CSE544
Introduction

Monday, April 1st, 2002

Staff

- Instructor: Dan Suciu
  - Sieg, Room 318, suciu@cs.washington.edu
  - Office hours: Monday, 3-4pm.
- TA: Peter Mork
  - Office hours: Wednesday, 12:30-1:20.
- Mailing list: cse544@cs.washington.edu
  - Send mail to majordomo@cs.washington.edu
    "subscribe cse544"
- Web page: (a lot of stuff already there)
  http://www.cs.washington.edu/544

Course Times

- Normally:
  - Mon, Wed, 1:30-2:50pm
- No classes:
  - Monday, May 6
  - Monday, June 3
  - Wednesday, June 6
- Make up classes (tentative)
  - Friday, May 10
  - Friday, May 31 (last lecture)
- Final: 2:30-4:20 p.m. Monday, Jun. 10, 2002

Goals of the Course

- Purpose:
  - Foundations of database management systems.
  - Issues in building database systems.
  - Introduction to current research in databases.

Grading

- Homeworks: 35%
  - Very little regurgitation.
  - Meant to be challenging (i.e., fun).
- Project: 30%
  - More later.
- Final: 30%
- Intangibles: 5%

Textbook

- **Database Management Systems**,
  Ramakrishnan and Gehrke.

Also:

- **Foundations of Databases**,
  Abiteboul, Hull & Vianu

Also:

- Some papers to read (see website)

Other Useful Texts

- Pair of books by Ullman, Widom and Garcia-Molina
- *Parallel and Distributed DBMS* (Ozsu and Valduriez)
- *Transaction Processing* (Gray and Reuter)
- *Data and Knowledge based Systems* (volumes I, II) (Ullman)
- *Data on the Web* (Abiteboul, Buneman, Suciu)
- *Readings in Database Systems* (Stonebraker and Hellerstein)
- Proceedings of SIGMOD, VLDB, PODS conferences.

Prerequisites

- Officially: none
- Real prerequisites:
  - Programming languages
  - Logic
  - Complexity theory
  - Algorithms and data structures

Traditional Database Application

Suppose we are building a system
to store the information about:

- students
- courses
- professors
- who takes what, who teaches what

Why use a DBMS ?

What we need from a database:

- store the data for a long period of time
  - large amounts (100s of GB)
  - protect against crashes, and unauthorized use
- allow users to query/update:
  - who teaches “CSE142”
- allow several (100s, 1000s) users to access the data simultaneously
- allow administrators to change the schema
  - add information about TAs

Trying Without a DBMS

Why Direct Implementation Won't Work:

- Storing data: file system is limited
  - size less than 4GB (on 32 bits machines)
  - when system crashes we may loose data
  - password-based authorization insufficient
- Query/update:
  - need to write a new C++/Java program for every new query
  - need to worry about performance
  - have you tried to update a file recently ?

- Concurrency: limited protection
  - need to worry about interfering with other users accessing *the same* file
  - need to offer different views to different users (e.g. registrar, students, professors)
- Schema change:
  - need to rewrite virtually all applications

DBMS

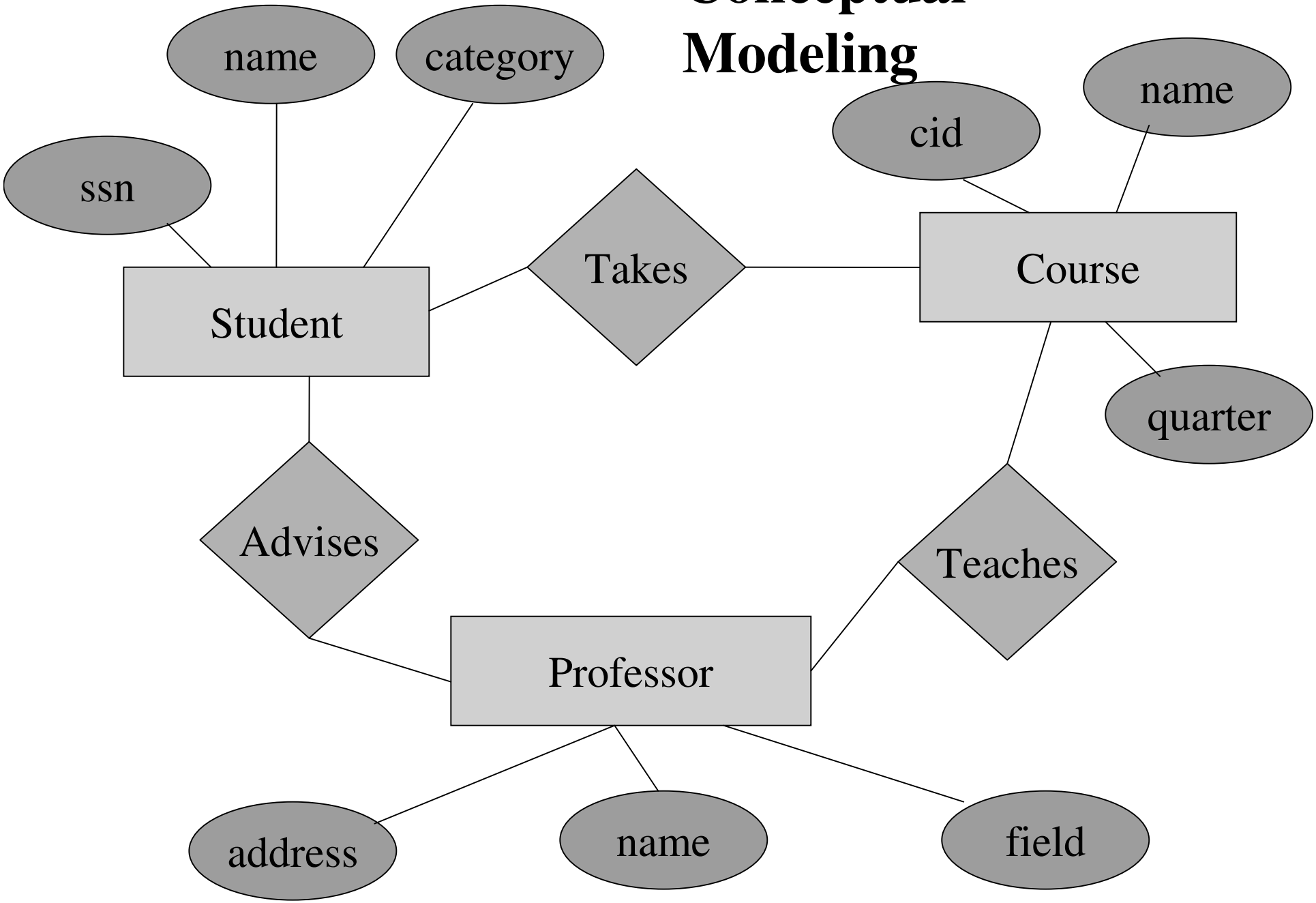
The DBMS offers:

- A Data Definition Language - DDL
- A Data Manipulation Language - DML
  - Aka “query language”. Always SQL
- Storage management
- Transaction Management
  - concurrency control
  - recovery

Building an Application with a DBMS

- Conceptual modeling (conceptual, pictures)
- Schema design and implementation
- Write application programs using the DBMS
  - way easier now that the data management is taken care of.

