

E/R Diagrams

September 11, 2013

Outline

Last time:

- Constraints and triggers (Sections 2.3, 7.1-7.5)

Today:

- E/R Diagrams (Sections 4.1-4.5)

Next:

- Functional Dependencies

Review

- Types of constraints
- Triggers
- Advantage of using triggers versus constraints

Example: A Trigger

- Instead of using a foreign-key constraint and rejecting insertions into `Purchase(ProdName, Store)` with unknown products, a trigger can add that product to `Product`, with a `NULL` category.

Example: Trigger Definition

```
CREATE TRIGGER ProductTrig
  AFTER INSERT ON Purchase
  REFERENCING NEW ROW AS NewTuple
  FOR EACH ROW
  WHEN (NewTuple.ProdName NOT IN
        (SELECT name FROM Product))
  INSERT INTO Product(name)
  VALUES(NewTuple.ProdName);
```

The event

The condition

The action

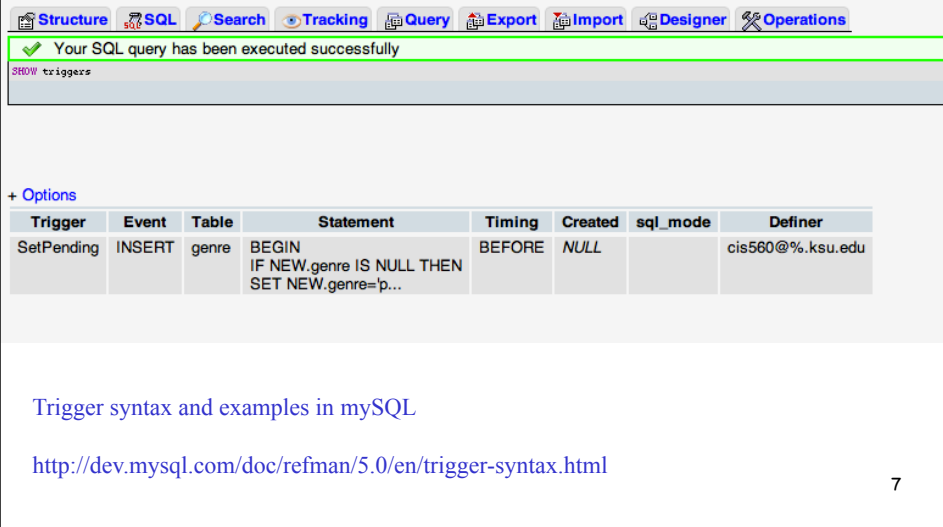
5

Example: Trigger Definition in mySQL

```
DELIMITER $
CREATE TRIGGER ProductTrig
  AFTER INSERT ON Purchase
  REFERENCING NEW ROW AS NewTuple
  FOR EACH ROW
  BEGIN
    IF (NEW.ProdName NOT IN
        (SELECT name FROM Product)) THEN
      INSERT INTO Product(name)
      VALUES(NEW.ProdName);
    END IF;
  END;
$
DELIMITER ;
```

6

SHOW TRIGGERS



The screenshot shows a MySQL database management tool interface. At the top, there is a toolbar with icons for Structure, SQL, Search, Tracking, Query, Export, Import, Designer, and Operations. Below the toolbar, a green status bar indicates "Your SQL query has been executed successfully". The main area displays the results of the "SHOW TRIGGERS" query. A table with 8 columns is shown: Trigger, Event, Table, Statement, Timing, Created, sql_mode, and Definer. The table contains one row of data for a trigger named "SetPending".

Trigger	Event	Table	Statement	Timing	Created	sql_mode	Definer
SetPending	INSERT	genre	BEGIN IF NEW.genre IS NULL THEN SET NEW.genre=p...	BEFORE	NULL		cis560@%.ksu.edu

Below the table, there is a link to "Trigger syntax and examples in mySQL" and a URL: <http://dev.mysql.com/doc/refman/5.0/en/trigger-syntax.html>. The page number "7" is in the bottom right corner.

