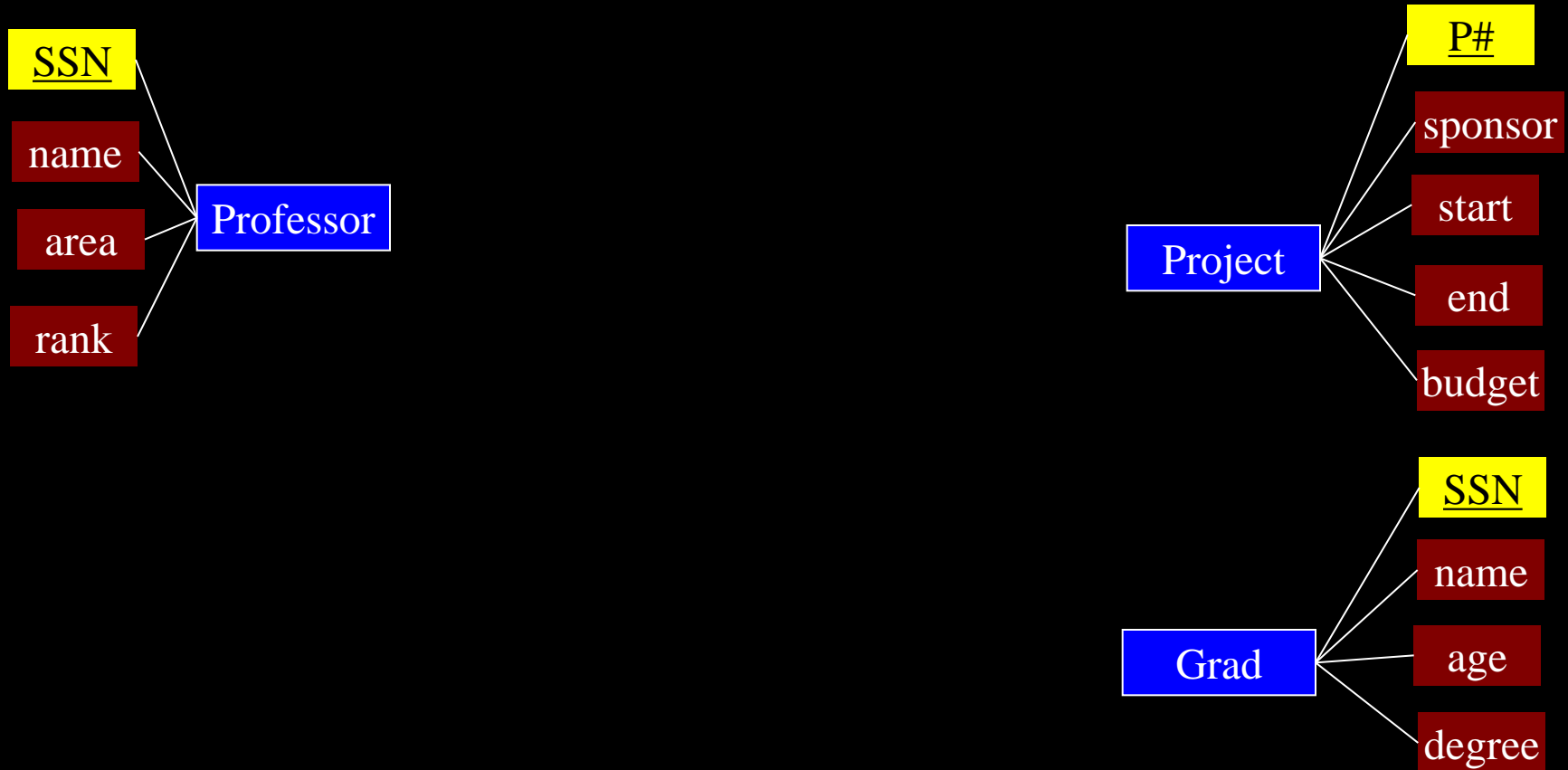
Model the Data Requirements for a University



3. Grad students have an ssn, name, age and degree program (M.S or Ph.D)

ER: Exercise

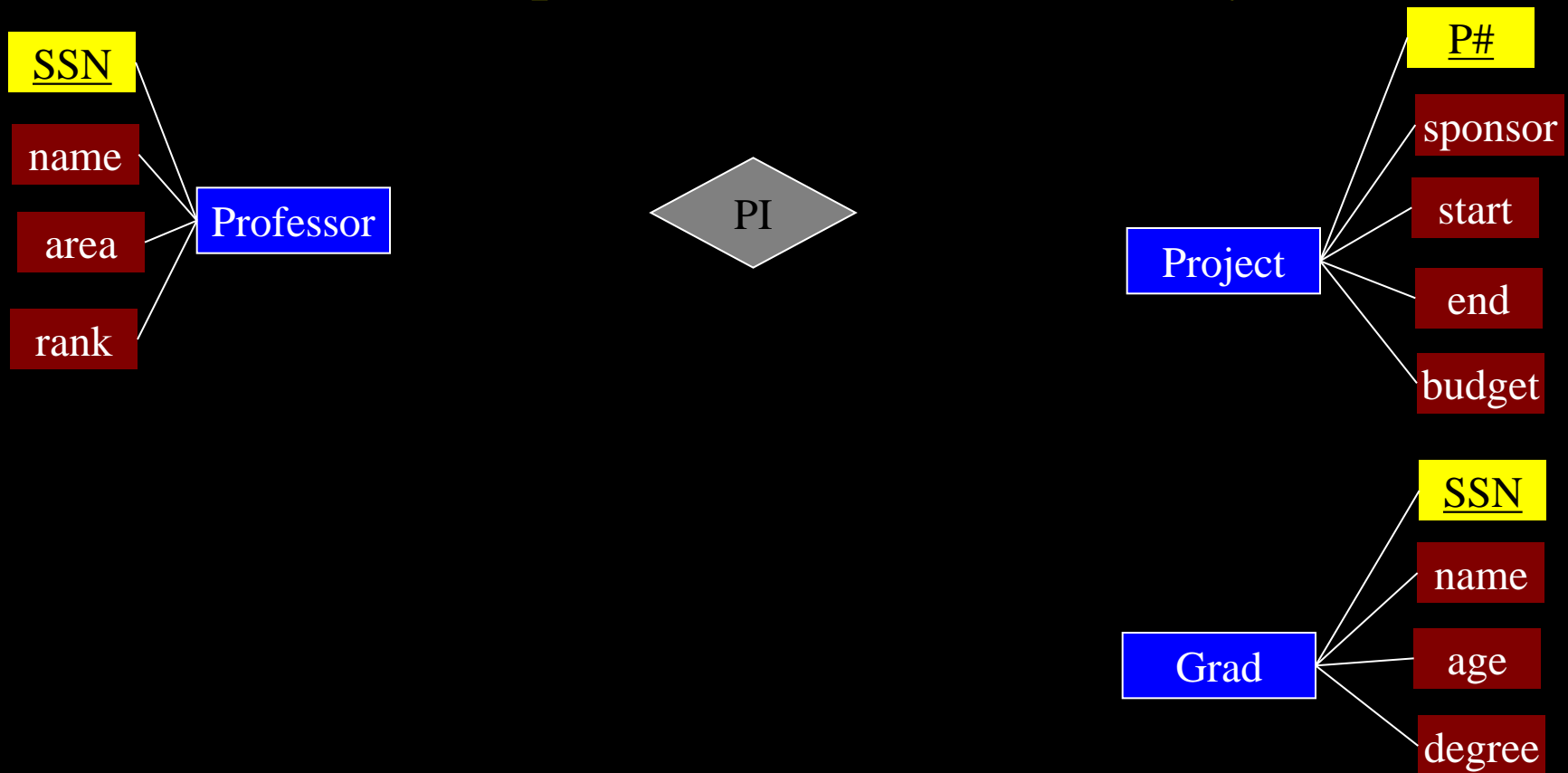
Model the Data Requirements for a University



4. Each project is managed by a professor, who can manage more than 1 project (PI)

ER: Exercise

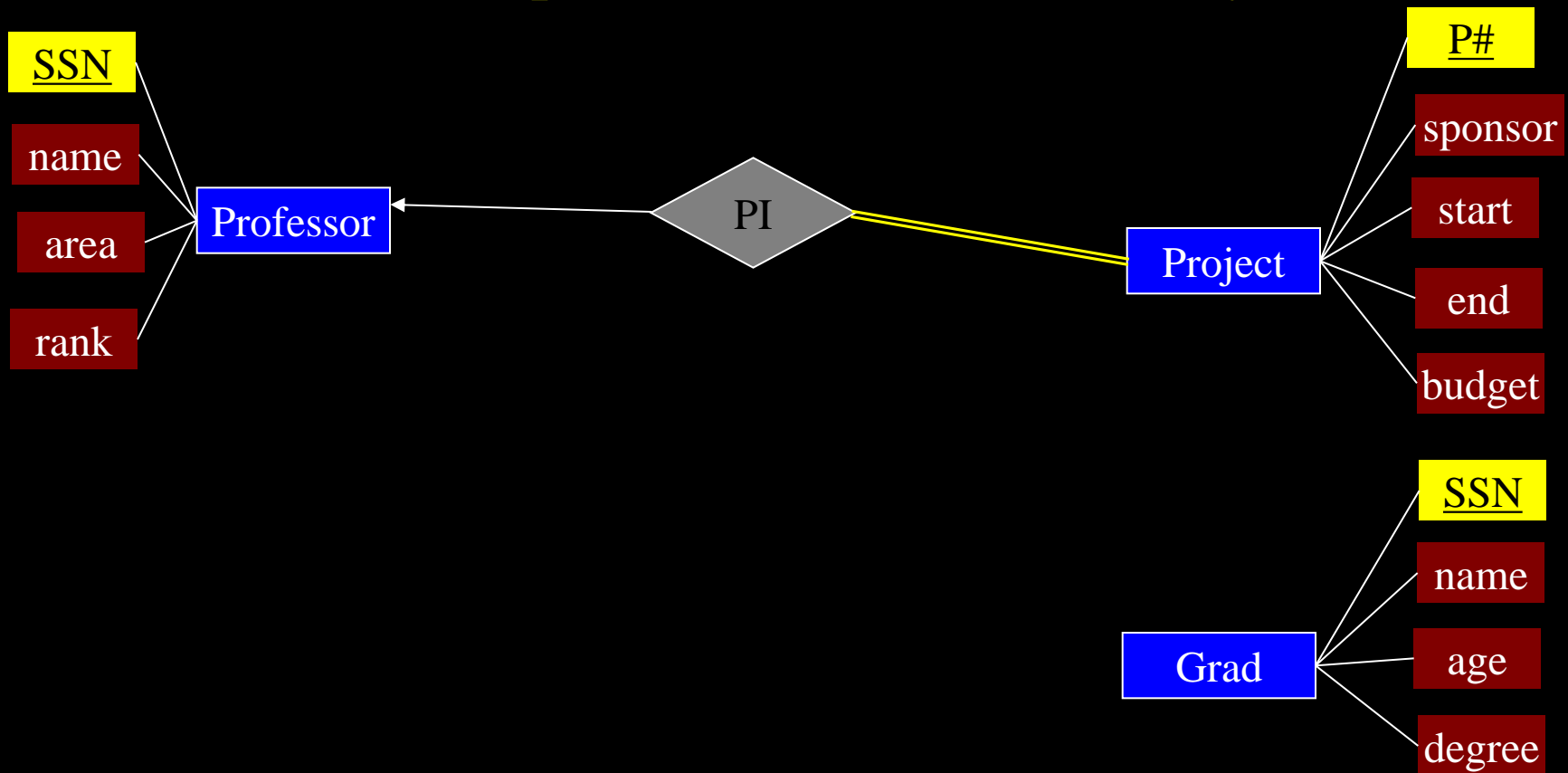
Model the Data Requirements for a University



4. Each project is managed by a professor, who can manage more than 1 project (PI)

ER: Exercise

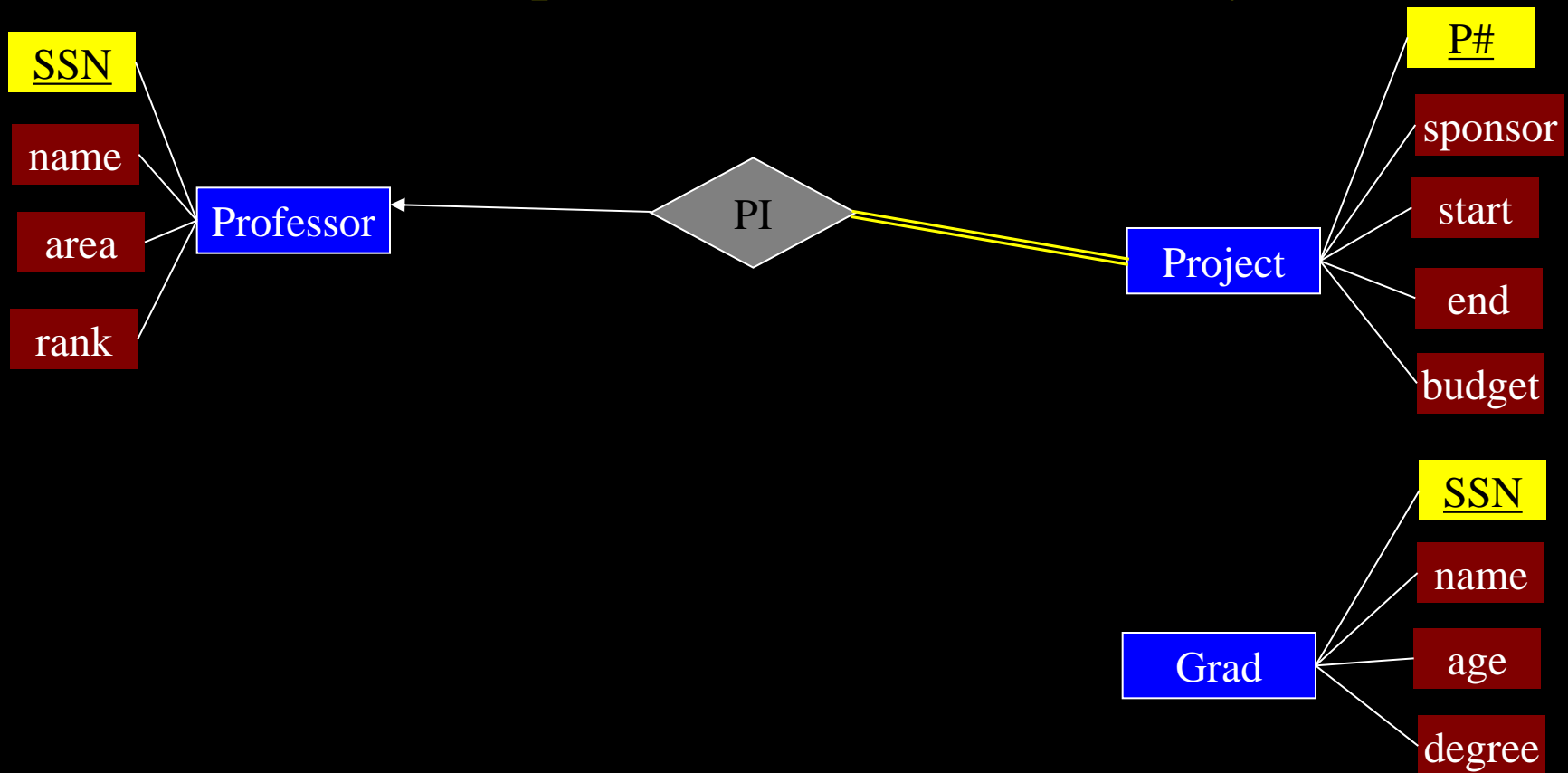
Model the Data Requirements for a University