Conceptual Modeling



Schema Design and Implementation

- Tables:

Students:

SSN	**Name**	**Category**
123-45-6789	Charles	undergrad
234-56-7890	Dan	grad
	…	…

Takes:

SSN	**CID**
123-45-6789	CSE444
123-45-6789	CSE444
234-56-7890	CSE142
	…

Courses:

CID	**Name**	**Quarter**
CSE444	Databases	fall
CSE541	Operating systems	winter

- Separates the logical view from the physical view of the data.

Querying a Database

- Find all courses that “Mary” takes
- S(tructured) Q(uery) L(anguage)

```
select  C.name
from Students S, Takes T, Courses C
where S.name=“Mary” and
    S.ssn = T.ssn and T.cid = C.cid
```

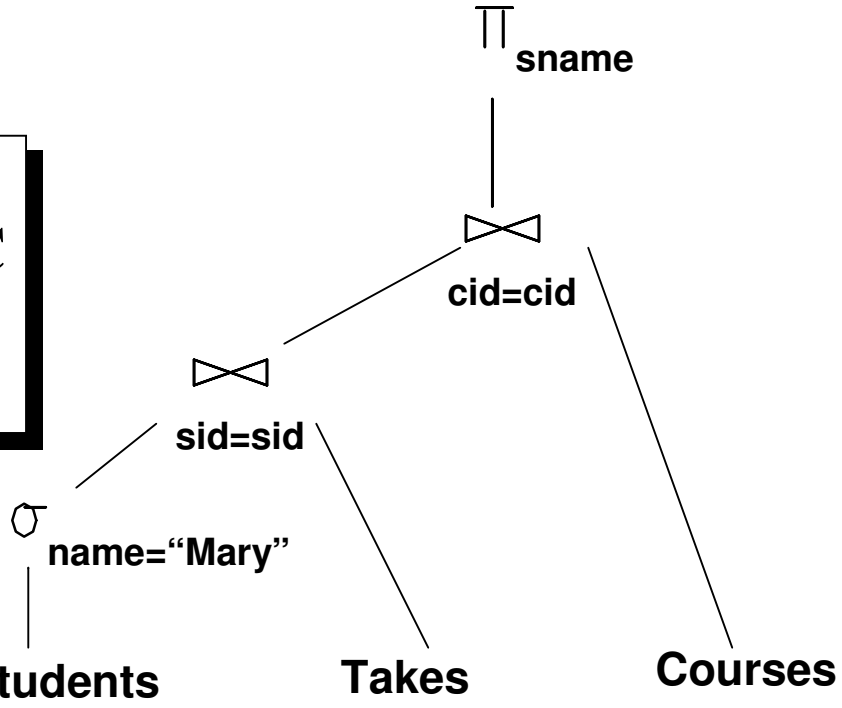
- Query processor figures out how to answer the query efficiently.

Query Optimization

Goal:

Declarative SQL query ⟶ ***Imperative query execution plan:***

```
select  C.name
from Students S, Takes T, Courses C
where S.name="Mary" and
    S.ssn = T.ssn and T.cid = C.cid
```



Plan: *Tree of Relational Algebra operators, with a choice of algorithm implementation for each operator*

Ideally: Want to find best plan. Practically: Avoid worst plans!

Database Industry

- Relational databases are a great success of theoretical ideas.
- Main players: Oracle, IBM, MS, Sybase
- Industry trends:
  - warehousing and decision support
  - data integration
  - XML, XML, XML.

What is the Field of Databases ?

- To a theoretical researcher (PODS/ICDT/LICS)
  - Focus on the query languages
  - Query language = logic = complexity classes
- To an applied researcher (SIGMOD/VLDB/ICDE)
  - Query optimization
  - Query processing (yet-another join algorithm)
  - Transaction processing, recovery (but most stuff is already done)
  - Novel applications: data mining, high-dimensional search
- To a systems programmer at Oracle:
  - Millions lines of code
- To an application builder:
  - E/R, SQL, ODBC/JDBC

Database Research

What is cool today:

- XML:
  - Theory: trees + logic + automata = ?
  - Implementation on top of a RDBMS
  - Native implementation, indexing
- Processing stream data
- Peer data management (p2p)

Course (Rough) Outline

- The basics: E/R, SQL
- Semistructured data and XML
- Some theory
- Query processing
- Query optimization

First Homework

- A small database application
- Provide a Web access to an anatomical database
- Required: SQL (Postgres), Java, JDBC, Website design

Projects

- Goal: apply some database principles to a new problem
- Suggested topics are from P2P and XML (on the website soon), but anything goes.
- Groups of 2-3
- Groups assembled end of week 2;
- Proposals, beginning of week 4
- Touch base with me: every two weeks.
- Start Early.

Final

- June 10, 2:30 – 4:20, same room
- Will be challenging

Today: Database Design

- E/R diagrams (Chapter 2 in the textbook)
- Why do we need it?
  - Agree on structure of the database before deciding on a particular implementation.
- Consider issues such as:
  - What entities to model
  - How entities are related
  - What constraints exist in the domain
  - How to achieve *good* designs

