Conceptual Data Modeling

Conceptual Data Modeling

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Kevin C.C. Chang, Professor

Computer Science @ Illinois

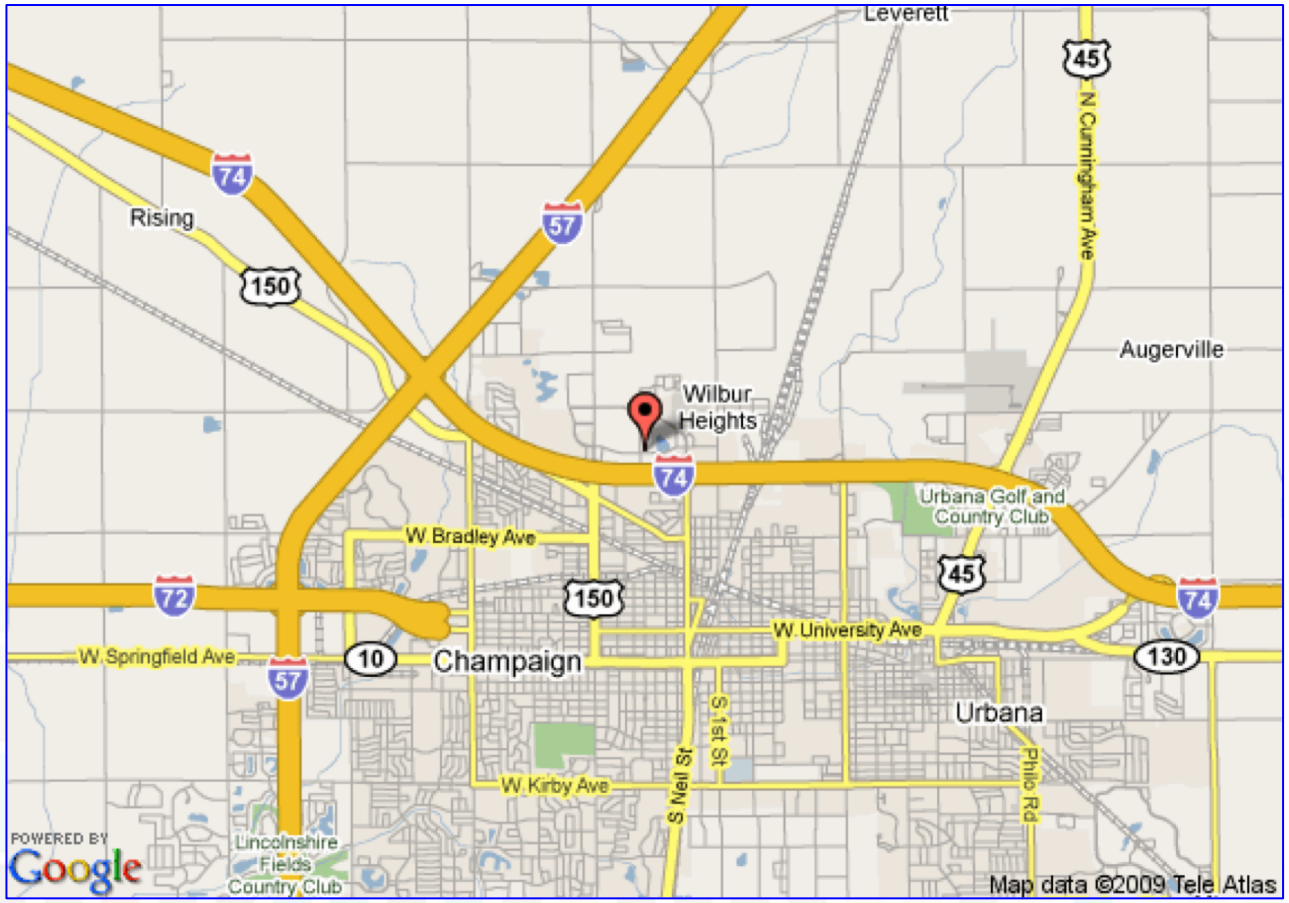
Learning Objectives

By the end of this video, you will be able to:

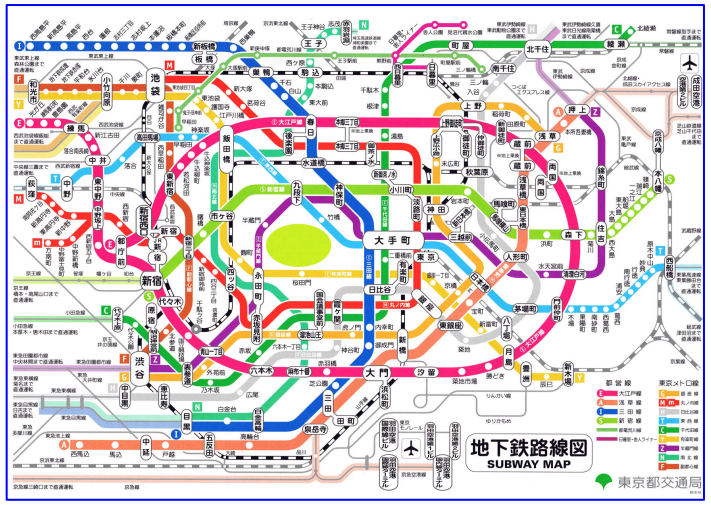
- Explain why building database applications starts with modeling data.
- Distinguish conceptual from physical data modeling.
- Name the general steps in creating a database.

Why Modeling Data?

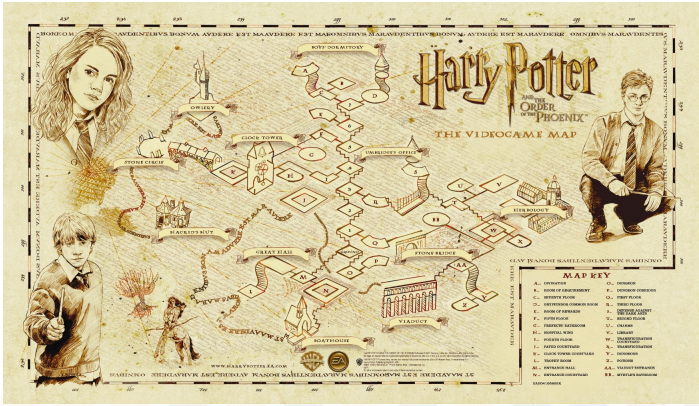
- How do you think about the real world?



Champaign-Urbana on Google Map (Google Maps, 2009)



Map of Tokyo subway (Printable Maps, 2012)



A Map of the interior of Hogwarts Castle (Nerdovore, 2013)

Why Modeling Data?

- **What part of the world** should we capture, for our applications?
- **How do we represent it**, to bring data to computation?
- There are many different ways.

From Idea to Database Application

Steps in Building a Database Application:

- Step 0: Plan application. Understand domain/users.
- Step 1: Conceptual data modeling.
- Step 2: Physical data modeling.
- Step 3: Create database. Develop application.

Why Conceptual Modeling First?

Since we start with knowing the application and the real world, conceptual modeling allows us:

- Focus on high-level description.
  - Think in terms of the main concepts– things and how they are related.
  - Close to how we think of the real world (and not the computer).
- Use natural expression.
  - Sketch diagrams (and not writing in computer languages).
- Bridge to physical modeling.
  - Easy translation/mapping to various physical data models.

References

- Google Maps, 2009. *Map of Champaign-Urbana.*
- Printable Maps, 2012. *Map of Tokyo Subway* [Online image]. Retrieved from http://printable-maps.blogspot.com/2012_02_01_archive.html.
- Nerdovore, 2013. *Fantasy world maps: Harry Potter* [Online image]. Retrieved from http://www.nerdovore.com/2013/01/fantasy-world-maps-harry-potter.html.

Our Example Applications

Conceptual Data Modeling

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Kevin C.C. Chang, Professor

Computer Science @ Illinois

Learning Objectives

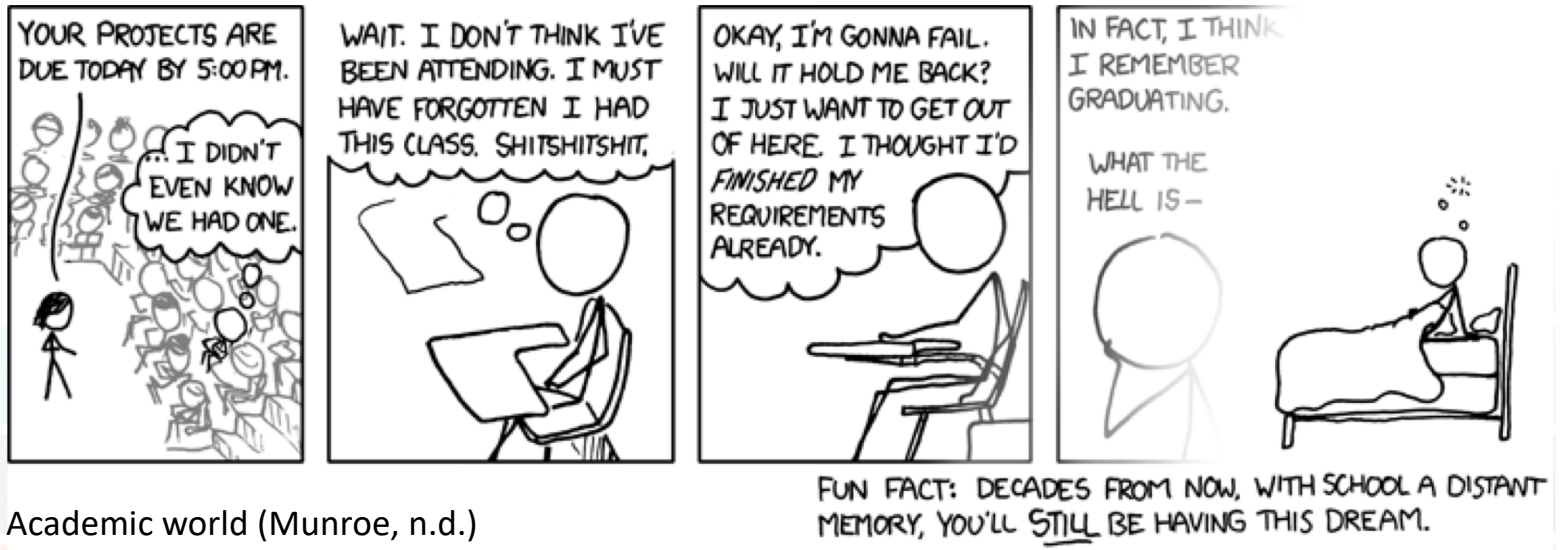
By the end of this video, you will be able to:

- Name the example application scenarios we will use throughout.
- Give more examples of database applications.

Example Application 1: *Academic World*

Academic World

- What you do in an academic world.
- Who they are: students, courses, and professors.
- What they do: students taking courses, professors teaching courses and advising students.



Academic world (Munroe, n.d.)

Example Application 2: *Friday Night*

Friday Night

- What *some* probably do on a Friday night.
- Who they are: drinkers, beers, and bars.
- What they do: drinkers frequenting bars and drinking favorite beers, bars selling beers.

Beer mug

Food for Thought

We use a database to store data for an application– a ***database application****.*

Most "interesting" applications have a database– Do you agree?

References

Beer mug image by Pixabay.com, used under CC BY [Online image]. Retrieved from https://pixabay.com/en/beer-mug-full-frothing-drink-26722/

Munroe, R. (n.d.). *xkcd 557* [Online image]. Retrieved from https://xkcd.com/557/

ER Model: Basics

Conceptual Data Modeling

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Kevin C.C. Chang, Professor

Computer Science @ Illinois

Learning Objectives

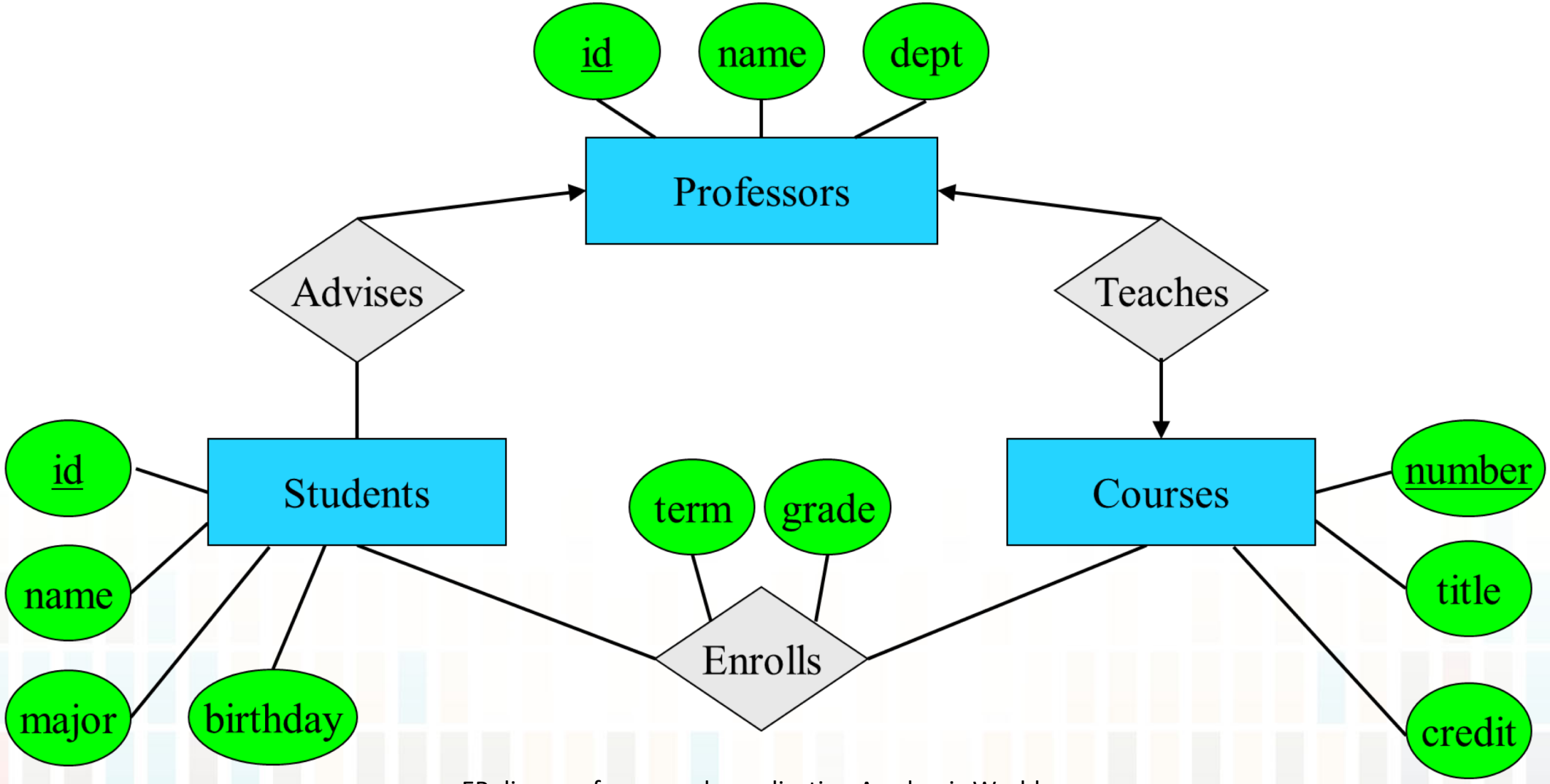
By the end of this video, you will be able to:

- Name the basic elements of the ER model.
- For an application domain, create ER diagrams using entity sets, relationships, and attributes.