4. Each project is managed by a professor, who can manage more than 1 project (PI)

ER: Exercise

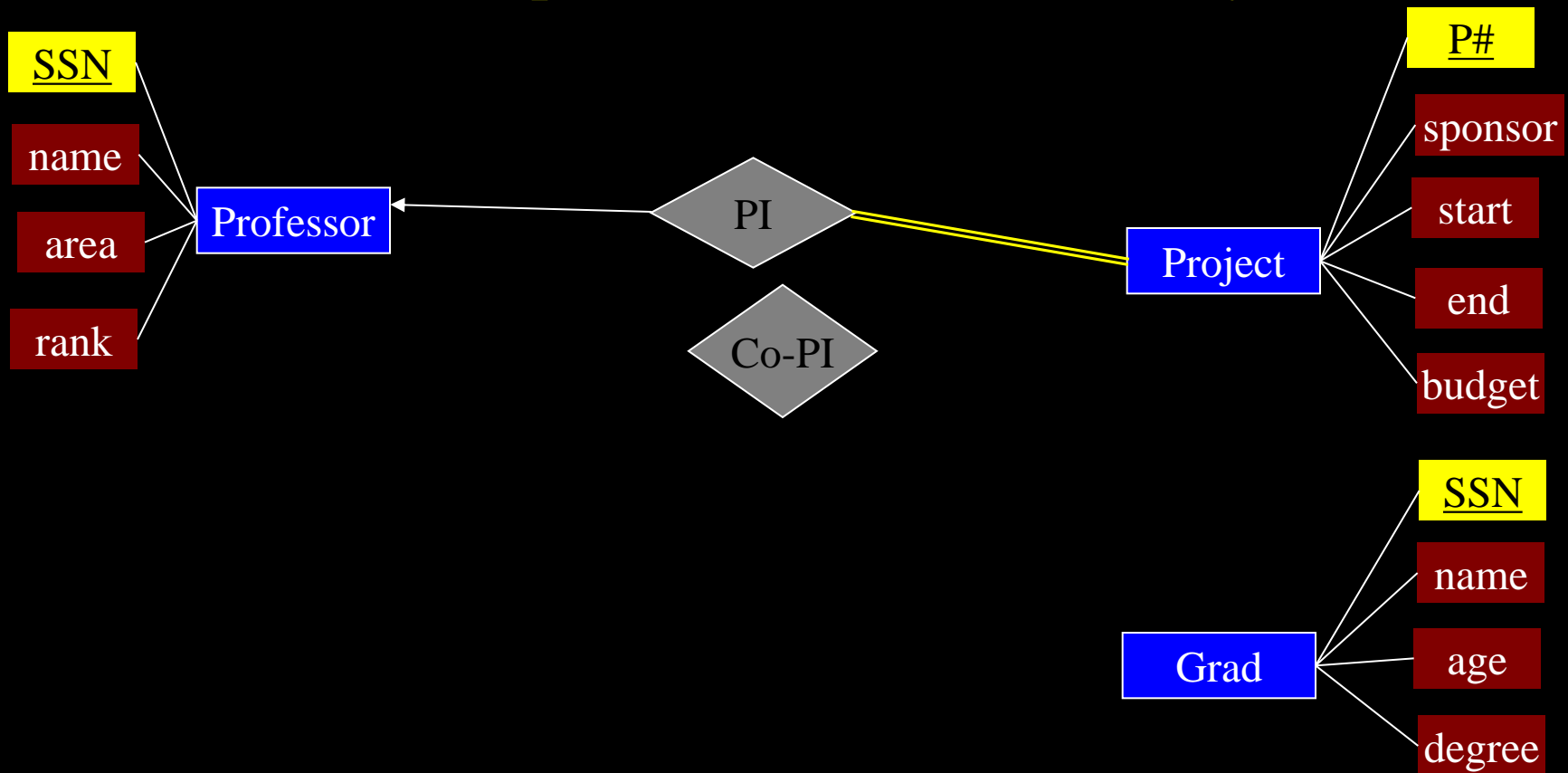
Model the Data Requirements for a University



5. Any project can be worked on by professors who aren't managers (co-PI's)

ER: Exercise

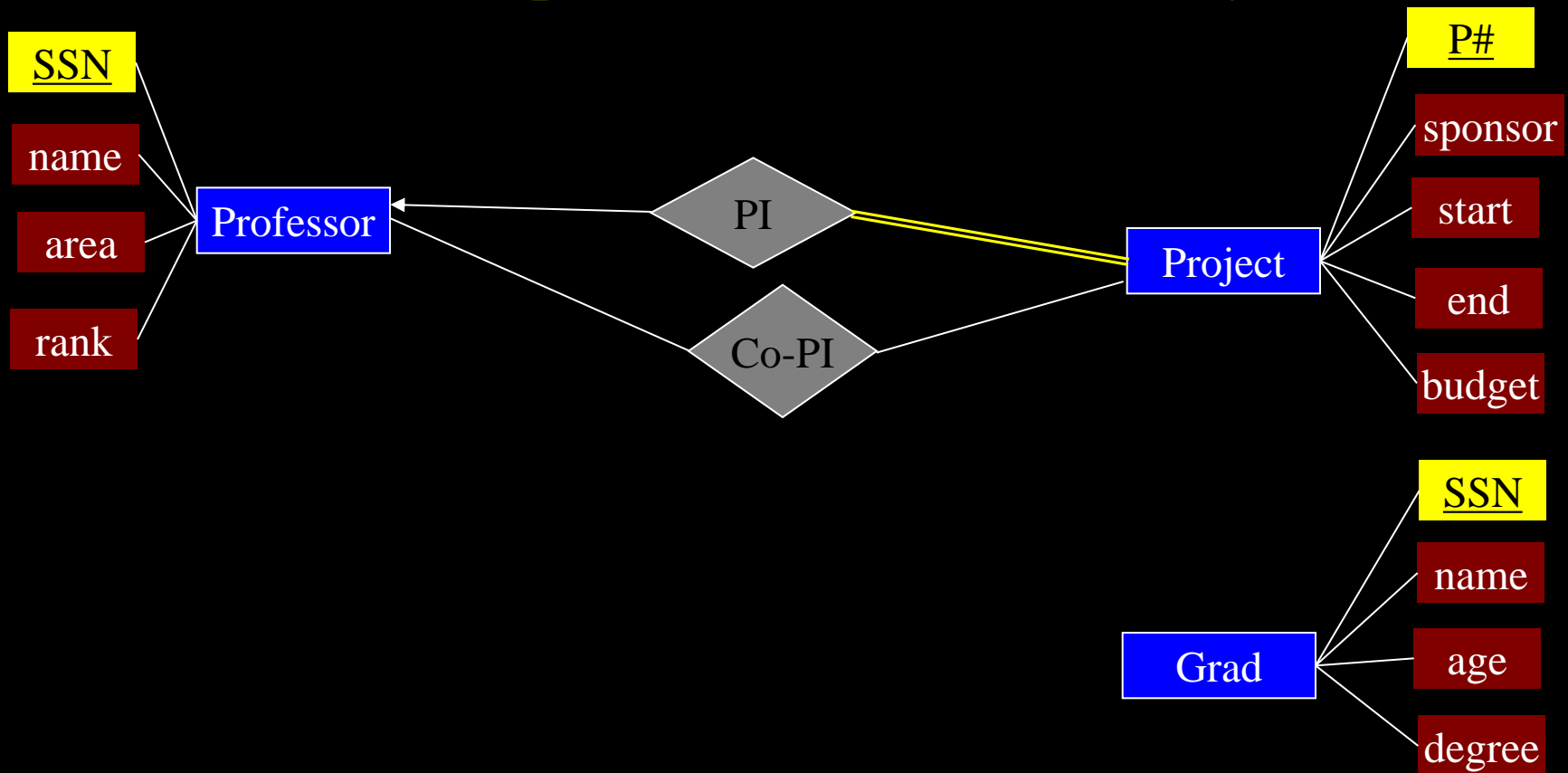
Model the Data Requirements for a University



5. Any project can be worked on by professors who aren't managers (co-PI's)

ER: Exercise

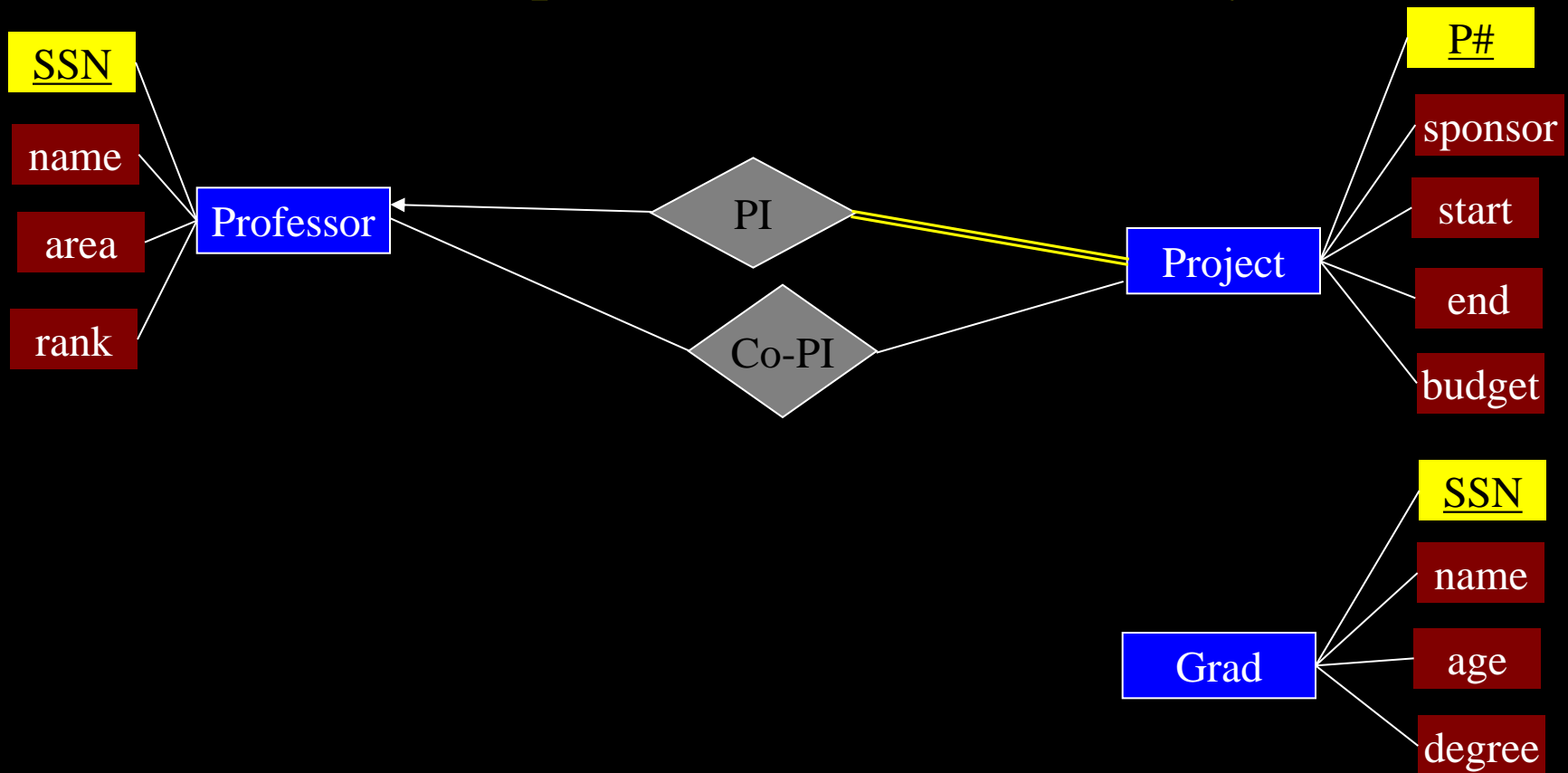
Model the Data Requirements for a University



5. Any project can be worked on by professors who aren't managers (co-PI's)

ER: Exercise

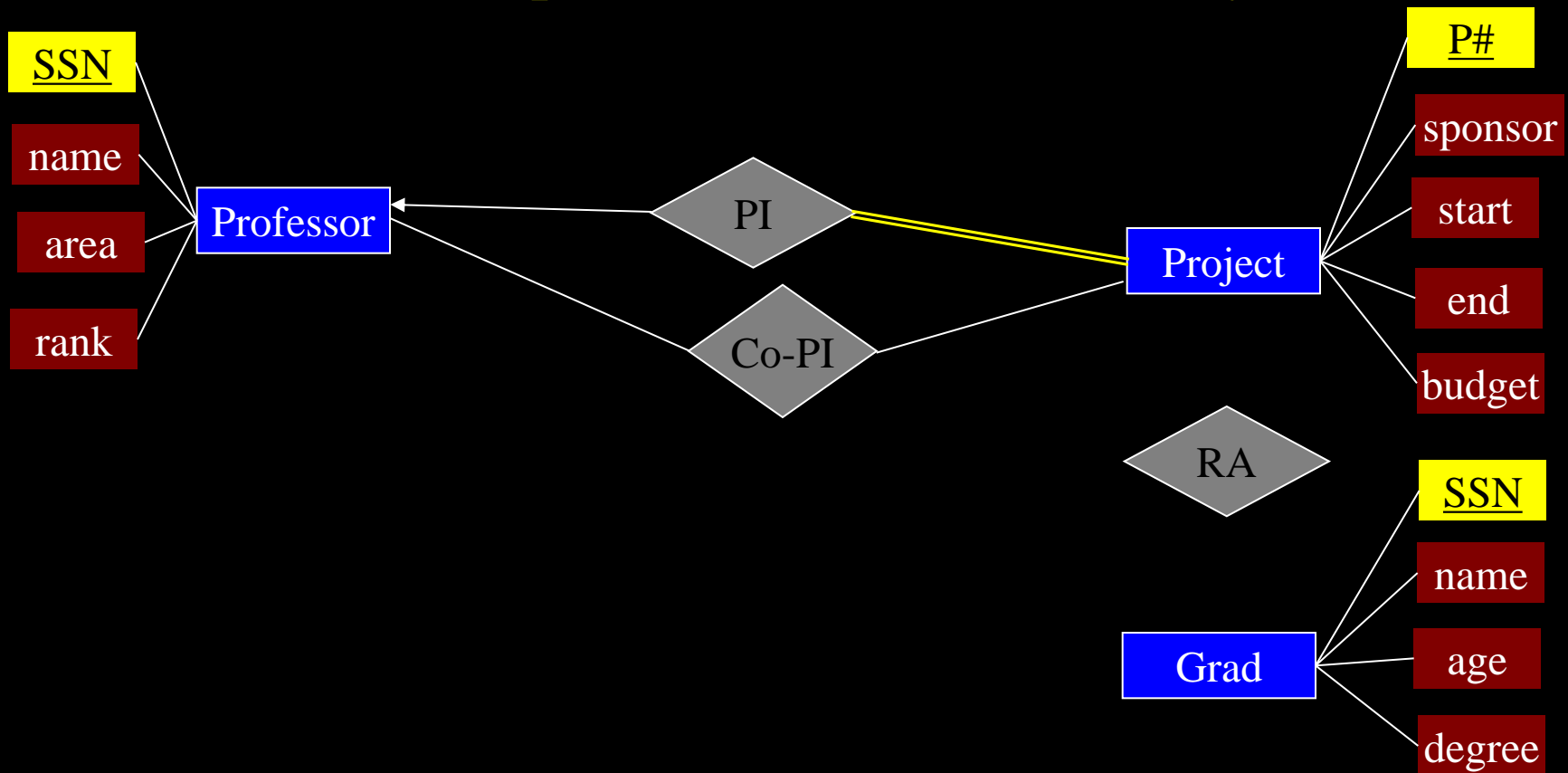
Model the Data Requirements for a University