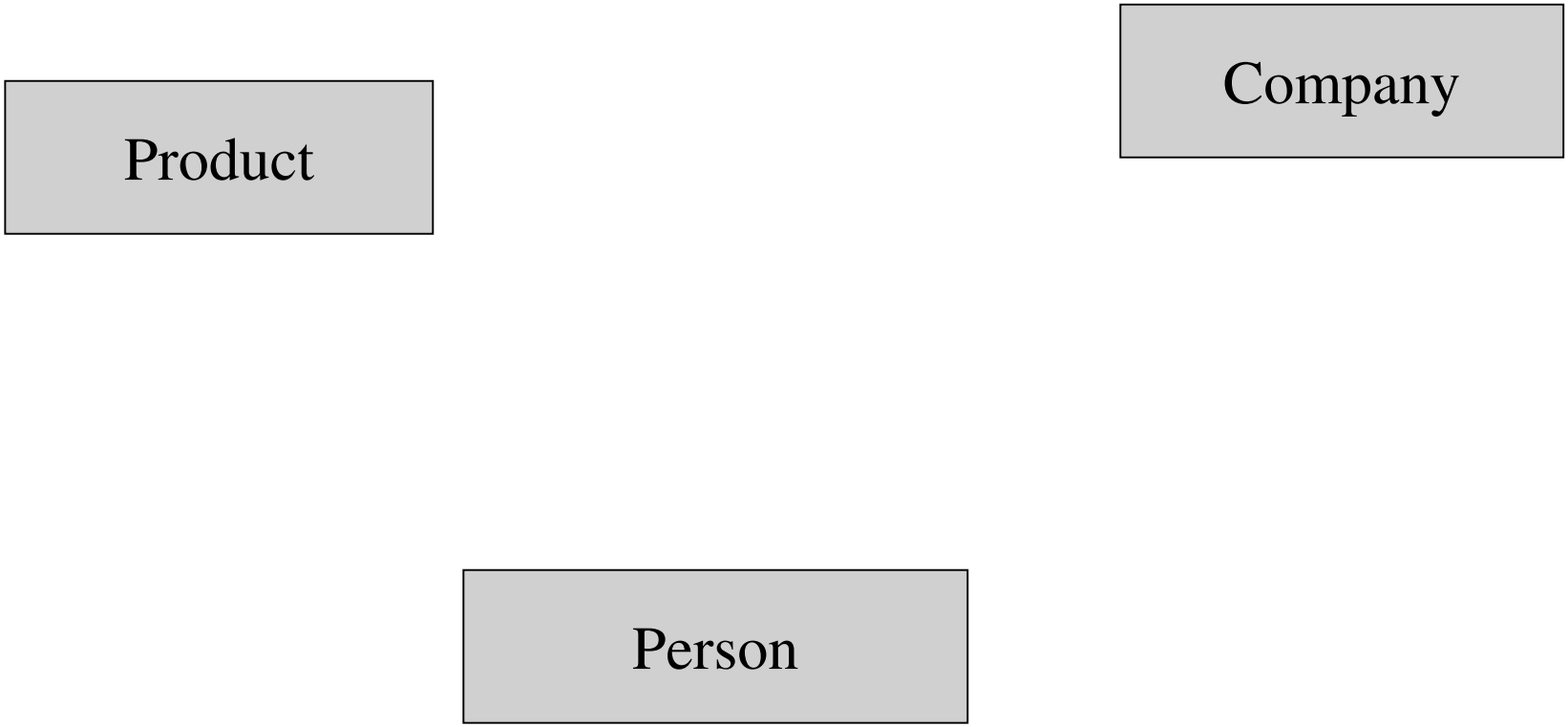
Entity-Relationship (E/R) Model

- Basic design paradigm in E/R:
  - Model ***entities*** and their ***properties.***
- For abstraction purposes:
  - Group objects into ***entity sets.***
- What qualifies as a ***good*** entity set ?
  - Entities in an entity set should have common properties.

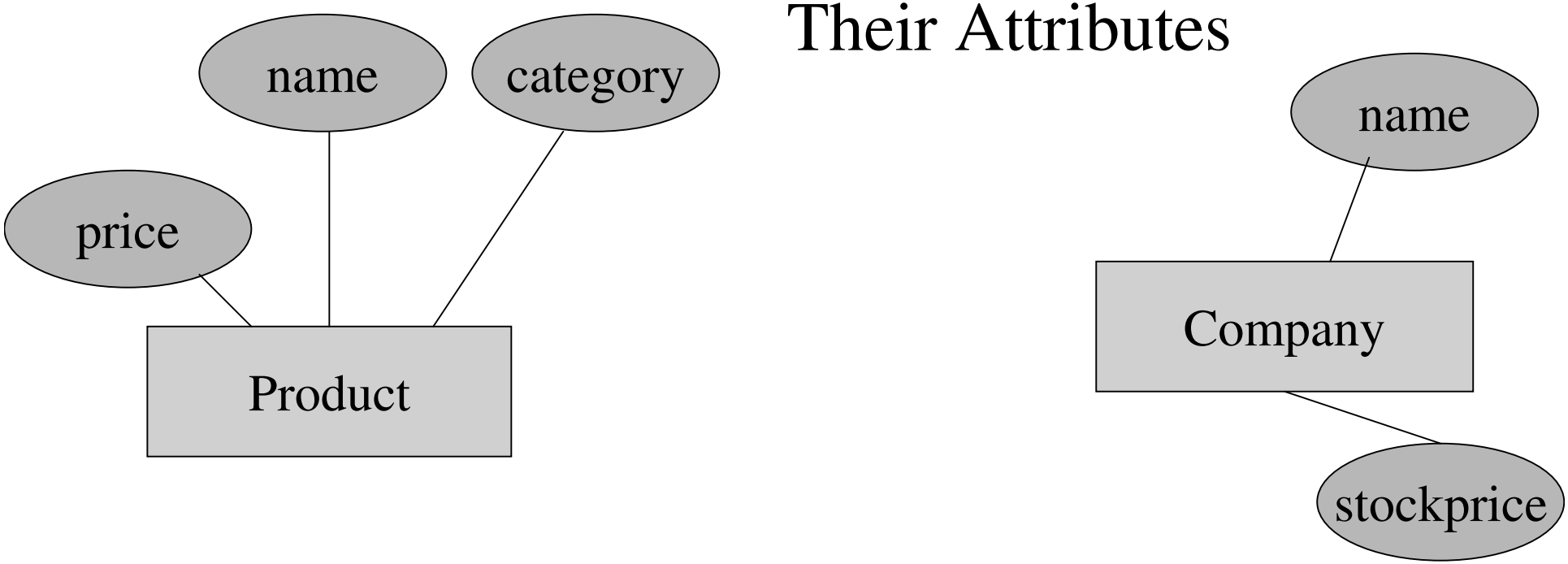
E/R Design

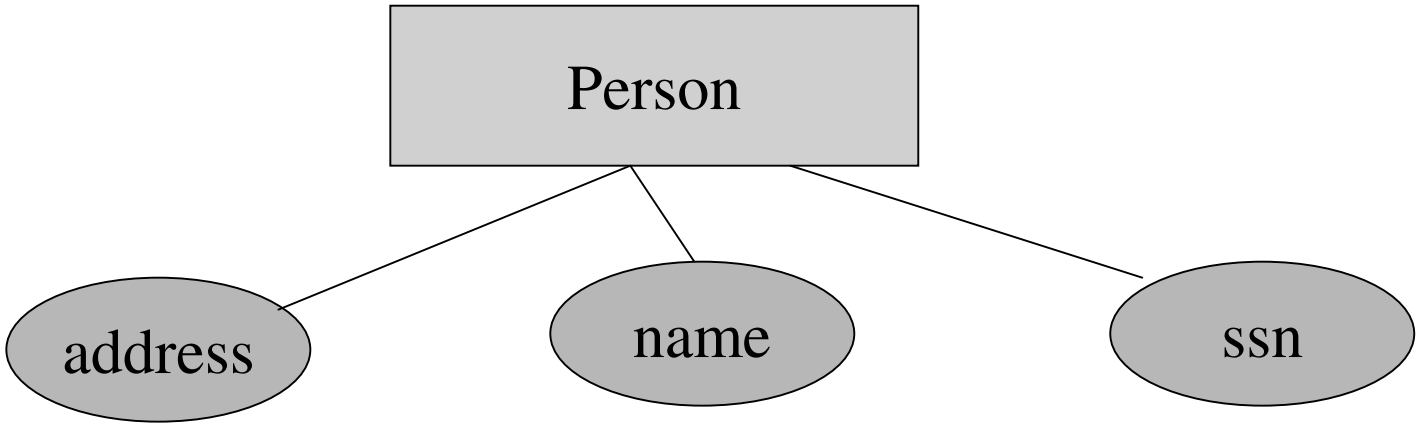
- Three steps:
  - Design the entity sets
  - Design their attributes
  - Design the relationships