ER Model: Entity-Relationship Model

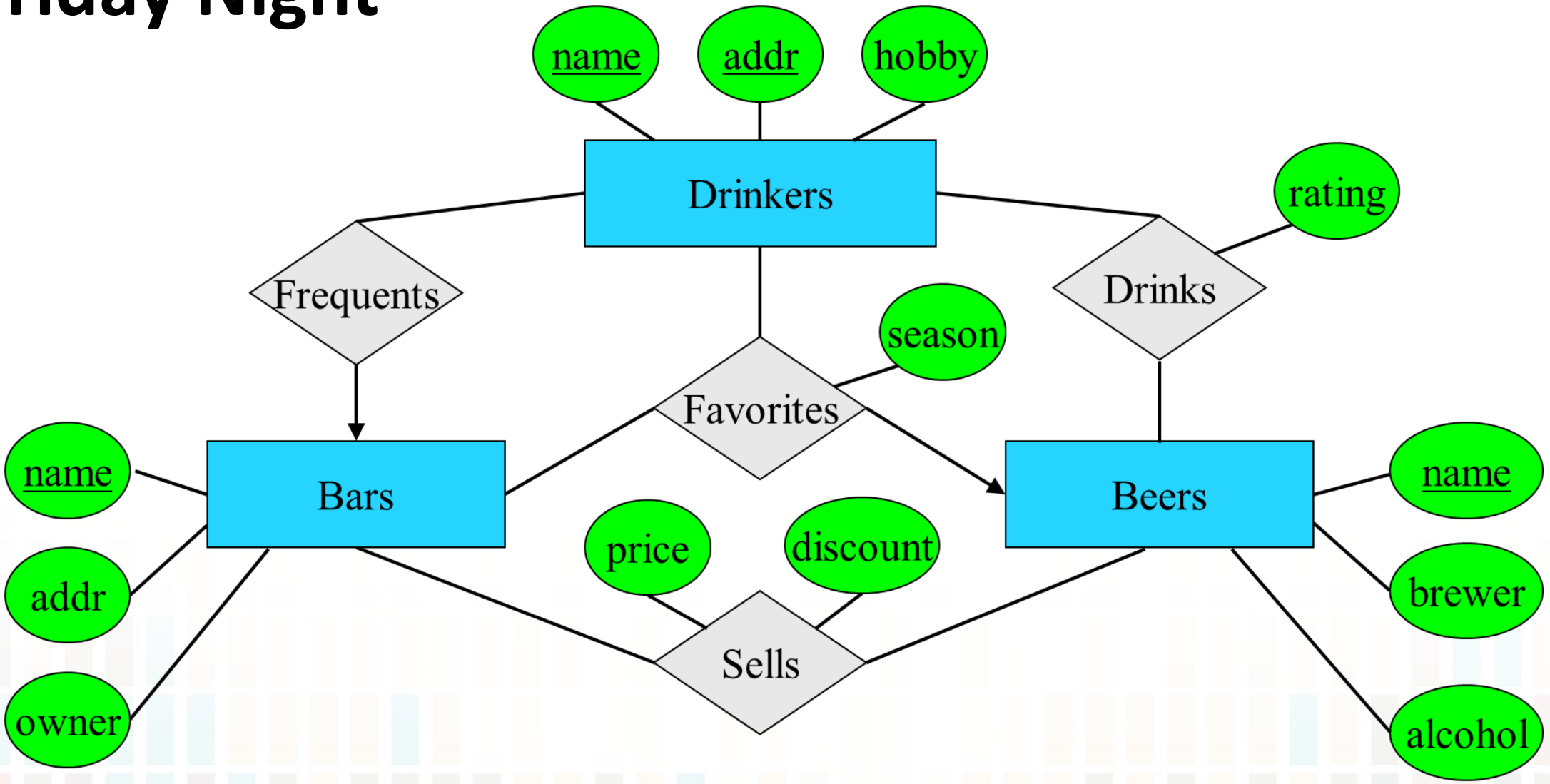
- Proposed by Peter Chen in 1976.
- A diagram language to specify:
  - What information the database must hold.
  - The components of that information.
  - The relationships among the components of that information.
- Output: **ER diagram**.

Academic World



ER diagram for example application Academic World

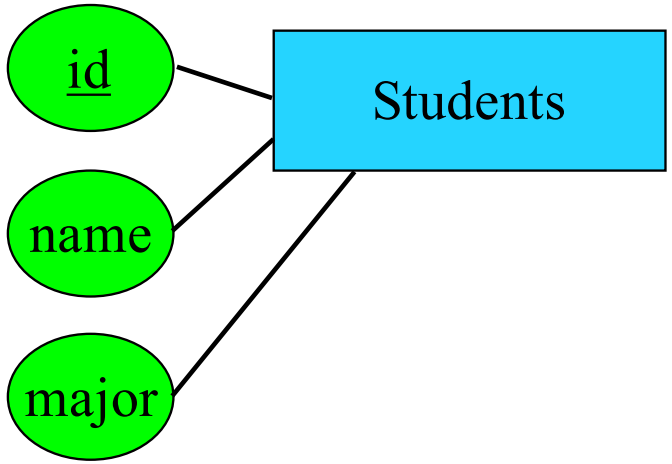
Friday Night



ER diagram for example application Friday Night

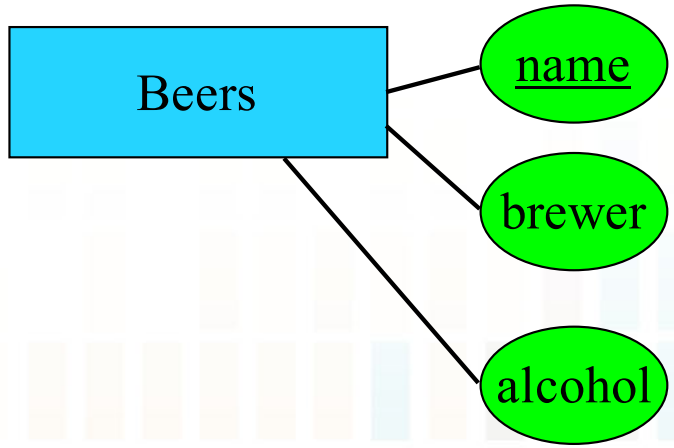
Entity Sets and Attributes

- Entity sets
  - Real-world objects distinguishable from others.
  - Characterized by using a set of attributes.



Entity set Students and attributes

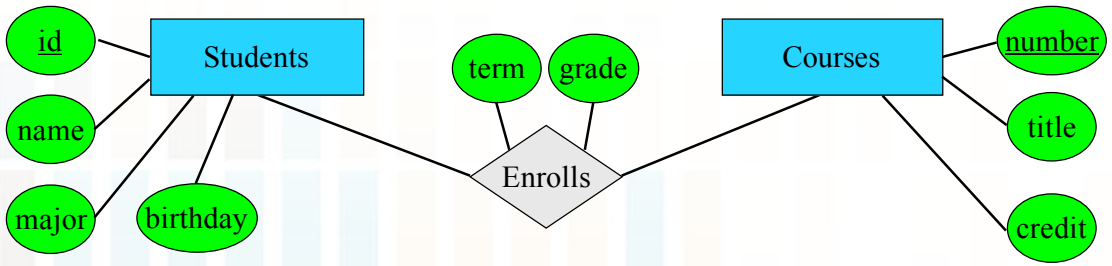
- Attributes
  - Each attribute is a property of an entity.
  - Each attribute has a simple value: string, integers, reals, etc.



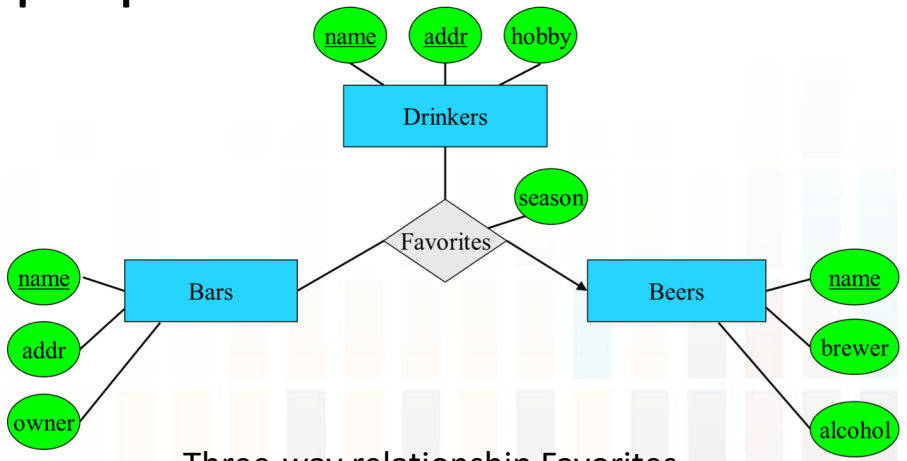
Entity set (beers) and attributes

Relationships

- Relates two or more entity sets for their interaction.
- **Cardinality** (or **degree**): Number of entity sets involved.
  - Binary: Two entity sets are related.
  - Multiway: More than two.
- A relationship can have attributes as its properties.



Binary relationship Enrolls



Three-way relationship Favorites

ER Model: More on Relationships

Conceptual Data Modeling

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Kevin C.C. Chang, Professor

Computer Science @ Illinois

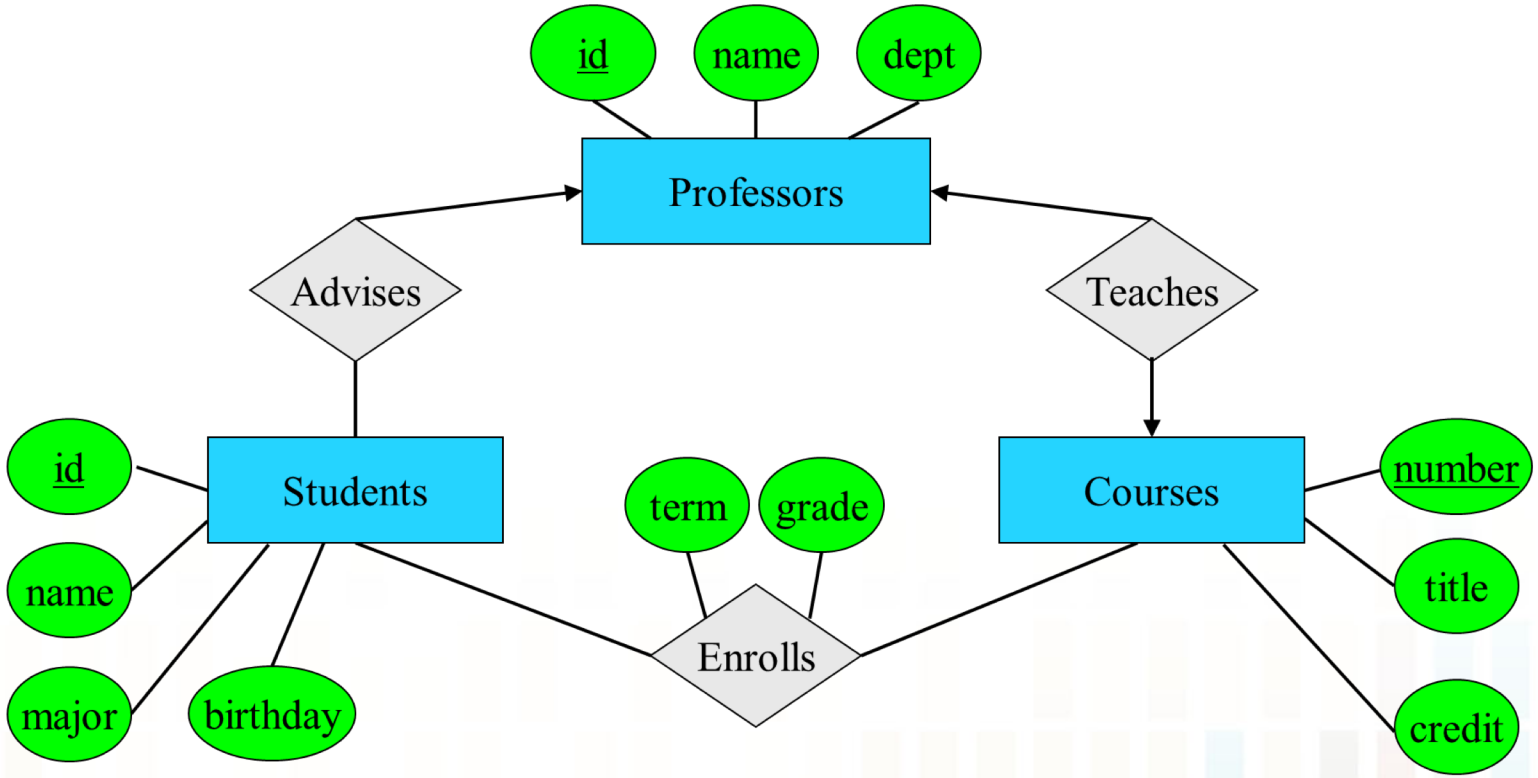
Learning Objectives

By the end of this video, you will be able to:

- Specify the multiplicity of a relationship.
- Identify the roles of the entity sets in a relationship.

