L12: Relational modeling 2

CS3200 Database design (fa18 s2)

https://northeastern-datalab.github.io/cs3200/

Version 10/18/2018

Announcements!

- BB organization cleaned up, no more separate bumping columns
- Continue to come to OHs, read Piazza posts (see Gradiance comment), SAMS book, videos
- Feedback:
  - Great way to share your views, please keep it up!
  - Why the emphasis on in-class participation? ⊳ participation teaches an important skill for real life; ⊳ there are alternative ways to contribute: e.g., Piazza, "Optional PPTX"; ⊳ better than checking class attendance
  - Why do we learn about so many ERD notations? ⊳ preparation for real life not just test
- FM exam1:
  - SQL witnesses vs. limits
  - Q1c

Breaking the "bamboo ceiling" (cp. glass ceiling)

We want you to be successful in real life. Thus, incentives in this class are such to force you to get out of your comfort zone to speak up and to contribute.

"'The loudest duck gets shot' is a Chinese proverb. 'The nail that sticks out gets hammered down' is a Japanese one. Its Western correlative: 'The squeaky wheel gets the grease.'"
http://nymag.com/news/features/asian-americans-2011-5/
http://en.wikipedia.org/wiki/Bamboo_ceiling

Can you solve witnesses with top-1?

Product (pname, price, cid)

Q: Find the most expensive product + its price:

```
SELECT  P2.pname, P2.price
FROM    Product P2
WHERE   P2.price =
        (SELECT  max(P1.price)
        FROM     Product P1)
```

```
SELECT  pname, price
FROM    Product
ORDER BY price
LIMIT 1
```

Can you solve witnesses with top-1?

Product (pname, price, cid)

Q: Find the most expensive product + its price:

```
SELECT  P2.pname, P2.price
FROM    Product P2
WHERE   P2.price =
        (SELECT  max(P1.price)
        FROM     Product P1)
```

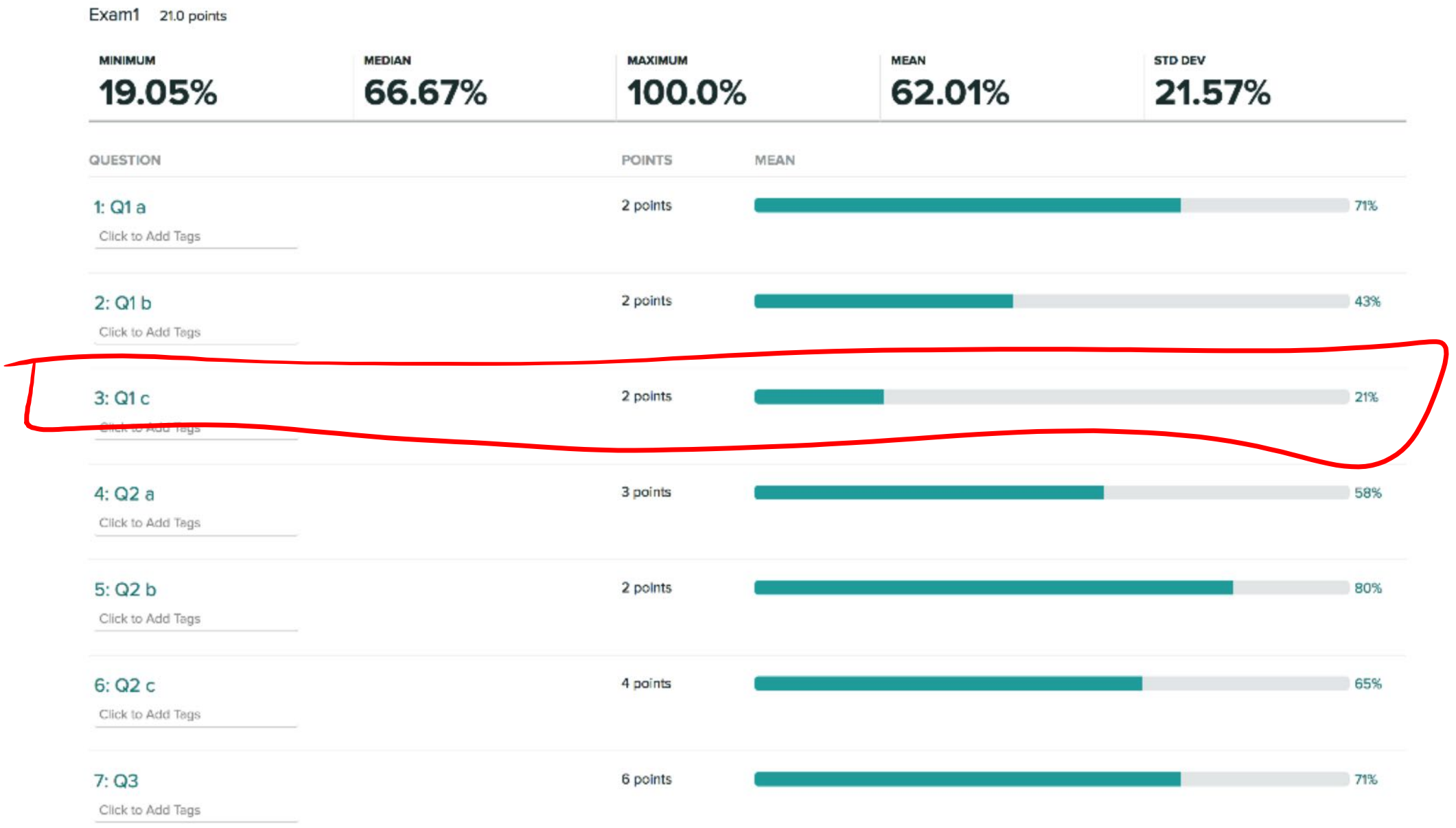
```
SELECT  pname, price
FROM    Product
ORDER BY price
LIMIT 1
```

Your query needs to return the correct result over *any* database

Product

PName	Price	cid
Gizmo	15	1
SuperGizmo	20	1
iTouch1	300	2
iTouch2	300	2

Q1-7 on exam 1 (gradescope interface)



Q1c

R

A	B
1	2
2	4
3	6
4	8

S

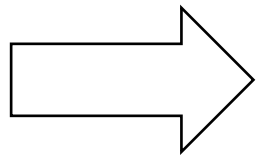
C	D
1	2
2	9
5	6
6	3

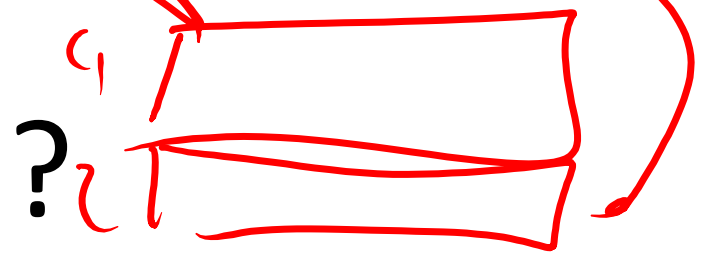
T

E	F
1	2
6	1
7	15
8	9

c. **(2 points)**

```
SELECT *
FROM R
UNION ALL (
        SELECT S.C, S.D
        FROM S
        JOIN T ON S.C=T.F)
```





Q1c

R

A	B
1	2
2	4
3	6
4	8

S

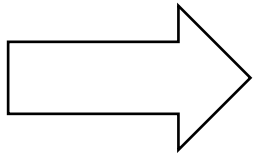
C	D
1	2
2	9
5	6
6	3

T

E	F
1	2
6	1
7	15
8	9

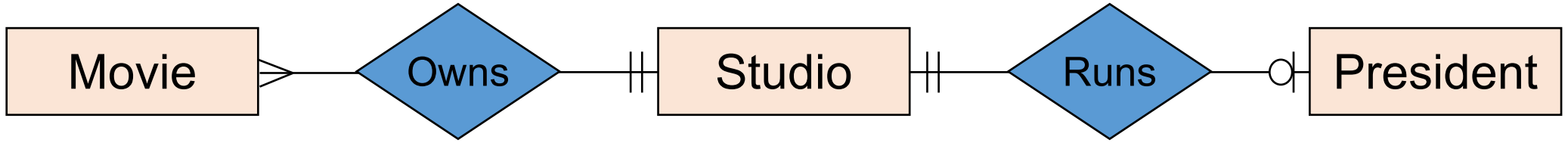
c. **(2 points)**

```
SELECT *
FROM R
UNION ALL (
        SELECT S.C, S.D
        FROM S
        JOIN T ON S.C=T.F)
```



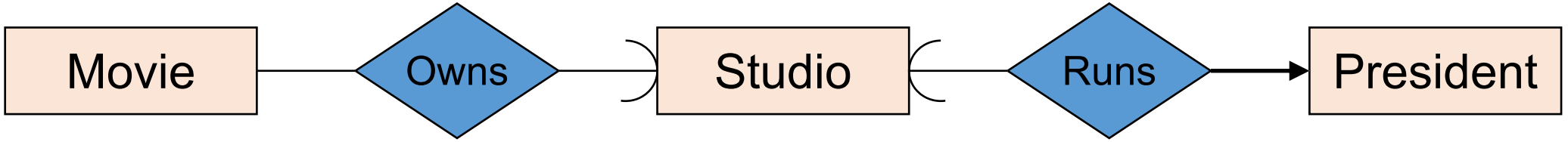
A	B
1	2
2	4
3	6
4	8
1	2
2	9

Read this ERD to your neighbor



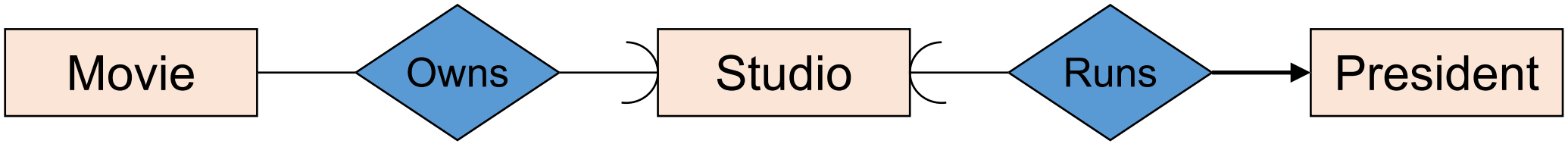
Read this ERD to your neighbor

same as ←



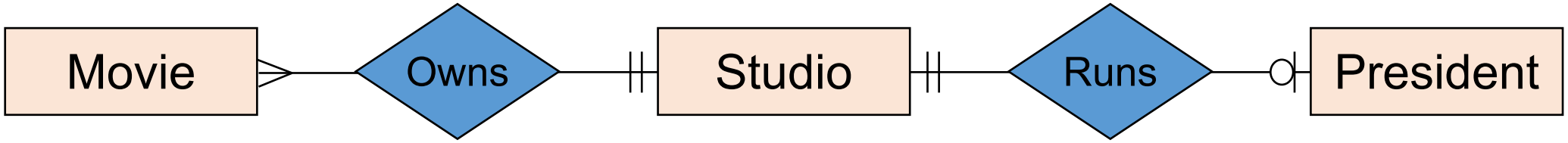
Two notations, same meaning

same as ←

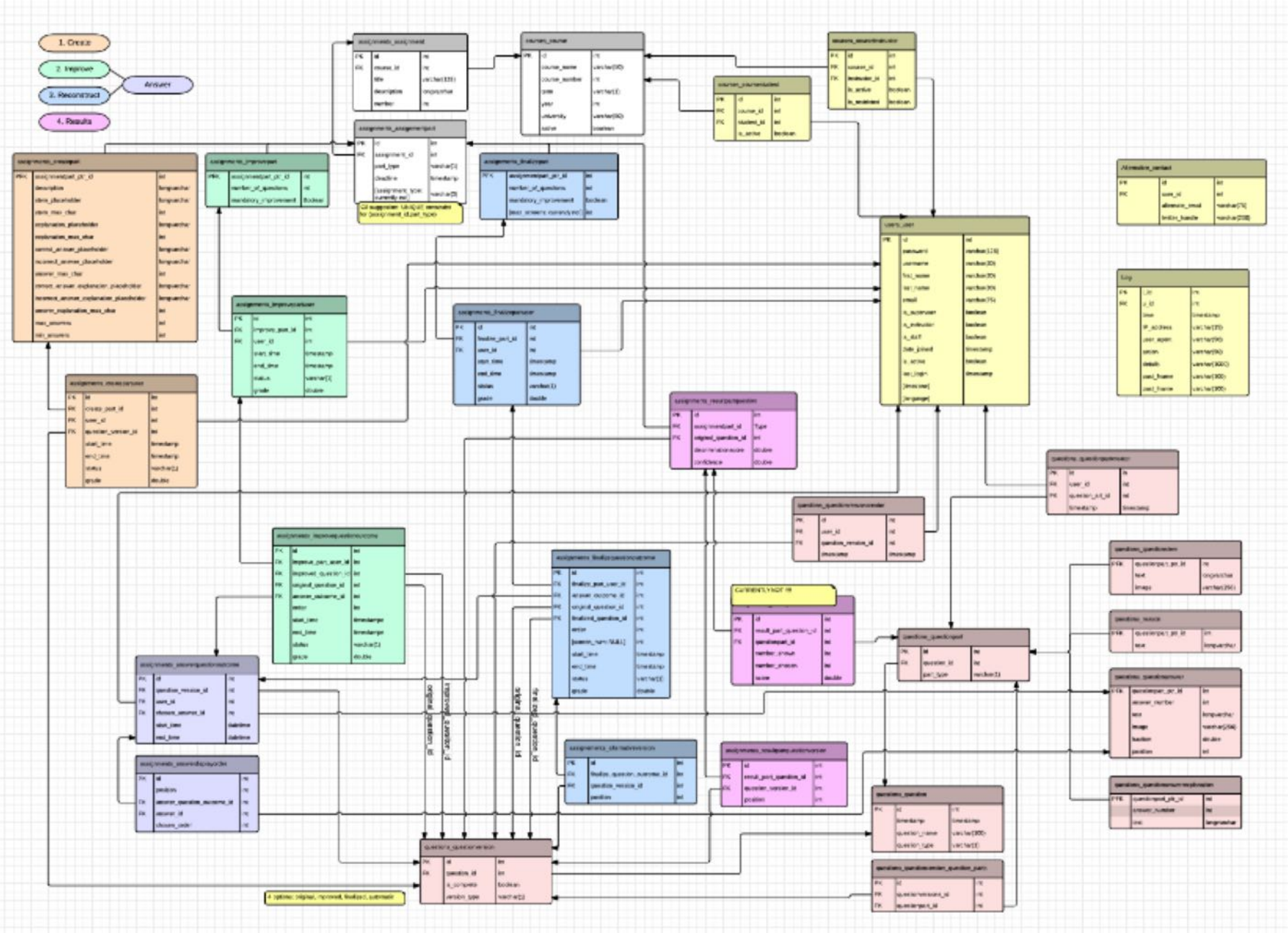


A studio can have at most one president

Each president must run exactly one studio (that exists in the studio entity set)