6. Each project is worked on by 1⁺ grad students (RA's). Each grad student works on 1⁺ project.

ER: Exercise

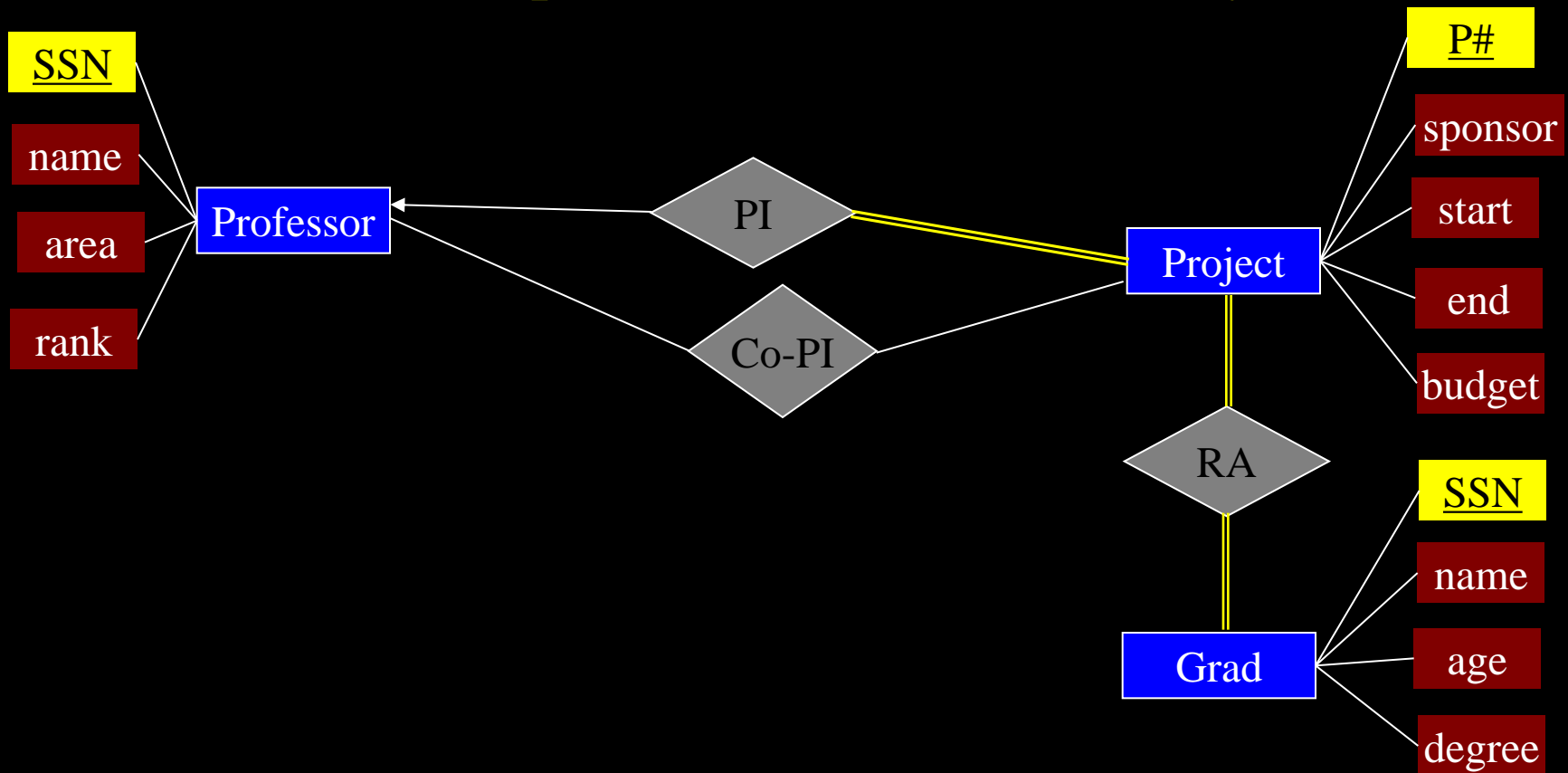
Model the Data Requirements for a University



6. Each project is worked on by 1⁺ grad students (RA's). Each grad student works on 1⁺ project.

ER: Exercise

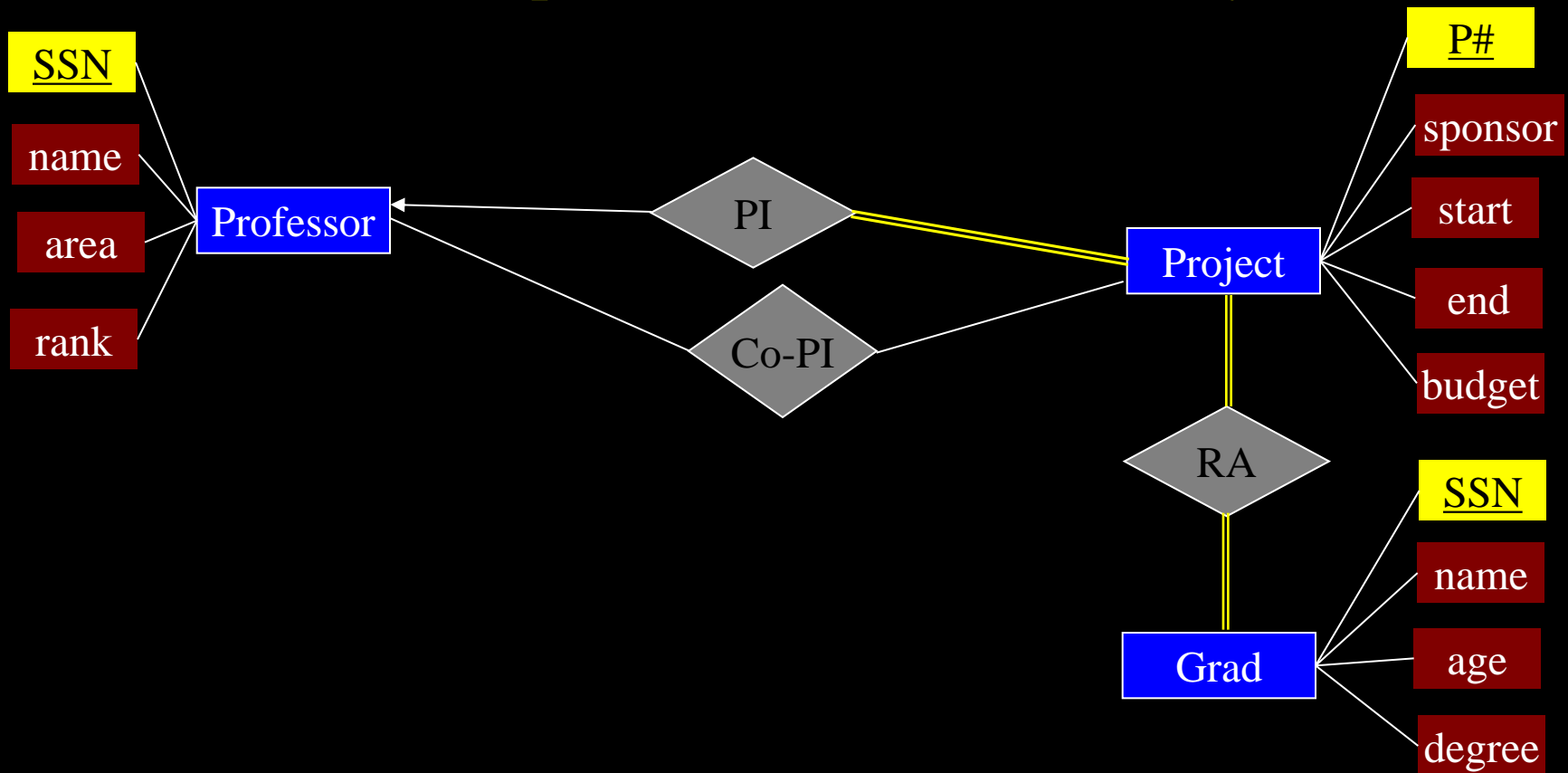
Model the Data Requirements for a University



6. Each project is worked on by 1⁺ grad students (RA's). Each grad student works on 1⁺ project.

ER: Exercise

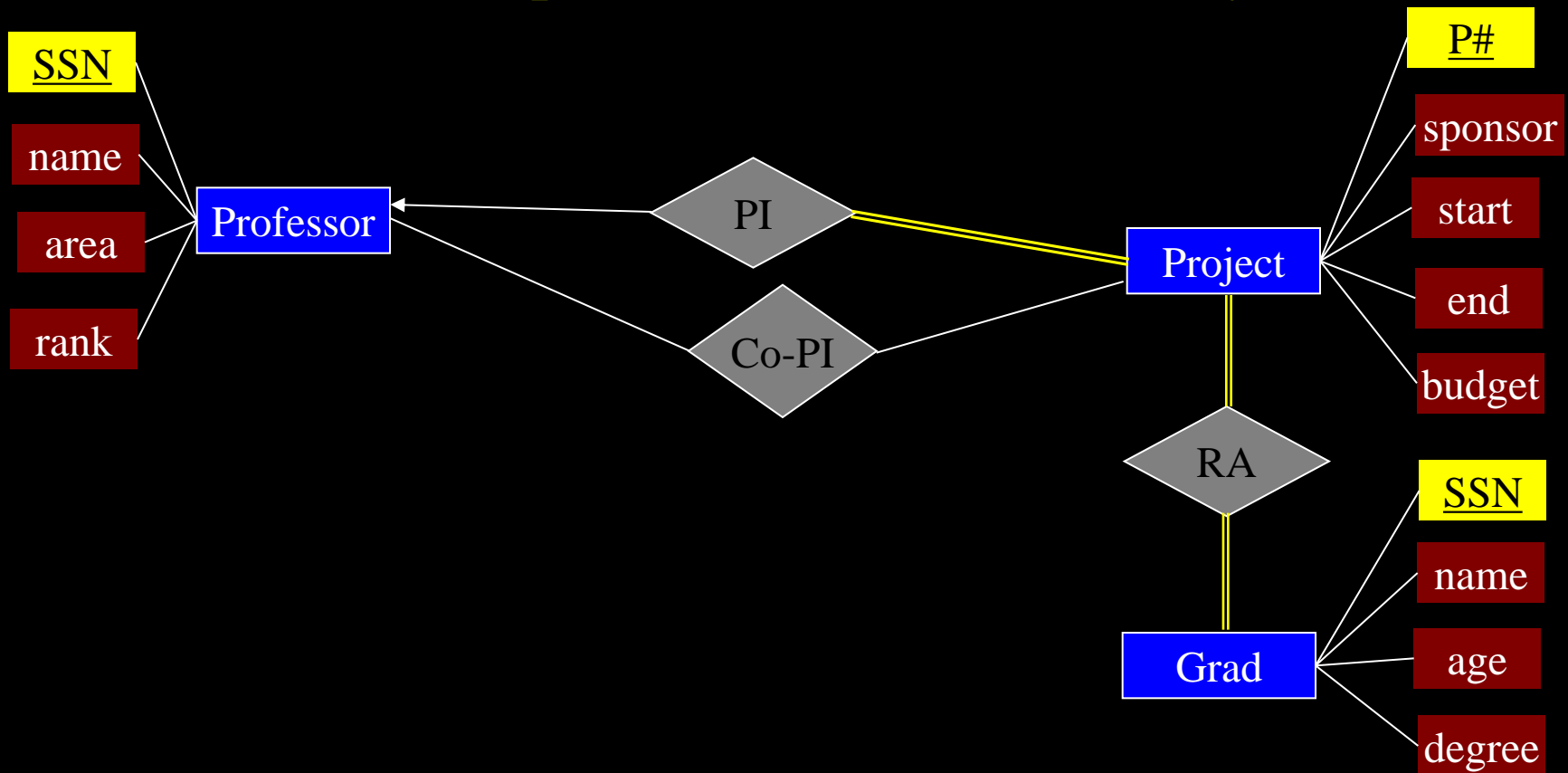
Model the Data Requirements for a University



7. Every RA appointment must be supervised by a professor.

ER: Exercise

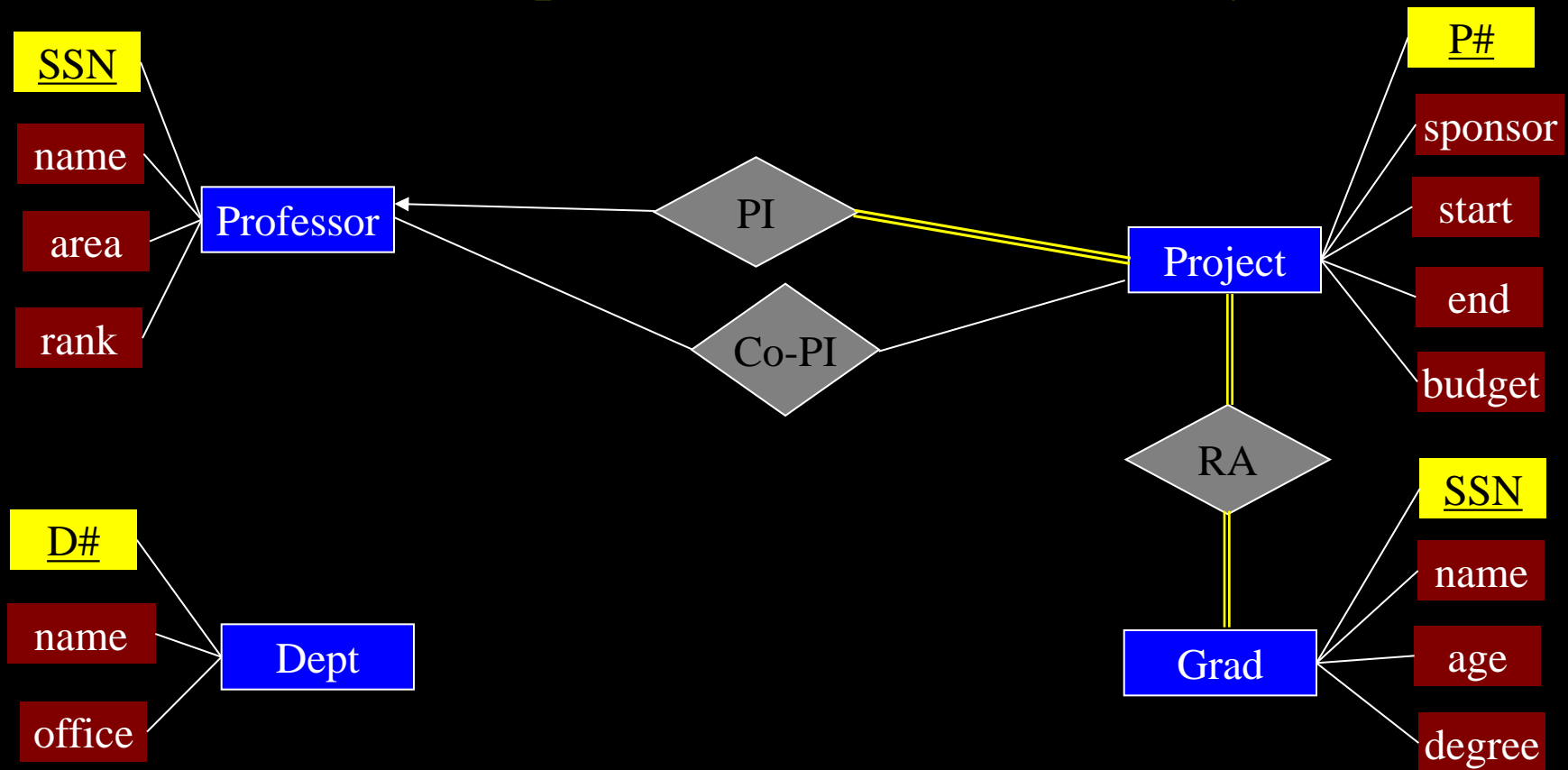
Model the Data Requirements for a University



