PT-1 Workshop 1
Database Design &
Google Cloud Platform

Abdu Alawini

University of Illinois at Urbana-Champaign

CS411: Database Systems

Design Principles

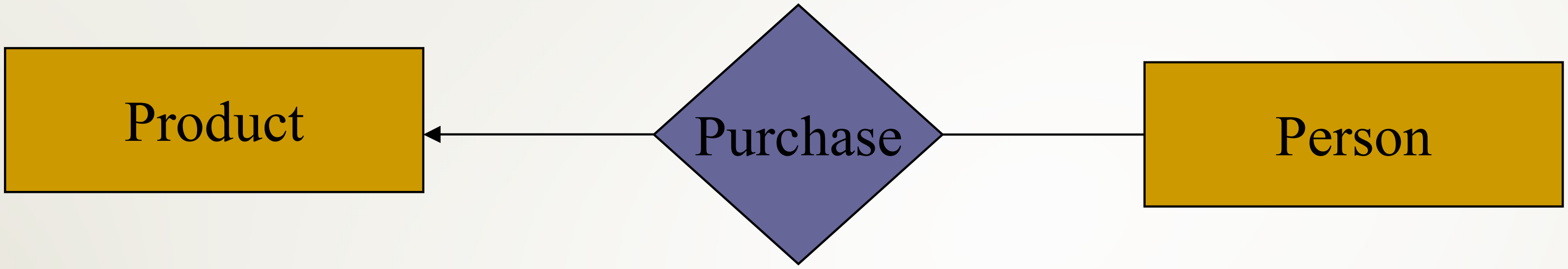
1. Be Faithful to Reality
2. Avoid Redundancy
3. Pick the Right Kind of Element

Also:

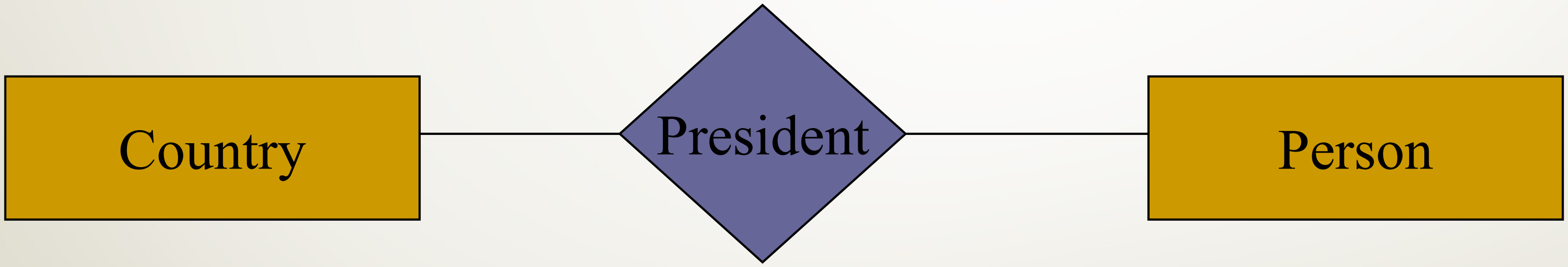
- Simplicity
- Choose the Right Relationships

(See Textbook for examples…)