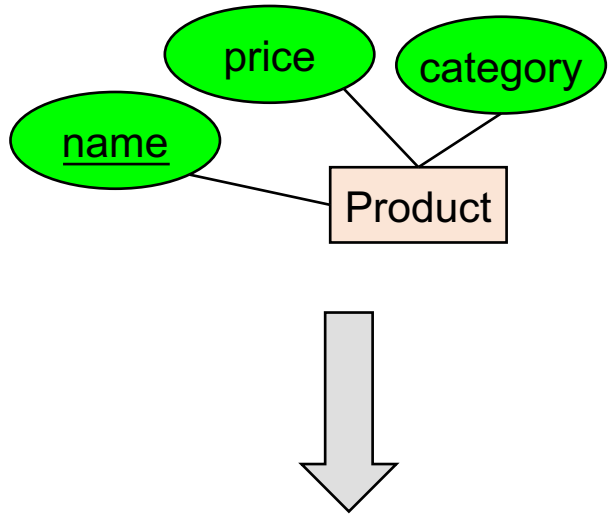
Why abstraction? Real relational schemas are huge



Back to "Relational modeling": From ERDs to Relations

From E/R Diagrams to Relational Schema

- An entity set becomes a relation (multiset of tuples / table)
  - Each tuple is one entity
  - Each tuple is composed of the entity's attributes, and has the same primary key

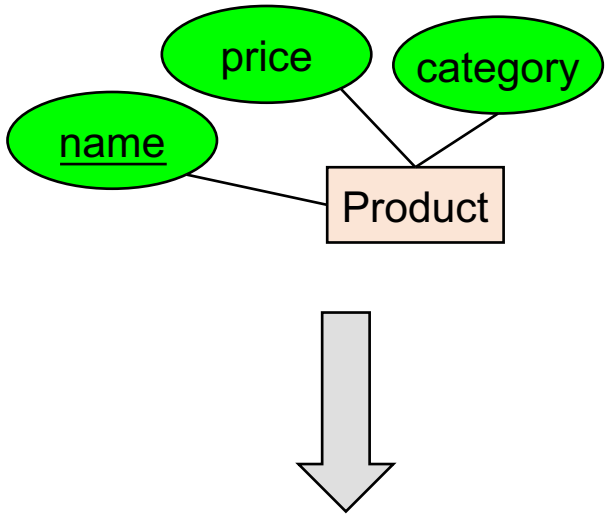


Product

name	price	category
Gizmo1	99.99	Camera
Gizmo2	19.99	Edible

From E/R Diagrams to Relational Schema

```
CREATE TABLE Product(
  name      CHAR(50) PRIMARY KEY,
  price     DECIMAL(8,2),
  category  VARCHAR(30)
)
```

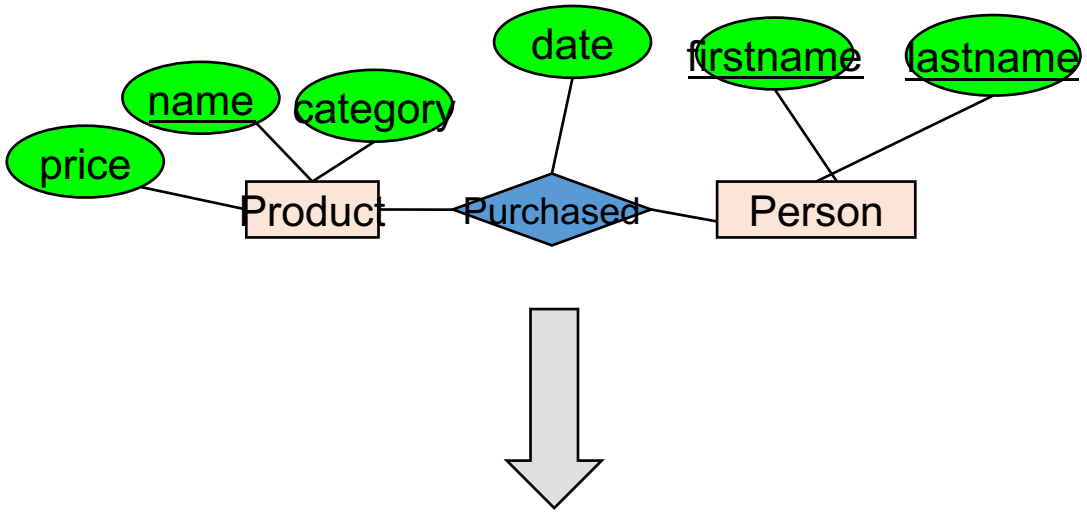


Product

name	price	category
Gizmo1	99.99	Camera
Gizmo2	19.99	Edible

From E/R Diagrams to Relational Schema

- A many-to-many relation between entity sets A_1, …, A_N *also* becomes a multiset of tuples / a table
  - Each row/tuple is one relation, i.e. one unique combination of entities $(a_1,…,a_N)$
  - Each row/tuple is
    - composed of the **union of the entity sets' keys**
    - has the entities' primary keys as foreign keys
    - has the union of the entity sets' keys as primary key

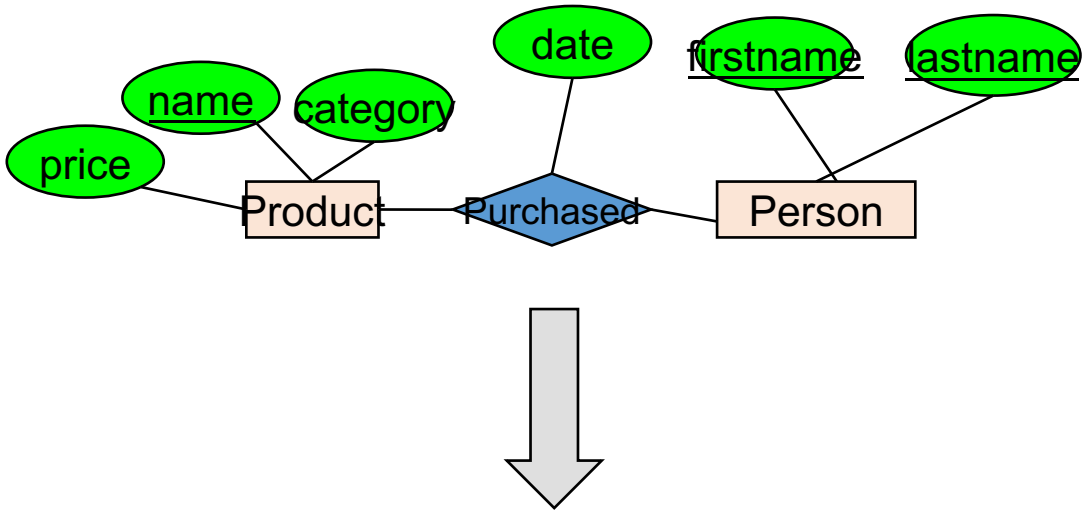


Purchased

name	firstname	lastname	date
Gizmo1	Bob	Joe	01/01/15
Gizmo2	Joe	Bob	01/03/15
Gizmo1	JoeBob	Smith	01/05/15

From E/R Diagrams to Relational Schema

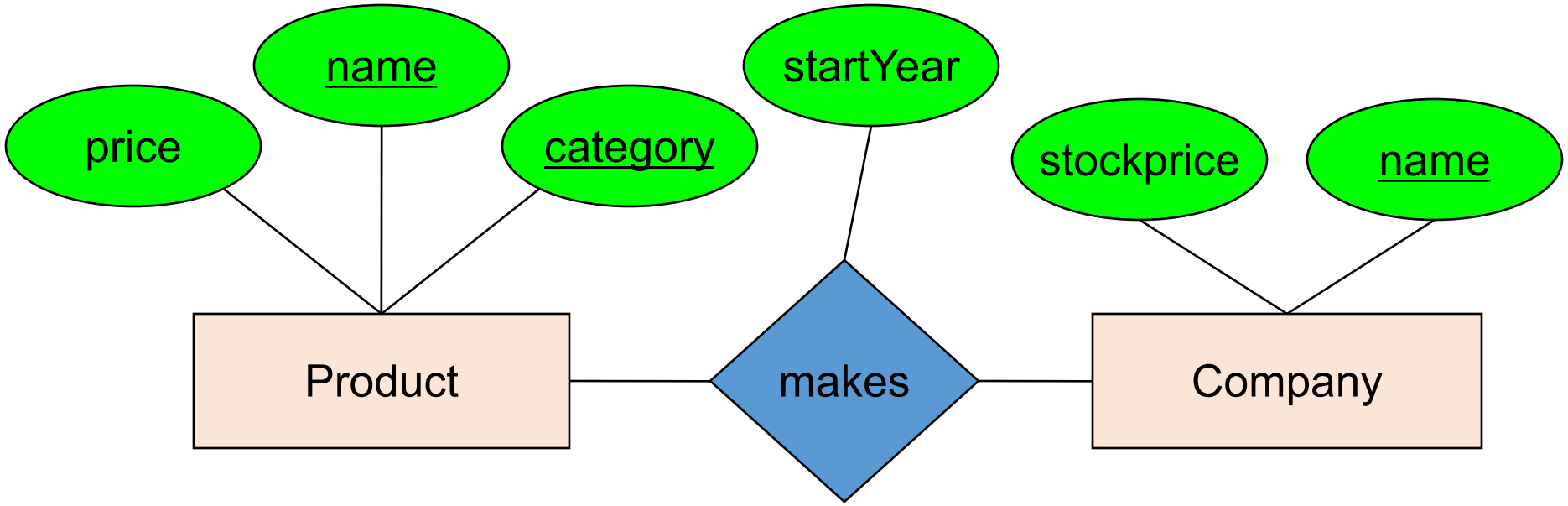
```
CREATE TABLE Purchased(
  name       CHAR(50),
  firstname  CHAR(50),
  lastname   CHAR(50),
  date       DATE,
  PRIMARY KEY (name, firstname, lastname),
  FOREIGN KEY (name)
      REFERENCES Product,
  FOREIGN KEY (firstname, lastname)
      REFERENCES Person
)
```



Purchased

name	firstname	lastname	date
Gizmo1	Bob	Joe	01/01/15
Gizmo2	Joe	Bob	01/03/15
Gizmo1	JoeBob	Smith	01/05/15