8. Depts have a dept number, name and office.

ER: Exercise

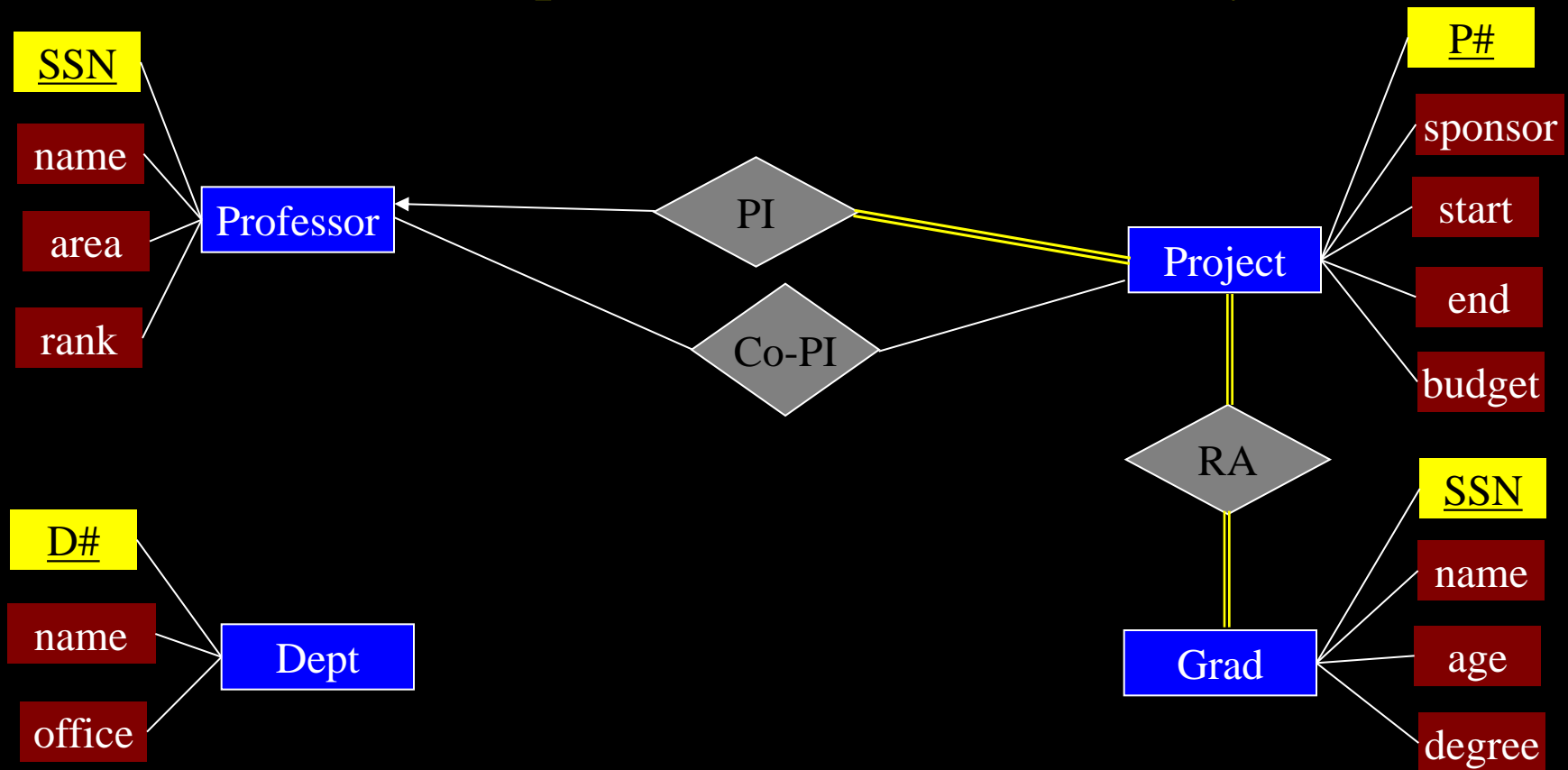
Model the Data Requirements for a University



8. Depts have a dept number, name and office.

ER: Exercise

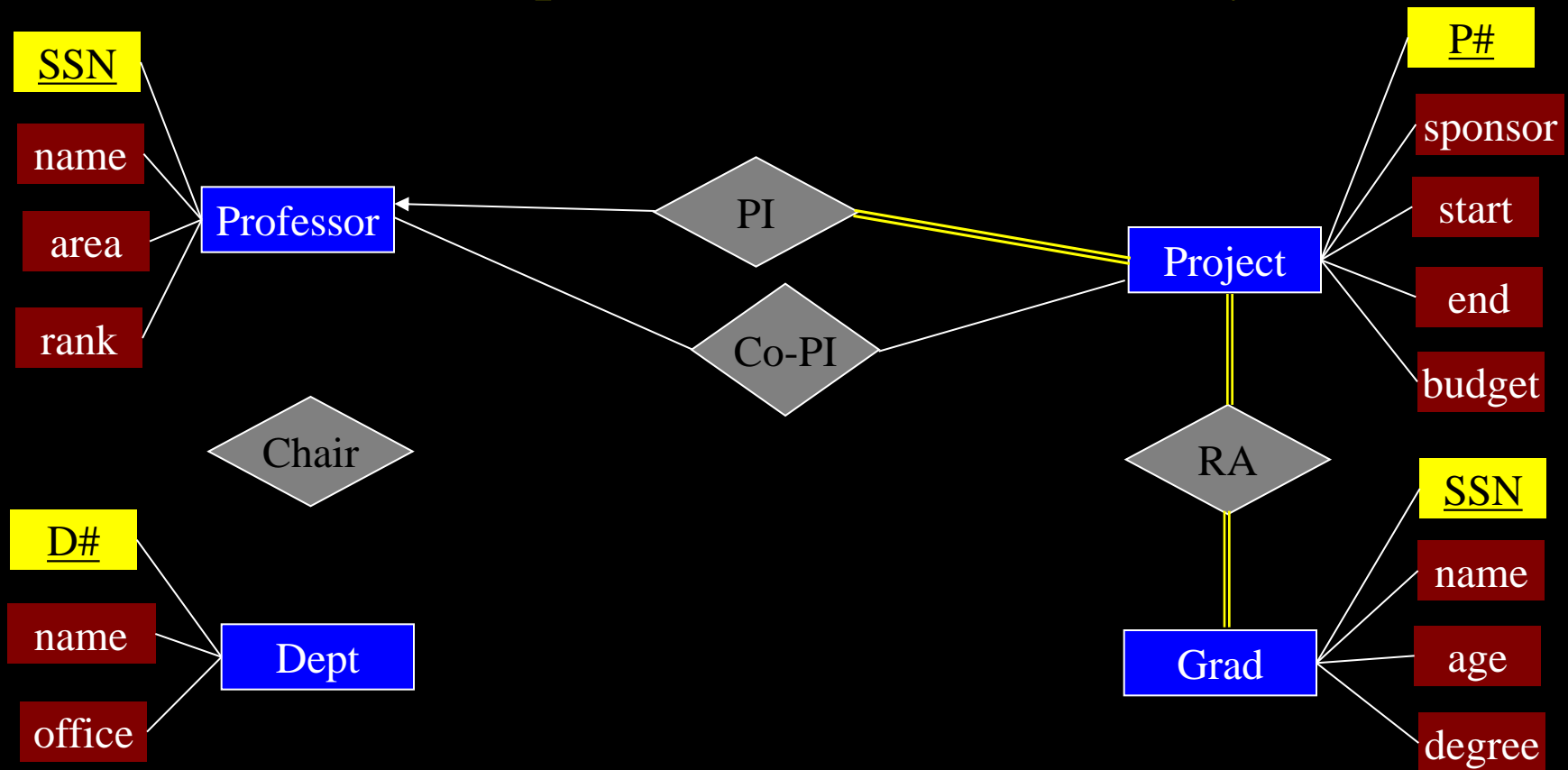
Model the Data Requirements for a University



9. Depts have a professor who is chairman.

ER: Exercise

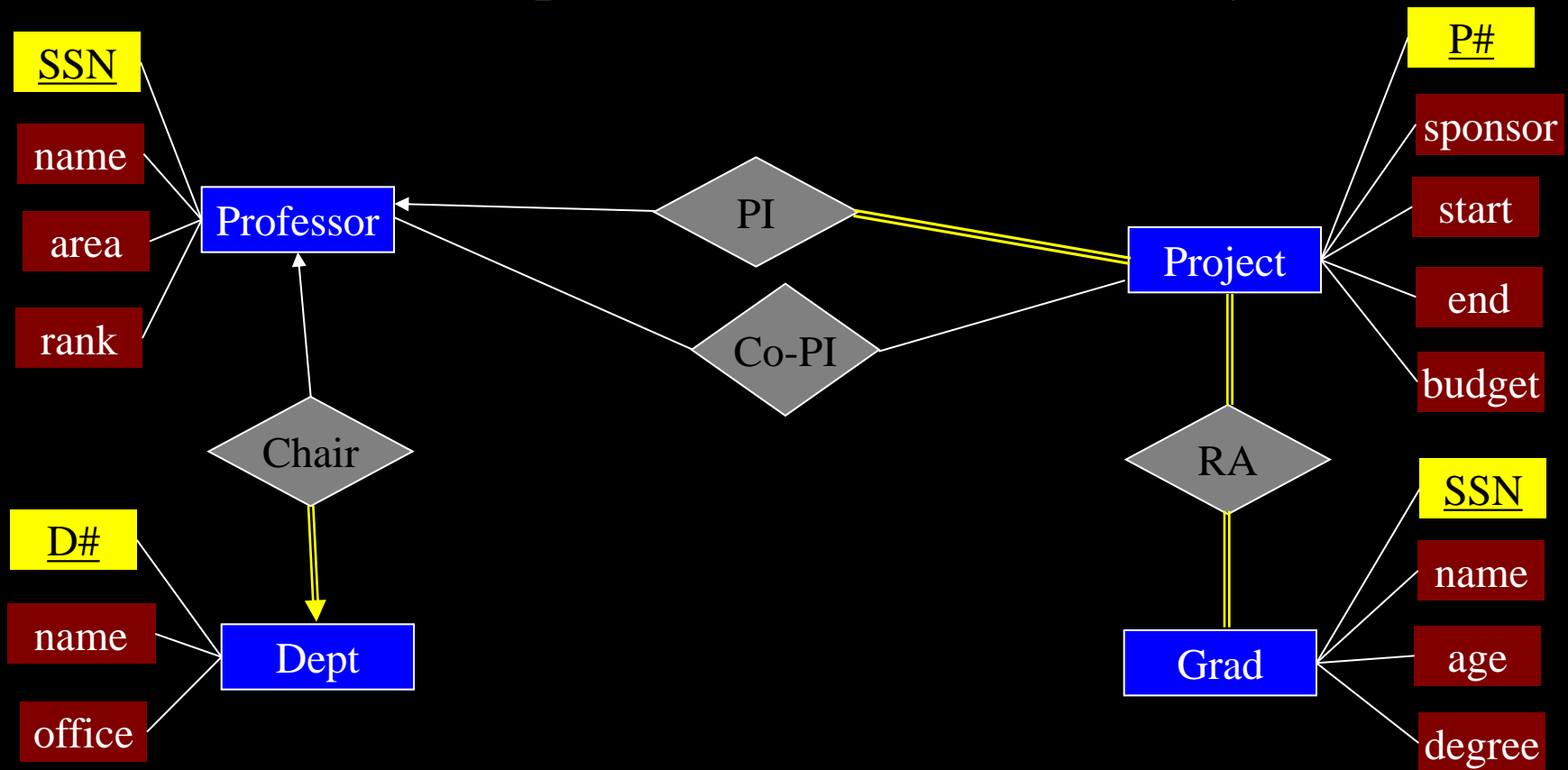
Model the Data Requirements for a University



9. Depts have a professor who is chairman.

ER: Exercise

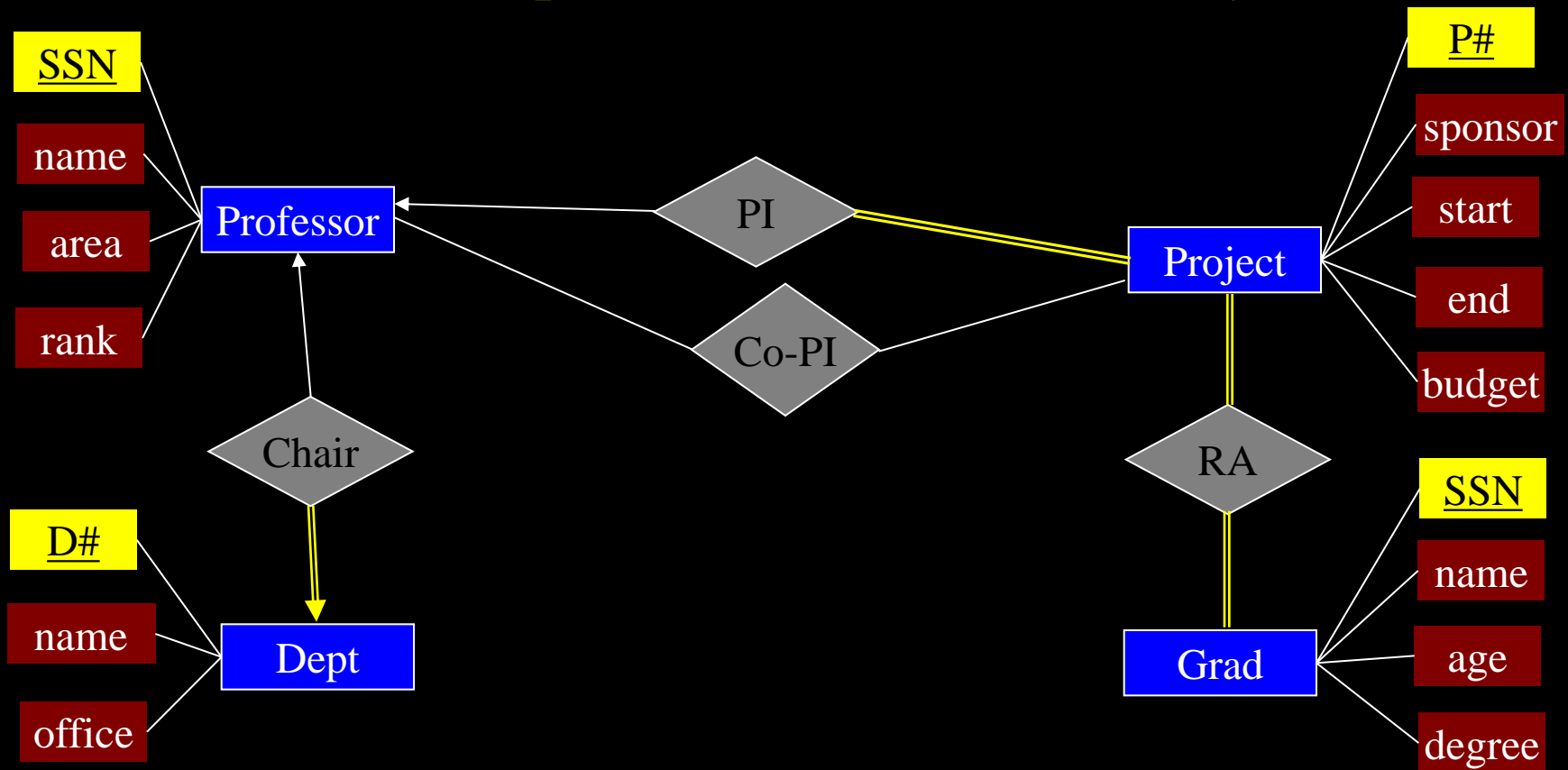
Model the Data Requirements for a University



