Administrivia

Project 1 Part 1 meetings this week and next

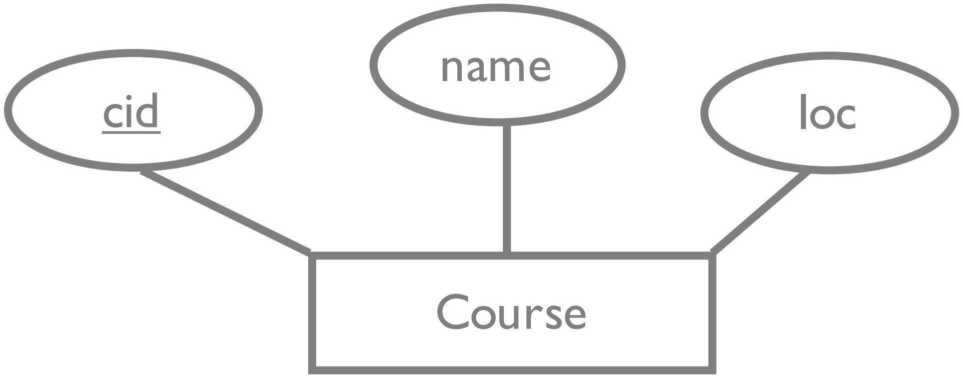
Who has not signed up yet?

Extra office hours after class today to go over DataBass

Make sure you've set up an environment before coming! (See DataBass readme)

Translating ER → Relational Models

Entity Set → Relation

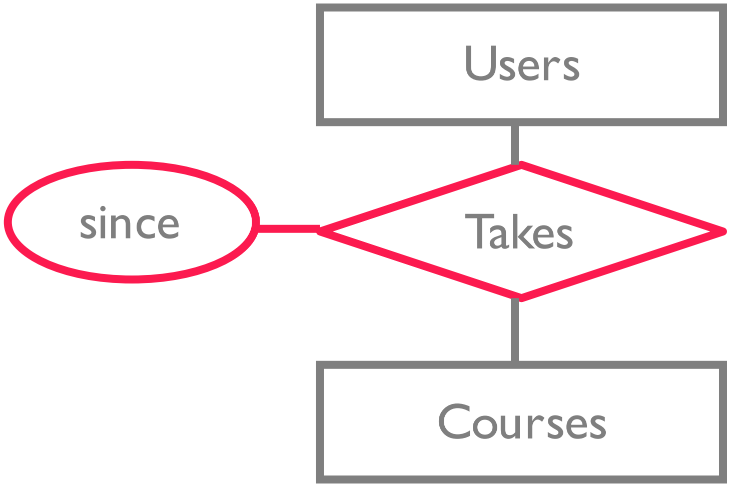


```
CREATE TABLE Course(
   cid int,
   name text,
   loc text,
   PRIMARY KEY (cid)
)
```

Relationship Set w/out constraint → Relation

Relation must include

- Keys for each entity set as foreign keys
  - these form superkey for relation
- All attributes of the relationship set



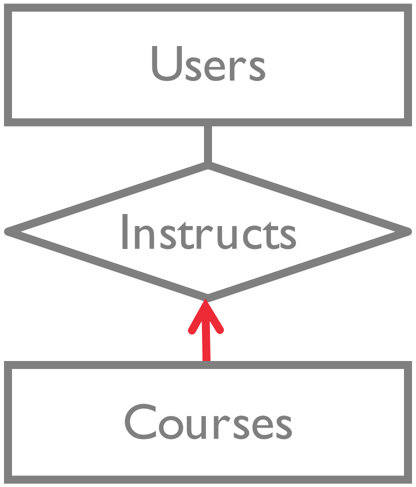
```
CREATE TABLE Takes(
    uid int,
    cid int,
    since date,
    PRIMARY KEY (uid, cid),
    FOREIGN KEY (uid) REFERENCES Users,
    FOREIGN KEY (cid) REFERENCES Courses
)
```

Key Constraint → Relation

Note only cid is a Key

Why don't we have UNIQUE(uid, cid)?