Relationships to Relations



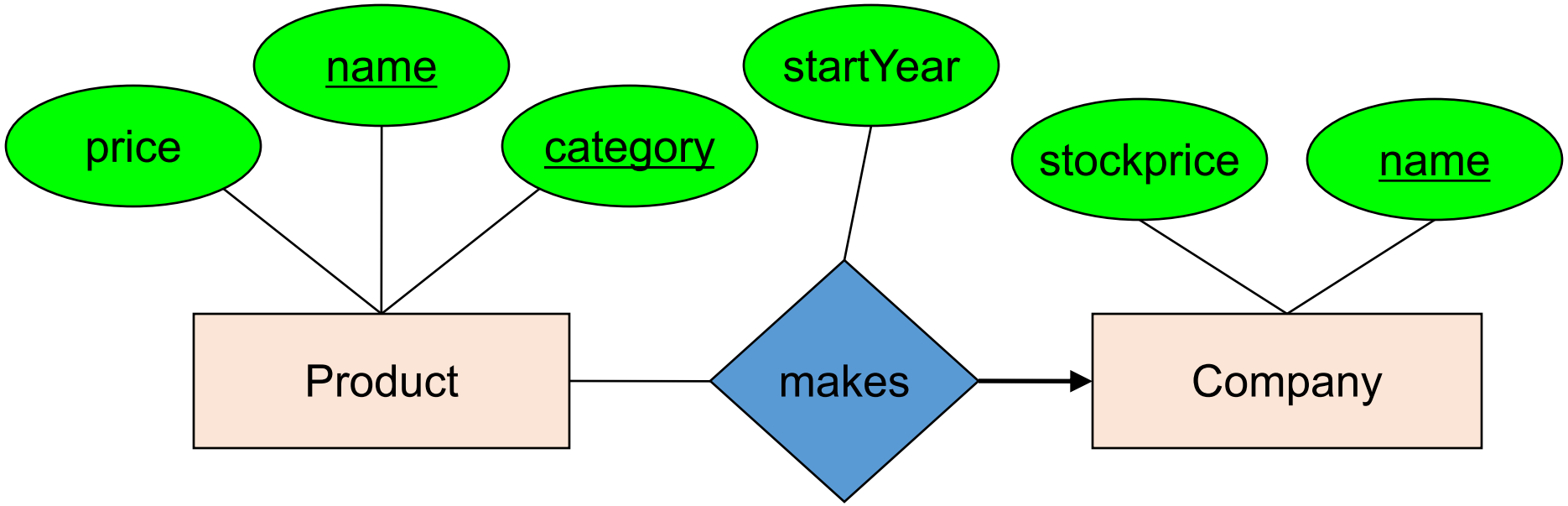
Watch out for attribute name conflicts

Makes(@productName, @category, @companyName, year)

Foreign keys

ProductName	ProductCategory	CompanyName	startYear
Gizmo	Gadgets	GizmoWorks	1963

Relationships to Relations (with constraints)



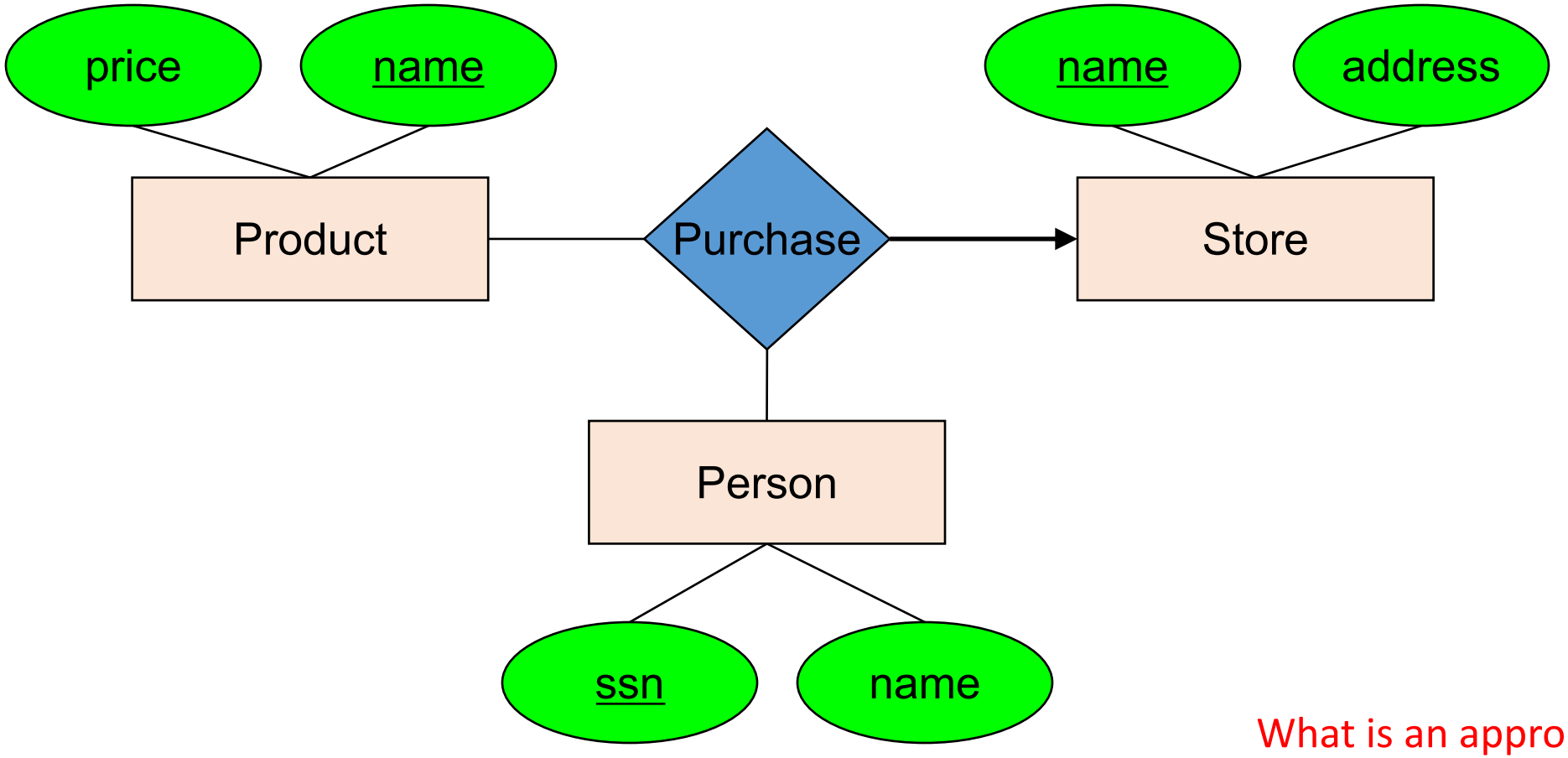
Only keep **Product** keys as primary key

Makes(@productName, @category, @companyName, year)

Better solution: get rid of **Makes**, modify **Product**:

prodName	Category	Price	startYear	CompanyName
Gizmo	Gadgets	$19.99	1963	GizmoWorks

Multi-way Relationships to Relations

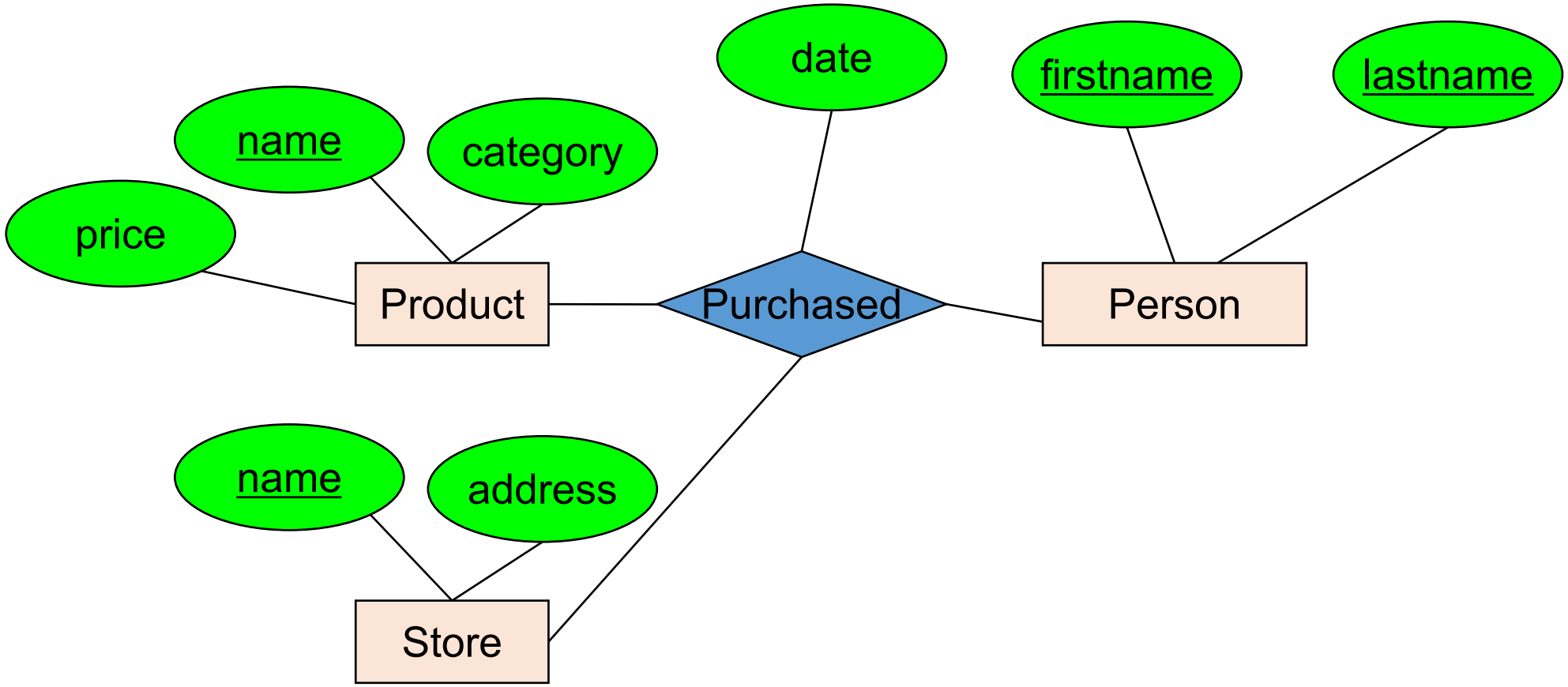


What is an appropriate key?

Purchase(prodName, storeName, ssn)

From E/R Diagram to Relational Schema

How do we represent this
as a relational schema?



In-Class Exercise (Part II): create an ERD

The following grade report below is mailed to students at the end of each semester. Prepare an ERD reflecting the data contained in the grade report (capturing Entities, Attributes, and Relationships). Assume that each course is taught by one instructor. Explain what you chose for the identifier of each entity type

MILLENNIUM COLLEGE
GRADE REPORT
FALL SEMESTER 200X

NAME: Emily Williams ID: 268300458
CAMPUS ADDRESS: 208 Brooks Hall
MAJOR: Information Systems

COURSE ID	TITLE	INSTRUCTOR NAME	INSTRUCTOR LOCATION	GRADE
IS 350	Database Mgt.	Codd	B104	A
IS 465	System Analysis	Parsons	B317	B

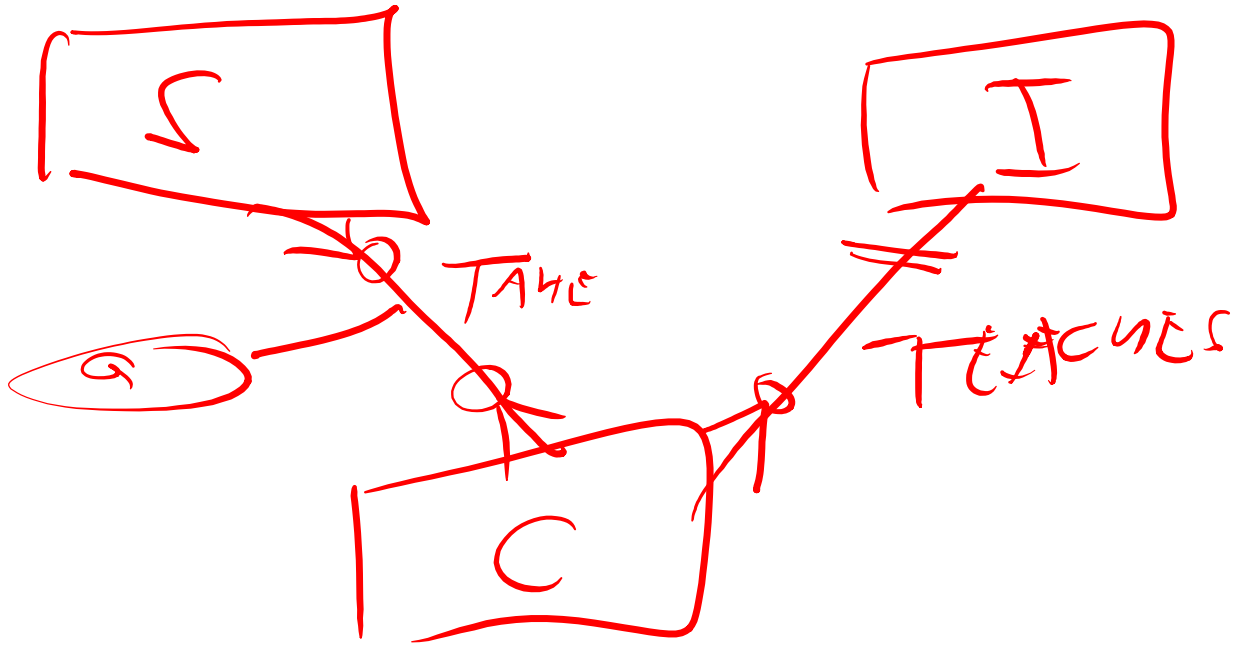
Source: Hoffer, Ramesh, Topi, "Modern database management," 10th ed, 2010.