Multiplicity of Binary Relationships

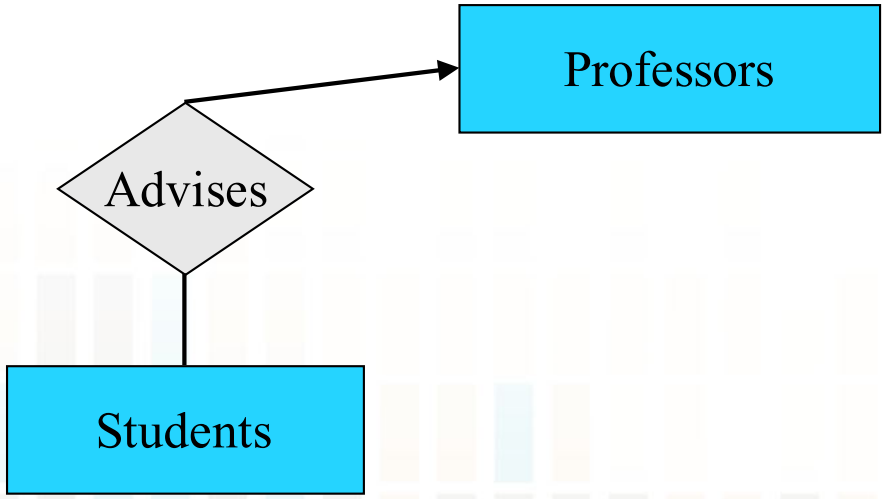
- Many-one
- One-one
- Many-many



ER diagram for example application Academic World

Many-One Relationships

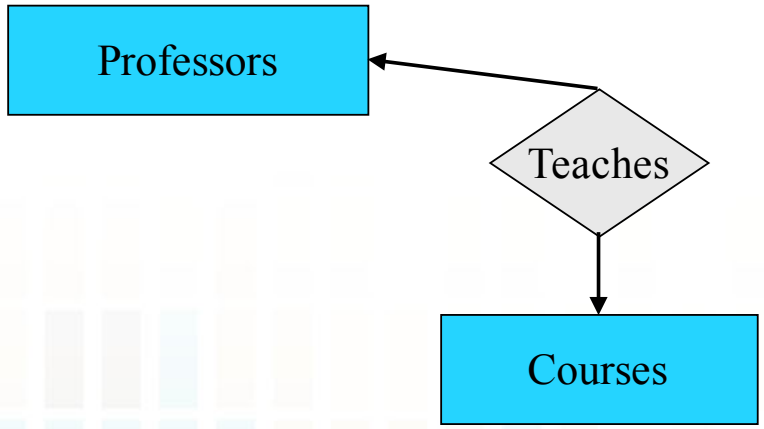
- The ONE side:
  - **Zero-or-one** entity at the “one” side can be related from an entity at the other side.
- The MANY side:
  - **Zero-or-more** entities at the “many” side can be related from an entity at the other side.



An example many-one relationship

One-One Relationships

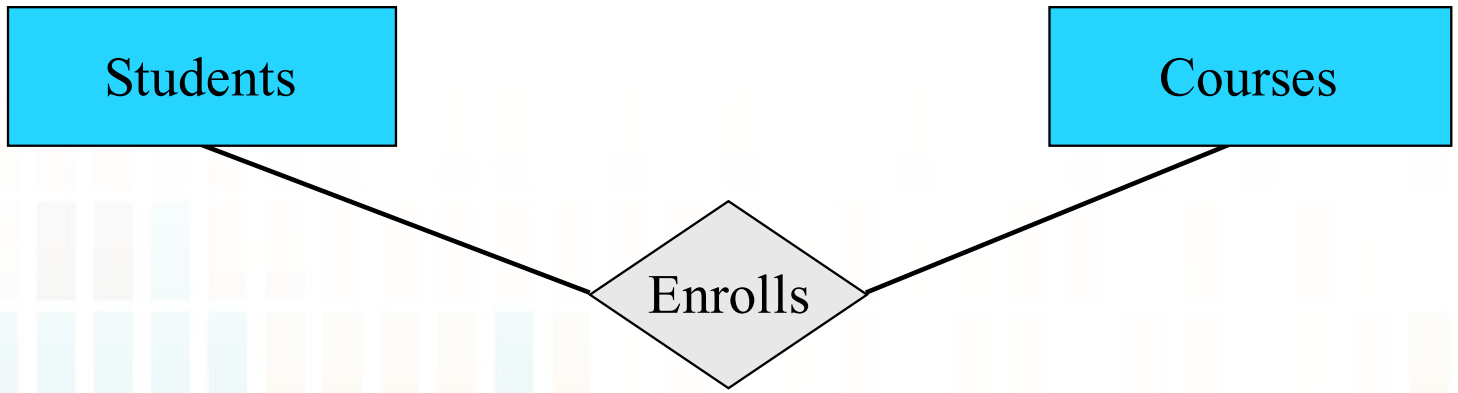
- Each entity at either side relates to **zero-or-one** entity of the other side.



An example one-one relationship

Many-Many Relationships

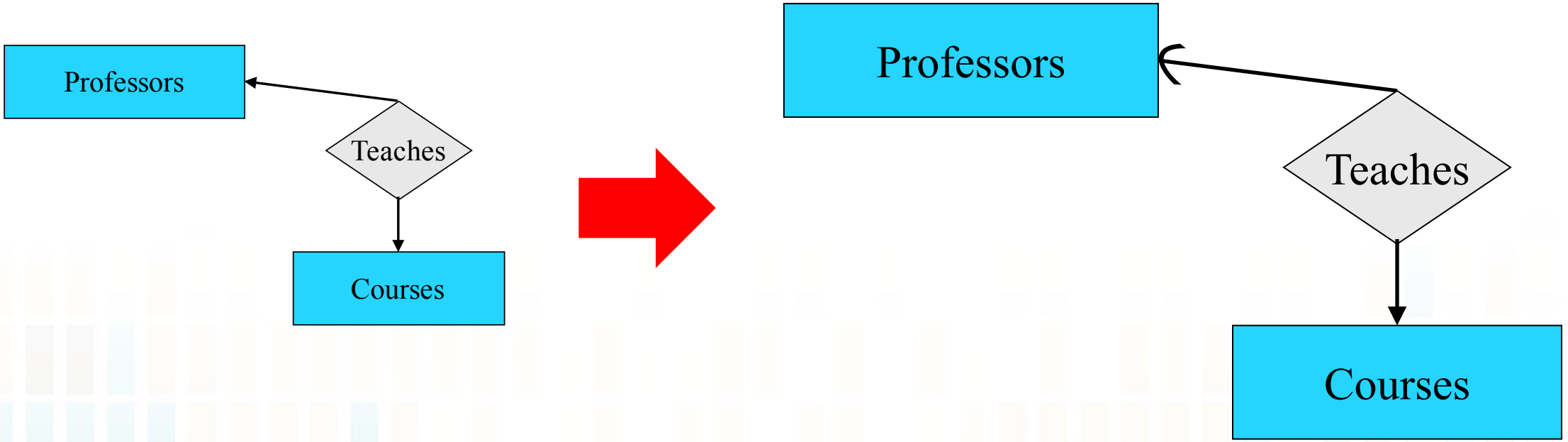
- Each entity at either side relates to **zero-or-more** entities at the other side.



An example many-many relationship

“Exactly-One” in a Relationship

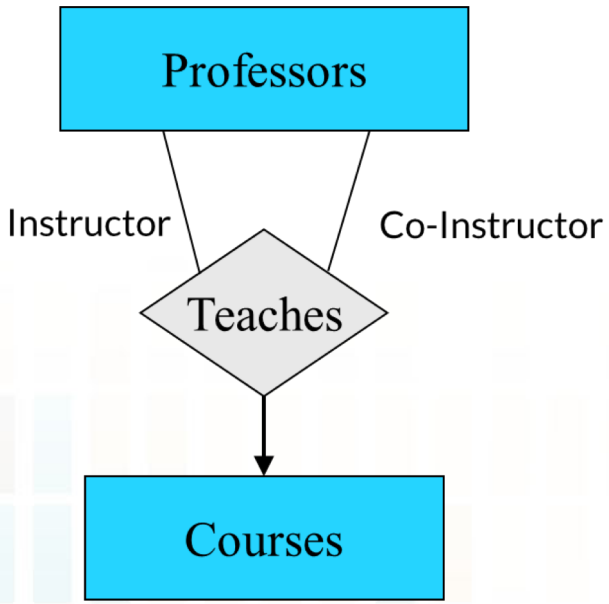
- Use a round arrow.

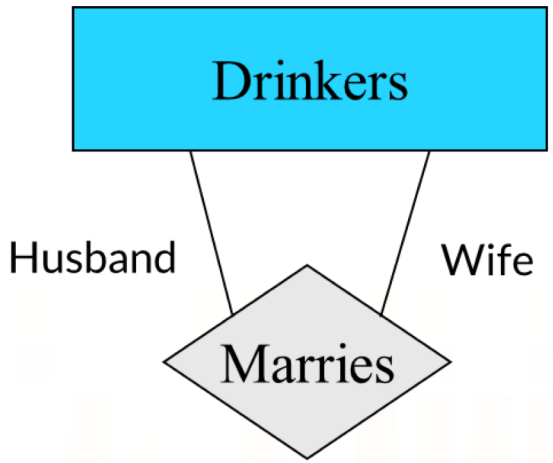


Using exactly-one in a relationship

Roles in Relationships

- When an entity set is involved multiple times in a relationship.
- Label each involvement with a **role**.

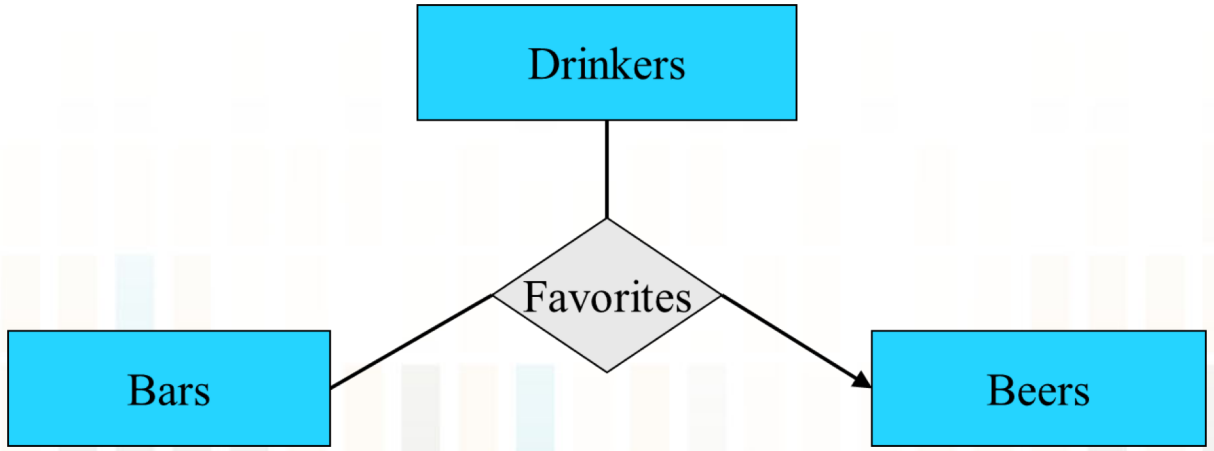




Labeling roles in relationships

Food for Thought

Does "multiplicity" apply to multiway relationships?
How do you define such multiplicity, e.g., many-many-one?



ER Model: Subclasses

Conceptual Data Modeling

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Kevin C.C. Chang, Professor

Computer Science @ Illinois

Learning Objectives

By the end of this video, you will be able to:

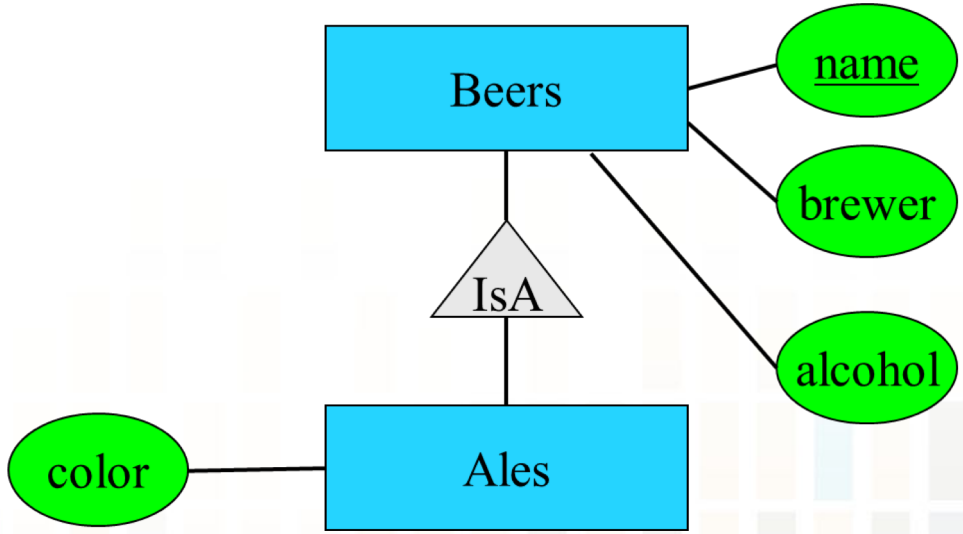
- Specify subclasses of an entity set by the IsA relationship.
- Explain how IsA is different from a general relationship.

Why Subclassing?

- Natural to think of real-world entities as specialization of others.
  - Graduate student is a specialization of student.
  - Ale is a specialization of beer.
- Specialization implies additional attributes.
  - Graduate student has research topic and research adviser.
  - Ale has (different) colors.
- Specialization implies fewer members.
  - Every grad student is a student, but not vice versa.
  - Every ale is a beer, but not vice versa.