User and Courses are separate relations

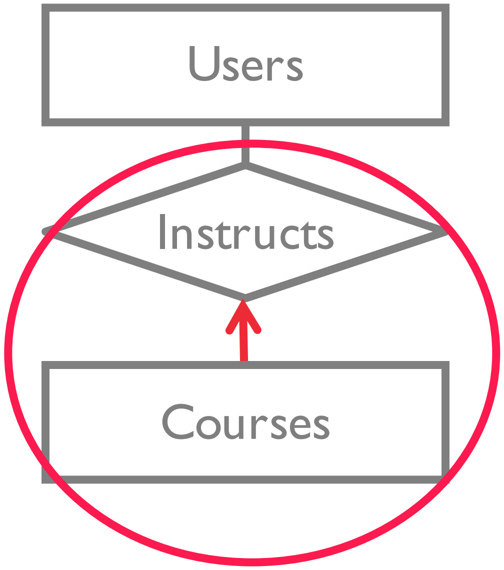


```
CREATE TABLE Instructs(
    uid int NOT NULL,
    cid int
    PRIMARY KEY (cid),
    FOREIGN KEY (uid) REFERENCES Users,
    FOREIGN KEY (cid) REFERENCES Courses
)
```

Key Constraint → Relation

Alternatively combine Courses and Users

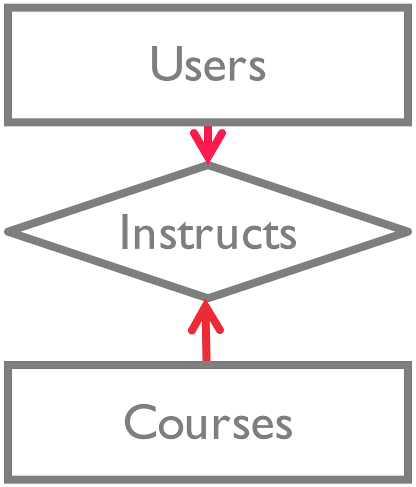
(this is the preferred way)



```
CREATE TABLE Course_Instructs(
    cid int
    uid int,
    name text,
    loc text,
    PRIMARY KEY (cid),
    FOREIGN KEY (uid) REFERENCES Users
)
```

Key Constraint → Relation

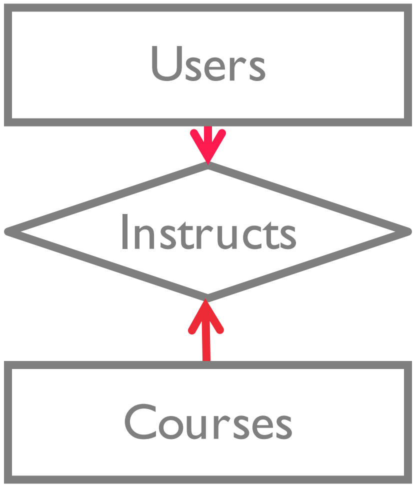
How to translate this ER diagram?



```
CREATE TABLE Course_Instructs(
    ????
)
```

Key Constraint → Relation

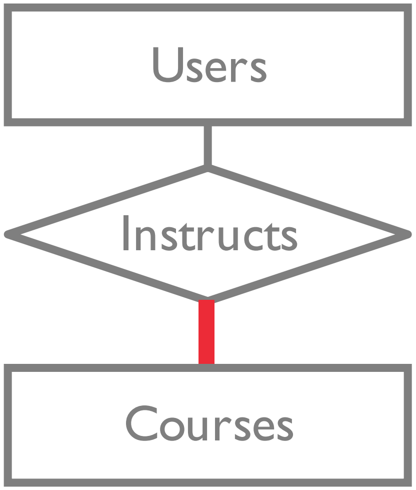
How to translate this ER diagram?



```
CREATE TABLE Course_Instructs_Users(
    cid int
    uid int,
    name text,
    loc text,
    username text,
    age text,
    PRIMARY KEY (cid, uid)
)
```

Participation Constraint → Relation

Only participation constraints with one entity set in binary relationship (others need CHECK constraint)