In-Class Exercise (Part III): create an ERD

MILLENNIUM COLLEGE
GRADE REPORT
FALL SEMESTER 200X

NAME: Emily Williams ID: 268300458
CAMPUS ADDRESS: 208 Brooks Hall
MAJOR: Information Systems

COURSE ID	TITLE	INSTRUCTOR NAME	INSTRUCTOR LOCATION	GRADE
IS 350	Database Mgt.	Codd	B104	A
IS 465	System Analysis	Parsons	B317	B



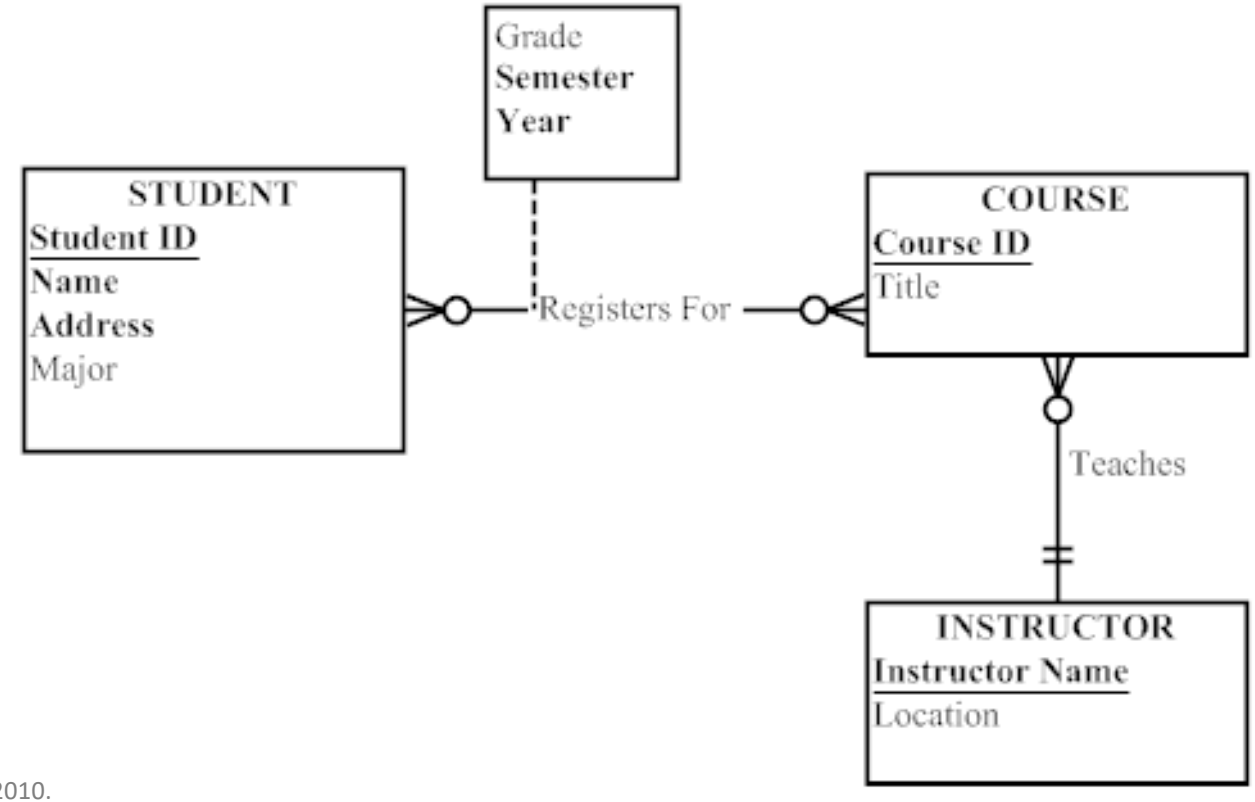
Source: Hoffer, Ramesh, Topi, "Modern database management," 10th ed, 2010.

In-Class Exercise (Part III): create an ERD

MILLENNIUM COLLEGE
GRADE REPORT
FALL SEMESTER 200X

NAME: Emily Williams ID: 268300458
CAMPUS ADDRESS: 208 Brooks Hall
MAJOR: Information Systems

COURSE ID	TITLE	INSTRUCTOR NAME	INSTRUCTOR LOCATION	GRADE
IS 350	Database Mgt.	Codd	B104	A
IS 465	System Analysis	Parsons	B317	B



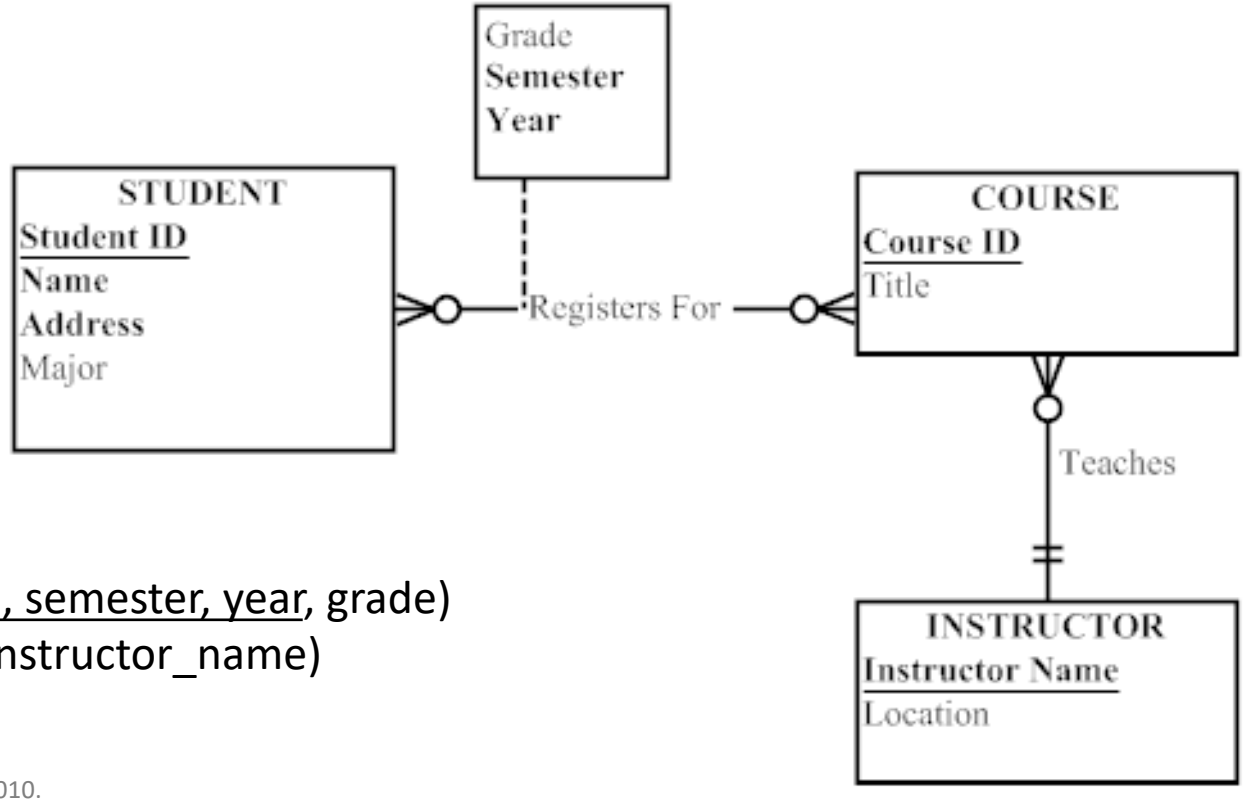
Source: Hoffer, Ramesh, Topi, "Modern database management," 10th ed, 2010.

In-Class Exercise (Part III): relational schema

MILLENNIUM COLLEGE
GRADE REPORT
FALL SEMESTER 200X

NAME: Emily Williams ID: 268300458
CAMPUS ADDRESS: 208 Brooks Hall
MAJOR: Information Systems

COURSE ID	TITLE	INSTRUCTOR NAME	INSTRUCTOR LOCATION	GRADE
IS 350	Database Mgt.	Codd	B104	A
IS 465	System Analysis	Parsons	B317	B



Register(@sid, @cid, semester, year, grade)
Course(cid, title, @instructor_name)

Source: Hoffer, Ramesh, Topi, "Modern database management," 10th ed, 2010.

Relational Modeling in more detail and with some repetition: Entities & Attributes

Relations

- A table consists of rows (records), and columns (attributes/fields)
- A relation is a named, two-dimensional table of data
- Six requirements for a table to qualify as a relation:
  1. The table must have a unique name.
  2. Columns (attributes) in tables must have unique names
  3. Every attribute value must be atomic (not multivalued, not composite)
  4. Every row must be unique (can't have two rows with exactly the same values for all their columns)
  5. The order of the columns must be irrelevant
     1. A(id, name) vs. A(name, id)
  6. The order of the rows must be irrelevant

Mapping ER Models To Relations

- Relations (tables) correspond to entity types and to many-to-many relationship types
- Rows correspond to entity instances and to many-to-many relationship instances
- Columns correspond to attributes

- relation (in relational database) ≠ relationship (in E-R model)

EMPLOYEE1

Emp_ID	Name	Dept_Name	Salary
100	Margaret Simpson	Marketing	48,000
140	Allen Beeton	Accounting	52,000
110	Chris Lucero	Info Systems	43,000
190	Lorenzo Davis	Finance	55,000
150	Susan Martin	Marketing	42,000

Relation Notation

- Here are two common notations for describing relations:

- Text statements – RELATION_NAME(attributes)
  - CUSTOMER(Customer_ID, Name, Address, City, State)
  - ORDER(Order_ID, Order_Date, Product_ID)

- Horizontal or Vertical graphical notation:

Key Fields

- Keys are special fields used to uniquely identify relations

- Primary keys are unique identifiers of the relation in question
  - Primary keys guarantee that all rows are unique
  - Examples
    - Employee ID numbers
    - Social security numbers
    - E-mail addresses

- Foreign keys are identifiers that refer to other primary keys
  - Useful as references

- Keys can be simple (single field) or composite (multiple fields)
- Keys are often used as indexes to speed up user queries

Mapping Regular Entities to Relations

- Simple attributes:
  - E-R attributes map directly onto the relation

- Composite attributes:
  - Use only their simple, component attributes

- Multivalued Attribute
  - Becomes a separate relation with a foreign key taken from the superior entity