Subclasses in ER: IsA Relationships

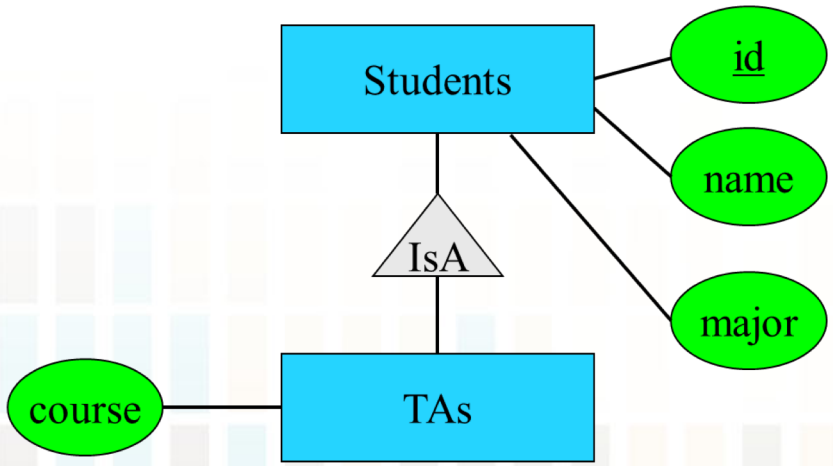
- An IsA triangle indicates a subclass relationship.
  - Directed– from subclass pointing to the superclass.
- Subclass inherits all attributes of superclass and some more.
- Assume subclasses form a tree.
  - I.e., no multiple inheritance.



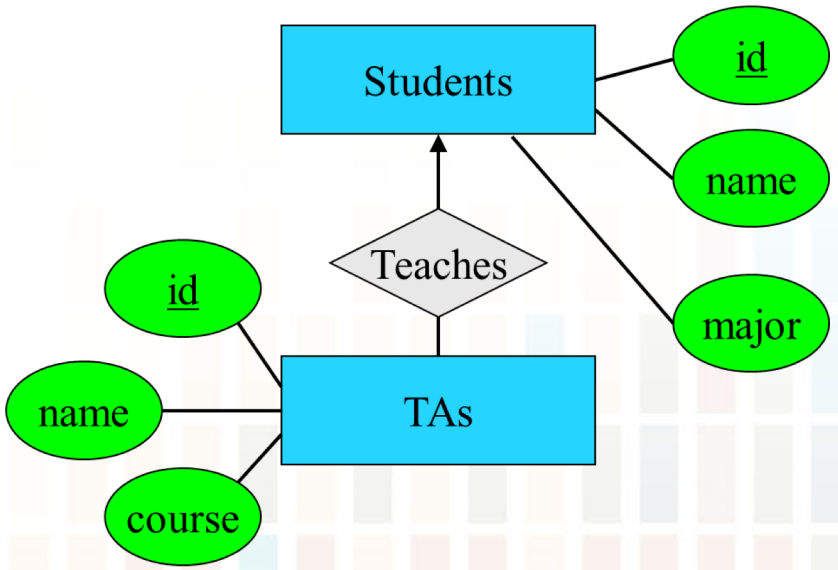
IsA relationship

Food for Thought

How does an IsA relationship different from a general relationship? Contrast the following two examples.



vs.



ER Model: Constraints

Conceptual Data Modeling

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Kevin C.C. Chang, Professor

Computer Science @ Illinois

Learning Objectives

By the end of this video, you will be able to:

- Define constraints used in the ER model.
- Identify different types of constraints.

Constraints

- An **assertion** about the database that must be true at all times.
- Part of the data modeling (and database “schema”).
- Important in database design.

Various Types of Constraints

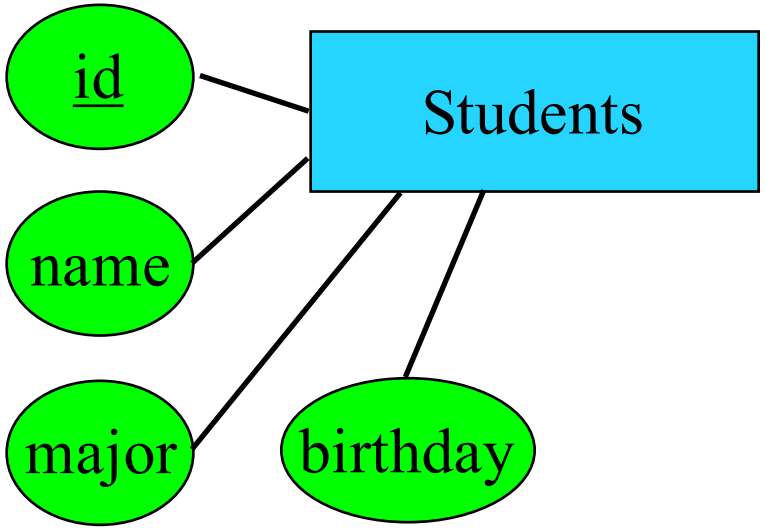
- Keys: Unique identifier
  - The “netid” at UIUC uniquely identifies a member of it.
- Referential integrity: A referenced entity must exist
  - If a drinker frequents a bar, the bar must exist.

- Many other constraints
  - Domain constraints: Grade $\in$ {A, B, C, D}. Price $\in$ [0, 100].
  - General constraints: BirthDate < EmploymentDate.

Why Constraints Are Important?

- Give more semantics to the data.
- So we can check correctness (consistency) of data.
- So we can enable efficient storage, data lookup, etc.

Keys

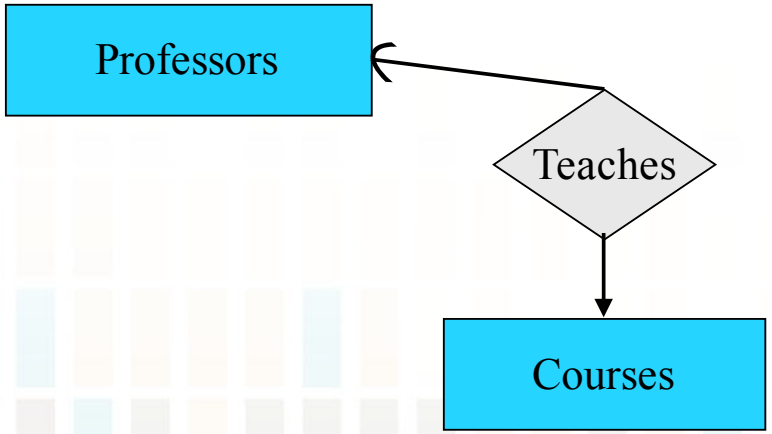


Labeling a key in an entity set

- A key can consist of one or more attributes.
  - {id} identifies a student.
  - {name, major} identifies a student.
- Every entity set must have a key.
  - At least the entire set of attributes would form a key.
  - We must capture sufficient attributes to distinguish entities.
- There can be more than one key.
  - We designate one key as the **primary key**, by underlining its attributes.
  - No formal way to specify multiple keys other than the primary.

Referential Integrity Constraint

- In a relationship, expressed by a round arrow:
  - A “referencing” entity must reference exactly-one “referenced” entity.
  - A referenced entity must exist.
- I.e., no “dangling pointer” (segmentation fault in programming).



Referential integrity constraint

Food for Thought

ER Model does not allow us to specify all kinds of constraints– but you can specify more than keys and referential integrity.

What other constructs we have learned also represent constraints?

Beyond ER Model: UML

Conceptual Data Modeling

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Kevin C.C. Chang, Professor

Computer Science @ Illinois

Learning Objectives

By the end of this video, you will be able to:

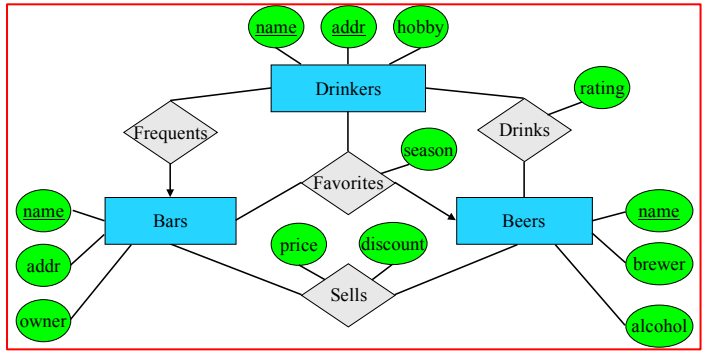
- Identify the key notions in the ER model.
- Describe how these notions are specified in UML.
- Create conceptual models in UML.

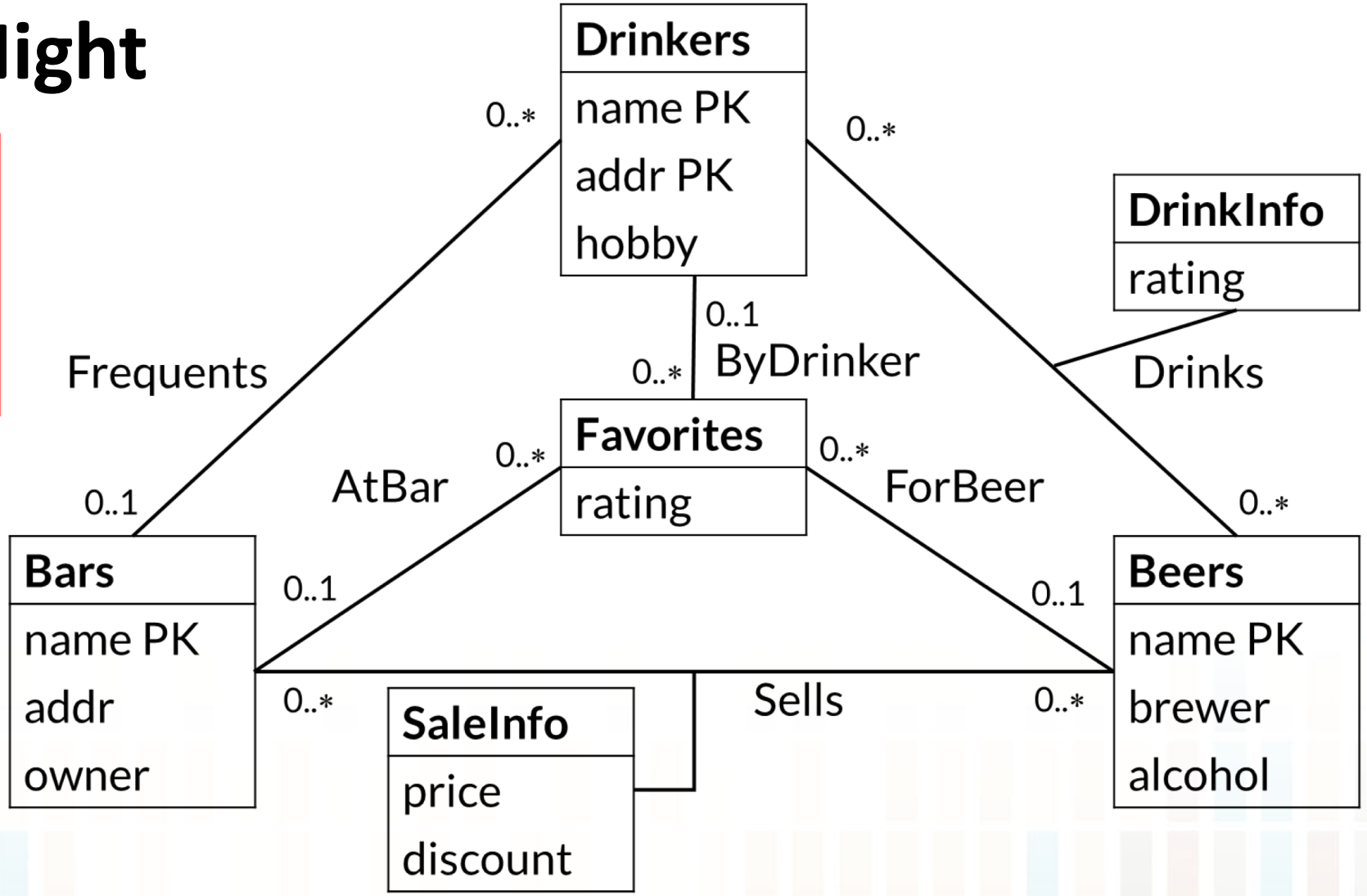
Other Models Exist, but Concepts Similar

- ER model is simple and natural.
- ER model corresponds well to relational model.
  - Relational model is a physical data model, used by relational databases.
  - ER model thus
    - becomes popular conceptual design before concretizing to relational model.
    - provides semantic interpretation for physical “relations”.
- ER model is graphical-- intuitive to create and view.
- Concepts of other models are similar.

- We will contrast with UML (Unified Modeling Language).