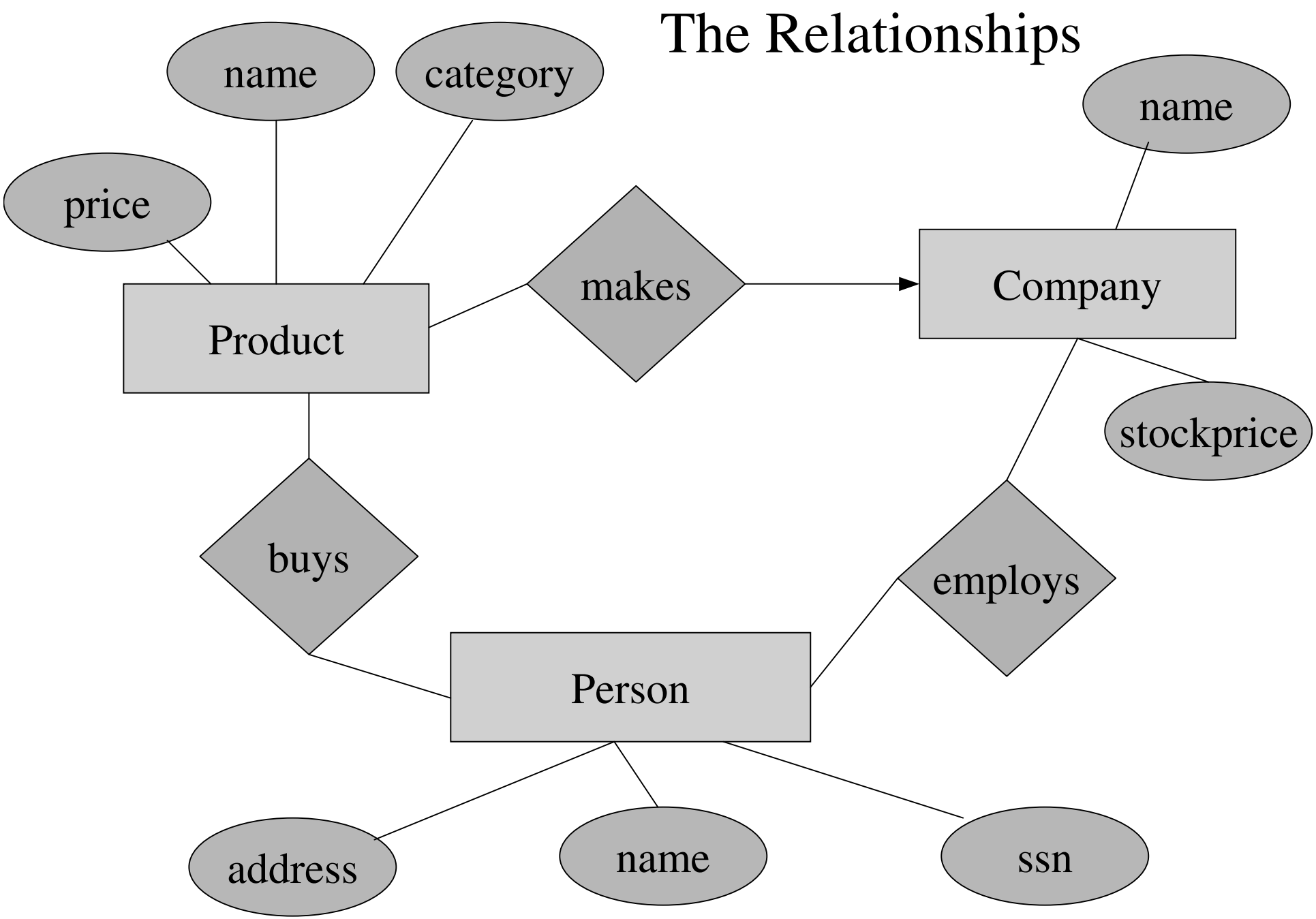
E/R Example: The Entity Sets



Their Attributes






The Relationships
name
category
price
Product
makes
Company
name
stockprice
buys
employs
Person
address
name
ssn

Entity / Relationship Diagrams in Summary

Entity sets:

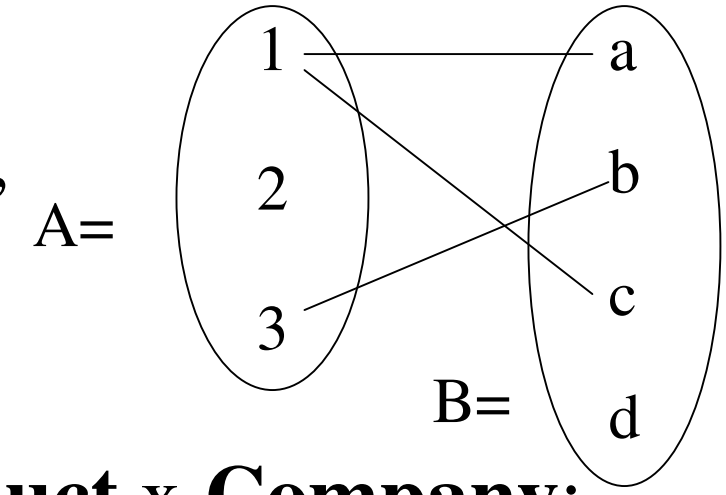
Properties:

Relationships:

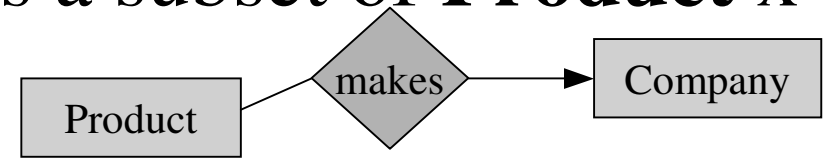
What is a Relation ?

- A mathematical definition:
  - if A, B are sets, then a relation R is a subset of A x B
- A={1,2,3}, B={a,b,c,d},

  R = {(1,a), (1,c), (3,b)}

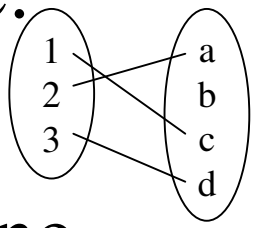


- **makes** is a subset of **Product** x **Company**:

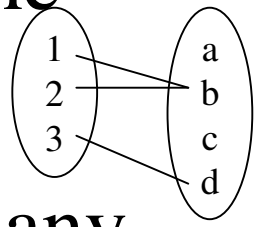


Multiplicity of E/R Relations

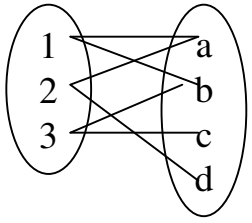
- one-one:

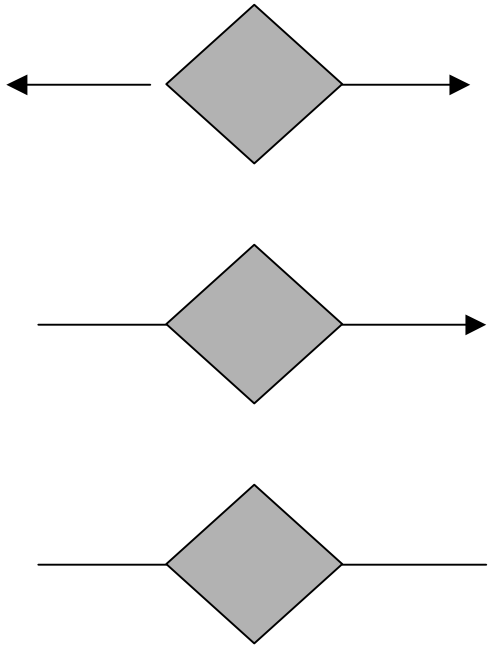


- many-one



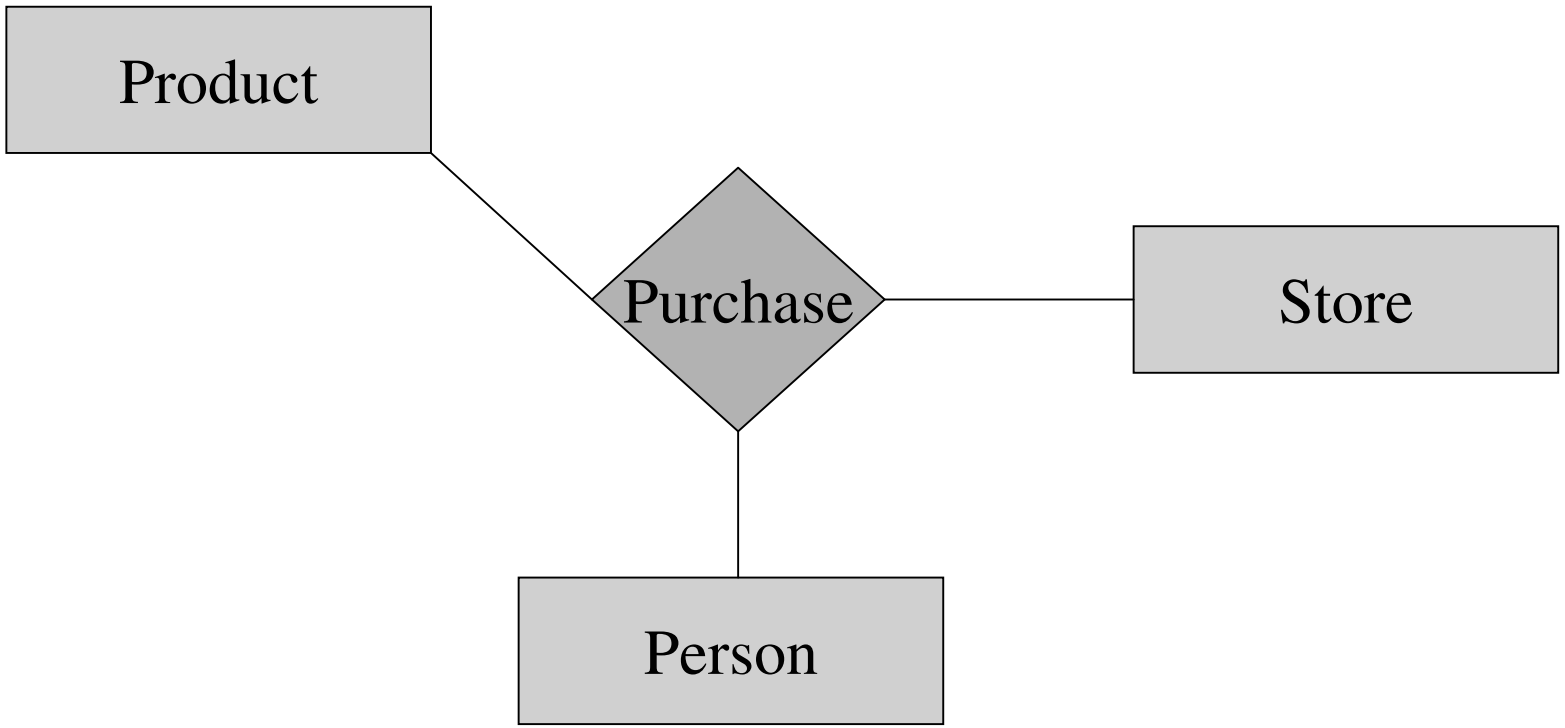
- many-many





Multi-way Relationships

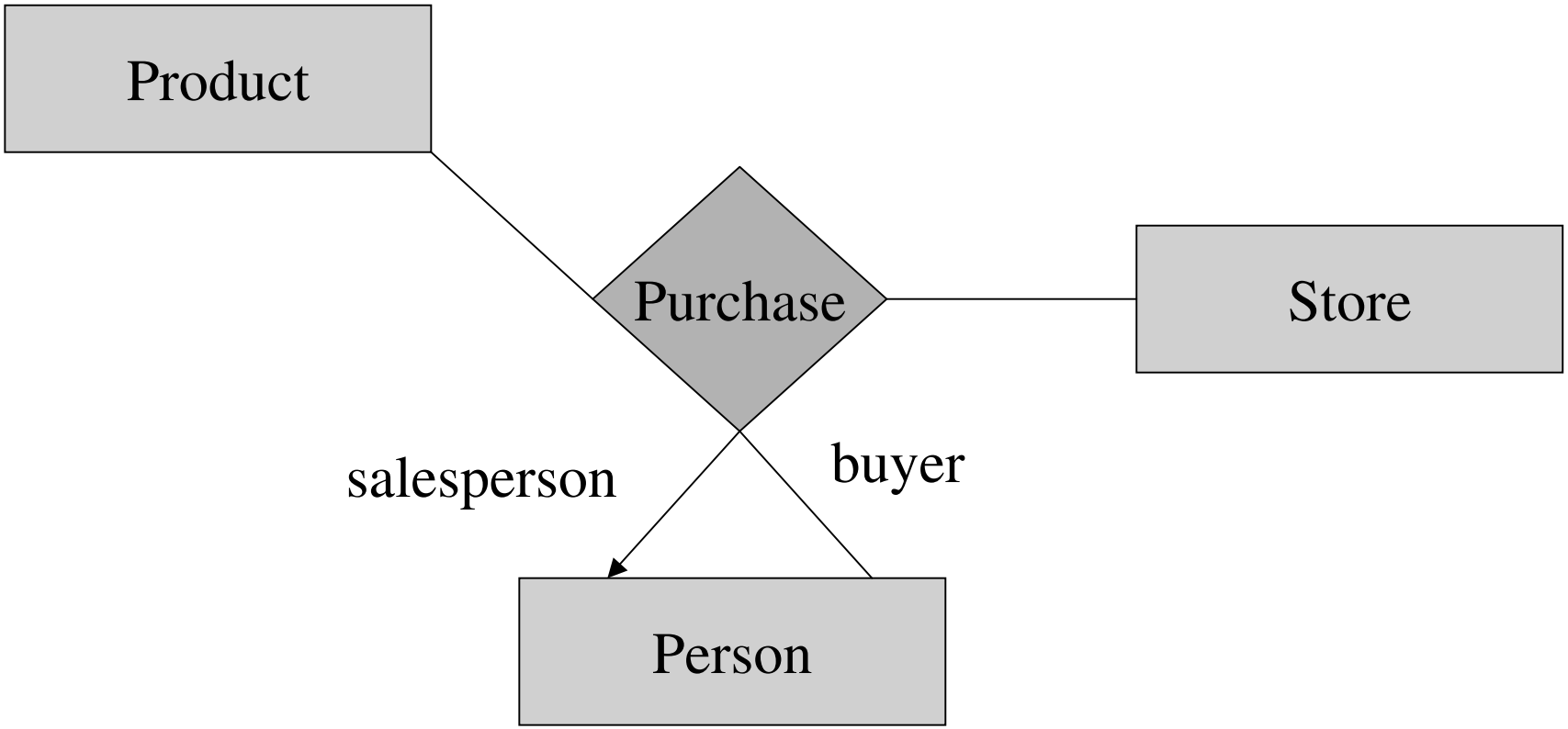
How do we model a purchase relationship between buyers, products and stores?