Map Simple ER Attributes Directly Onto Relation

CUSTOMER entity type with simple attributes

Map Simple ER Attributes Directly Onto Relation

CUSTOMER entity type with simple attributes

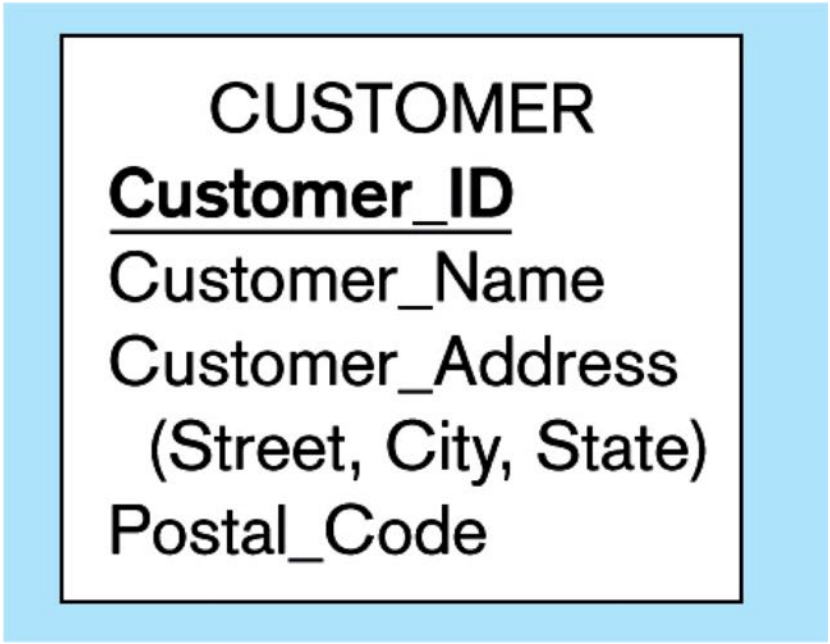
CUSTOMER relation

CUSTOMER

Customer_ID	Customer_Name	Customer_Address	Postal_Code

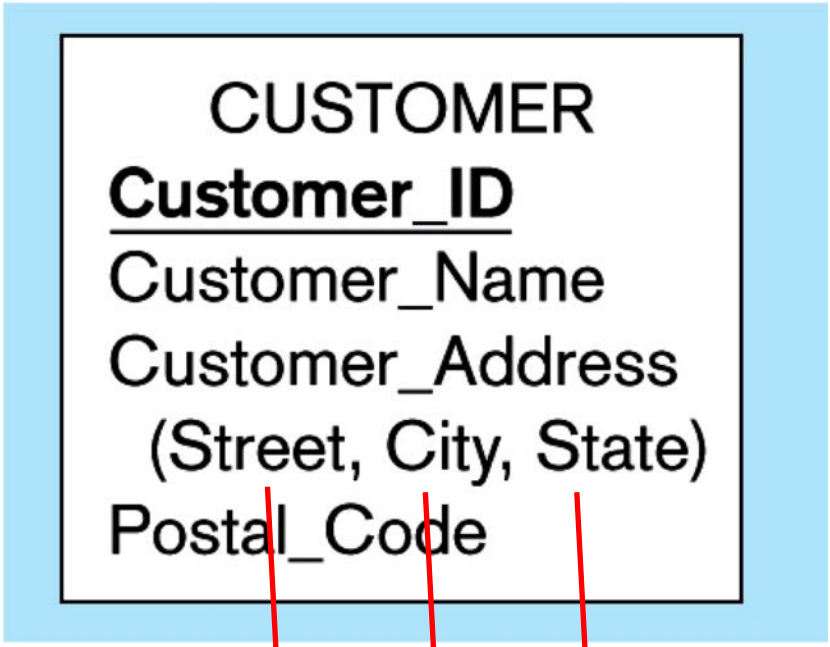
Example: Mapping A Composite Attribute

CUSTOMER entity type with composite attribute

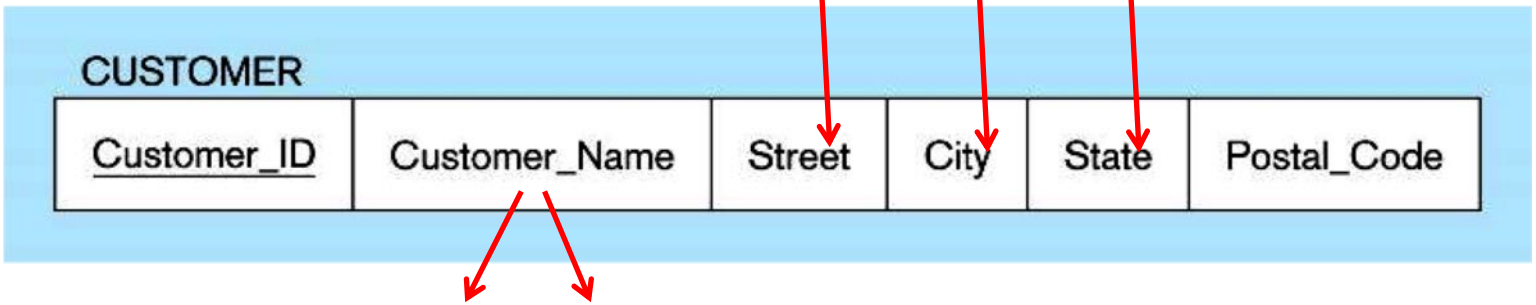


Example: Mapping A Composite Attribute

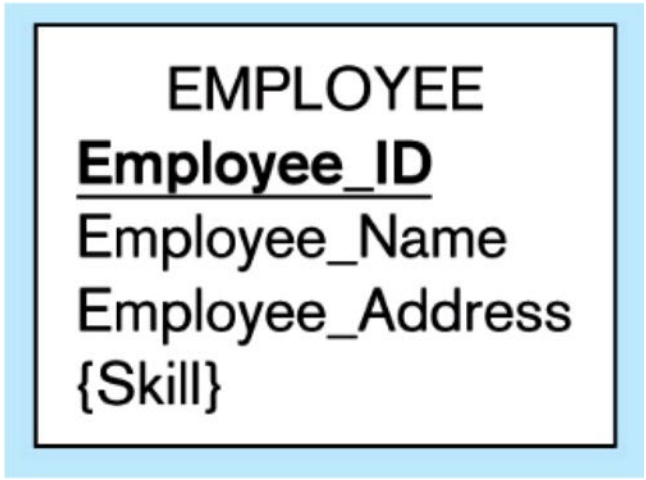
CUSTOMER entity type with composite attribute



CUSTOMER relation with address detail

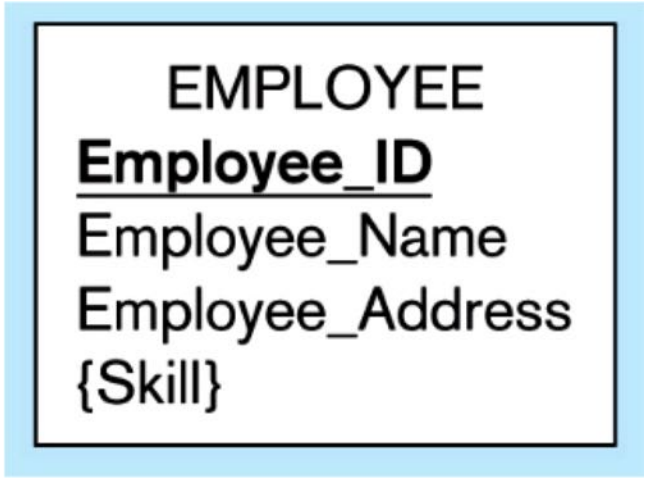


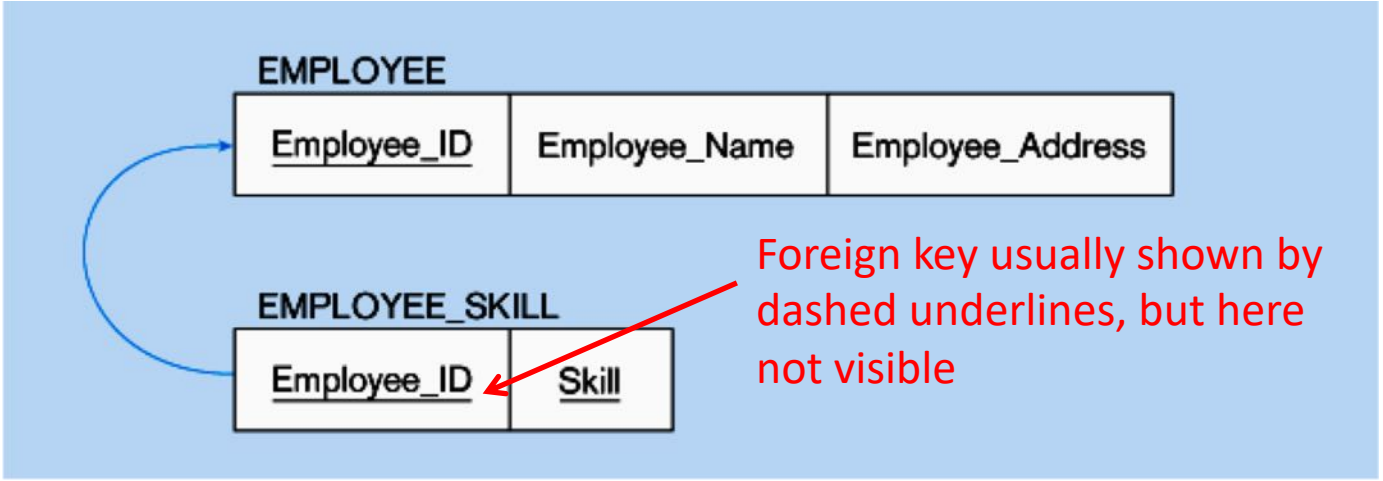
Example: Mapping A Multivalued Attribute



Example: Mapping A Multivalued Attribute

Multivalued Attribute becomes a separate relation with foreign key





1–to–many relationship between original entity and new relation

Relational Modeling: Relationships

Mapping Binary Relationships

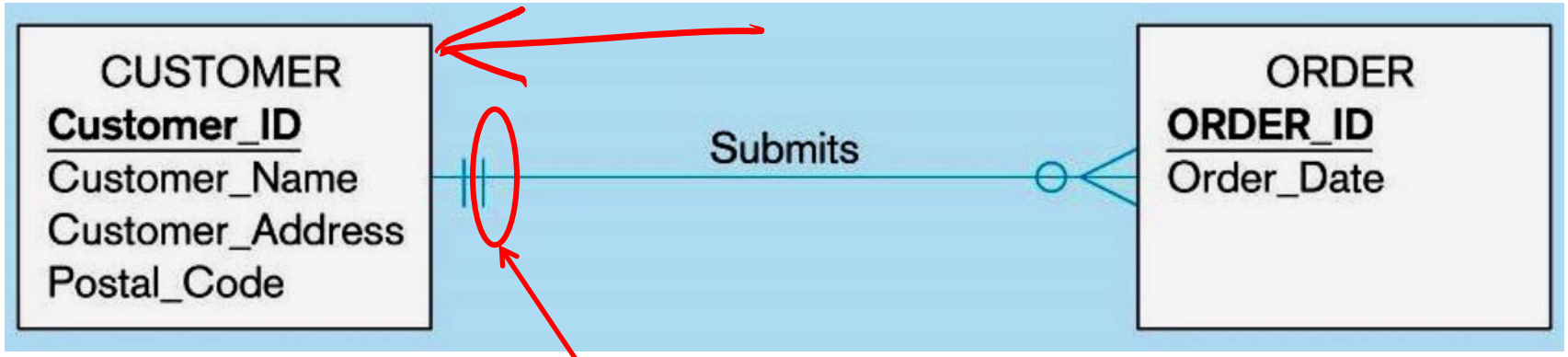
- 1. **One-to-Many**: Primary key on the one side becomes a foreign key on the many side

- 2. **Many-to-Many**: Create a new relation with the primary keys of the two entities as its primary key

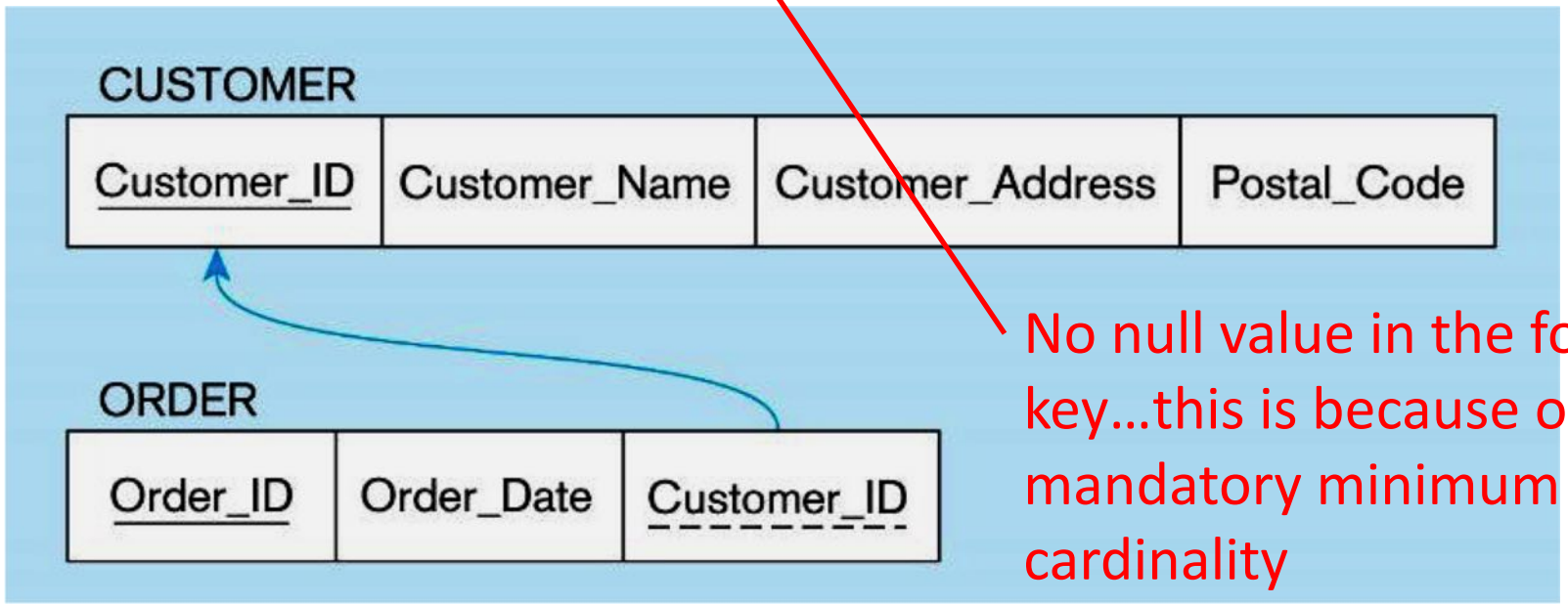
- 3. **One-to-One**: Primary key on the mandatory side becomes a foreign key on the optional side

1) Mapping a 1:M Relationship

1) Mapping a 1:M Relationship



Rule: Primary key on the one side becomes a foreign key on the many side



No null value in the foreign key…this is because of the mandatory minimum cardinality

Referential integrity constraints and NULL

SQLQuery12.sql - j..._Assignment (1220))* | SQLQuery11.sql - (...8OR\Wolfgang (59))

```
select *
from product;
```

```
--------------------------
-- Create the tables
--------------------------
create table Company (
    CName char(20) PRIMARY KEY,
    StockPrice int,
    Country char(20) );

create table Product (
    PName char(20),
    Price decimal(9, 2),
    Category char(20),
    Manufacturer char(20),
PRIMARY KEY (PName),
FOREIGN KEY (Manufacturer) REFERENCES Company(CName) );
```