E/R Model

- Gives us a language to specify
 - what information the db must hold
 - what are the relationships among components of that information
- Proposed by Peter Chen in 1976
- What we will cover
 - basic stuff
 - constraints
 - weak entity sets
 - design principles

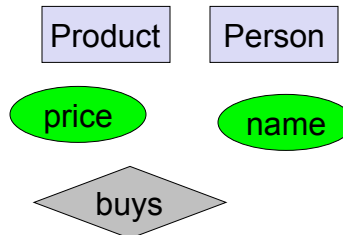
Entity / Relationship Diagrams

Objects → entities
Classes → entity sets

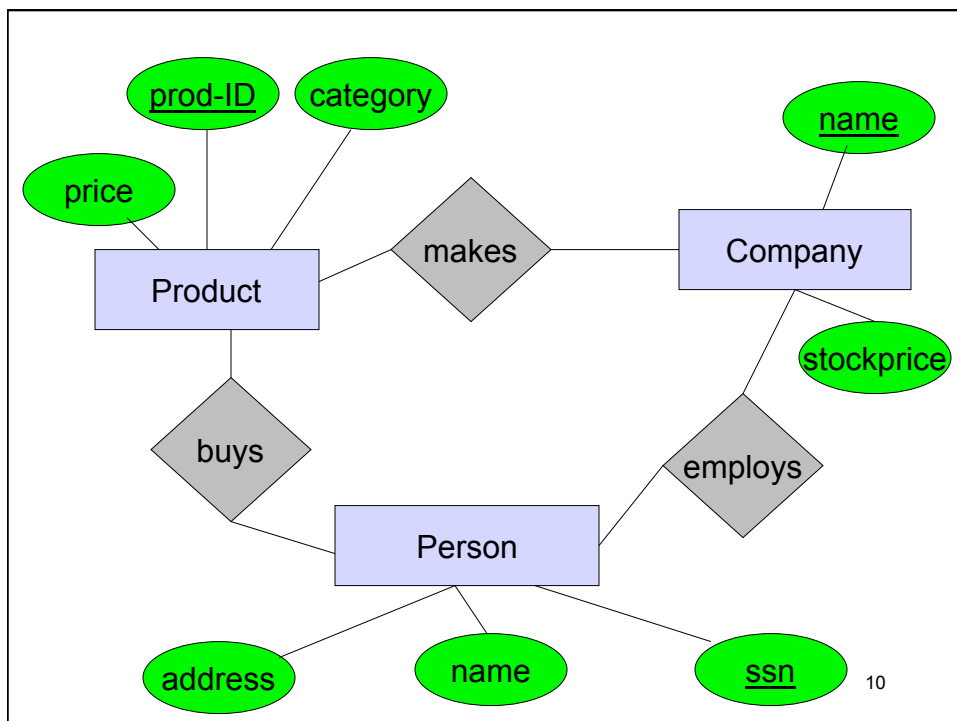
Attributes:

Relationships

- first class citizens (not associated with classes)
- not necessarily binary

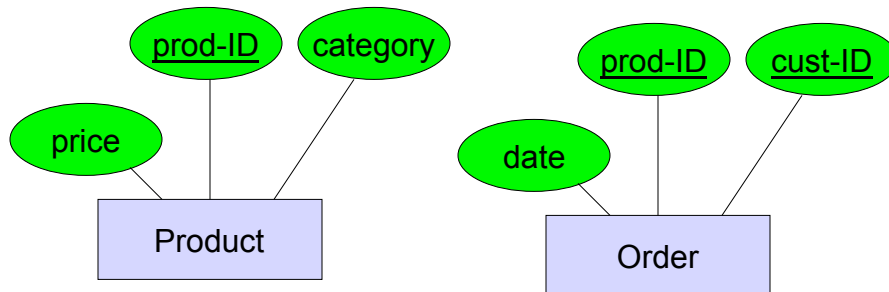


9



Keys in E/R Diagrams

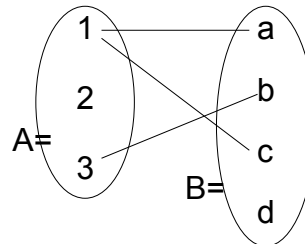
- Every entity set must have a key
- May be a *multi-attribute key*:



11

What is a Relation?