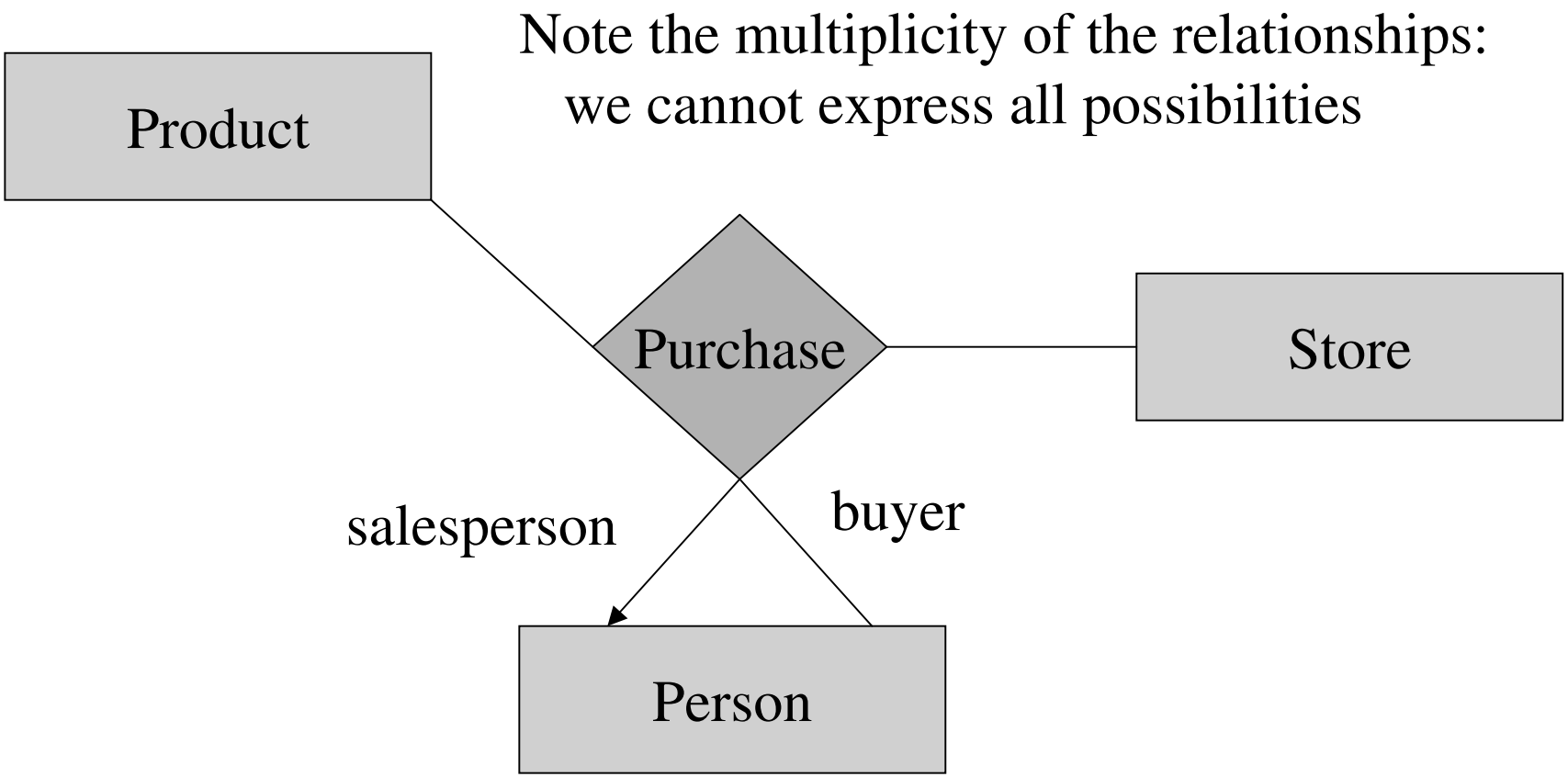
Can still model as a mathematical set (how ?)

Roles in Relationships

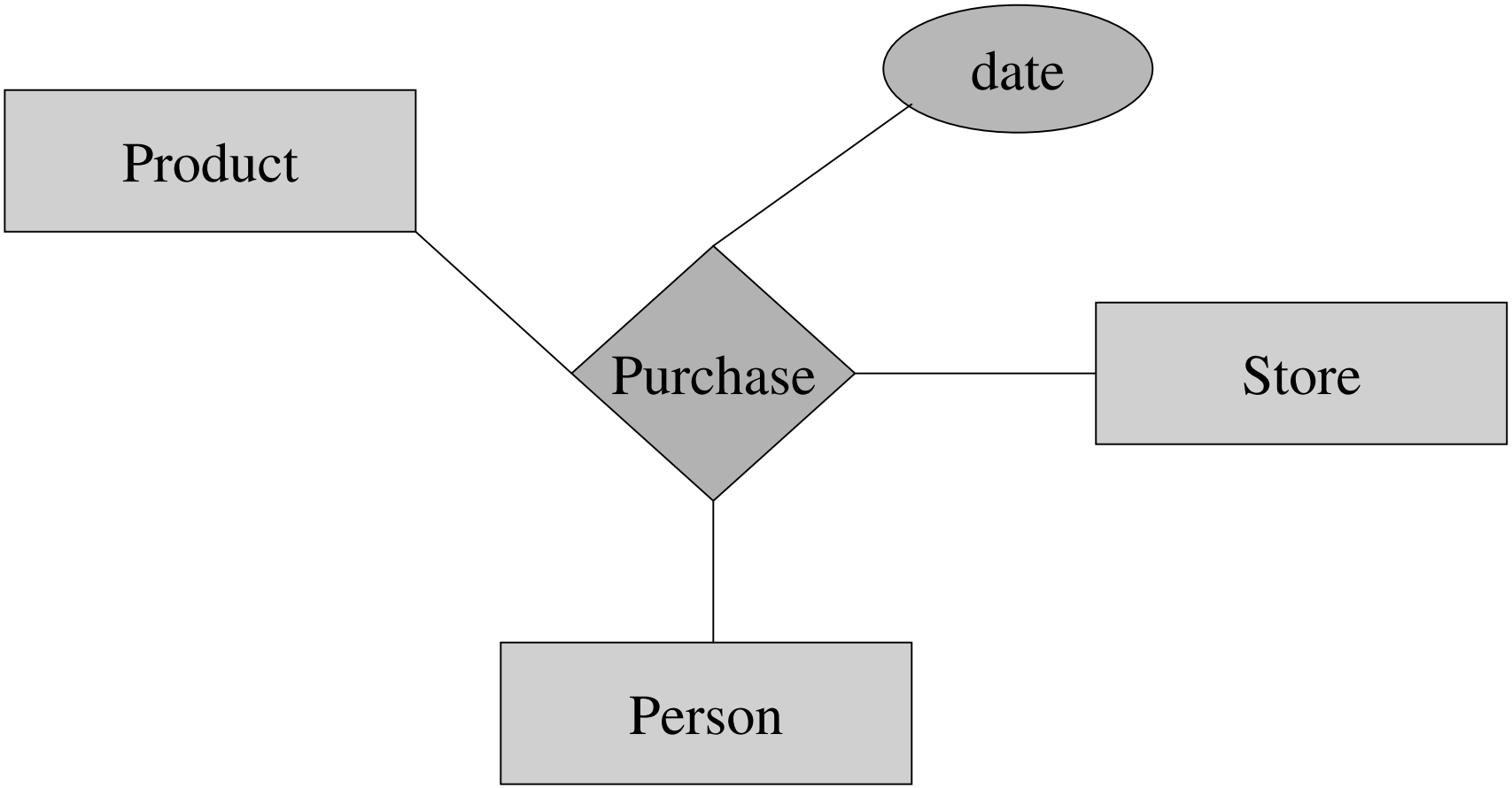
What if we need an entity set twice in one relationship?