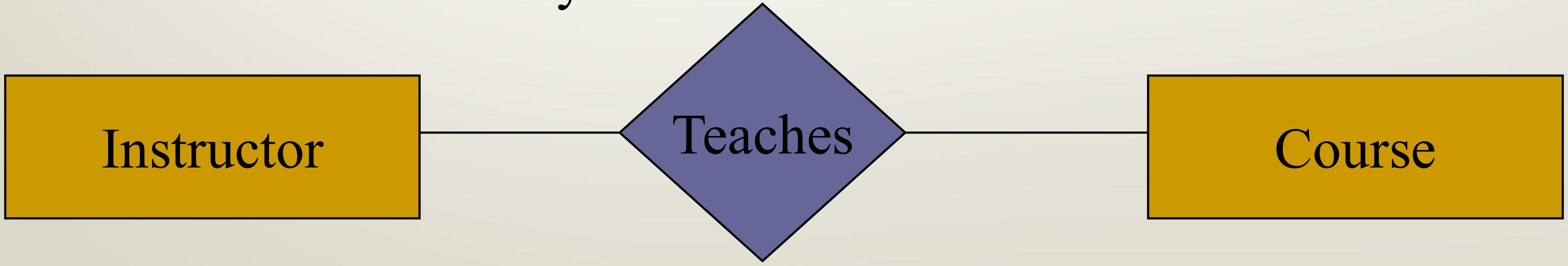
Design Principles:
Principle 1: Be Faithful To Reality



No: A Person may purchase multiple Products



No: A Country can have at most one President

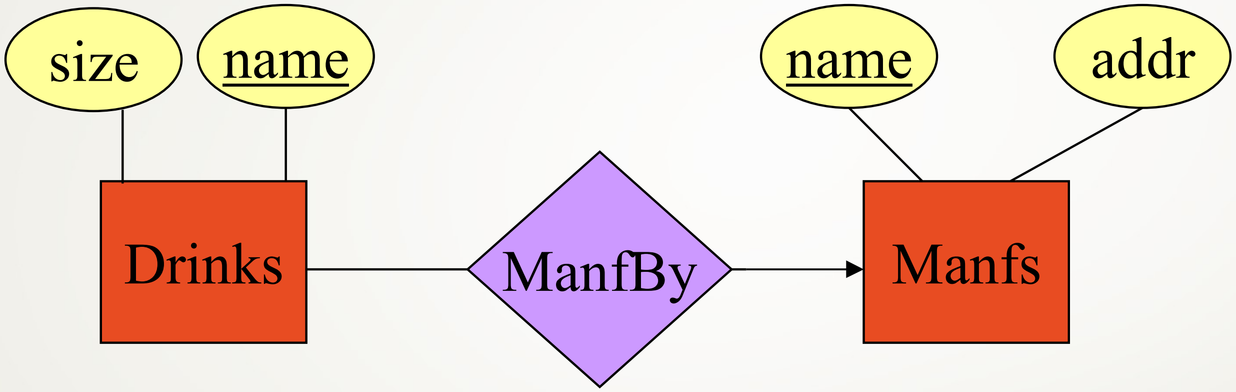


Yes if multiple instructors, No if not.

Principle 2: Avoiding Redundancy

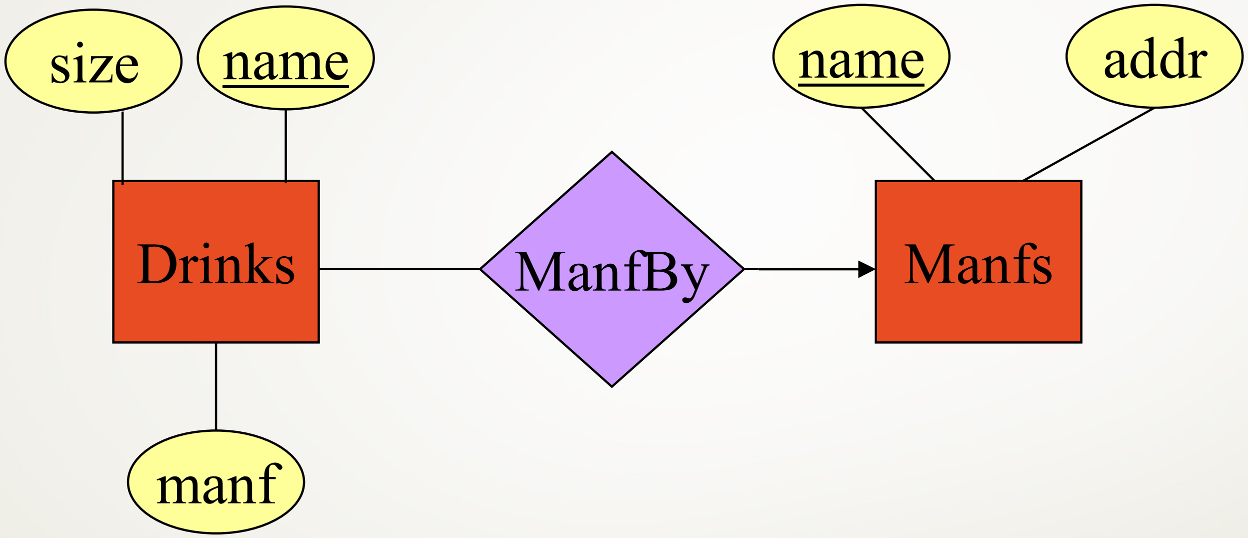
- Redundancy occurs when we say the same thing in two different ways.
- Redundancy wastes space and (more importantly) encourages inconsistency.
  - The two instances of the same fact may become inconsistent if we change one & forget to change the other.