- A mathematical definition:
 - if A, B are sets, then a relation R is a subset of $A \times B$
- $A = \{1, 2, 3\}$, $B = \{a, b, c, d\}$,
 $A \times B = \{(1, a), (1, b), \dots, (3, d)\}$
 $R = \{(1, a), (1, c), (3, b)\}$



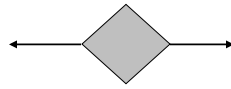
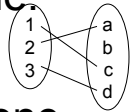
- **makes** is a subset of **Product** \times **Company**:



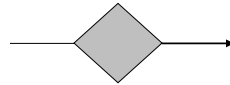
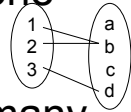
12

Multiplicity of E/R Relations

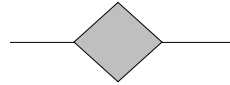
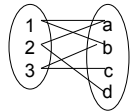
- one-one:



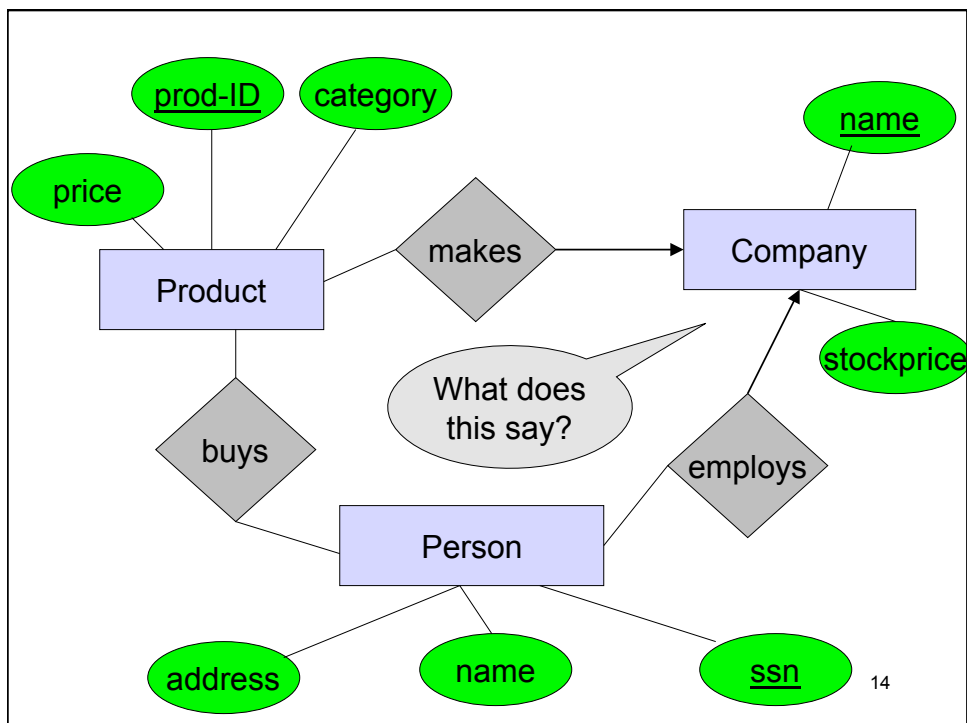
- many-one



- many-many

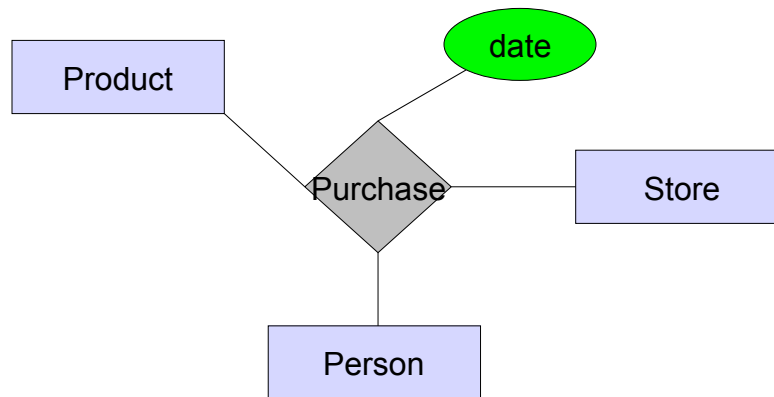


13



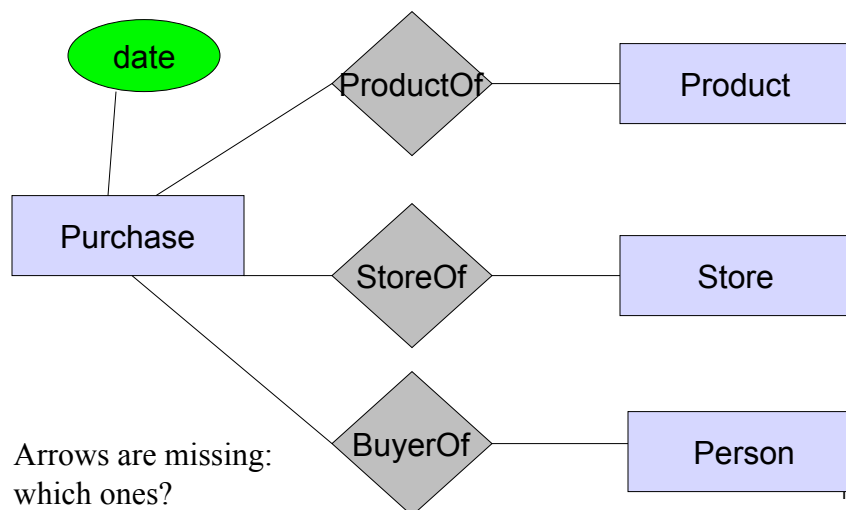
14

Multi-way Relationships



15

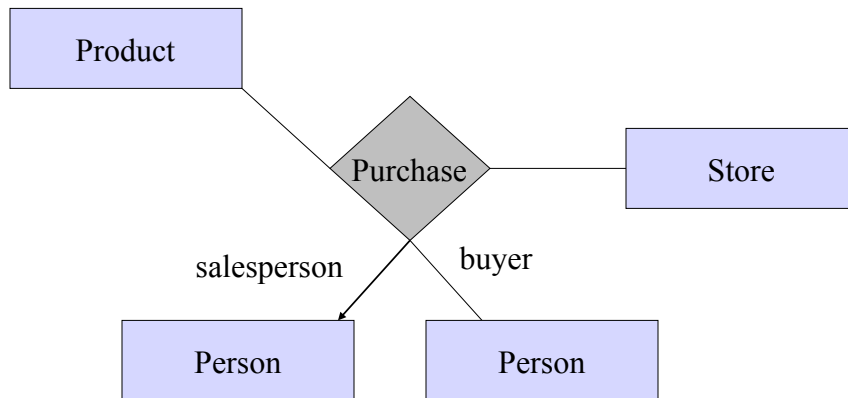
Converting Multi-way Relationships to Binary