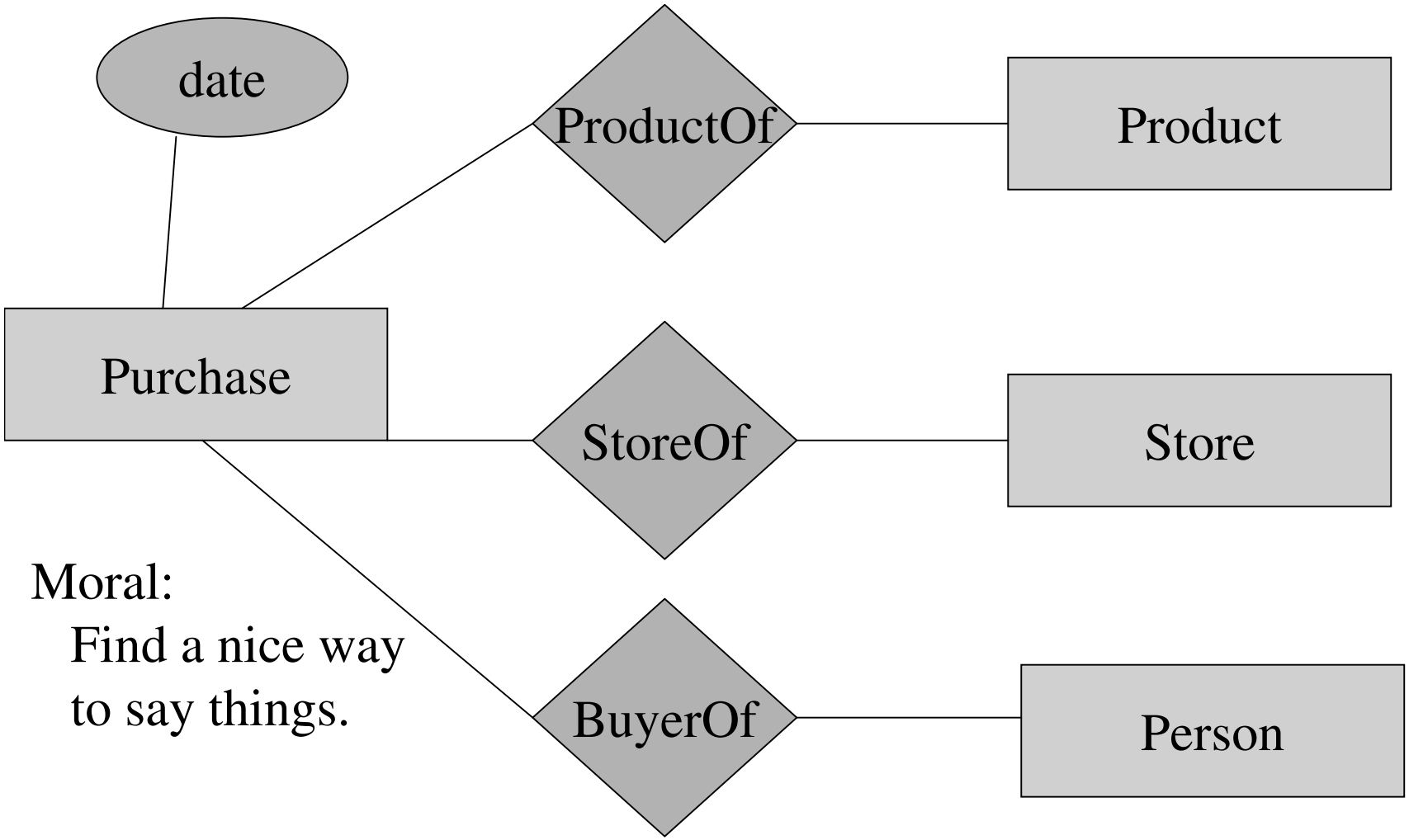
Roles in Relationships



Attributes on Relationships



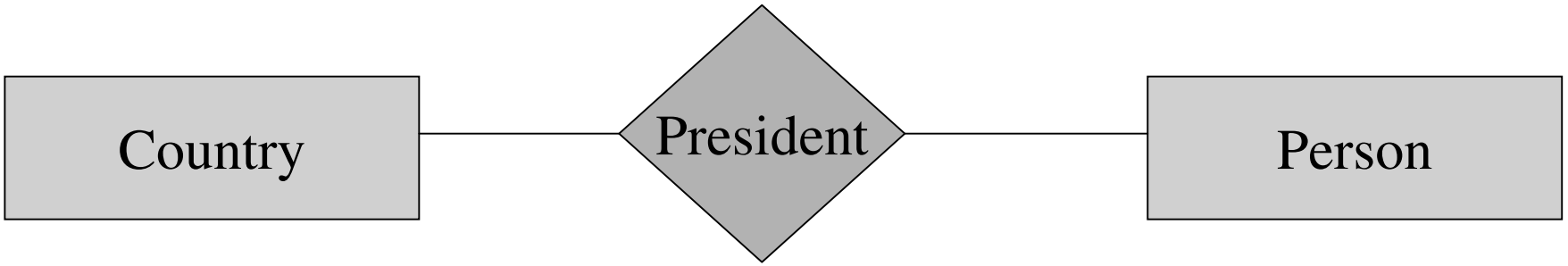
Converting Multi-way Relationships to Binary



Moral:
Find a nice way
to say things.