Example: Good



This design gives the address of each manufacturer exactly once.

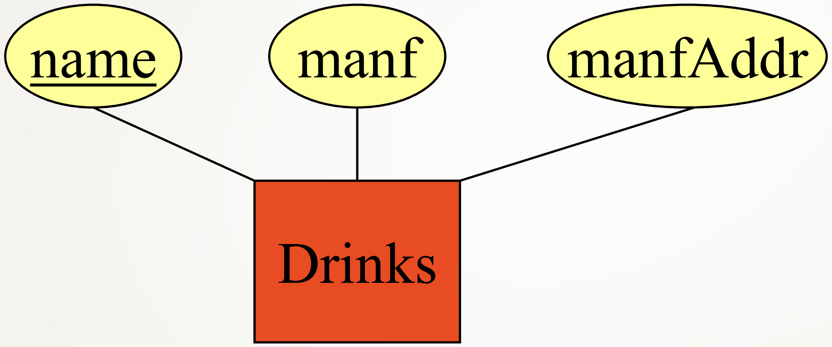
Example: Bad



This design states the name of the manufacturer of a drink twice: as an attribute and as a related entity.

Update issues, Wasteful, …

Example: Bad



This design repeats the manufacturer's address once for each drink (wasteful, update anomalies);

Also loses the address if there are temporarily no drinks for a manufacturer.

Principle 3: Entity Sets Versus Attributes

- An entity set should satisfy at least one of the following conditions:
  - It is more than the name of something (i.e., it has at least one non-key attribute)
    Or
  - It is the "many" in a many-one or many-many relationship.

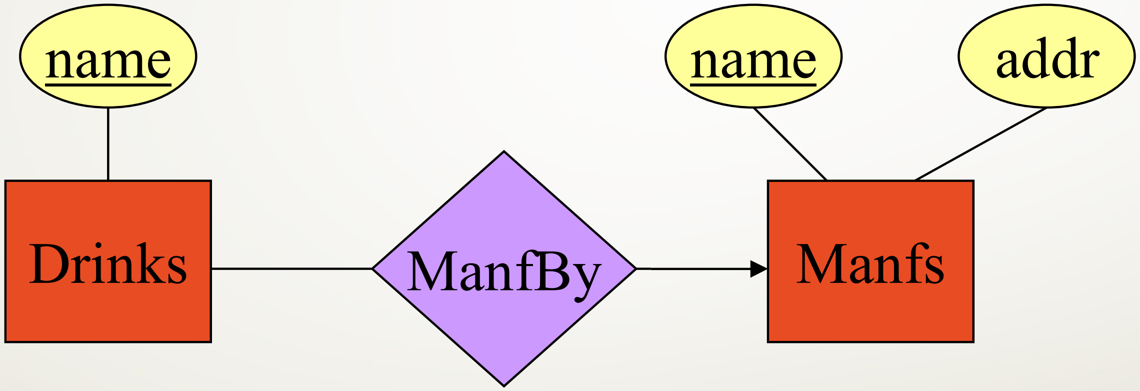
Examples will illustrate why, but also think why each of these rules actually make sense.