9. Depts have a professor who is chairman.

ER: Exercise

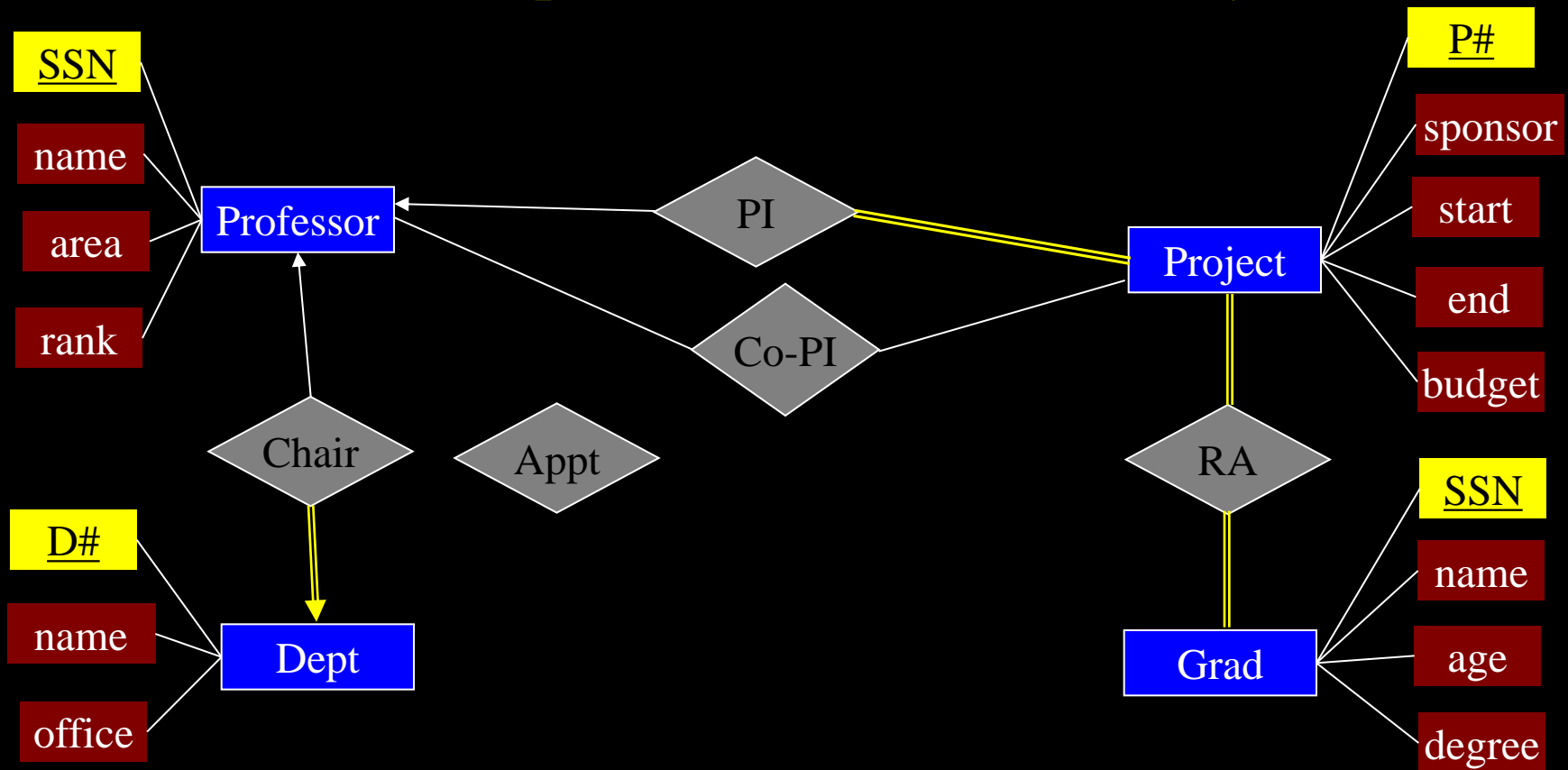
Model the Data Requirements for a University



10. Professors can be cross-appointed to 1+ depts. Each appointment has an associated time %.

ER: Exercise

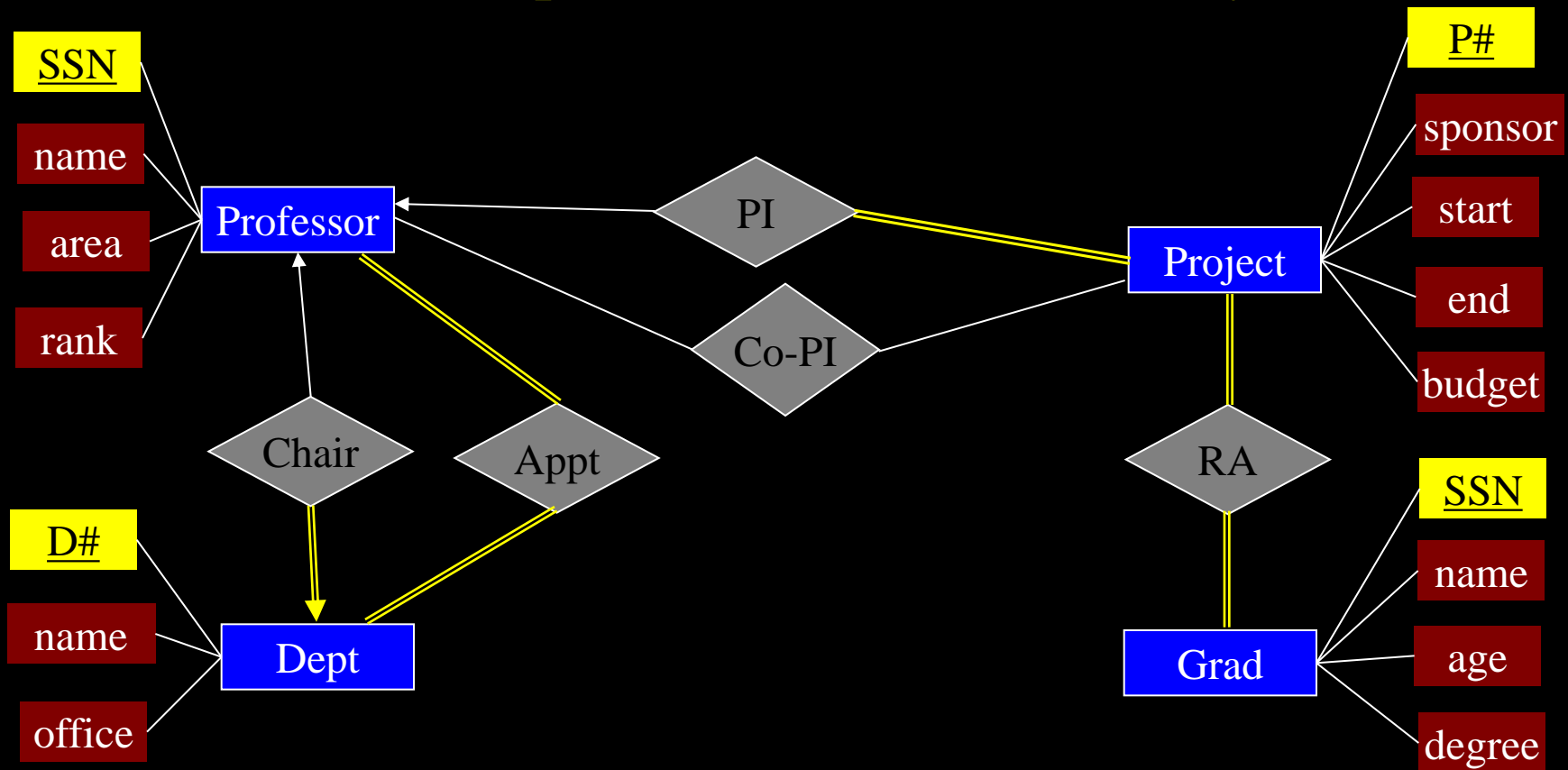
Model the Data Requirements for a University



10. Professors can be cross-appointed to 1+ depts. Each appointment has an associated time %.

ER: Exercise

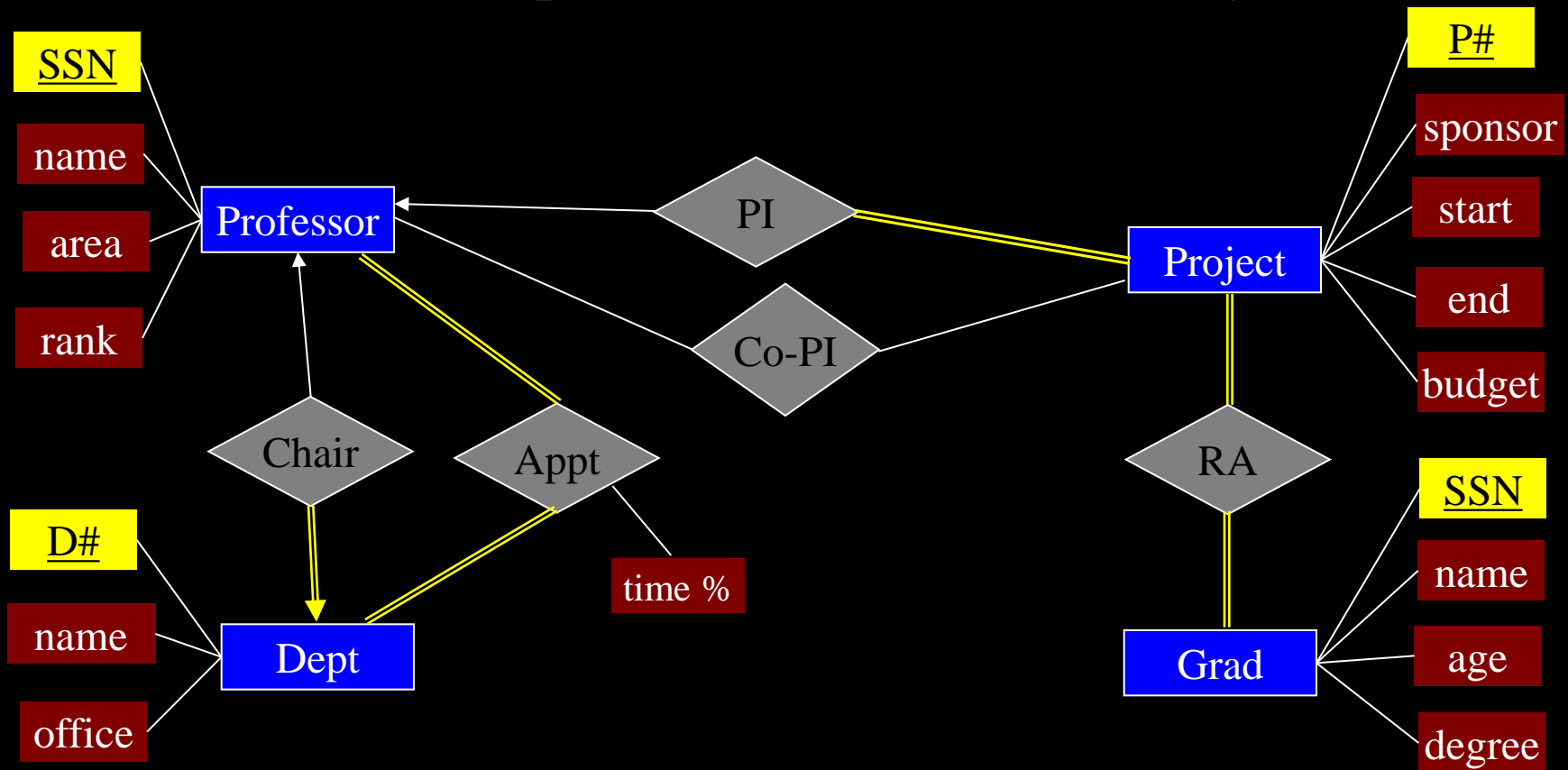
Model the Data Requirements for a University



10. Professors can be cross-appointed to 1+ depts. Each appointment has an associated time %.

ER: Exercise

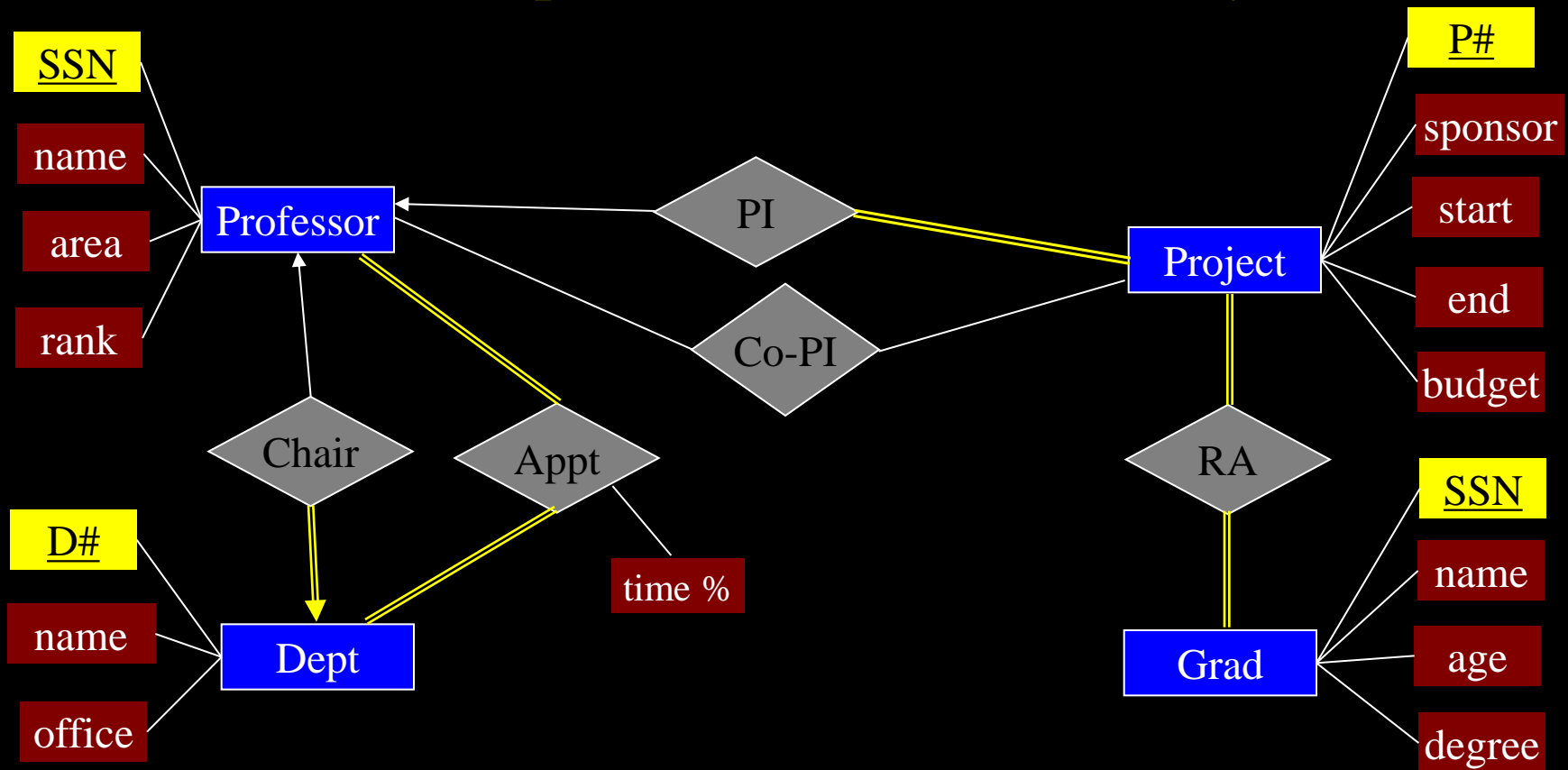
Model the Data Requirements for a University