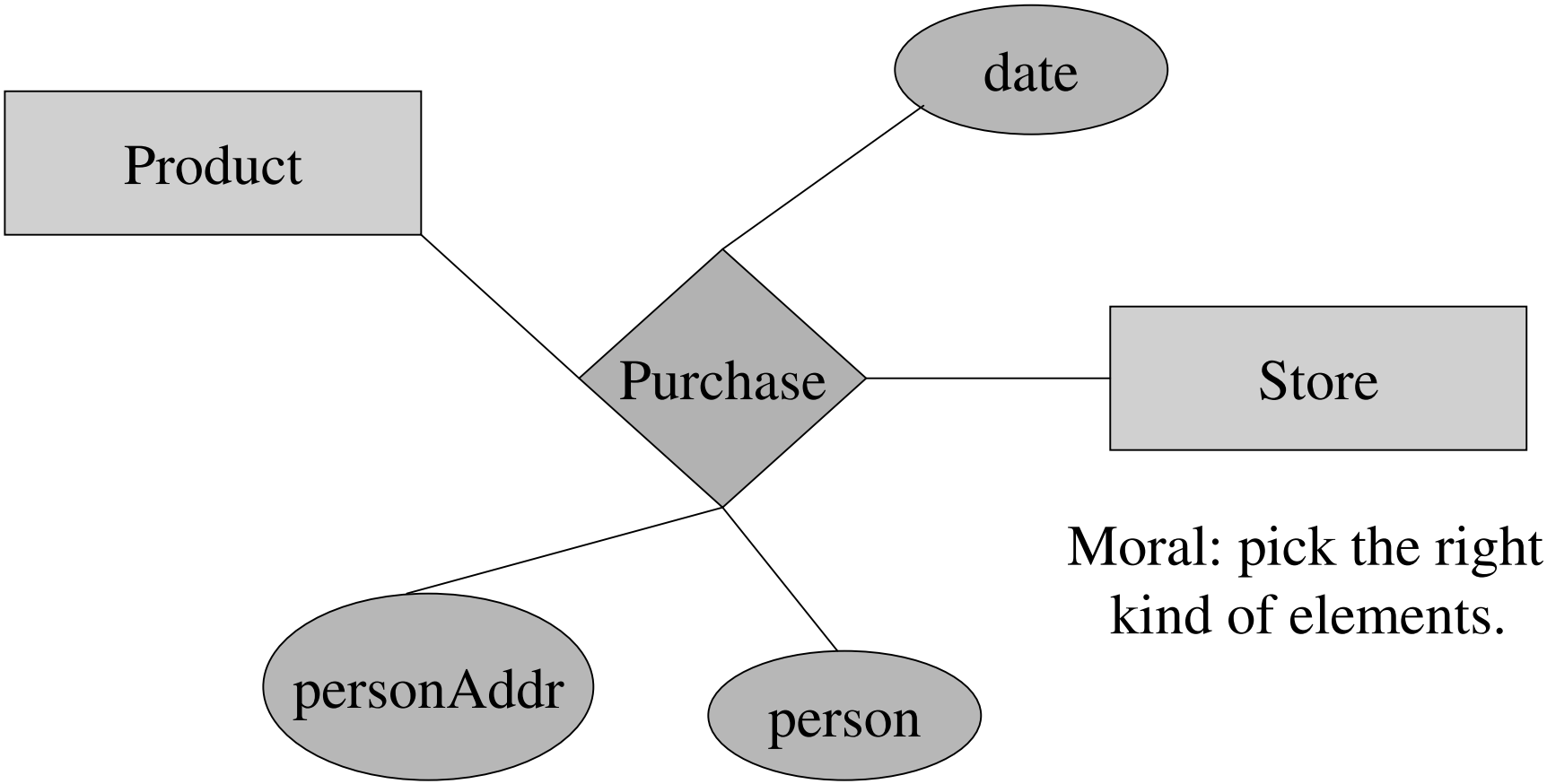
Design Principles

What's wrong?



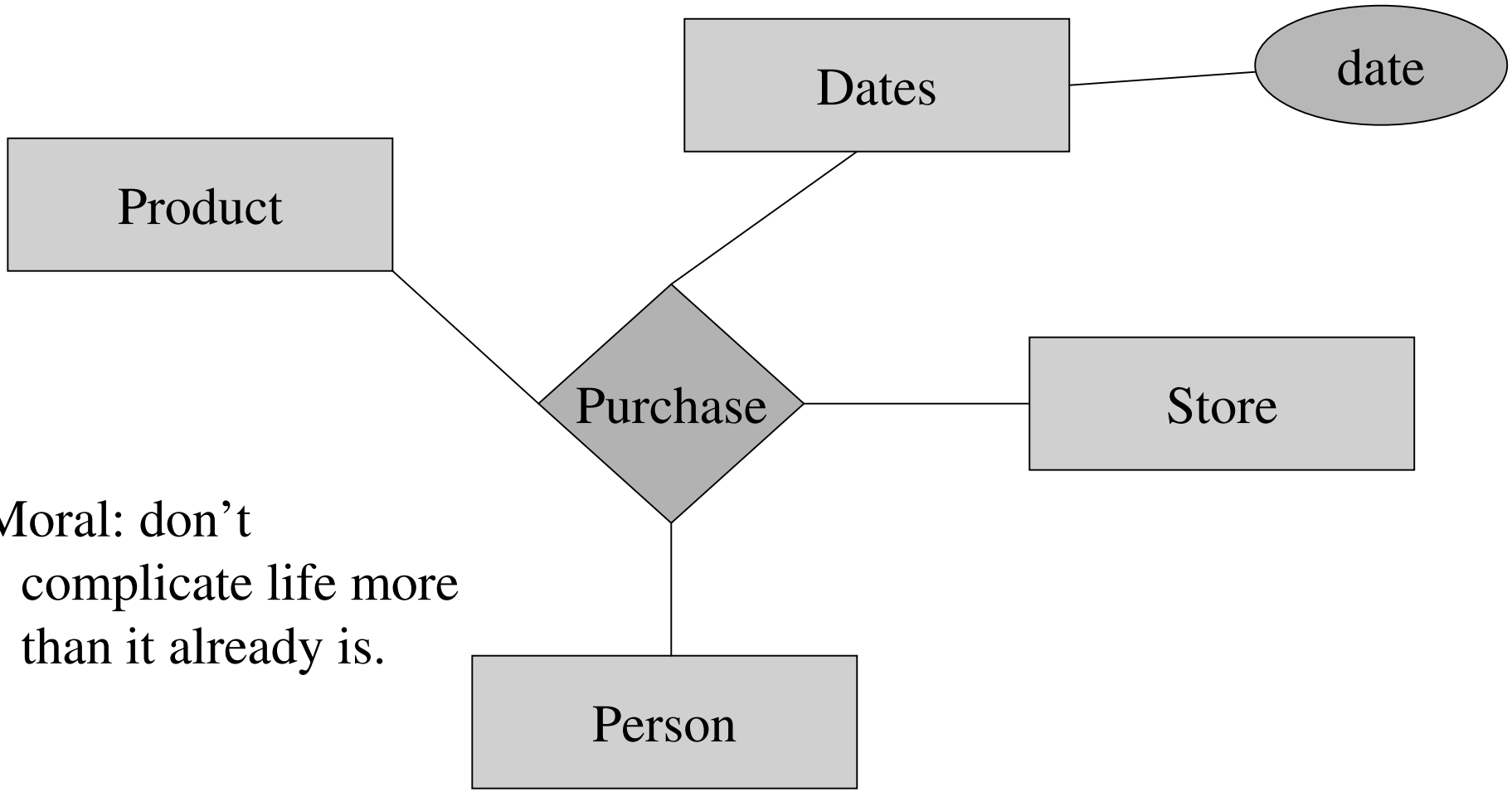
Moral: be faithful!

What's Wrong?



Moral: pick the right kind of elements.

What's Wrong?



Moral: don't complicate life more than it already is.