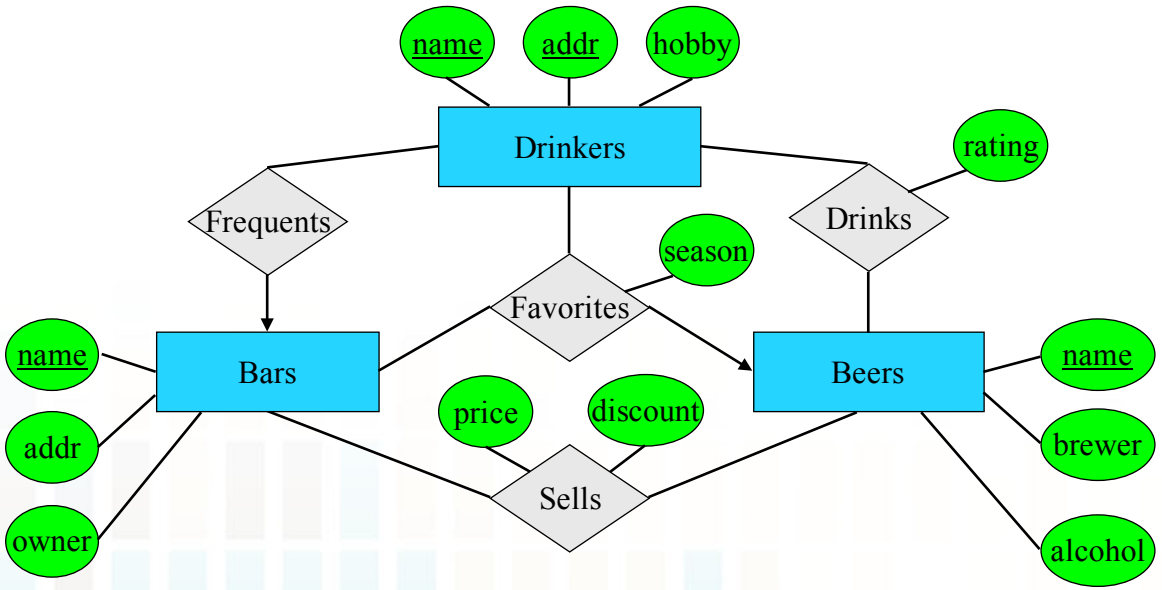
UML: Friday Night



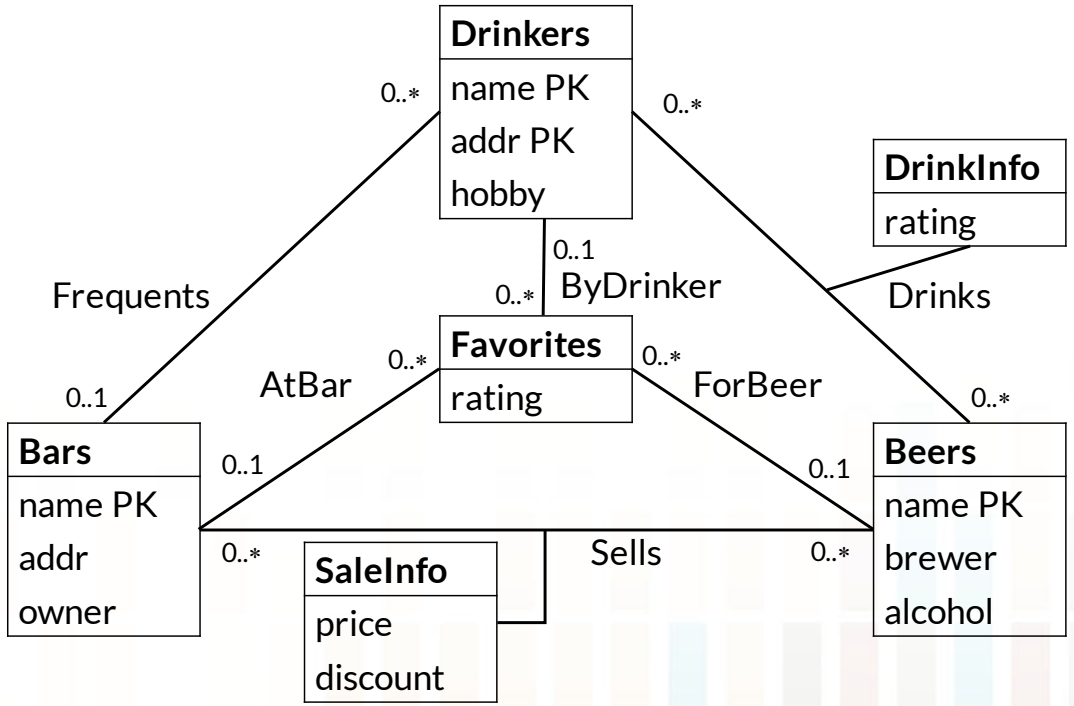


From ER to UML diagram for example application Friday Night

ER Model vs. UML: *Entity Set → Class*

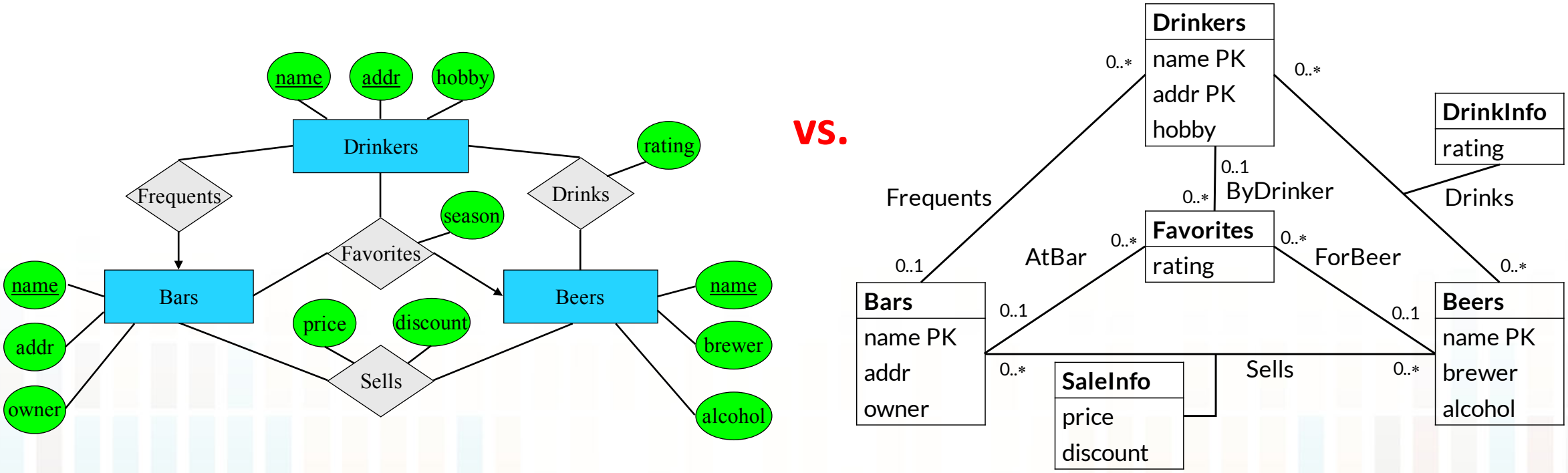


vs.



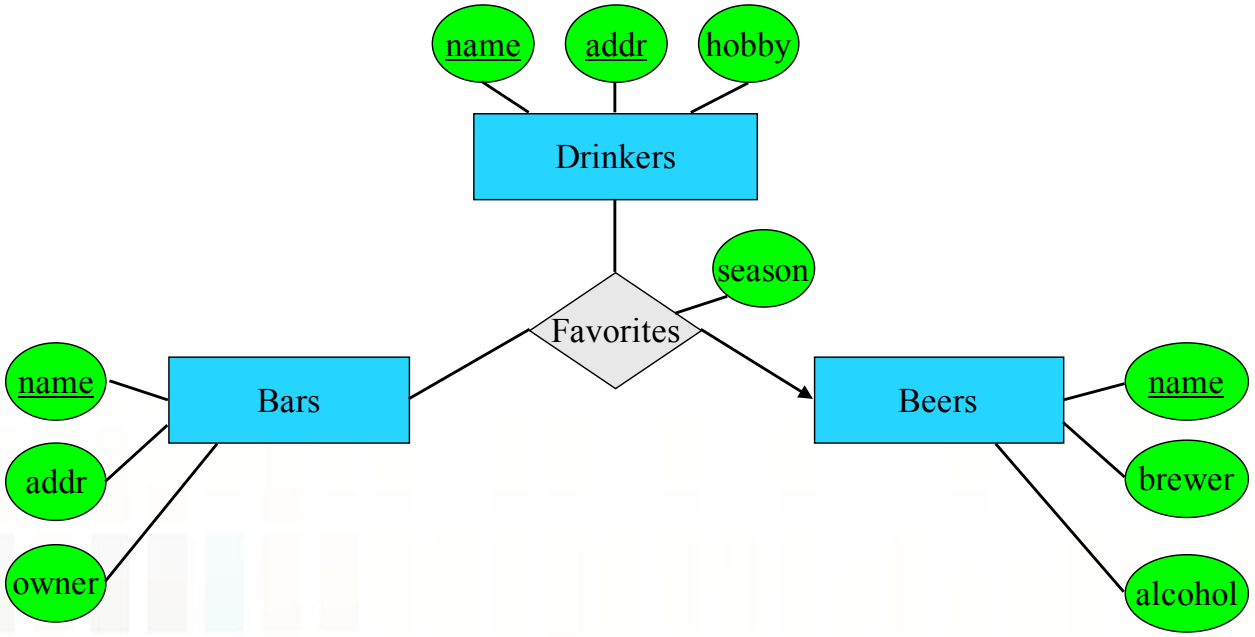
ER model vs. UML for example application Friday Night

ER Model vs. UML:
Relationship → Association and Association Classes



ER model vs. UML for example application Friday Night

UML Allows Binary Associations Only

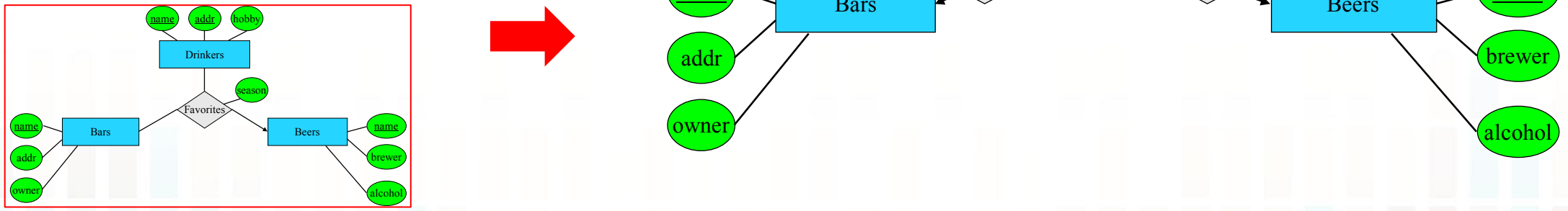


vs. ?

Example three-way relationship Favorites

K-way Can Be Expressed as *K* Binaries

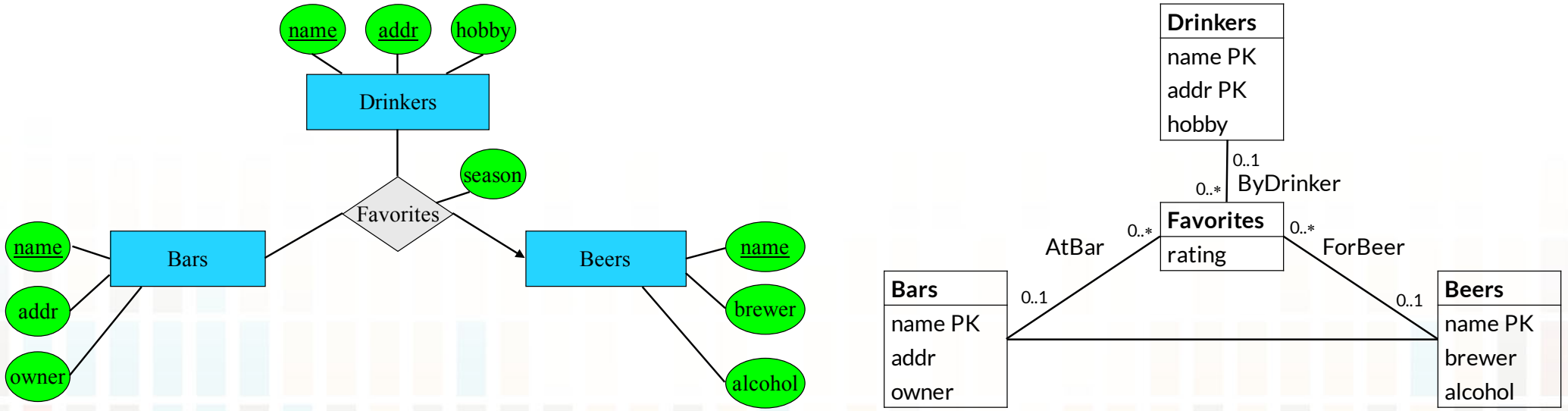
- A multiway relationship is just a combination of (zero or) one entity from each entity set.
- We can represent that combination as an entity.
- That new entity will connect to each entity set with arrows.



Converting multiway relationship to binary ones

ER Relationship vs. UML Association

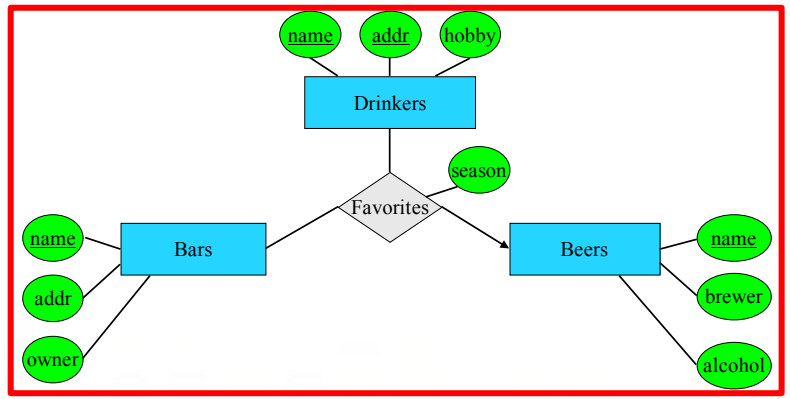
	ER Model	UML
Cardinality	binary, multiway	binary
Multiplicity	arrow, round arrow, <= n	m..n

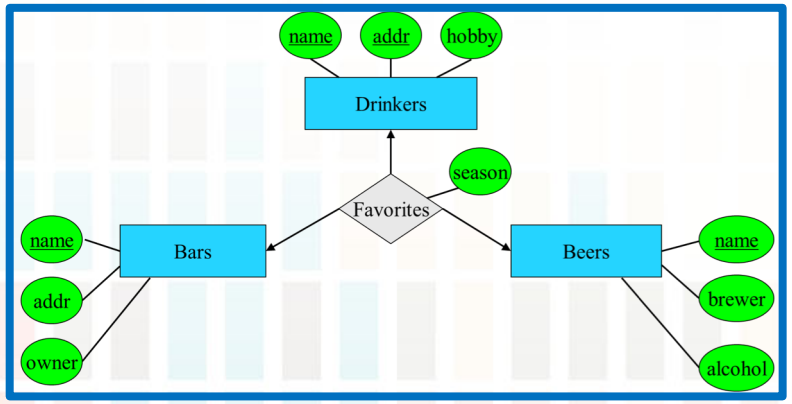


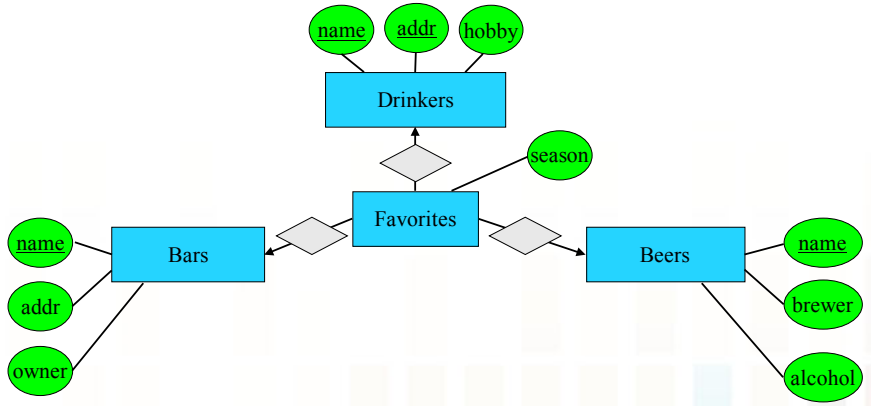
ER relationship vs. UML relationship

Food for Thought

It may not be quite right to say that the conversion from k-way to k-binaries as we did is fully equivalent. Compare the following two conversions. What happens?