Results | Messages

	PName	Price	Category	Manufacturer
1	Gizmo	19.99	Gadgets	GizmoWorks
2	MultiTouch	203.99	Household	Hitachi
3	PowerGizmo	29.99	Gadgets	GizmoWorks
4	SingleTouch	149.99	Photography	Canon

Referential integrity constraints and NULL

SQLQuery12.sql - j..._Assignment (1220))* | SQLQuery11.sql - (...8OR\Wolfgang (59))

```
select *
from product;

insert into product values('hallo', 10, 'Gadgets', NULL);
```

```
--------------------------
-- Create the tables
--------------------------
create table Company (
    CName char(20) PRIMARY KEY,
    StockPrice int,
    Country char(20) );

create table Product (
    PName char(20),
    Price decimal(9, 2),
    Category char(20),
    Manufacturer char(20),
PRIMARY KEY (PName),
FOREIGN KEY (Manufacturer) REFERENCES Company(CName) );
```

Results | Messages

	PName	Price	Category	Manufacturer
1	Gizmo	19.99	Gadgets	GizmoWorks
2	MultiTouch	203.99	Household	Hitachi
3	PowerGizmo	29.99	Gadgets	GizmoWorks
4	SingleTouch	149.99	Photography	Canon

Referential integrity constraints and NULL

SQLQuery12.sql - j..._Assignment (1220))* | SQLQuery11.sql - (...8OR\Wolfgang (59))

```
select *
from product;

insert into product values('hallo', 10, 'Gadgets', NULL);

select *
from product;

delete from product
where manufacturer is null;
```

100 %

Results | Messages

	PName	Price	Category	Manufacturer
1	Gizmo	19.99	Gadgets	GizmoWorks
2	MultiTouch	203.99	Household	Hitachi
3	PowerGizmo	29.99	Gadgets	GizmoWorks
4	SingleTouch	149.99	Photography	Canon

	PName	Price	Category	Manufacturer
1	Gizmo	19.99	Gadgets	GizmoWorks
2	hallo	10.00	Gadgets	NULL
3	MultiTouch	203.99	Household	Hitachi
4	PowerGizmo	29.99	Gadgets	GizmoWorks
5	SingleTouch	149.99	Photography	Canon

```
--------------------------
-- Create the tables
--------------------------
create table Company (
    CName char(20) PRIMARY KEY,
    StockPrice int,
    Country char(20) );

create table Product (
    PName char(20),
    Price decimal(9, 2),
    Category char(20),
    Manufacturer char(20),
PRIMARY KEY (PName),
FOREIGN KEY (Manufacturer) REFERENCES Company(CName) );
```

Referential integrity constraints and NULL

SQLQuery12.sql - j..._Assignment (1220))* | SQLQuery11.sql - (...8OR\Wolfgang (59))

```
select *
from product;

insert into product values('hallo', 10, 'Gadgets', NULL);

select *
from product;

delete from product
where manufacturer is null;
```

100 %

Results | Messages

	PName	Price	Category	Manufacturer
1	Gizmo	19.99	Gadgets	GizmoWorks
2	MultiTouch	203.99	Household	Hitachi
3	PowerGizmo	29.99	Gadgets	GizmoWorks
4	SingleTouch	149.99	Photography	Canon

	PName	Price	Category	Manufacturer
1	Gizmo	19.99	Gadgets	GizmoWorks
2	hallo	10.00	Gadgets	NULL
3	MultiTouch	203.99	Household	Hitachi
4	PowerGizmo	29.99	Gadgets	GizmoWorks
5	SingleTouch	149.99	Photography	Canon

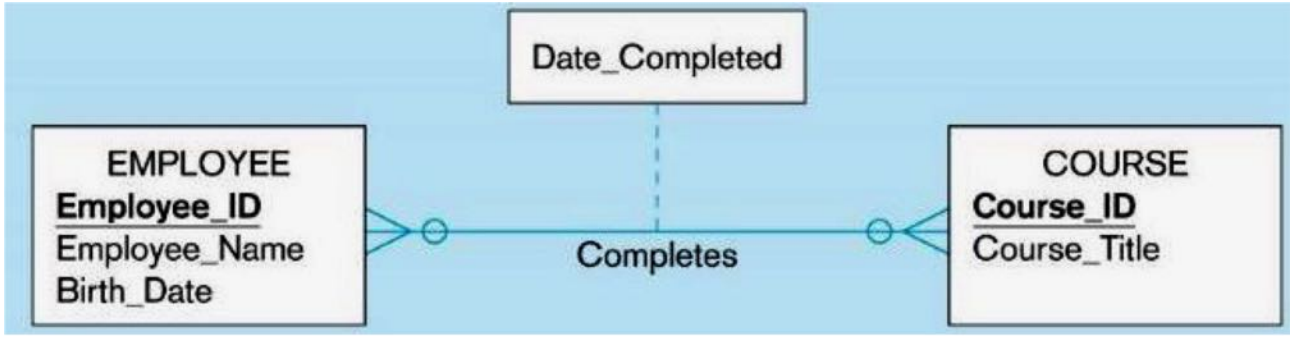
```
--------------------------
-- Create the tables
--------------------------
create table Company (
    CName char(20) PRIMARY KEY,
    StockPrice int,
    Country char(20) );

create table Product (
    PName char(20),
    Price decimal(9, 2),
    Category char(20),
    Manufacturer char(20),
PRIMARY KEY (PName),
FOREIGN KEY (Manufacturer) REFERENCES Company(CName) );
```

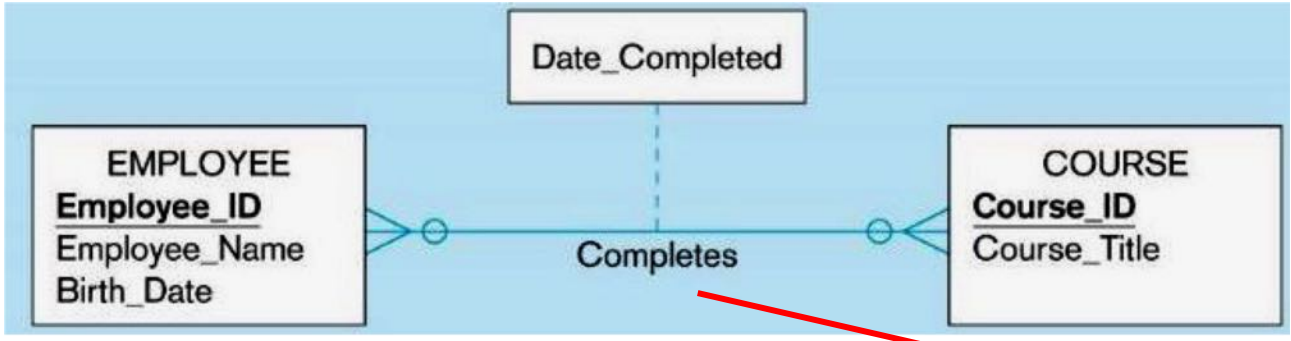
What you want:

```
create table Product (
    PName char(20),
    Price decimal(9, 2),
    Category char(20),
    Manufacturer char(20) not null,
PRIMARY KEY (PName),
FOREIGN KEY (Manufacturer) REFERENCES Company(CName) );
```