Example: Good

- An E.S. is more than the name of something; (i.e., it has at least one non-key attribute)

  OR

- An E.S is the “many” in a many-one or many-many relationship.



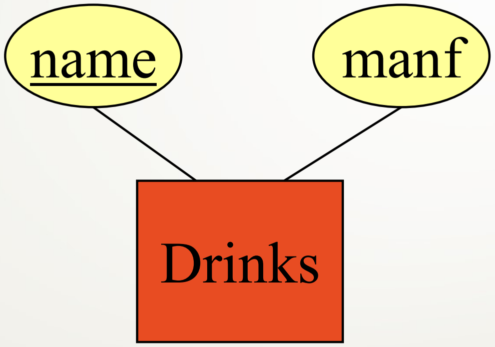
- *Manfs* deserves to be an entity set because of the nonkey attribute *addr*.
- *Drinks* deserves to be an entity set because it is the “many” of the many-one relationship *ManfBy*. Can you see why?

Example: Good

- An E.S. is more than the name of something; (i.e., it has at least one non-key attribute)

  OR

- An E.S is the “many” in a many-one or many-many relationship.



If we had no manufacturer address information…

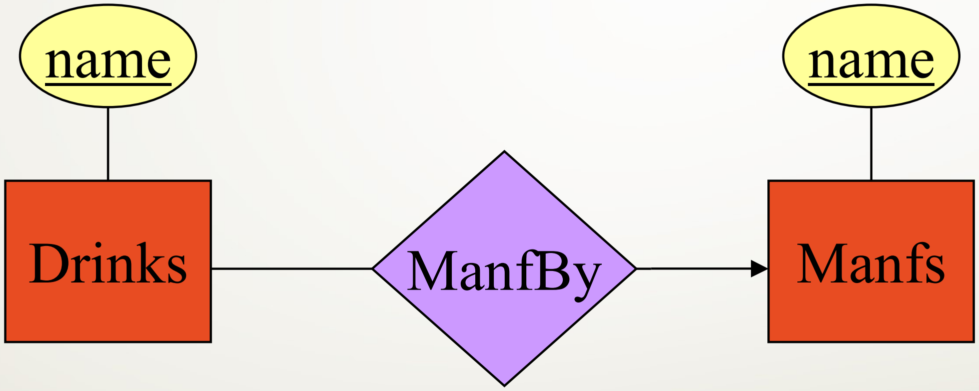
There is no need to make the manufacturer an entity set, because we record nothing about manufacturers besides their name.

Example: Bad

- An E.S. is more than the name of something; (i.e., it has at least one non-key attribute)

  OR

- An E.S is the "many" in a many-one or many-many relationship.



Since the manufacturer is nothing but a name, and is not at the "many" end of any relationship, it should not be an entity set.

Google Cloud Platform

Abdu Alawini

University of Illinois at Urbana-Champaign

CS411: Database Systems

Thanks to Google!

- We received a generous educational grant from Google that includes $50 credit for Google Cloud Platform **for each student**
  - Web-based GUI that provides a suite of cloud computing services which runs on Google's infrastructure

Google Cloud Platform (GCP)

- Web-based GUI that provides a suite of cloud computing services which runs on Google's infrastructure