The End

Step 2: Physical Data Modeling

- Choose a database management system (DBMS) to use.
  - We will use relational databases such as MySQL or PostgreSQL.
- Translate your conceptual model to the data model of the DBMS.
  - Mechanical process to translate from ER to relational schema.
- Refine it to a good relational schema, using a set of rules.

An example database system

Professors(<u>id</u>, name, dept, course)
Students(<u>id</u>, name, major, birthday, advisor)
Courses(<u>number</u>, title, credit)
Enrolls(<u>id</u>, <u>number</u>, term, grade

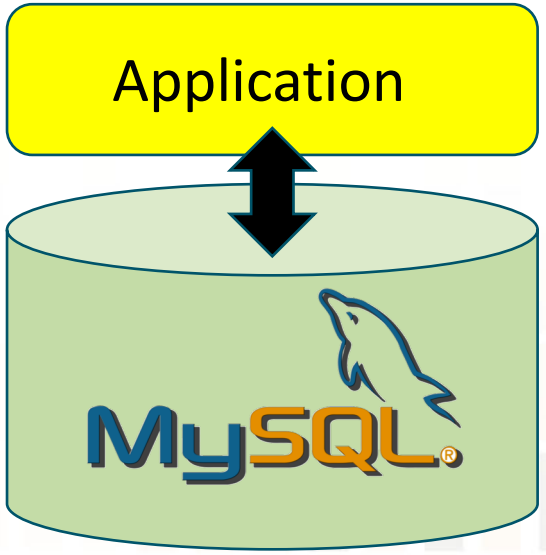
An example relational schema

Step 3: Create Database. Develop Application.

- Open DBMS. Create database tables according to the schema.
- Develop application code that runs on top of DBMS.

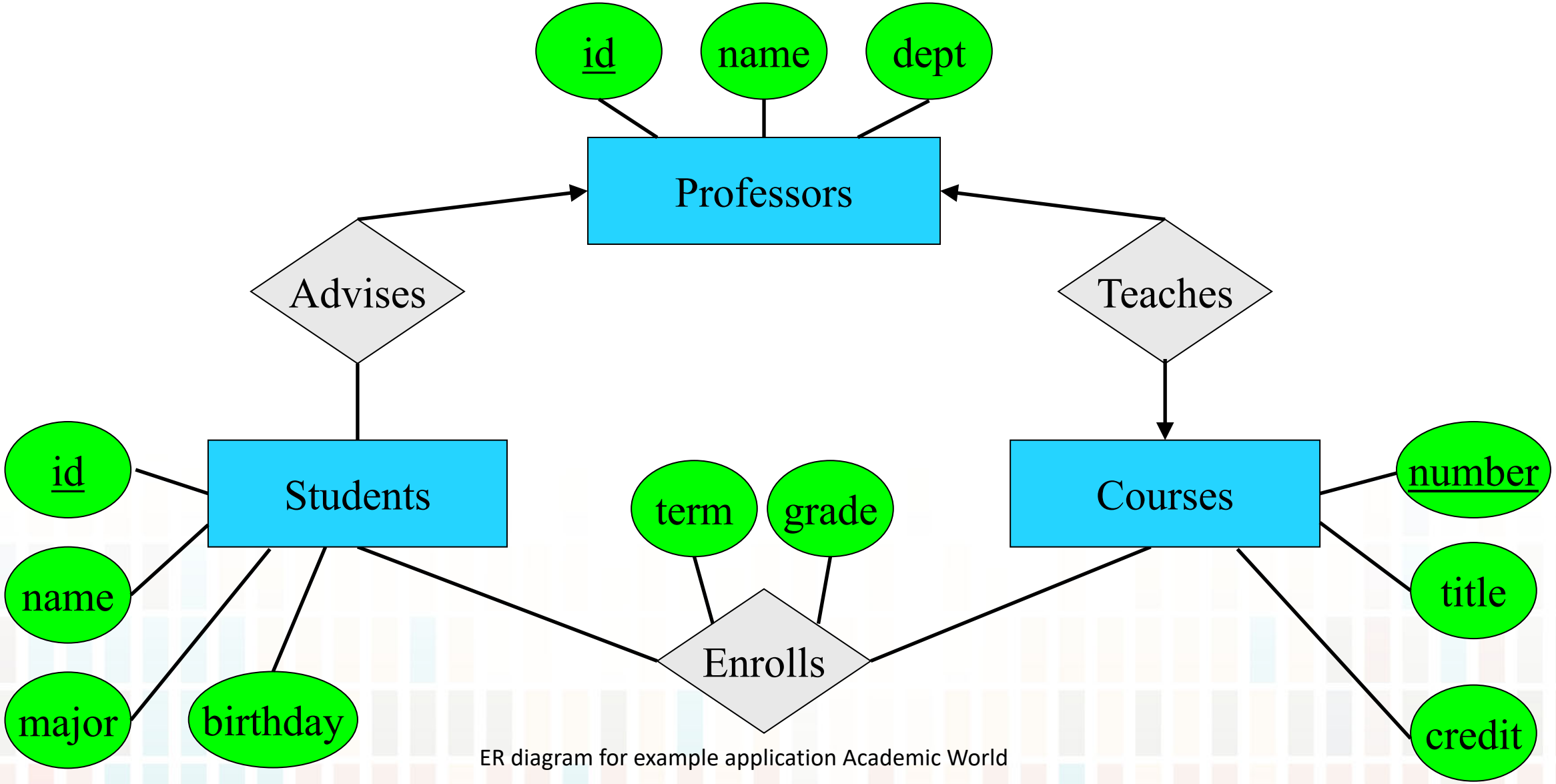
```
CREATE TABLE Courses (
number text,
title text,
credit integer);

CREATE TABLE Students (      name
text,
major text,
…
…
```

SQL commands for creating database tables



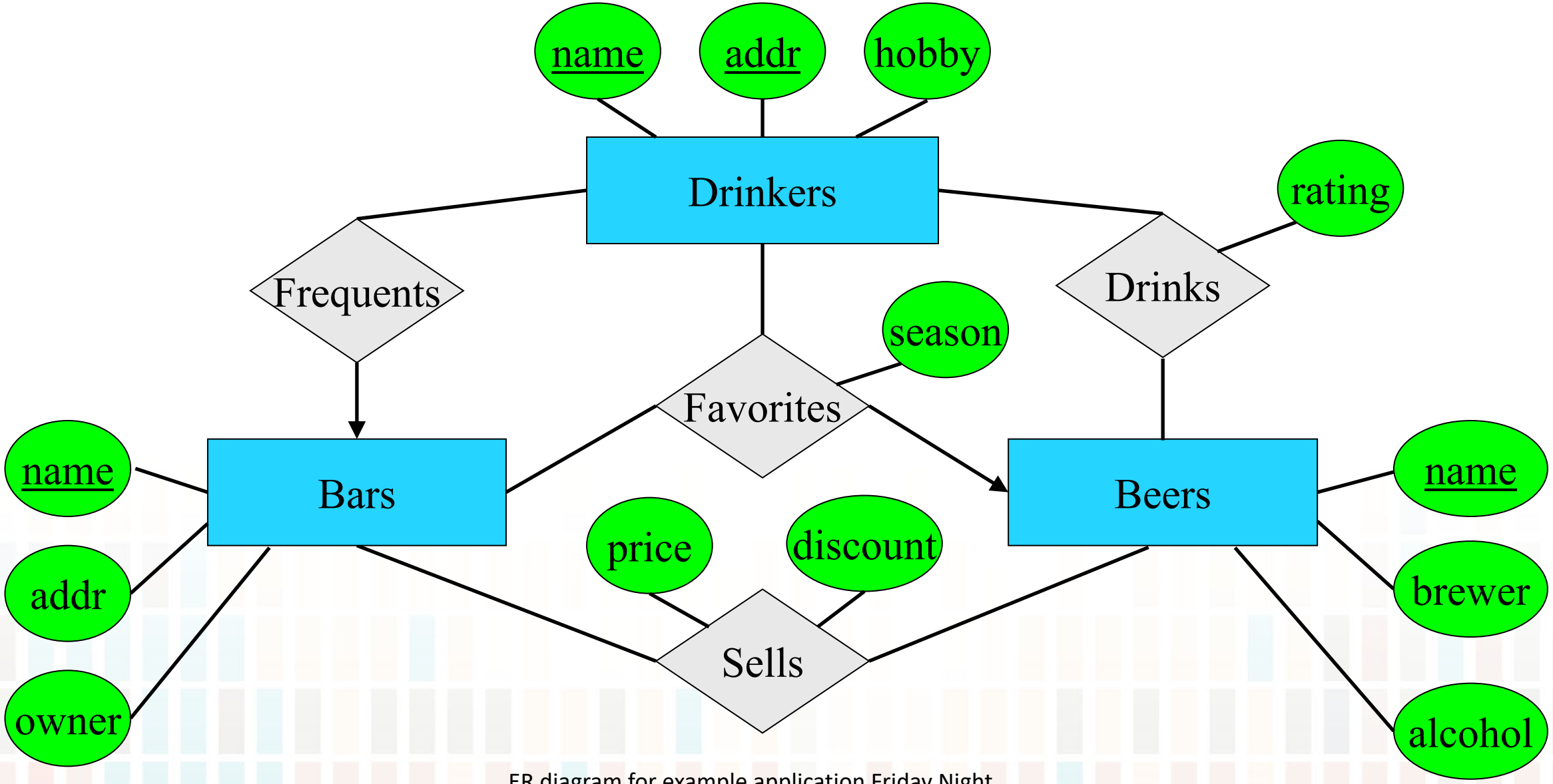
Developing an application on top of DBMS