10

Roles in Relationships

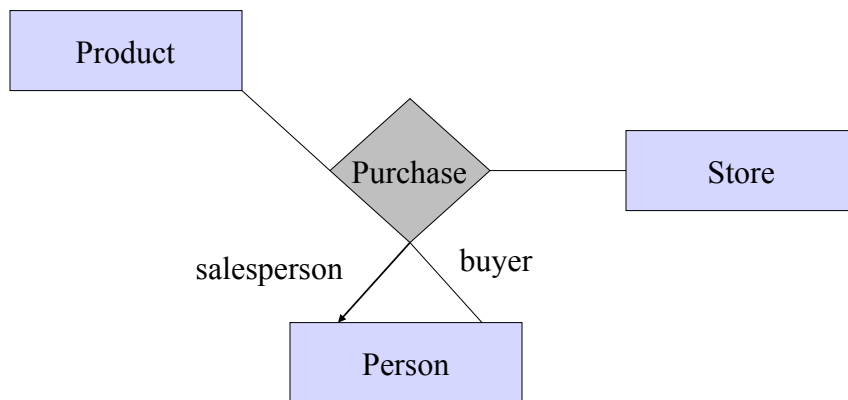
What if we need an entity set twice in one relationship?



17

Roles in Relationships

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



18

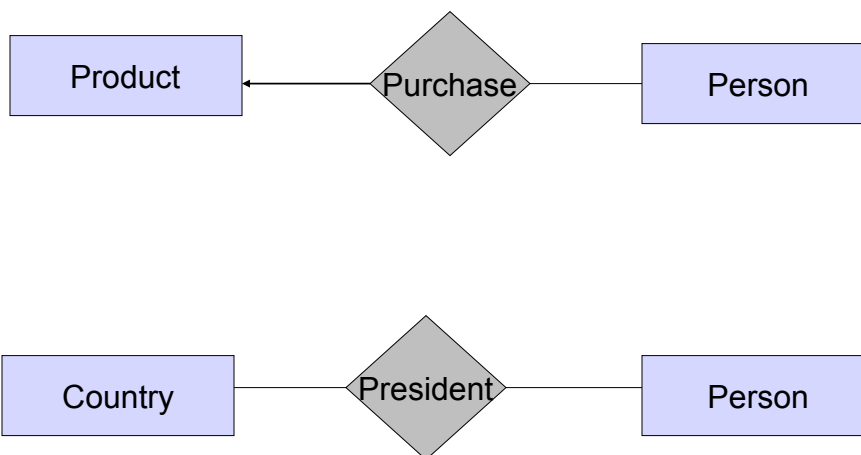
Design Principles

- Faithfulness: Entity sets and their attributes should reflect reality
- Avoid redundancy: say something only once
- Simplicity: Avoid introducing more elements into your design than is absolutely necessary.
- Don't use an entity set when an attribute will do
- Choose the right relationships
- Limit the use of weak entity sets (Section 4.4).

19

Design Principles

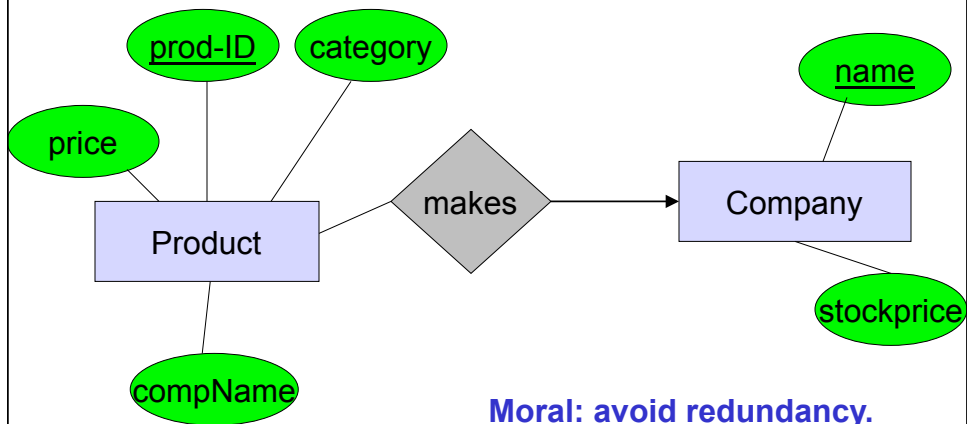
What's wrong?