10. Professors can be cross-appointed to 1+ depts. Each appointment has an associated time %.

ER: Exercise

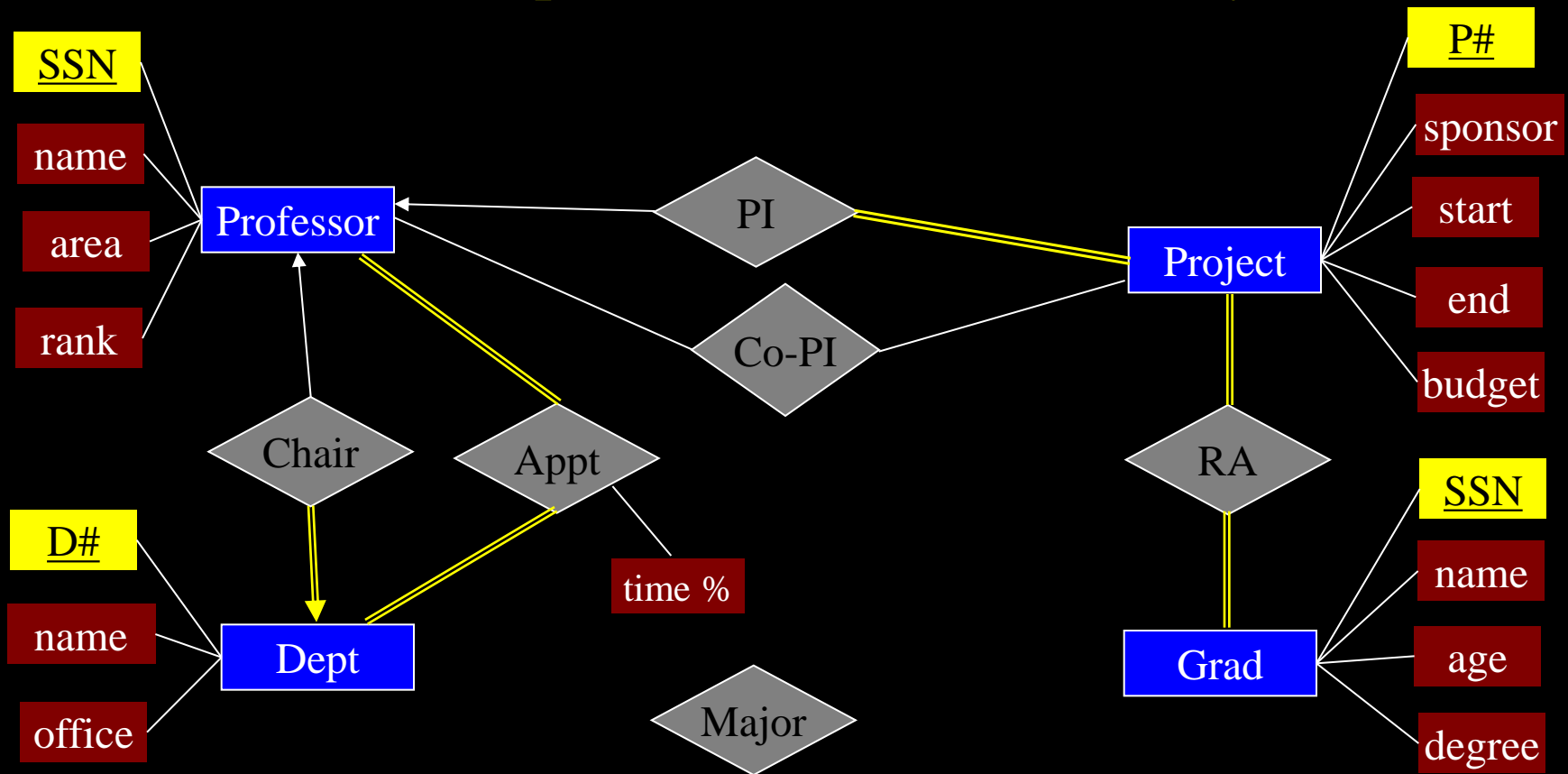
Model the Data Requirements for a University



11. Grad students have 1 major department. Every department has at least 1 major.

ER: Exercise

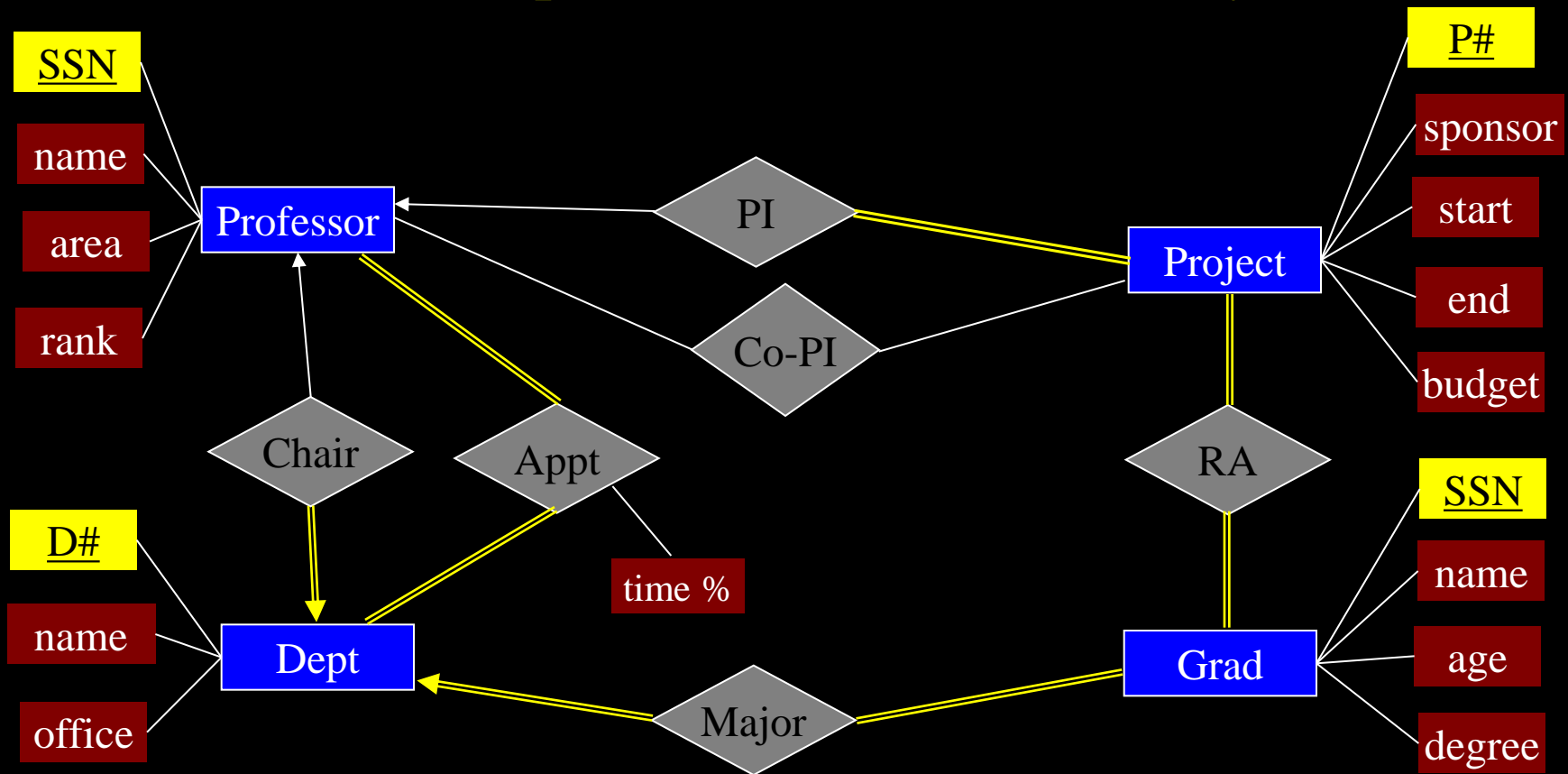
Model the Data Requirements for a University



11. Grad students have 1 major department. Every department has at least 1 major.

ER: Exercise

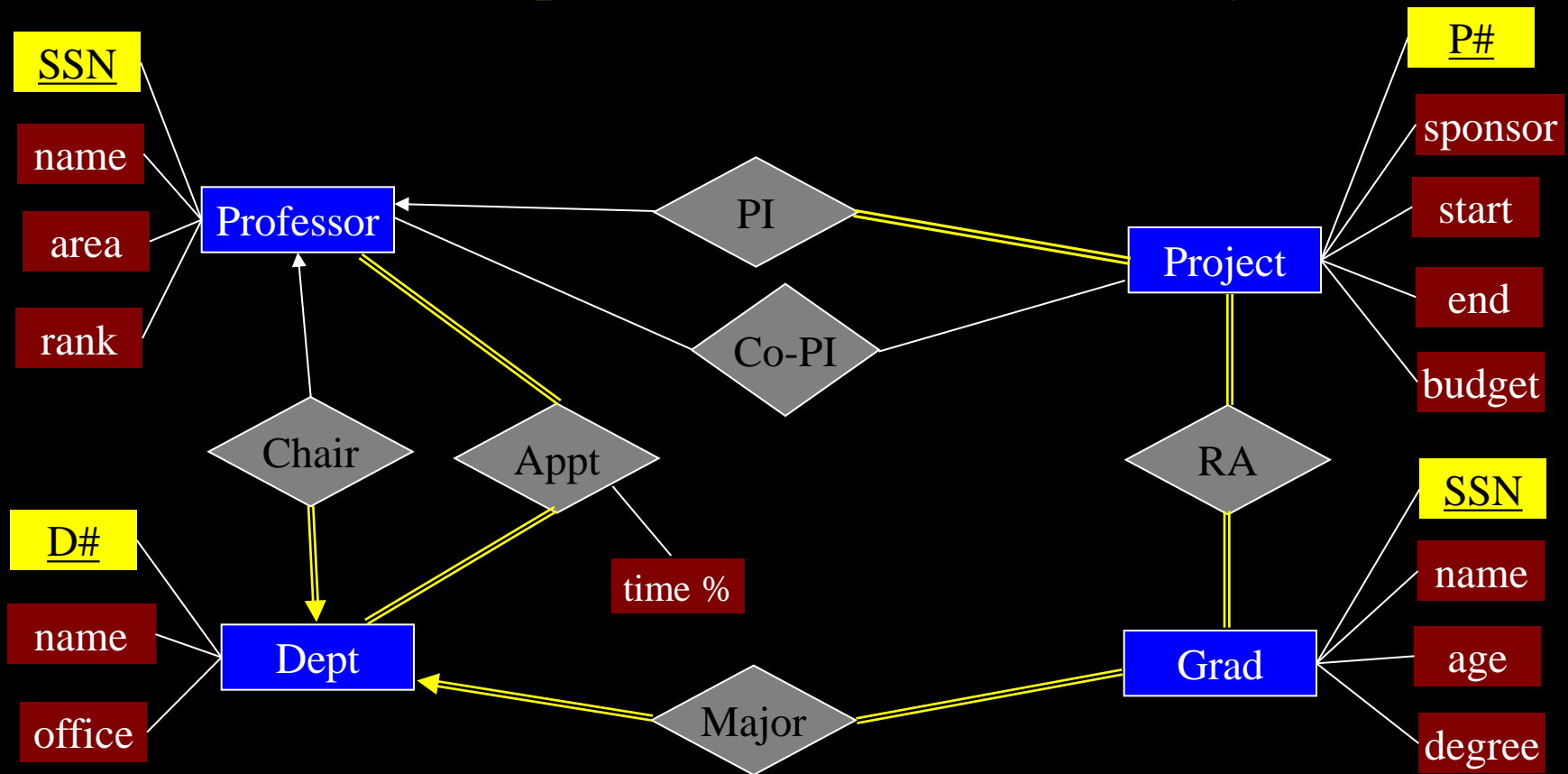
Model the Data Requirements for a University



11. Grad students have 1 major department. Every department has at least 1 major.

ER: Exercise

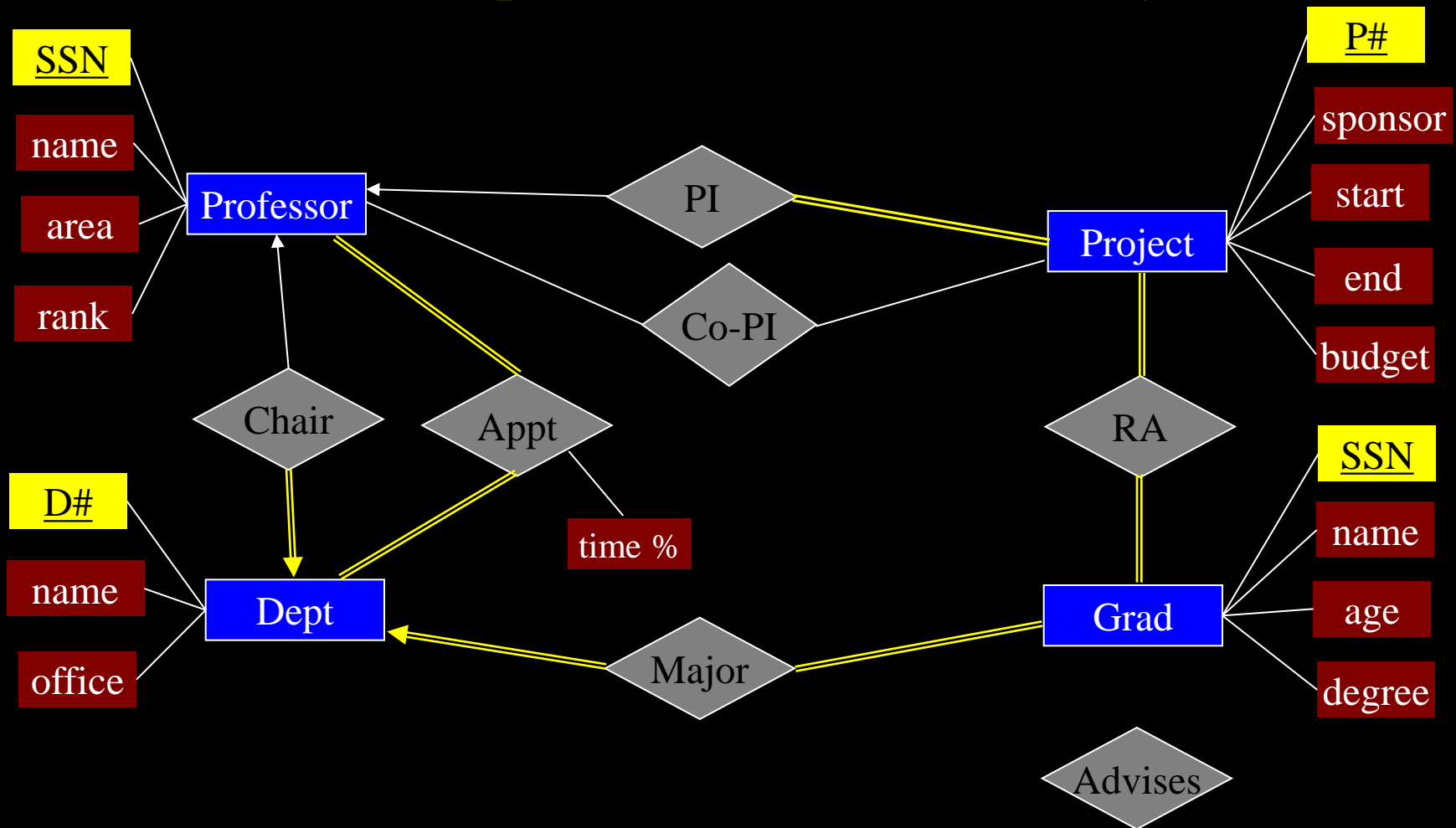
Model the Data Requirements for a University