Academic World

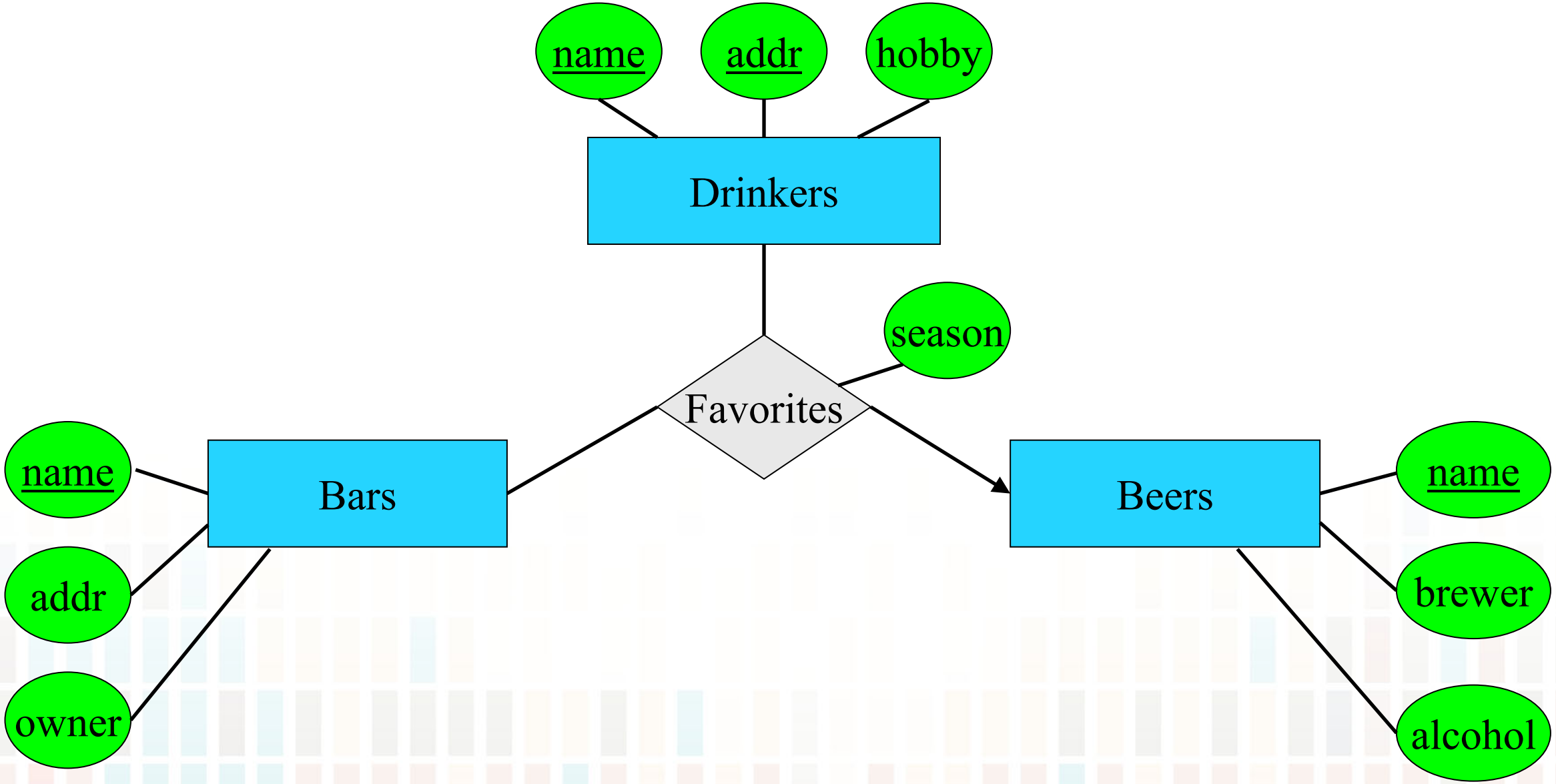


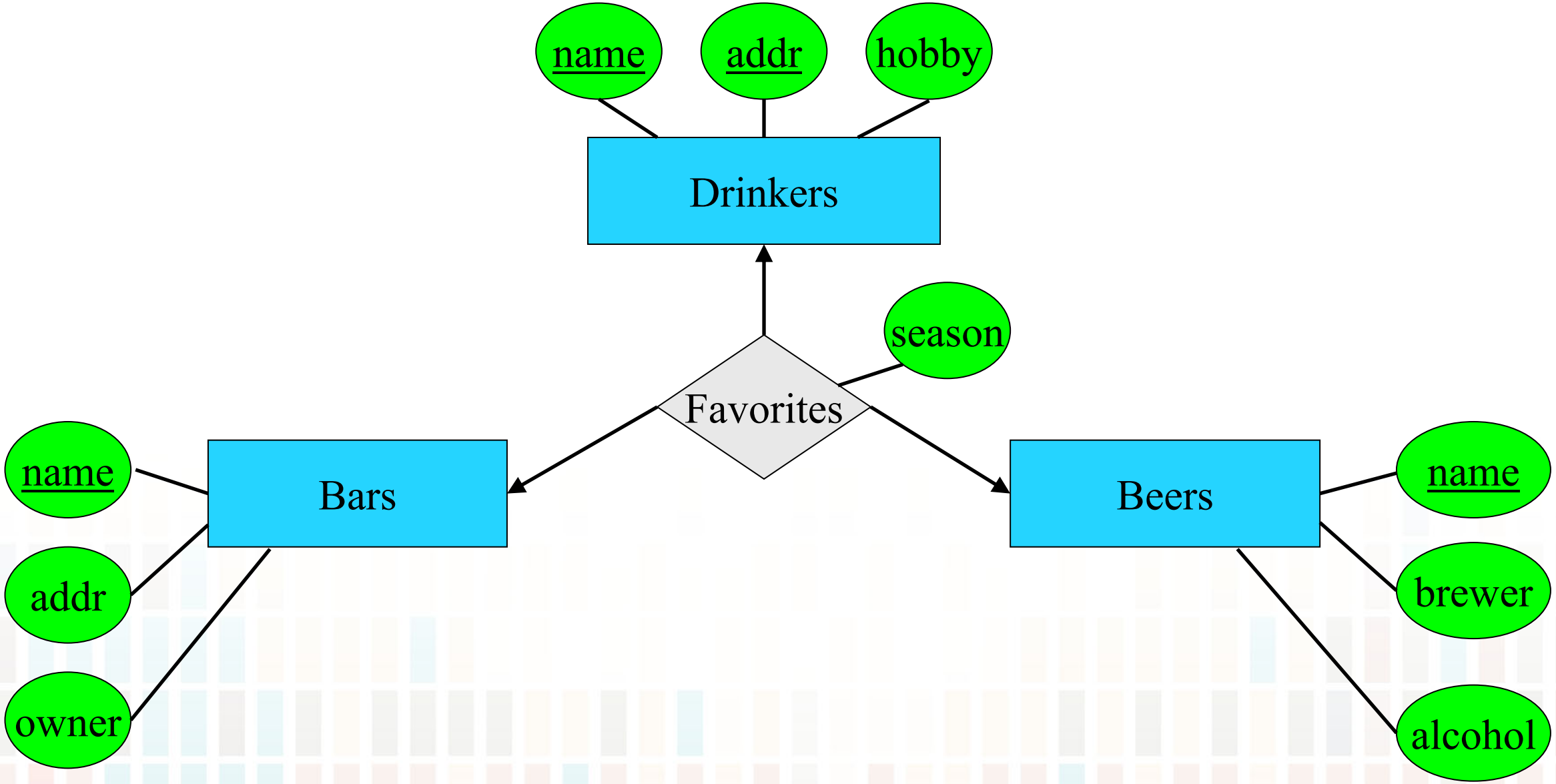
ER diagram for example application Academic World

Friday Night

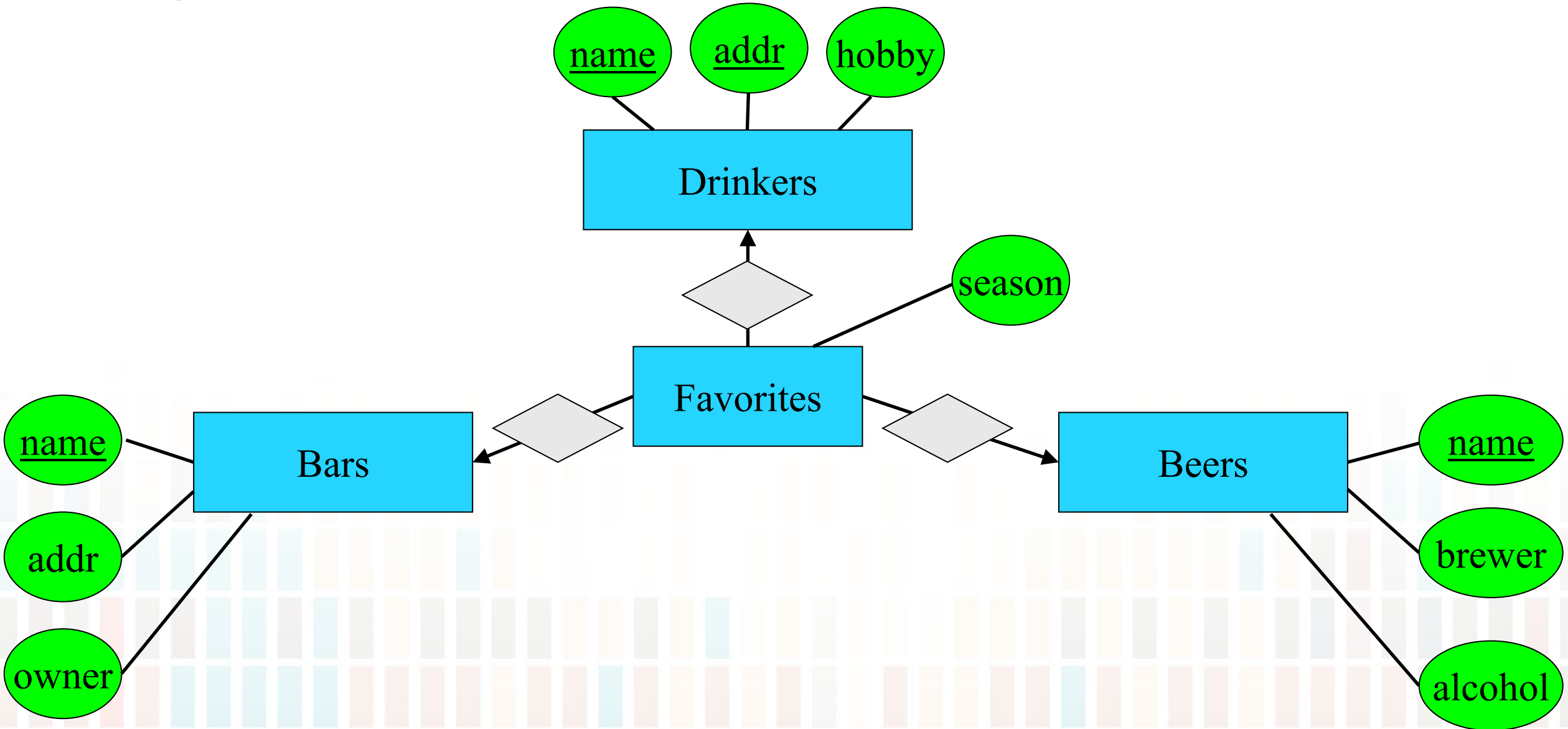


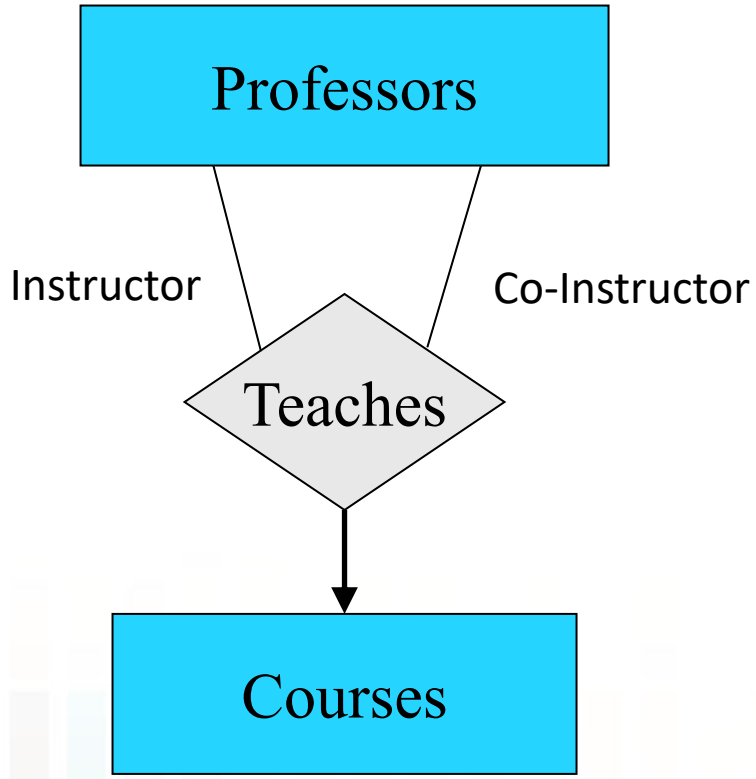
ER diagram for example application Friday Night

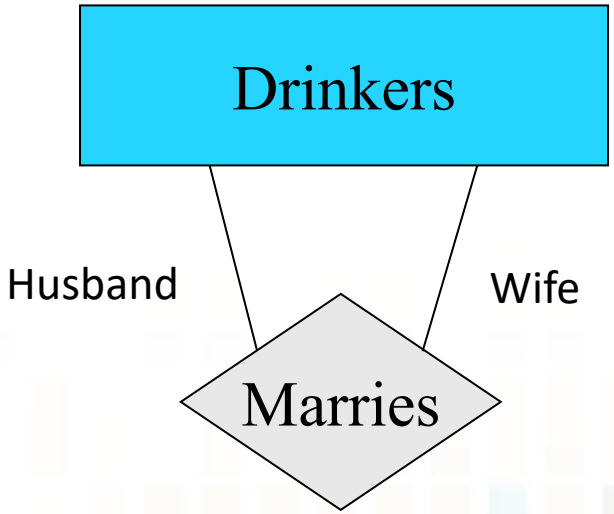

name
addr
hobby
Drinkers
season
Favorites
name
Bars
addr
owner
Beers
name
brewer
alcohol

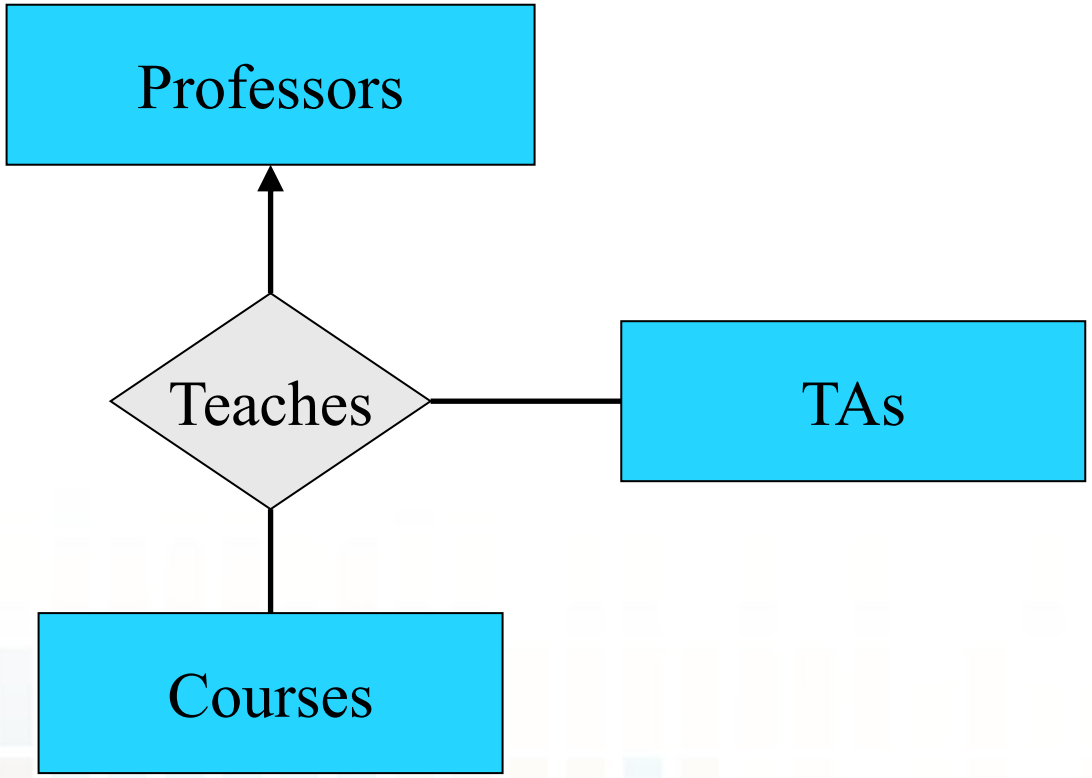

name
addr
hobby
Drinkers
season
Favorites
name
Bars
addr
owner
Beers
name
brewer
alcohol