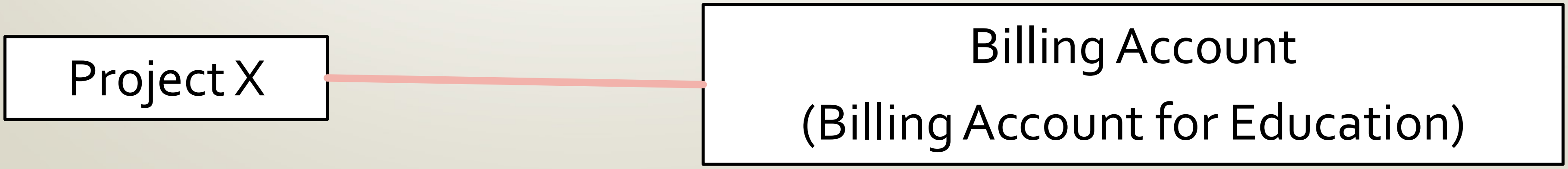
- You will use GCP for PT1.
  - Host your database in GCP.
  - Avoid the "database on my teammate's machine" problem!

How to redeem GCP coupon?

- We will post instructions on our course discussion forum
- Please follow these steps:
  1. Visit the student coupon retrieval link
  2. Enter your name and **school email**
     - A **confirmation email** will be sent to you **with a coupon code**.
  3. You must use your **personal Google account** when redeeming GCP
     - Unfortunately, Campus IT department has disabled GCP for Google Apps @ Illinois accounts.
- After signing up, you'll get $50 credit ☺

GCP: basic workflow

1. go to the GCP console at https://console.cloud.google.com
2. create a new project OR select an existing one
3. set billing for this project ***VERY IMPORTANT***



Project X

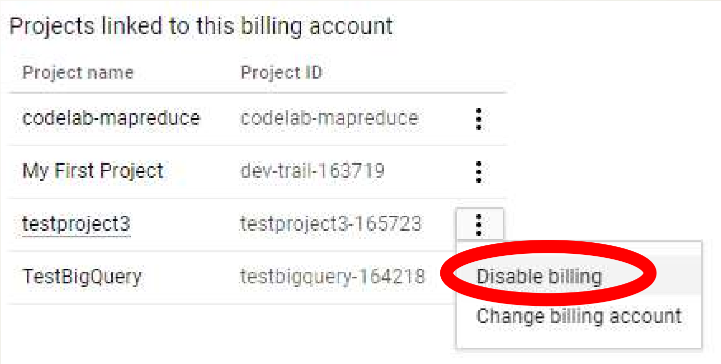
Billing Account
(Billing Account for Education)

GCP: billing

Go to Main Menu → Billing → Account Management

Enable/change billing: if a project's billing has not been enabled, set it to the billing account that you redeemed

Disable billing: if you are worried about overspending, it's easiest to just disable billing from any project that you don't use



Projects linked to this billing account

Project name	Project ID
codelab-mapreduce	codelab-mapreduce
My First Project	dev-trail-163719
testproject3	testproject3-165723
TestBigQuery	testbigquery-164218

Budget Alerts

- Create a budget.
  - GCP Console -> Billing -> Budgets & Alerts
- When your spending exceeds your budget, you get a warning email.
- The budget does not stop your projects from consuming resources.

MySQL in GCP

- Main menu -> SQL
- Choose MySQL
- Create instance (with a name, such as myInstanceName)
- Open gcloud shell
- Commands to access MySQL from GCloud Shell:
  - `gcloud config set project project-id`
  - `gcloud sql connect myInstanceName --user=root`

- IMPORTANT: Do not forget to stop or delete your mysql instance after you finish your work.

Accessing MySQL in GCP

- MySQL Workbench
- From a programming language

PT1 – Workshop 1

Your team will:

1. Review your ER/UML draft
   - Your project TA will help with reviewing your draft and providing comments.
2. Translate your ER/UML into a logical design (relational database schema)
3. Create a MySQL instance for your project on GCP
4. Create your project database schema on MySQL@GCP
5. Access your GCP MySQL from:
   - a programming language.
   - From MySQL Workbench

GCP MySQL Pricing

- For information about pricing please follow this link. https://cloud.google.com/sql/docs/mysql/pricing