2) Mapping An M:N Relationship

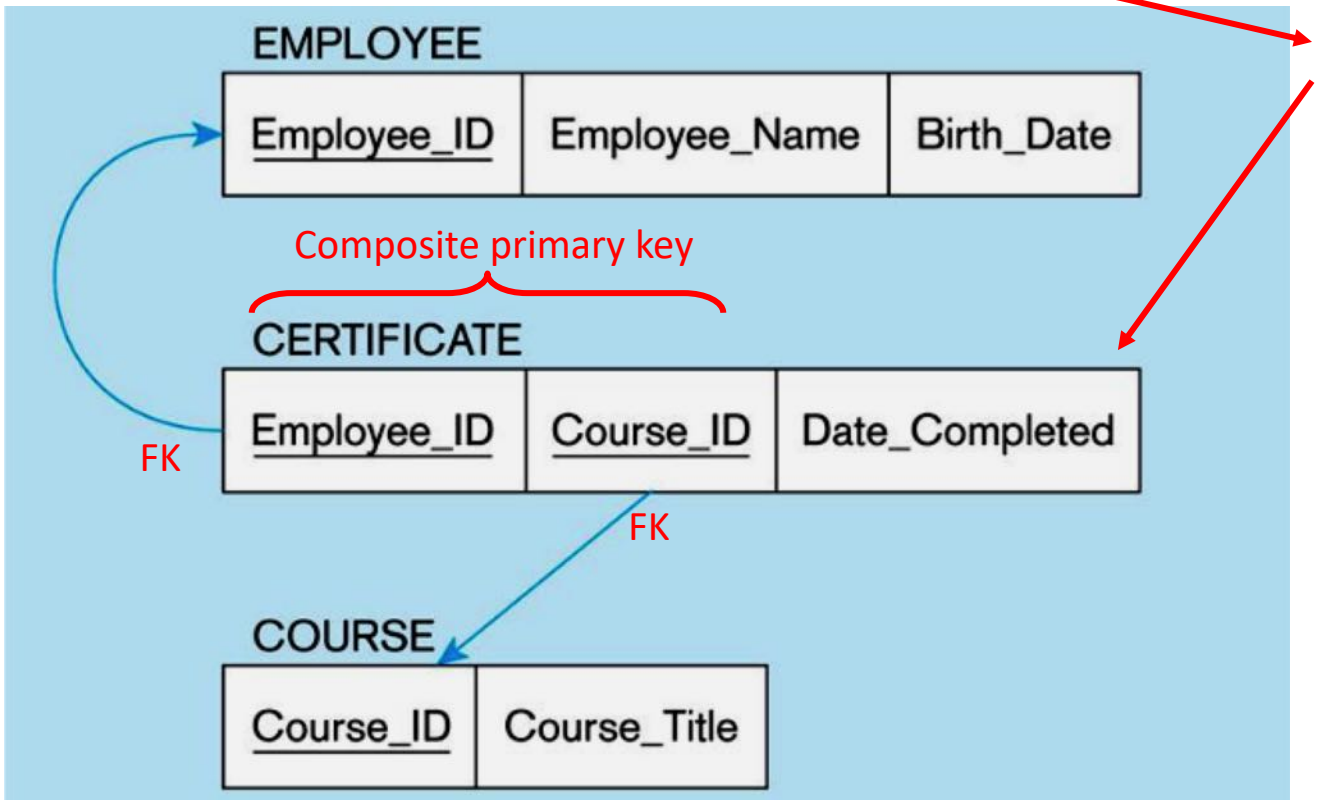


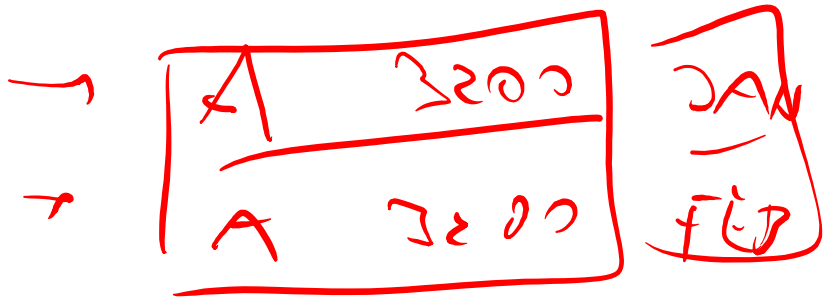
2) Mapping An M:N Relationship



The *Completes* relationship will need to become a separate relation

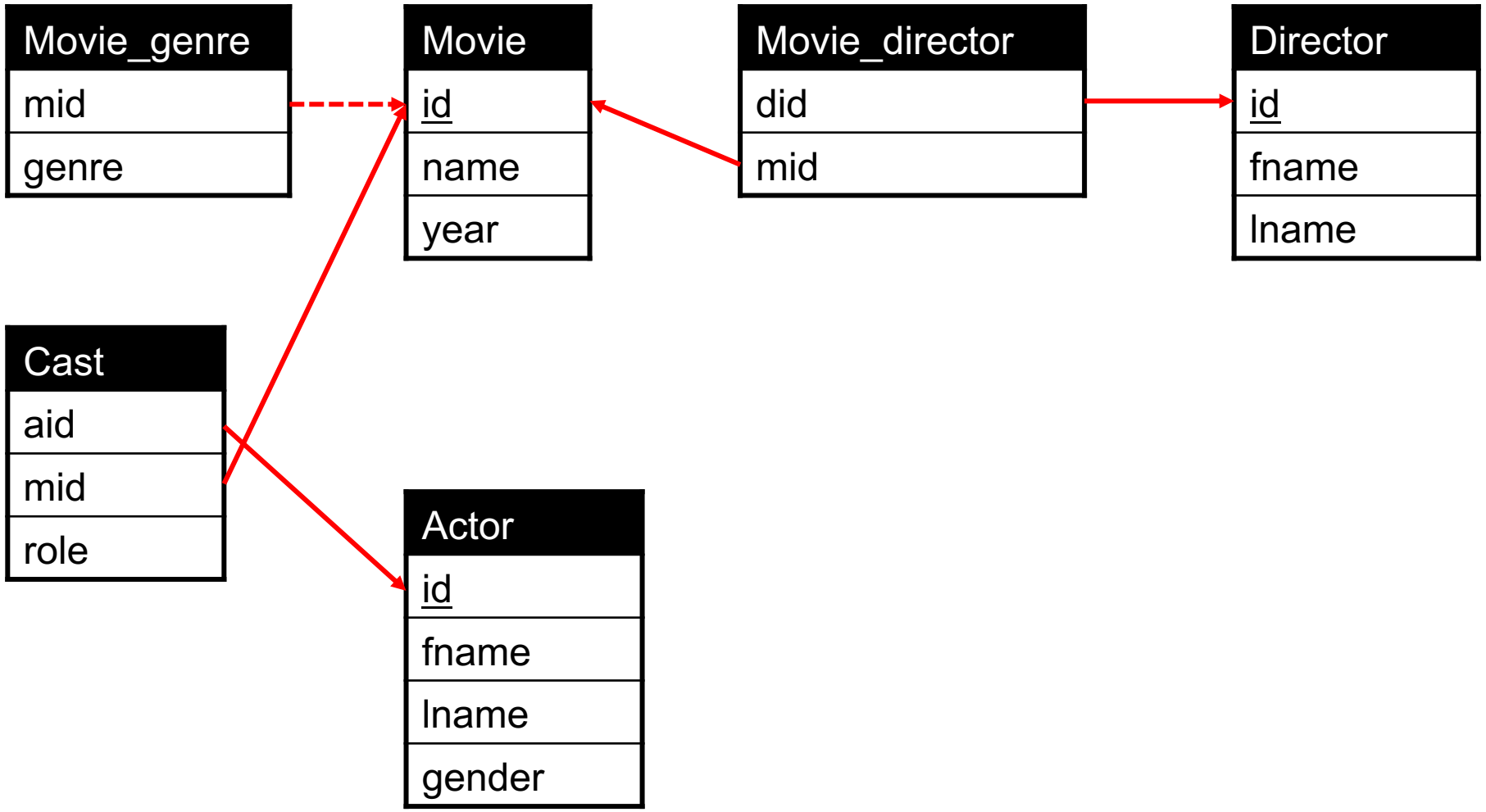
Rule: Create a new relation with the primary keys of the two entities as its primary key



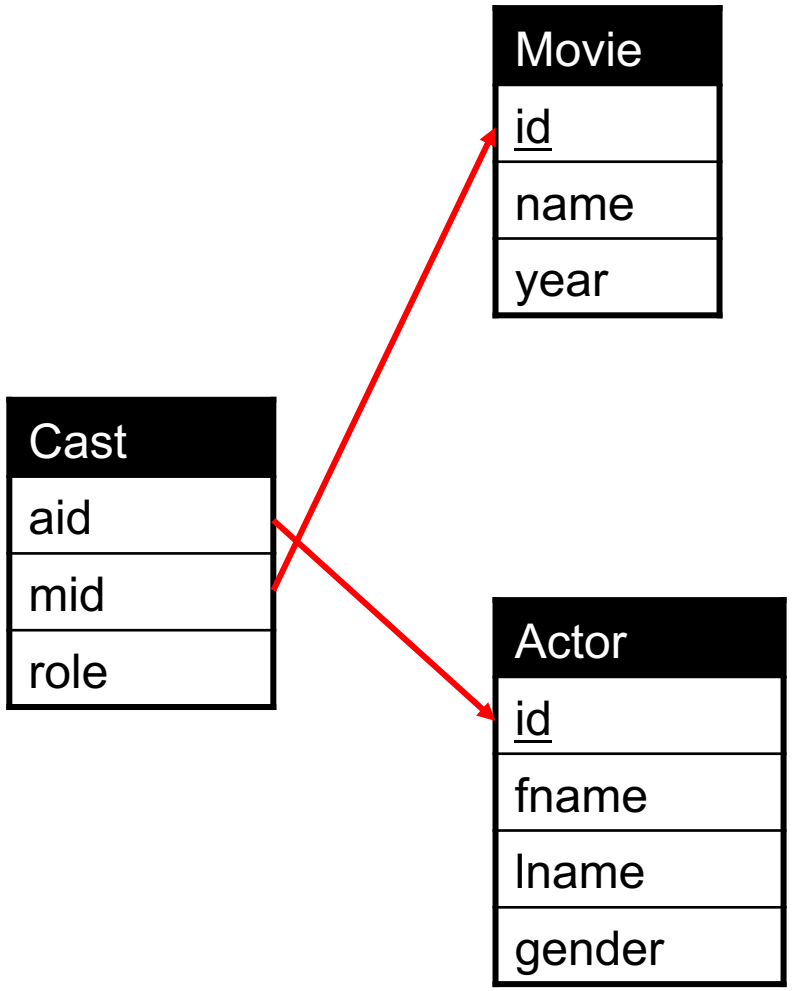


Difference to IMDB cast table?

Small IMDB Movie Database: Schema



Small IMDB Movie Database: Schema




```
-- Table structure for table 'cast'
DROP TABLE IF EXISTS 'Cast';
CREATE TABLE 'Cast' (
  'aid' int(11) default NULL,
  'mid' int(11) default NULL,
  'role' varchar(100) default NULL,
  FOREIGN KEY(aid) REFERENCES actor(id),
  FOREIGN KEY(mid) REFERENCES movie(id)
);
```

Enter SQL

```
insert into 'cast'
values (NULL, NULL, NULL)
```

Run SQL | Actions | Last Error: not an error

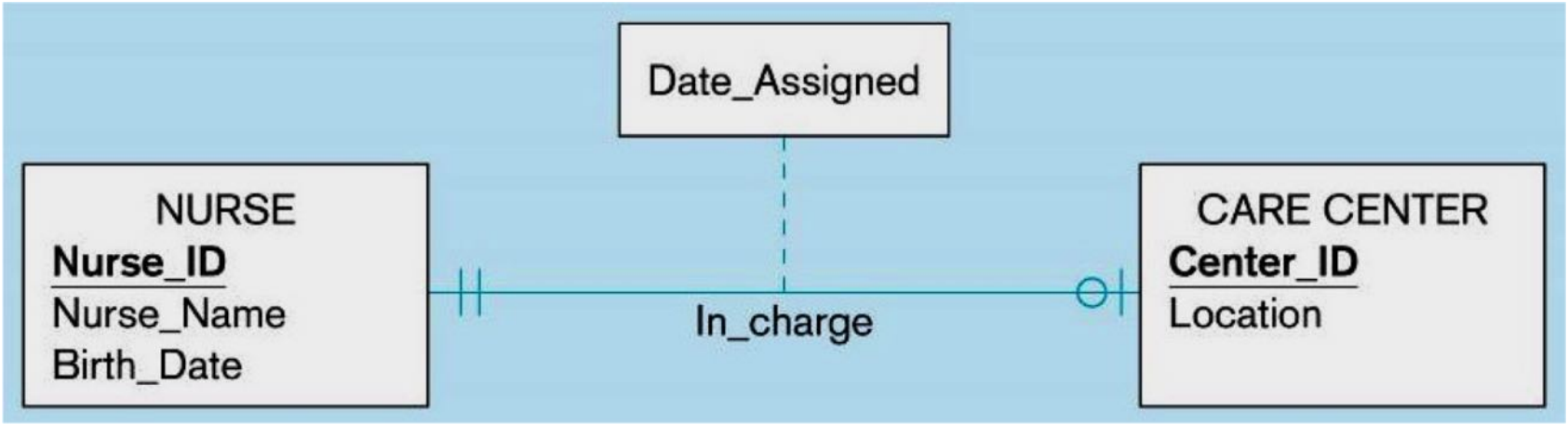
Enter SQL

```
select *
from 'cast'
where role is null
```

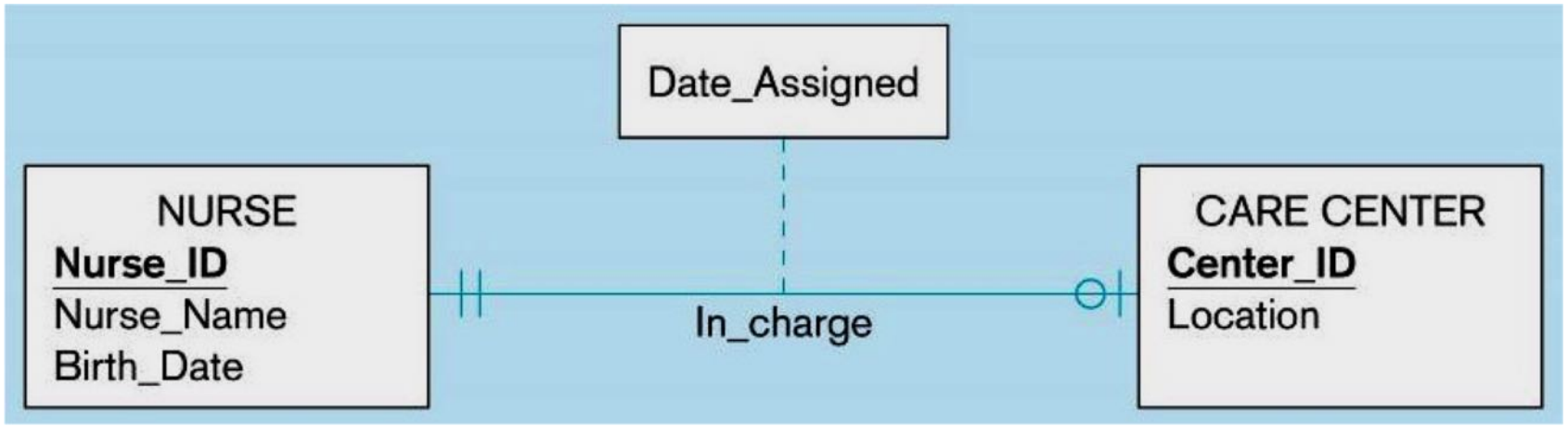
Run SQL | Actions | Last Error: not an error