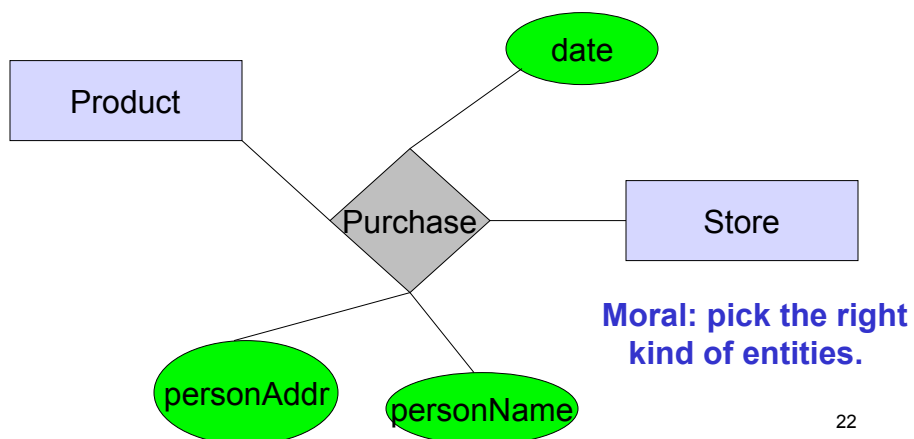
Moral: be faithful!

20

Design Principles: What's Wrong?

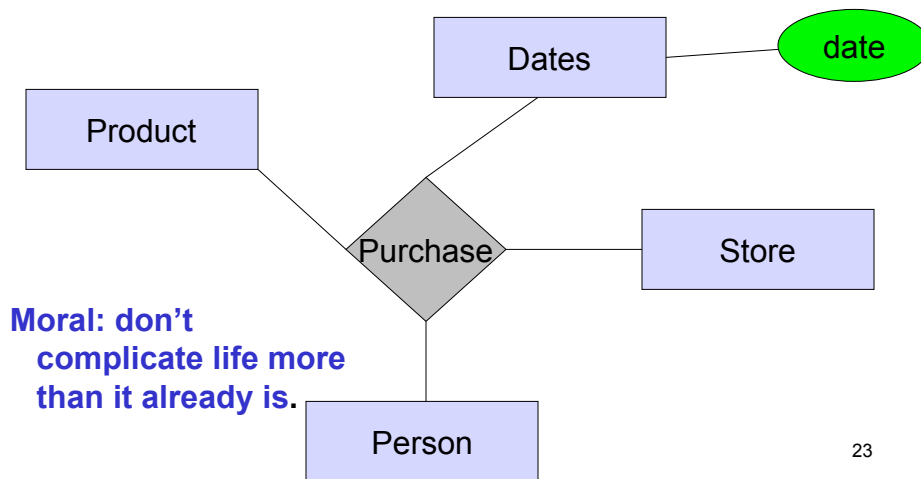


Design Principles: What's Wrong?

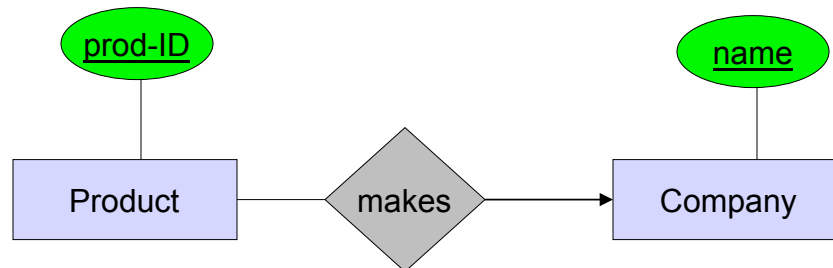


22

Design Principles: What's Wrong?



Design Principles: What's Wrong?



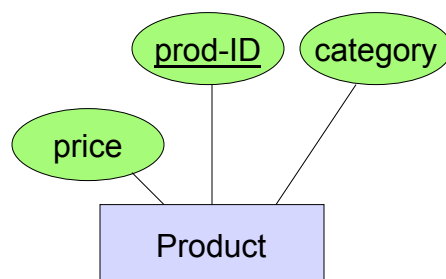
Moral: don't complicate life more than it already is.