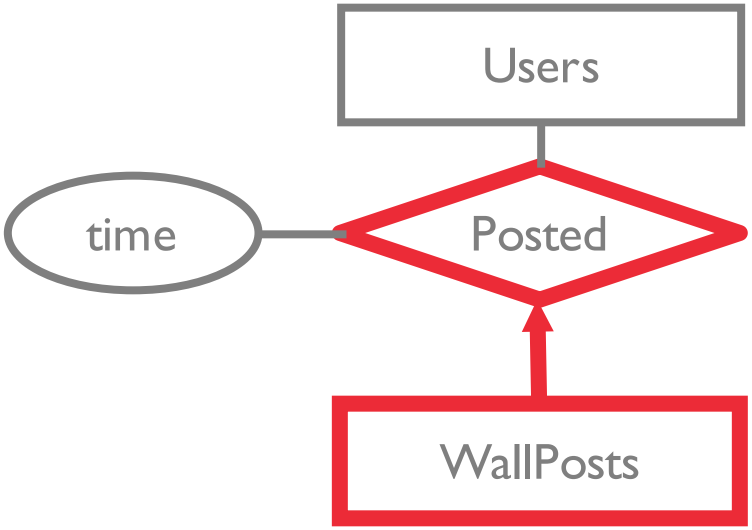


```
CREATE TABLE Course_Instructs(
    cid int
    uid int NOT NULL,
    name text,
    loc text,
    PRIMARY KEY (cid),
    FOREIGN KEY (uid) REFERENCES Users
       ON DELETE NO ACTION
)
```

Weak Entity → Relation

Weak entity set and identifying relationship set are translated into a single table.

- When the owner entity is deleted, all owned weak entities must also be deleted.



```
CREATE TABLE Wall_Posted(
    pid int
    post_text text,
    posted_at DATE,
    uid int NOT NULL,
    PRIMARY KEY (pid, uid),
    FOREIGN KEY (uid) REFERENCES Users
        ON DELETE CASCADE
)
```

ISA Hierarchies

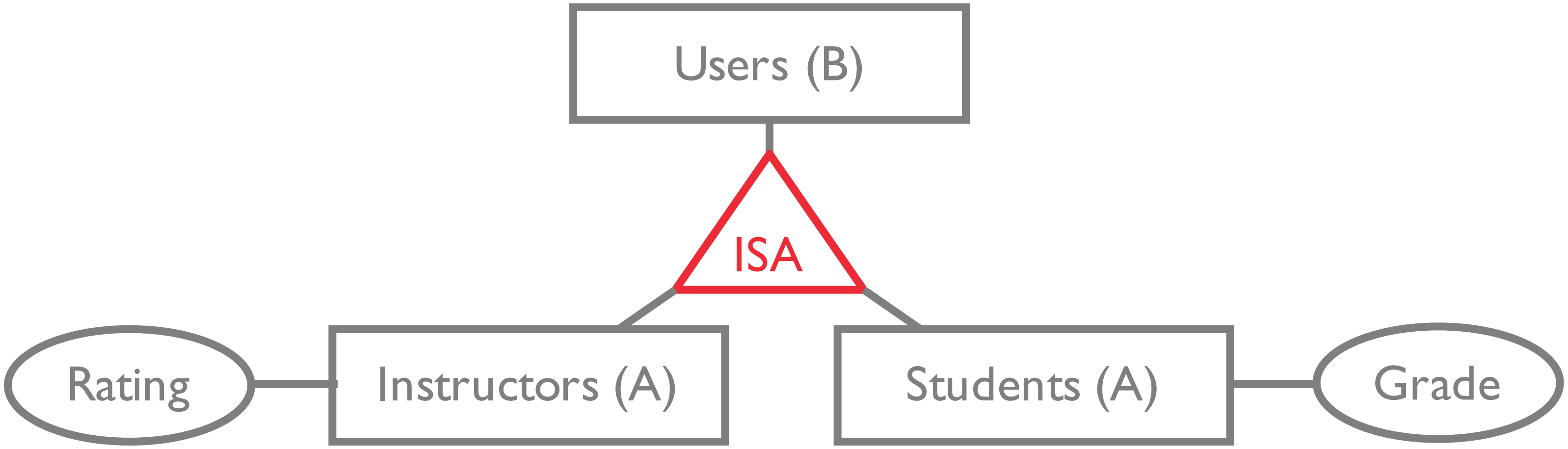
Option 1: Keep base relation

- Instructors & Students recorded in Users
- Extra info in Instructors or Students relation
- JOIN between child and base relations for all attributes