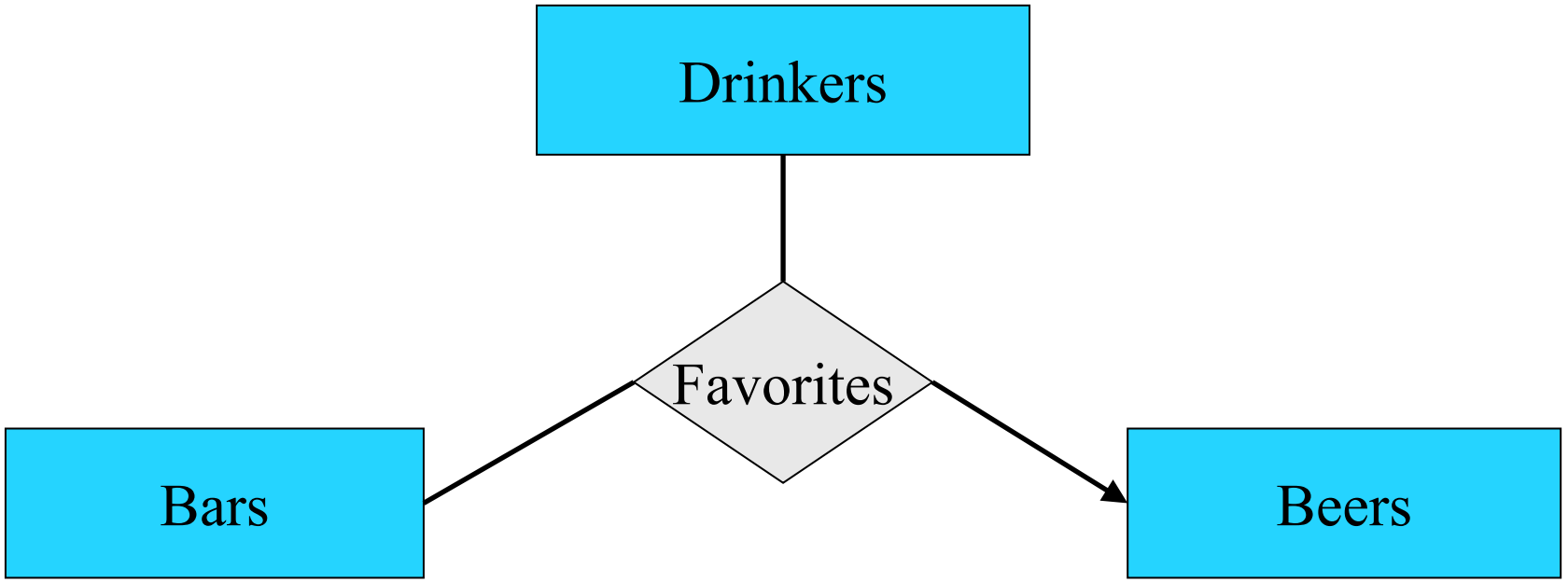
Fig

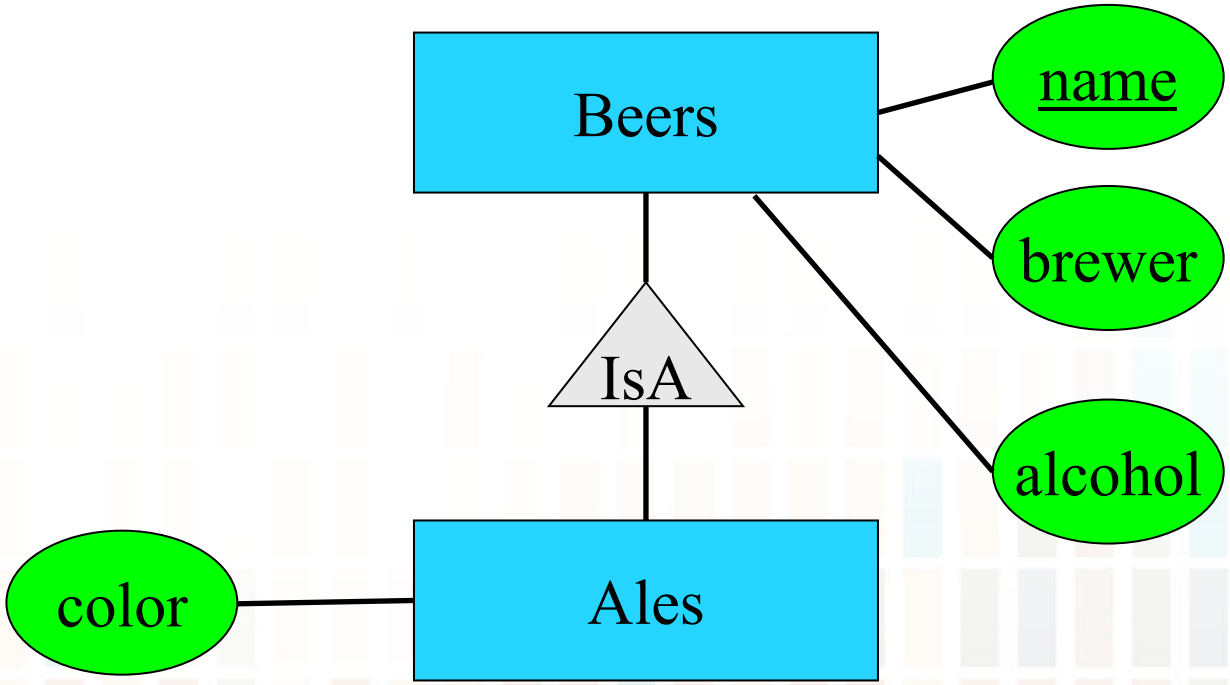

name
addr
hobby
Drinkers
season
Favorites
name
Bars
addr
owner
Beers
name
brewer
alcohol

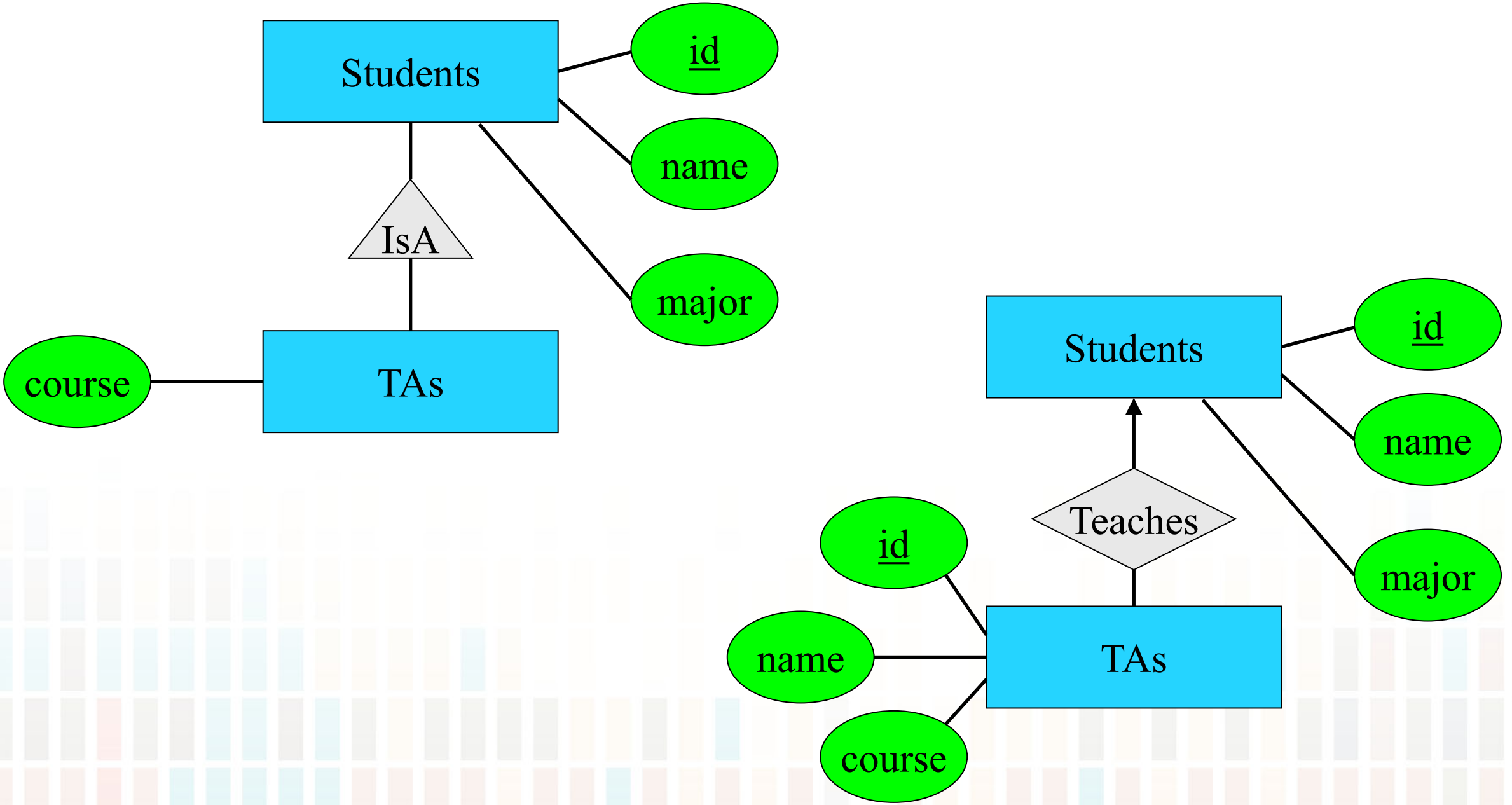

Professors
Instructor
Co-Instructor
Teaches
Courses

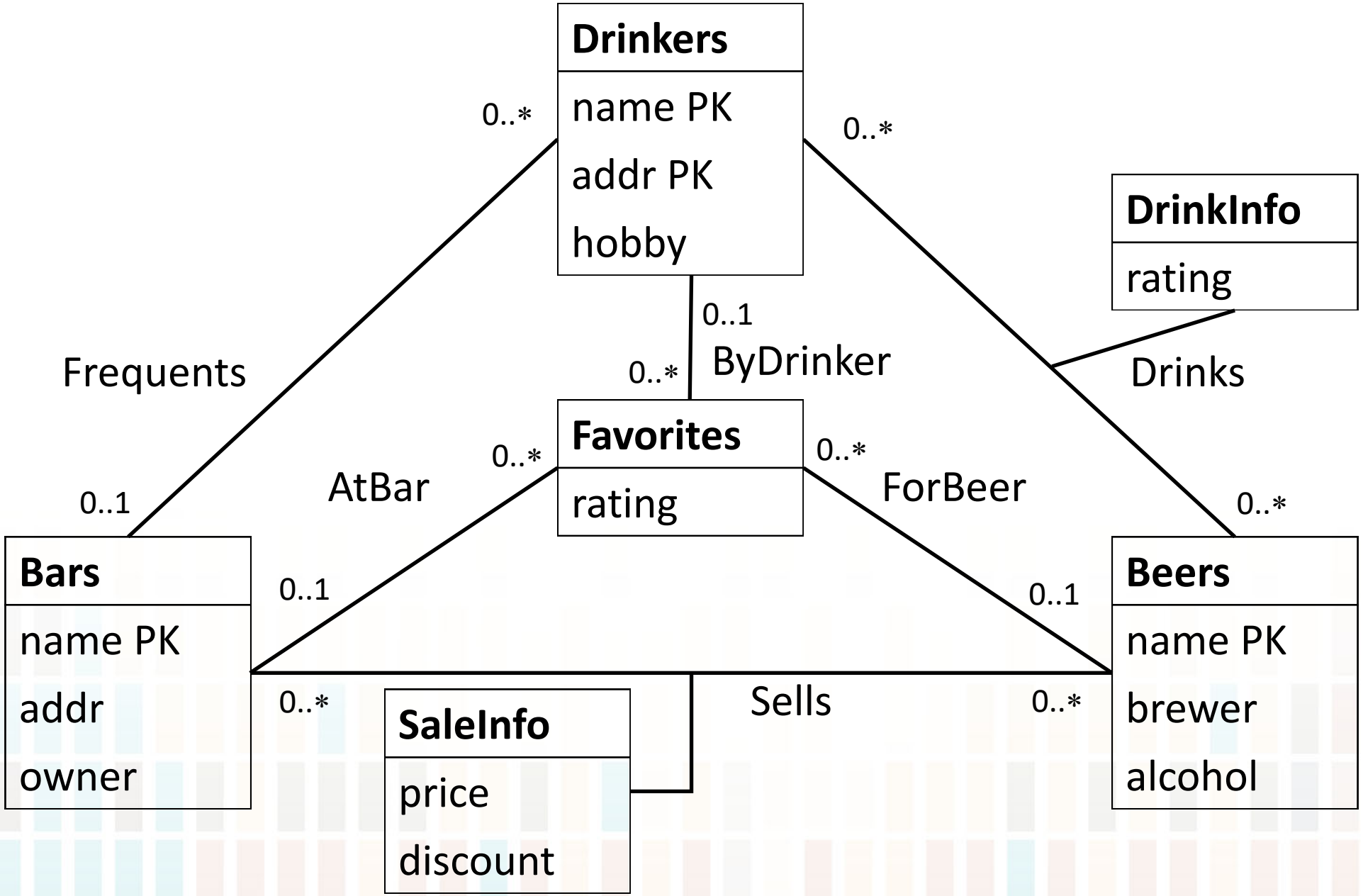

Drinkers
Husband
Wife
Marries


Professors
Teaches
TAs
Courses


Drinkers
Favorites
Bars
Beers


Beers
name
brewer
alcohol
IsA
color
Ales


Students
id
name
major
IsA
course
TAs
Students
id
name
major
Teaches
id
name
course
TAs


Drinkers
name PK
addr PK
hobby
DrinkInfo
rating
Frequents
0..*
0..*
0..1
0..*
ByDrinker
Drinks
Favorites
rating
0..*
AtBar
0..*
ForBeer
0..1
0..*
Bars
name PK
addr
owner
0..1
0..1
Beers
name PK
brewer
alcohol
0..*
Sells
0..*
SaleInfo
price
discount