From E/R Diagrams to Relational Schema

- Entity set E \rightarrow relation with attributes of E
- Relationship R \rightarrow relation with attributes being keys of related entity sets + attributes of R

25

Entity Set to Relation



Product(prod-ID, category, price)

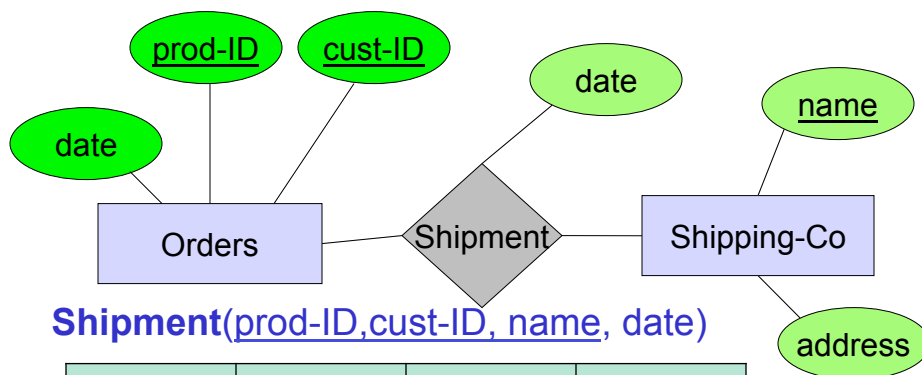
<u>prod-ID</u>	category	price
Gizmo55	Camera	99.99
Pokemn19	Toy	29.99

Create Table (SQL)

```
CREATE TABLE Product (  
  prod-ID CHAR(30) PRIMARY KEY,  
  category VARCHAR(20),  
  price double)
```

27

Relationships to Relations



Shipment(prod-ID, cust-ID, name, date)

<u>prod-ID</u>	<u>cust-ID</u>	<u>name</u>	date
Gizmo55	Joe12	UPS	4/10/2010
Gizmo55	Joe12	FEDEX	4/9/2010

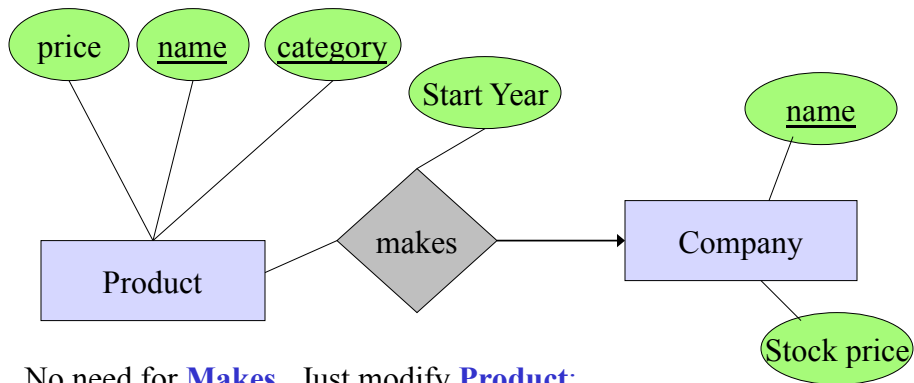
28

Create Table (SQL)

```
CREATE TABLE Shipment(
    name CHAR(30)
    REFERENCES Shipping-Co,
    prod-ID CHAR(30),
    cust-ID VARCHAR(20),
    date DATETIME,
    PRIMARY KEY (name, prod-ID, cust-ID),
    FOREIGN KEY (prod-ID, cust-ID)
    REFERENCES Orders
)
```

29

Many-One: Combining Two Relations



No need for **Makes**. Just modify **Product**:

<u>name</u>	<u>category</u>	<u>price</u>	<u>StartYear</u>	<u>companyName</u>
gizmo	gadgets	19.99	1963	gizmoWorks

30