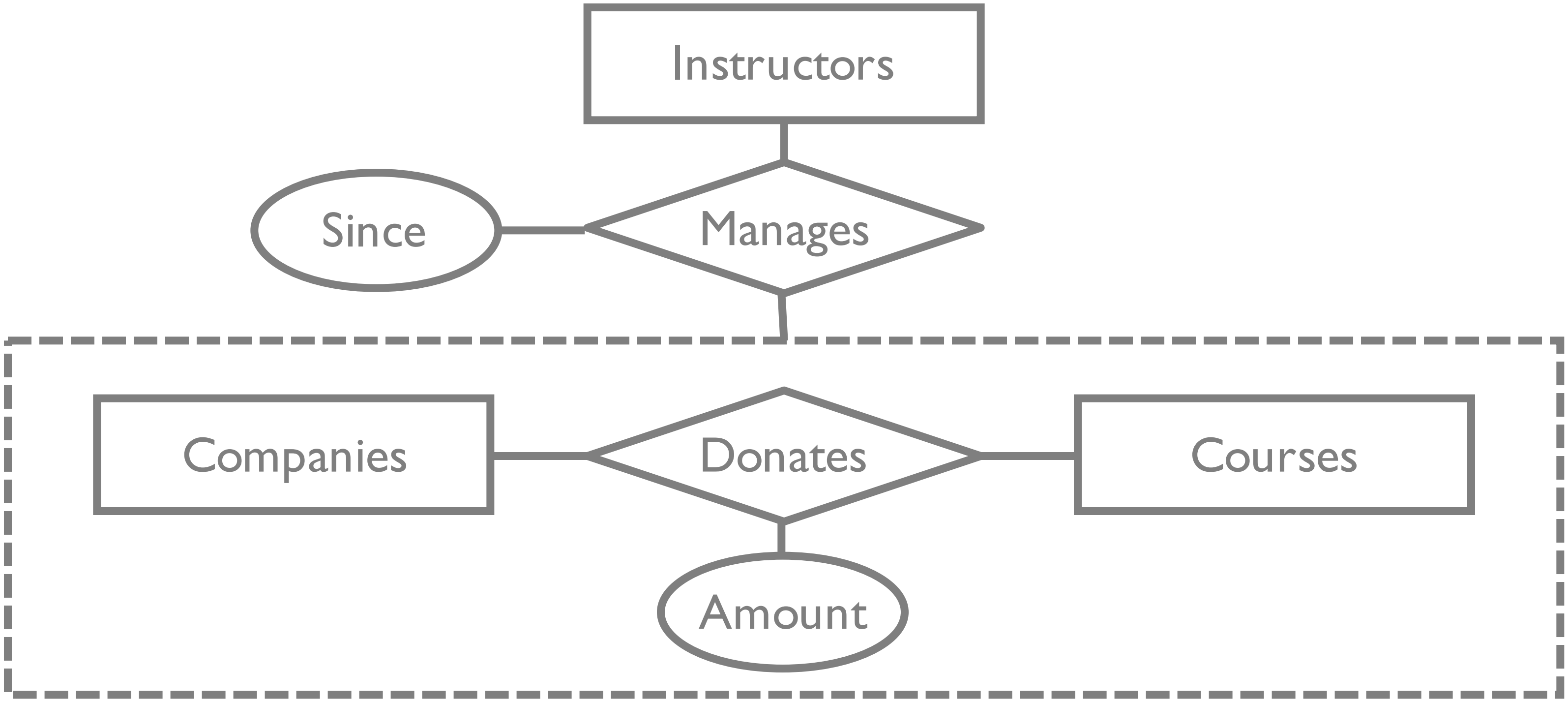
Option 2: Only keep child relations

- Instructors copies attributes from Users
- Instructors(uid, name, age, …, rating)