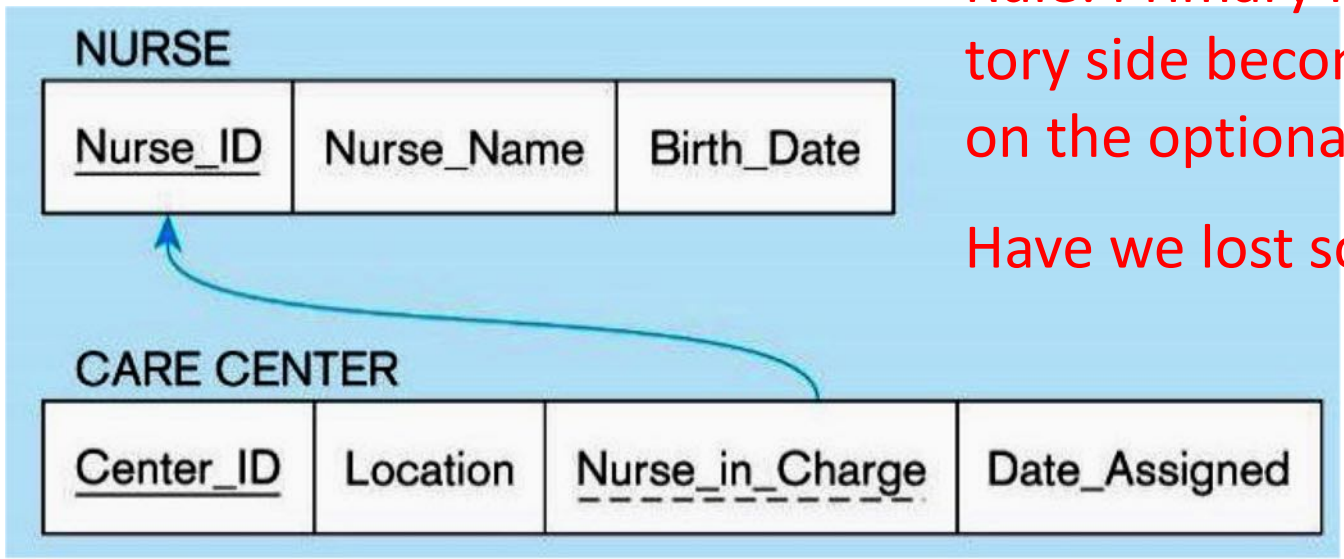
aid	mid

3) Mapping A Binary 1:1 Relationship



3) Mapping A Binary 1:1 Relationship

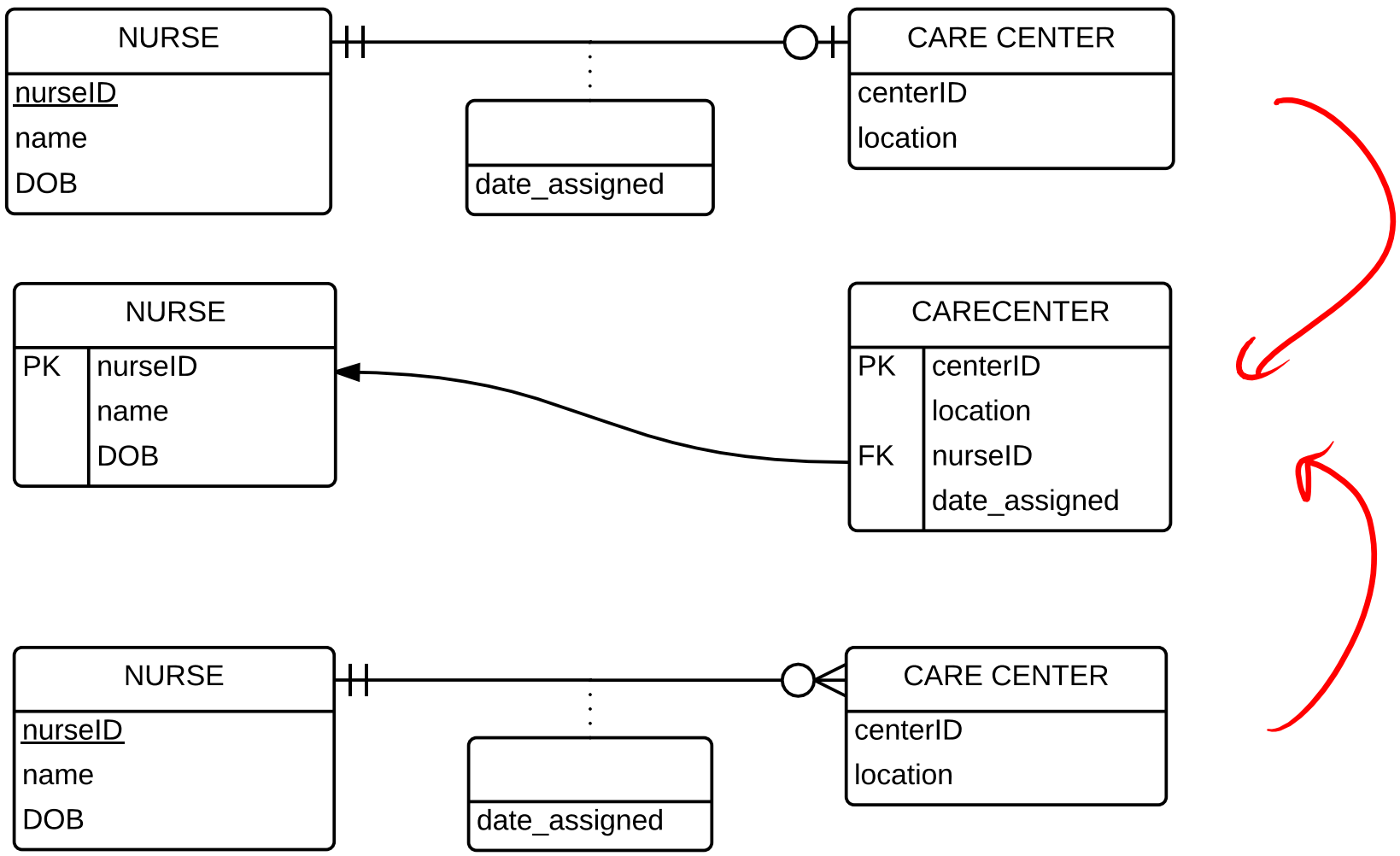




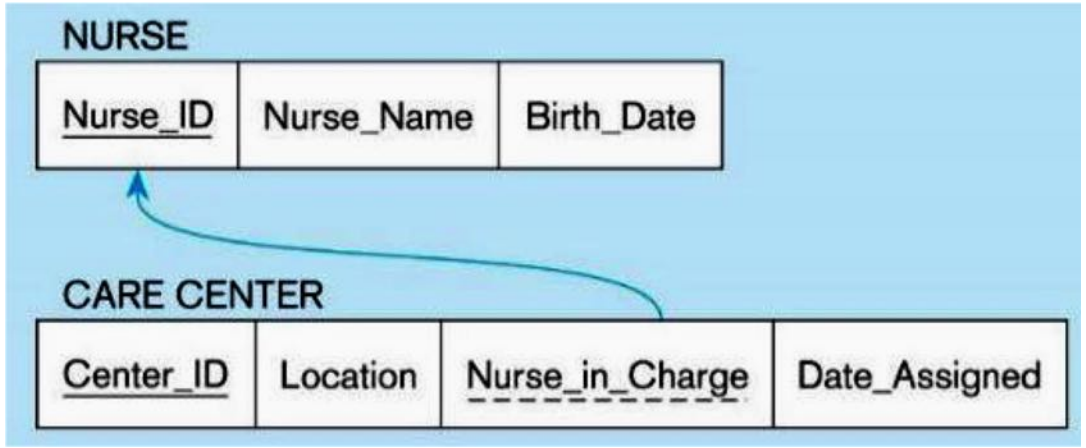
Rule: Primary key on the mandatory side becomes foreign key on the optional side

Have we lost some information?

Transform the ERD into the appropriate schema



Nurses: Instance



Nurse

nid	name	birth_date
1	Alice	1/1/1980
2	Bob	1/1/1970
3	Clarissa	1/1/1975
4	Dora	1/1/1972

Carecenter

cid	location	nurseid@	date_assigned
1	Boston	1	1/1/2016
2	New York	3	3/1/2016

Nurse

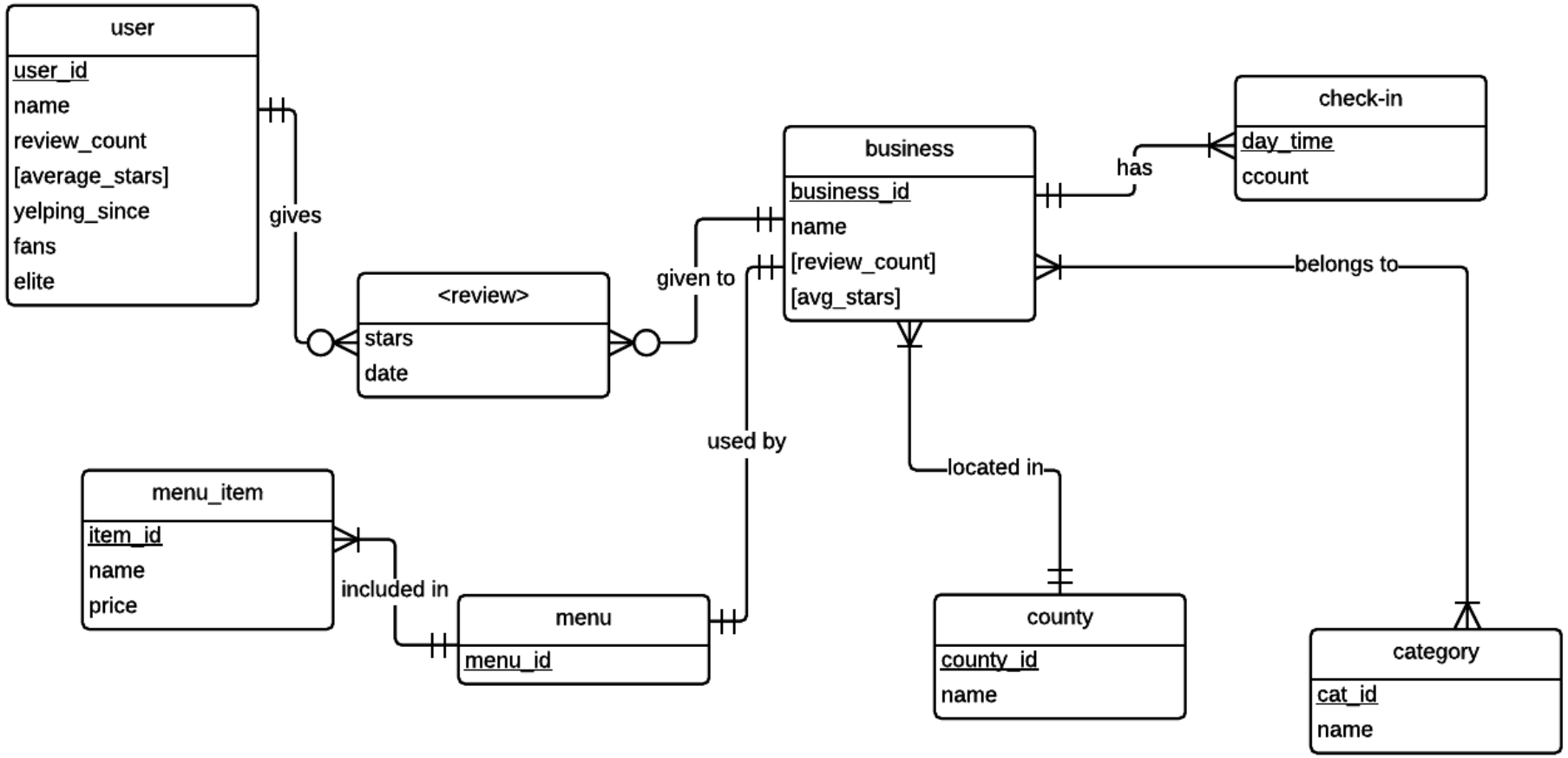
nid	name	birth_date	carecenter@	date_assigned
1	Alice	1/1/1980	1	1/1/2016
2	Bob	1/1/1970	NULL	NULL
3	Clarissa	1/1/1975	3	3/1/2016
4	Dora	1/1/1972	NULL	NULL

Carecenter

cid	location
1	Boston
2	New York

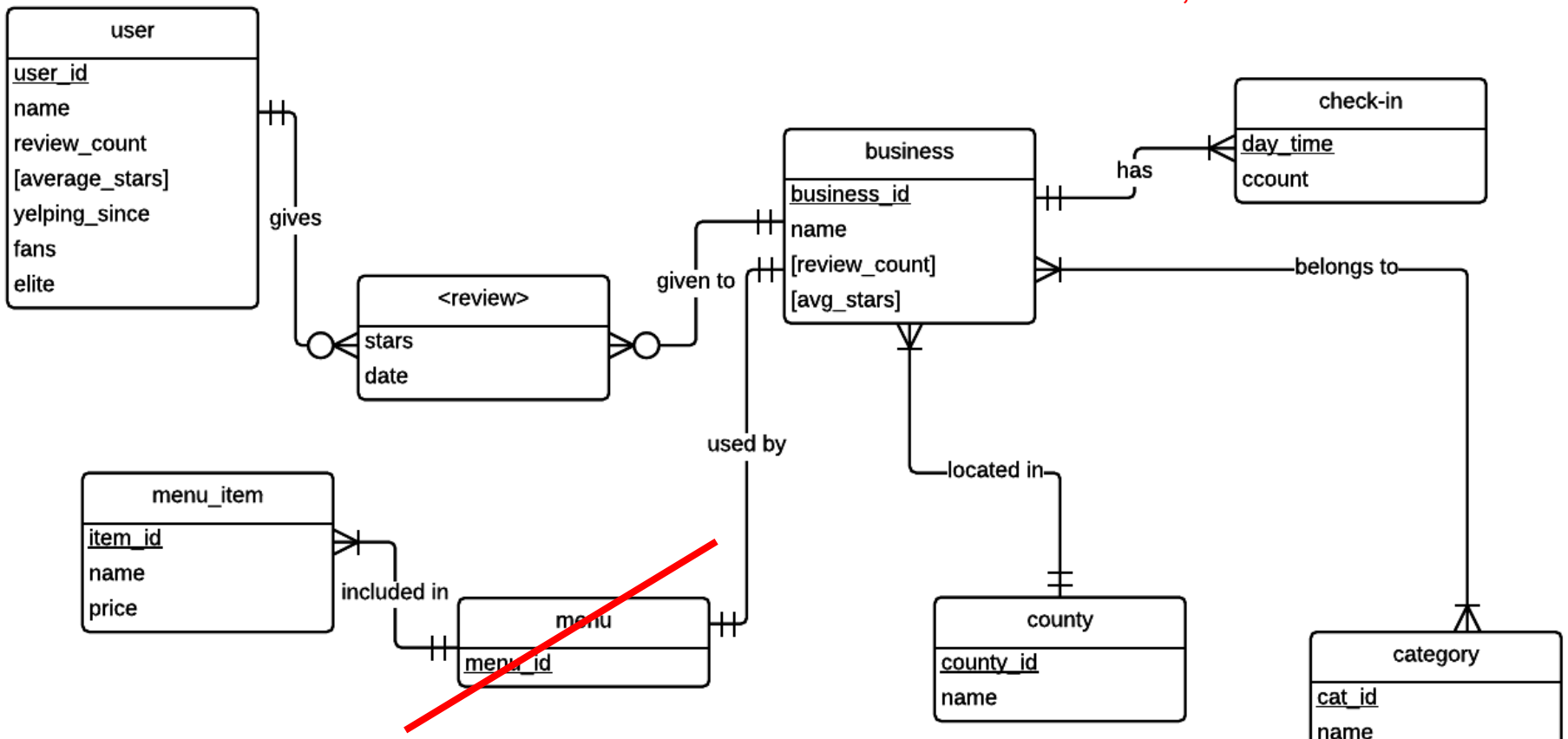
Yelp: 1:1 relationships

?



Yelp: 1:1 relationships

Also, who checks in?



No need for separate "menu" given mandatory 1:1 relationship with business. In other words, you will never have a (1,1)-to-(1,1) relationship