12. Each grad student is advised by another (more senior) grad student.

ER: Exercise

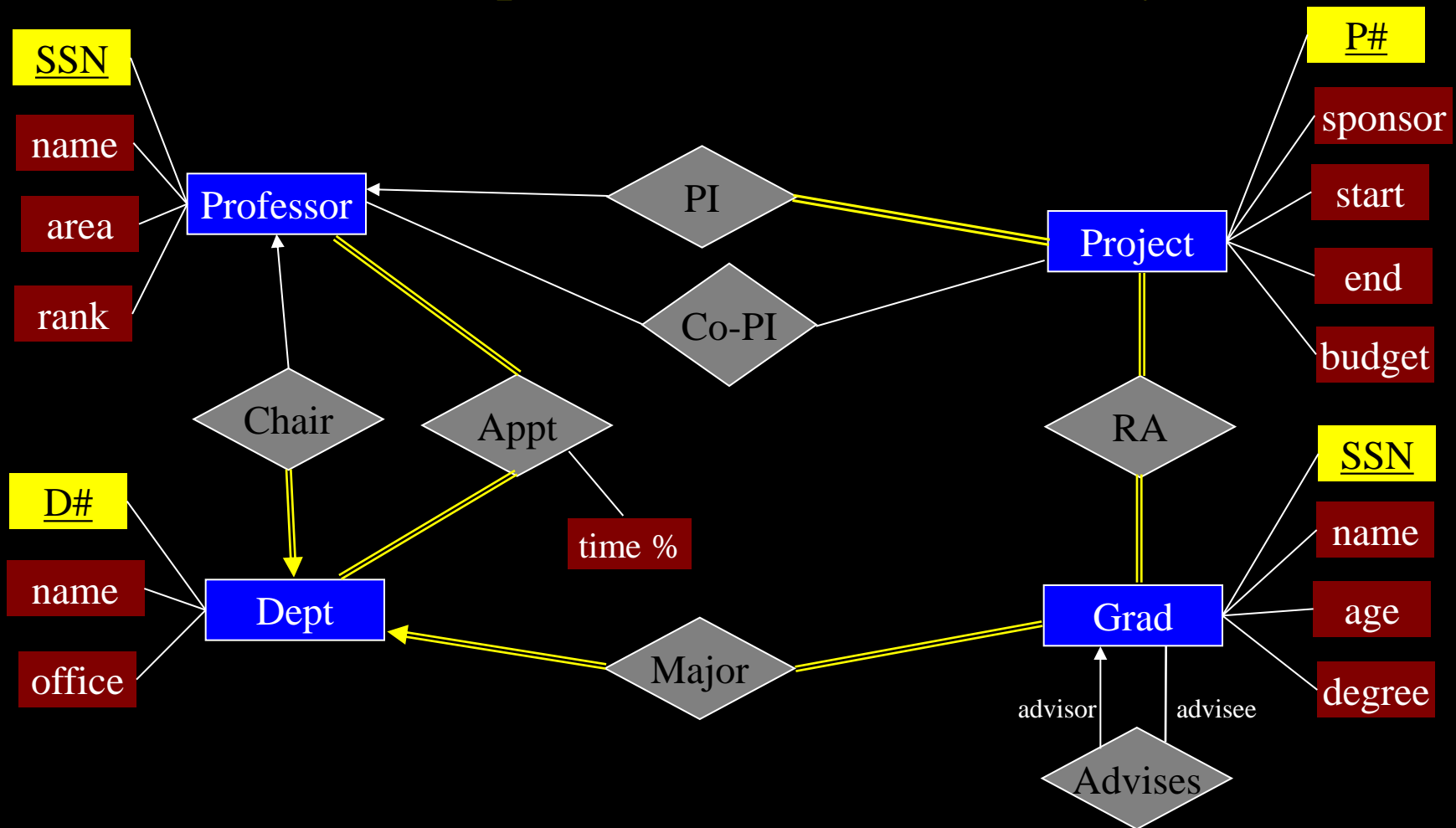
Model the Data Requirements for a University



12. Each grad student is advised by another (more senior) grad student.

ER: Exercise

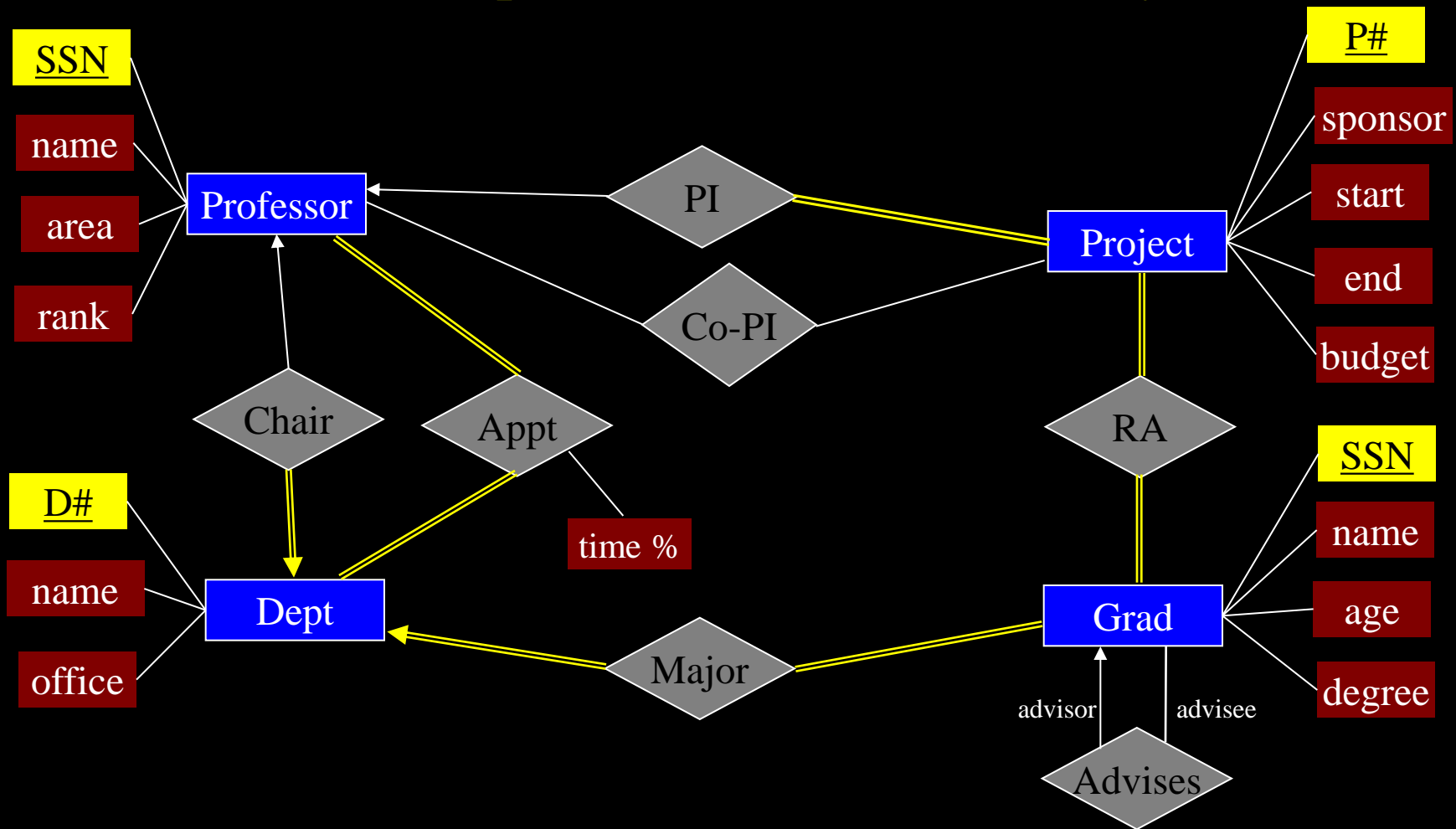
Model the Data Requirements for a University



12. Each grad student is advised by another (more senior) grad student.

ER: Exercise

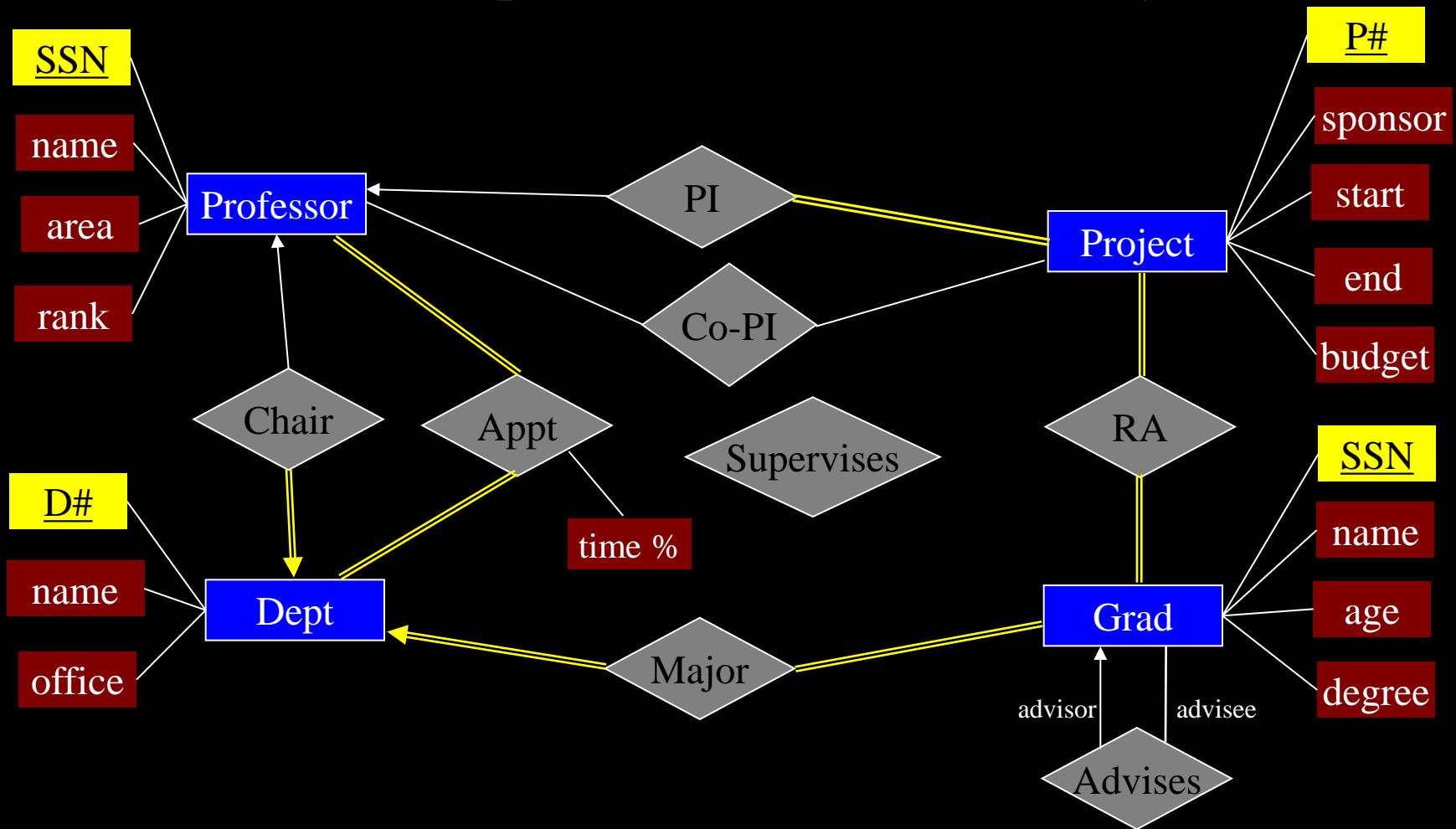
Model the Data Requirements for a University



7. Every RA appointment must be supervised by a professor.

ER: Exercise

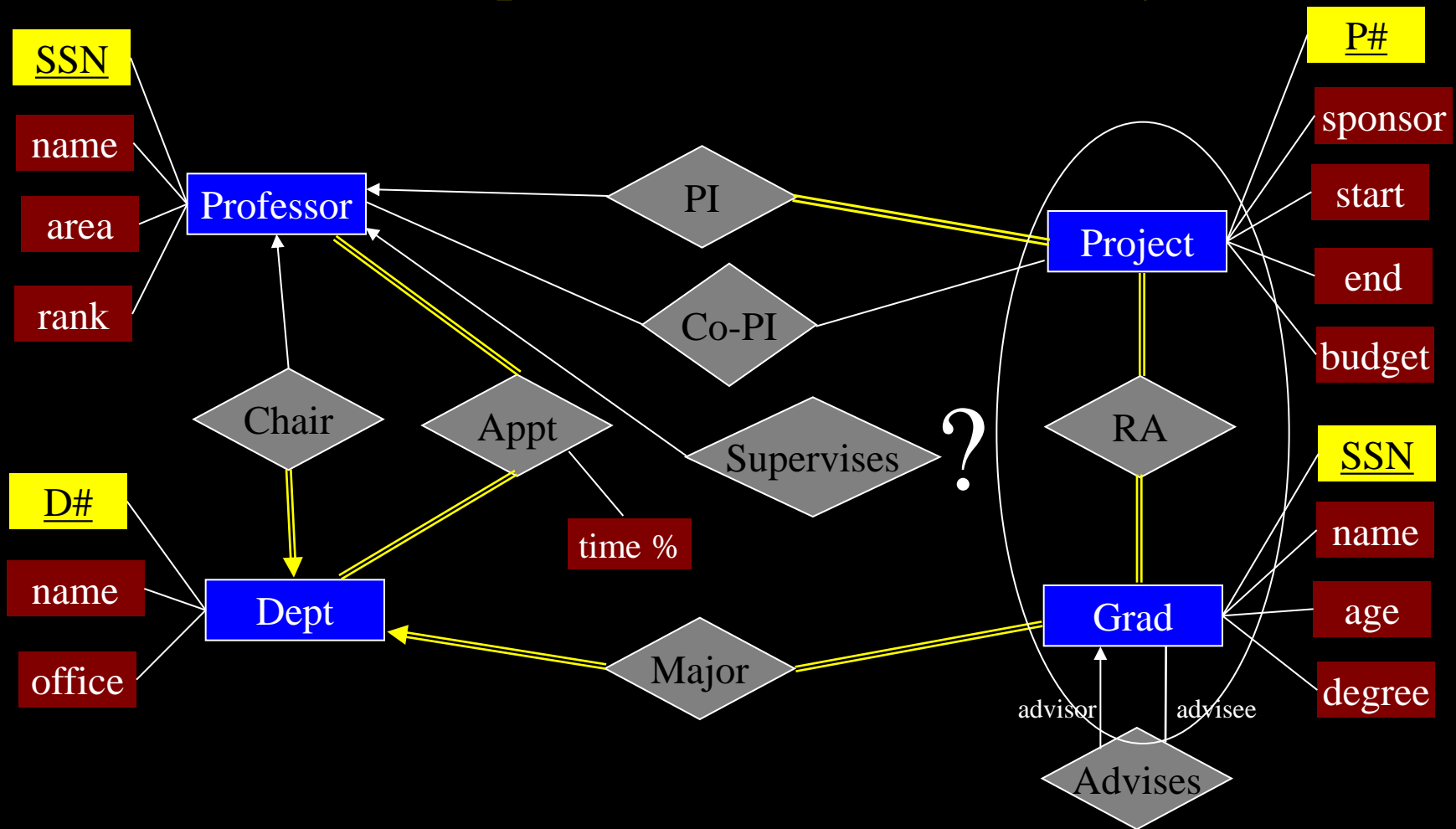
Model the Data Requirements for a University



7. Every RA appointment must be supervised by a professor.

ER: Exercise

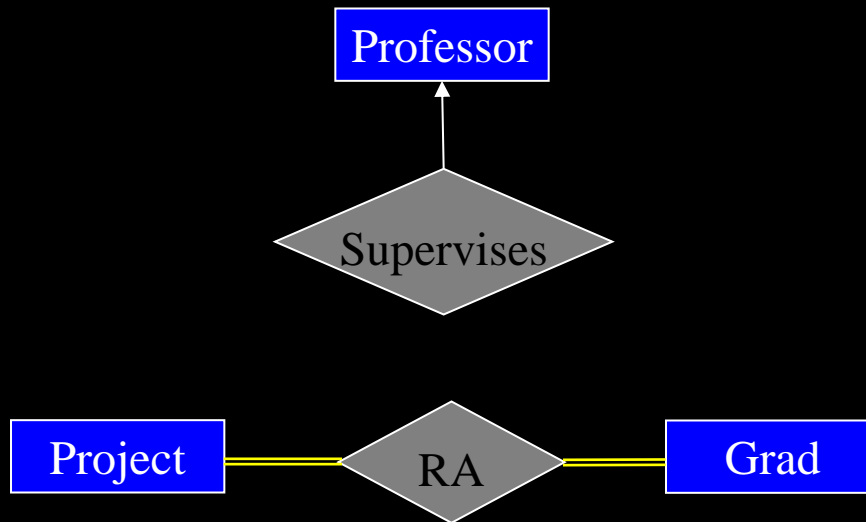
Model the Data Requirements for a University