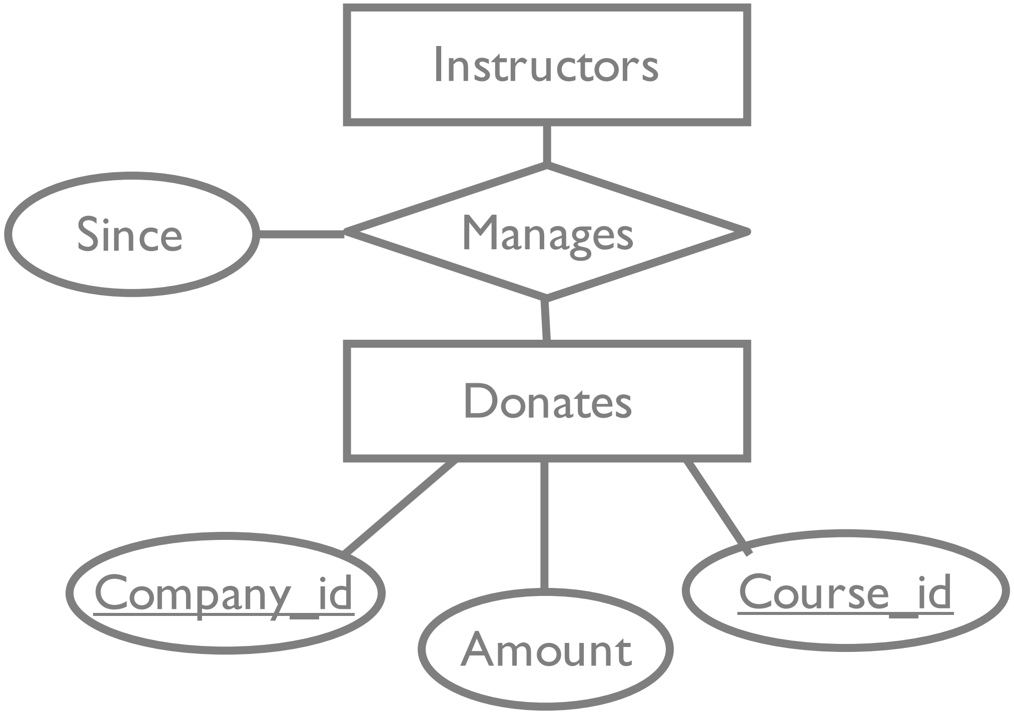
Only if covering constraint = yes



Aggregation



Aggregation



in-class demo on instabase

https://www.instabase.com/user/w4111-nb/notebooks/ewu/w4111-public/fs/Instabase%20Drive/Examples/lec5.ipynb

REVIEW OF ER AND RELATIONAL

ER Overview

- Ternary relationships
- Relationships constraints
    - At most one
    - At least one
    - Exactly one
    - Weak entities
- Aggregation