7. Every RA appointment must be supervised by a professor.

ER: Exercise

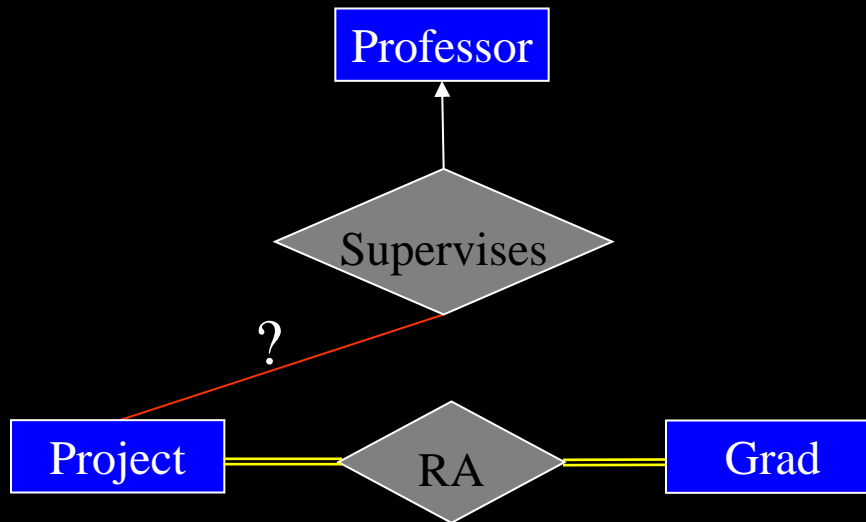
Model the Data Requirements for a University



e.g., want every RA appointment supervised by a professor

ER: Exercise

Model the Data Requirements for a University

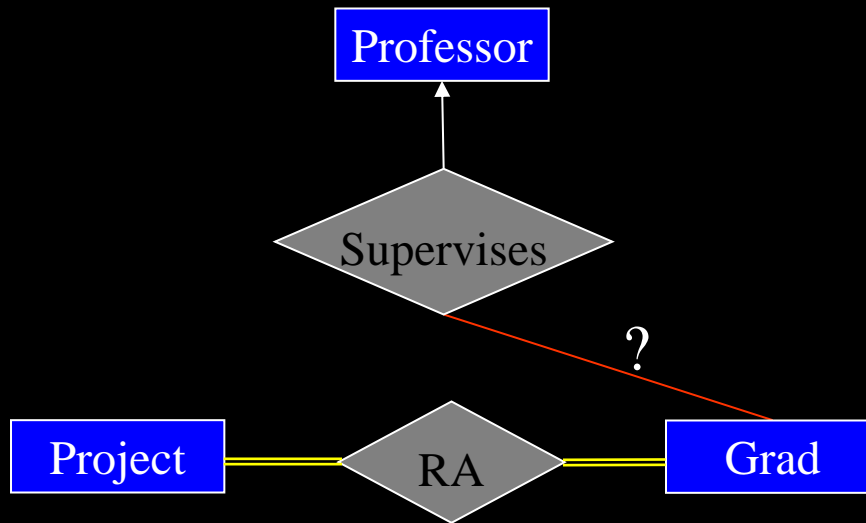


e.g., want every RA appointment supervised by a professor

Nope: Project may have different professors supervising different grad students

ER: Exercise

Model the Data Requirements for a University

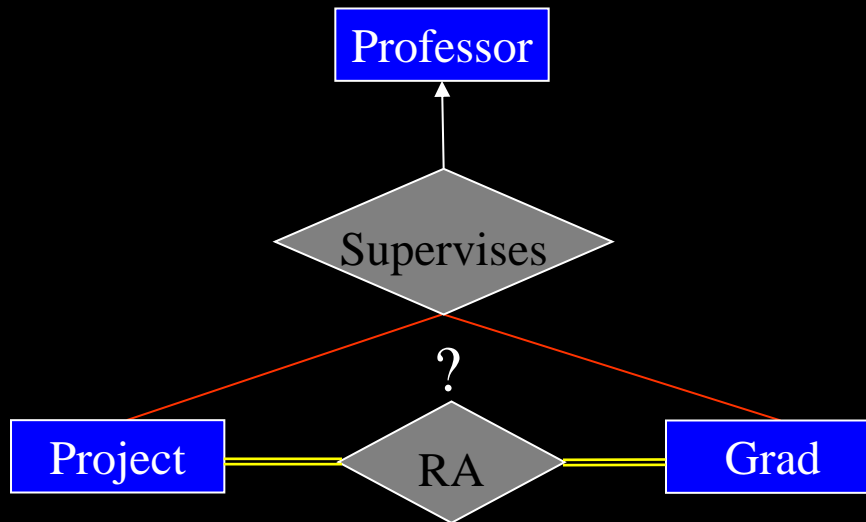


e.g., want every RA appointment supervised by a professor

Nope: Grad student may have different professors supervising for different projects

ER: Exercise

Model the Data Requirements for a University

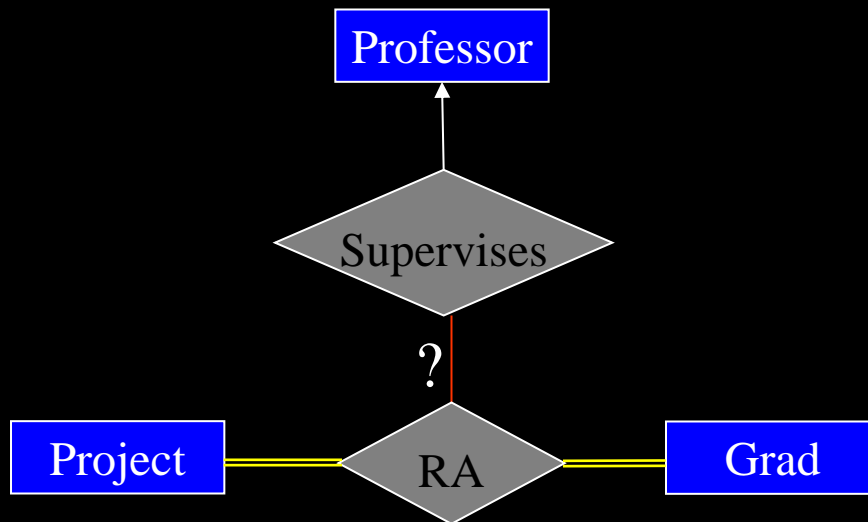


e.g., want every RA appointment supervised by a professor

Nope: Doesn't restrict supervision to project, grad student appointments

ER: Exercise

Model the Data Requirements for a University

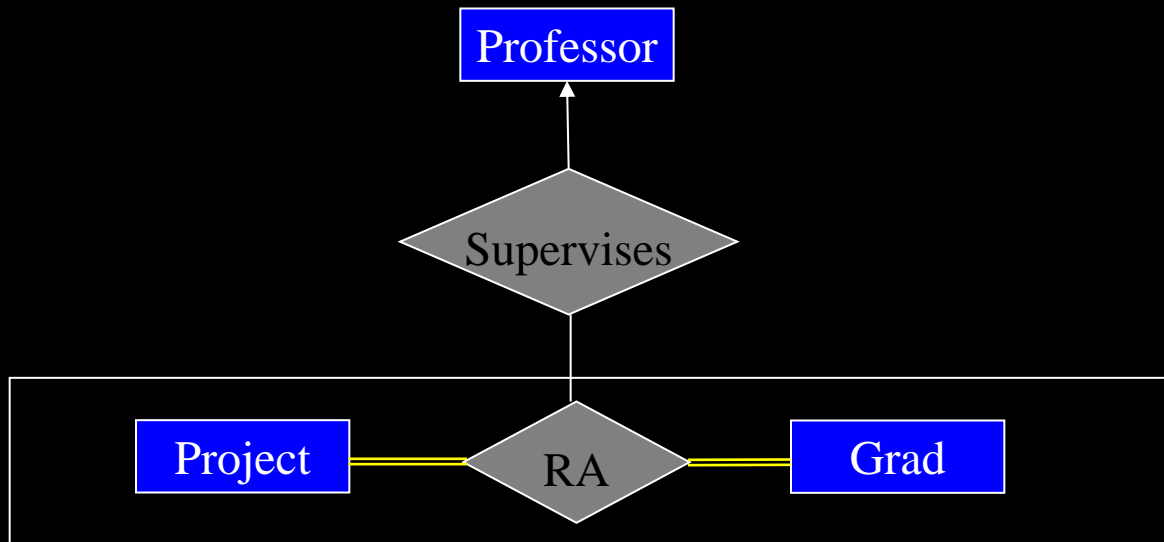


e.g., want every RA appointment supervised by a professor

Nope: What we want but E/R model only allows relationships between entities

ER: Exercise

Model the Data Requirements for a University

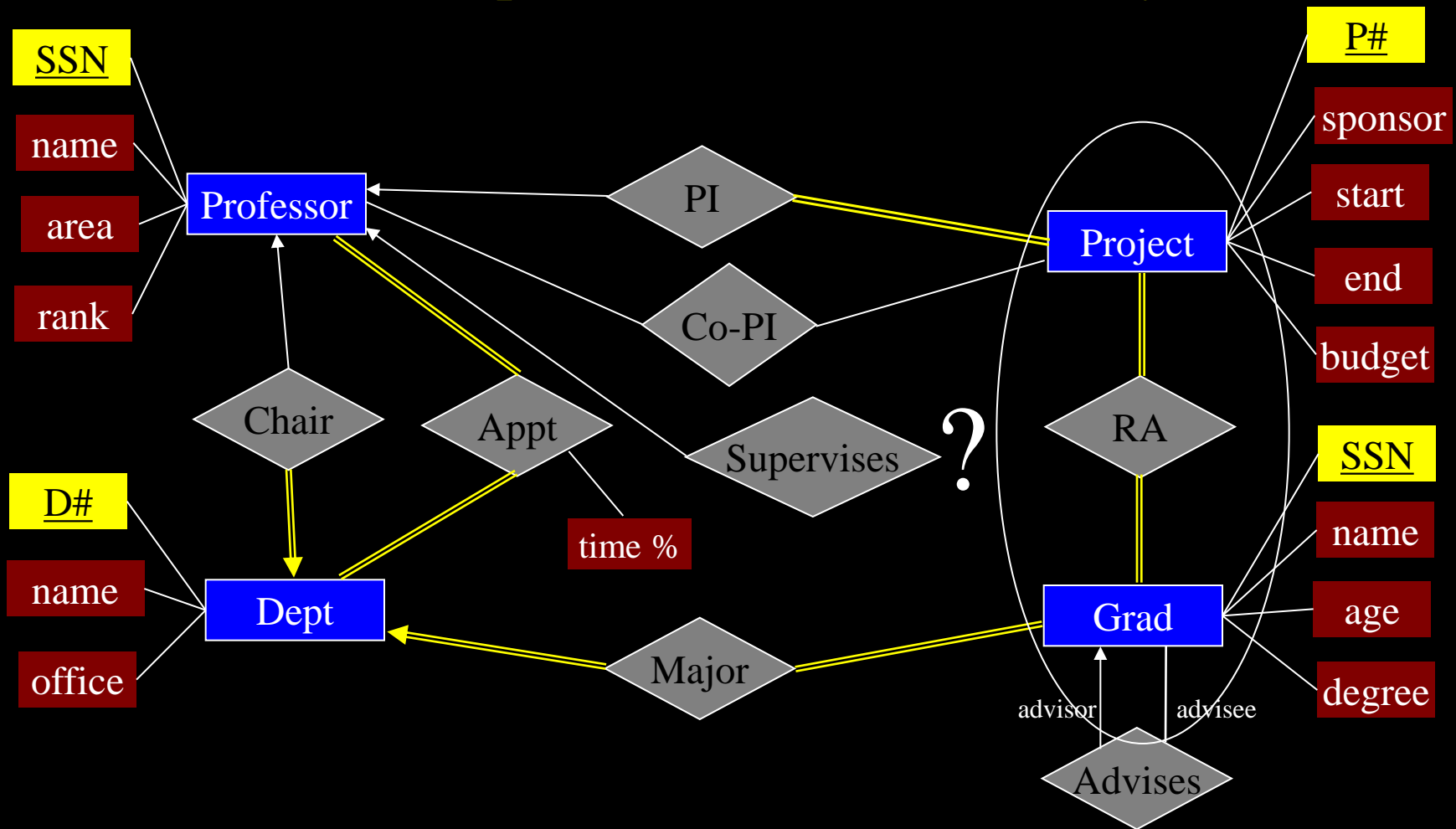


To associate Professor with RA...

must first **aggregate** (creates an entity set from a relationship set)

ER: Exercise

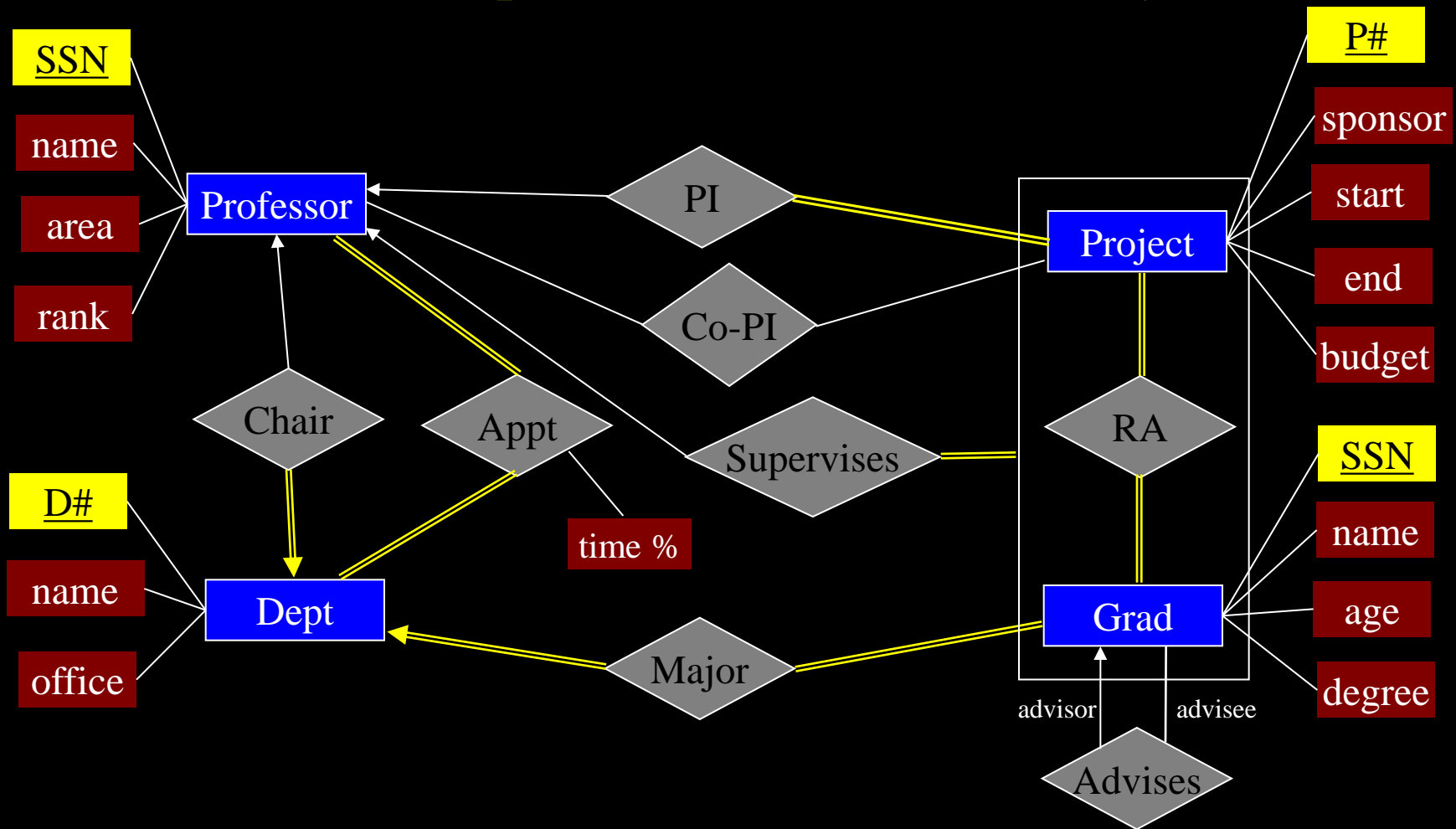
Model the Data Requirements for a University



7. Every RA appointment must be supervised by a professor.

ER: Exercise

Model the Data Requirements for a University



7. Every RA appointment must be supervised by a professor.

ER: Exercise

Model the Data Requirements for a University