Relational Overview

Relations ~= Sets

- Schema = structure
- Instance = the data

Relational Review

Relations ~= Sets

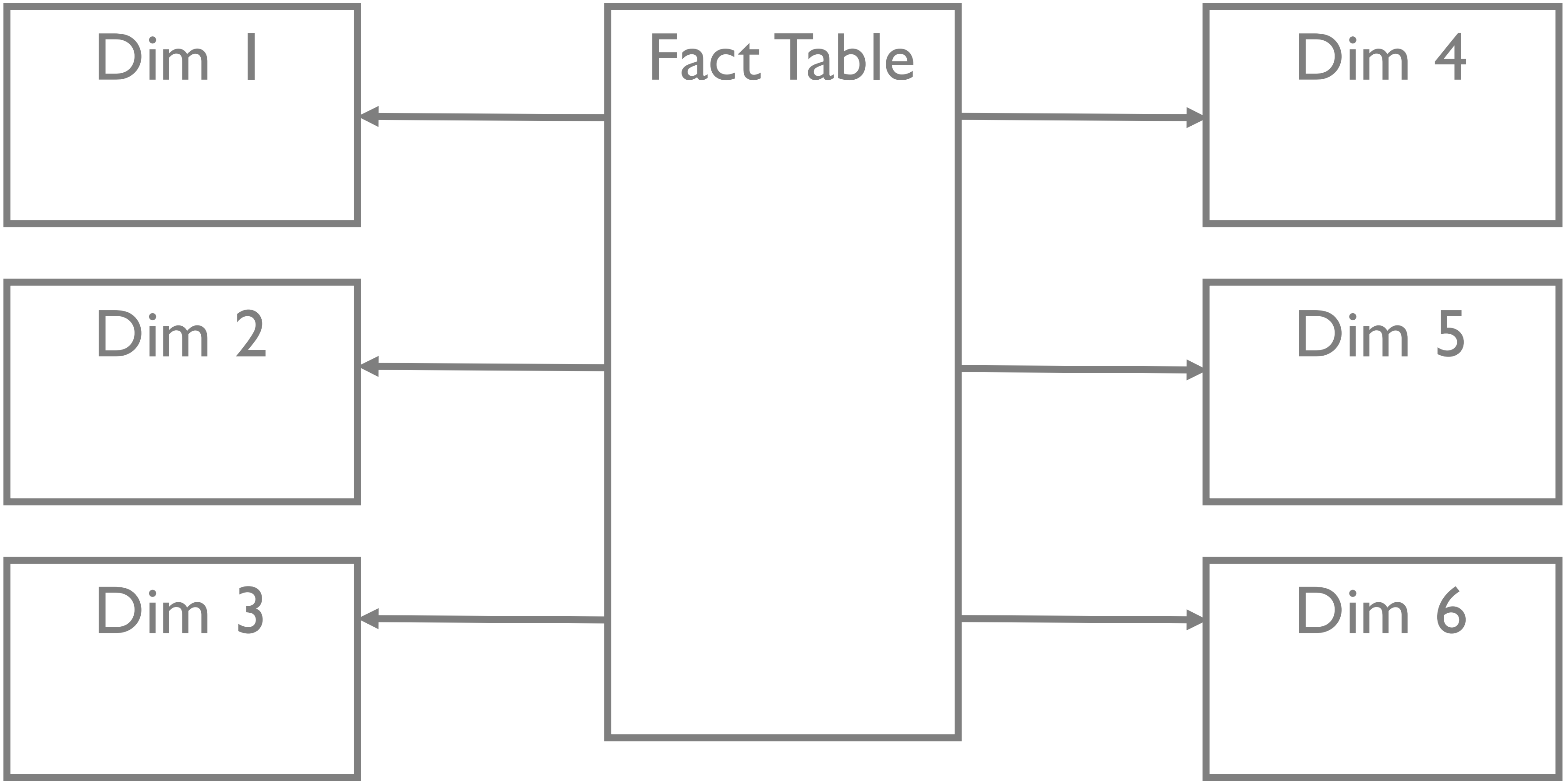
- Schema = structure
- Instance = the data
- "every relation is guaranteed to have a key” ?

Integrity Constraints

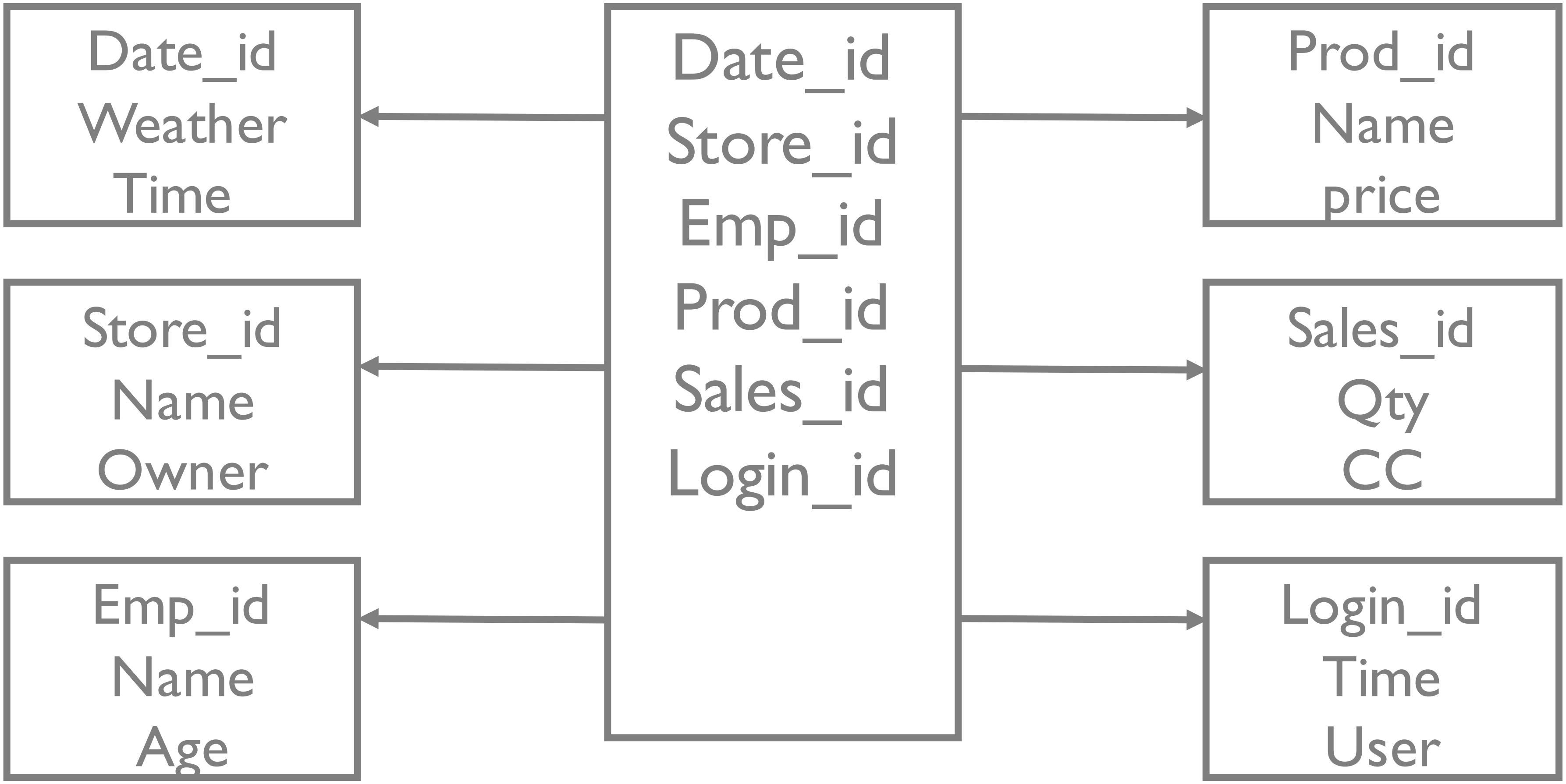
- Candidate and primary keys
- NOT NULL
- Referential integrity

How are foreign keys managed by the DBMS?

Star Schema



Star Schema



Date_id
Weather
Time

Store_id
Name
Owner

Emp_id
Name
Age

Date_id
Store_id
Emp_id
Prod_id
Sales_id
Login_id

Prod_id
Name
price

Sales_id
Qty
CC

Login_id
Time
User

So What Happened?

1970 heated debates about CODASYL vs Relational

Network arguments

- low level languages more efficient (performance)
- relational queries would never be fast (performance)

Relational arguments

- data independence
- high level *simpler* languages

Market spoke.

Other models beyond relational!

Summary

Better than IMS/CODASYL

- allows us to talk about constraints!
- allows us to talk at a logical level
- declarative queries better than navigational programs

Everything is a relation (table)

DBA specifies ICs based on app, DBMS enforces

- Primary and Foreign Keys most used
- Types == Domain constraints

SQL