

## Databases

- sounds complicated, but is very intuitive
- just a way of storing information
  - large amounts
  - easily accessed
  - easily managed
  - easily updated
- special type of data structure

## Table Analogy

Name	SID	year	major
Joe	25	1	CS
Mary	47	2	English

} students

## DBMS (Database Management System)

- when people say 'DB' they often mean DBMS
- software that interacts with the DB database structure, in a user-friendly way
- allows creation, modification, administration, etc...

- ex: MySQL, PostgreSQL, Oracle, Access

## "Relational" DB

- distinction b/w relational DB and non-relational
- not referring to "relationships"
- technical term from relational algebra
  - ↳ related to set theory
- "relation" is just a set of tuples
  - finite, ordered list of elements
- every tuple has the same # of elements and every element in the same position is a member of the same domain
- "domain"
  - a set of every possible value that a given element of a tuple could have
- in the table analogy
  - tuple = row
  - attribute/element = column
  - domain = data type (int, string, boolean, etc)
    - ↳ defined by
  - relation = table

- examples of relational DBs:  
OracleDB, MySQL, PostgreSQL, SQLite, etc.
- examples of non-relational DBs:
  - key-value stores: Redis
  - document store: MongoDB

## Schema

- structure of a DB
- map of how it's constructed and how things relate to each other
- some schema are static (non-changing) and some are dynamic

## Query

- a single action taken on the database
  - finding information
  - updating or creating data (or deleting)
  - creating or changing the schema
- looks like a line of code

## Transaction

- sequence of queries run against a DB
- single unit of work (dependent on each other)

ex: "        +

name	money
Bob	\$40
Alice	\$80

Bob gives Alice \$20

- subtract 20 from Bob } transaction
- add 20 to Alice }

ACID (Atomicity, Consistency, Isolation, Durability)

- transactions must demonstrate all 4 to be

ACID-compliant

Atomicity

- everything succeeds or fails together

Consistency

- all queries leave the DB in a consistent state

Isolation

- no transaction interferes with another transaction

Durability

- changes made by a transaction are available in the DB in case of system failure

SQL (Structured Query Language)

- a programming language based on relational algebra

- used to communicate with a DBMS

- this a specific abstraction used in specific DBMS's (MySQL, SQLite, PostgreSQL....)

## Normalization (de-)

- process of structuring the schema to improve:
  - reducing redundancy
  - data integrity (consistent, up-to-date, accurate)
  - efficiency

<u>author</u>	<u>ISBN</u>
king, steven	10234, 24116, 38058
fields, sally	1143, 53216
.....	

was book # 38058 written by  
Stephen King?

<u>author</u>	<u>ISBN</u>
king, steven	10234
king, steven	24116
king, steven	38058
:	:

-denormalization

-adding redundancy for some effect

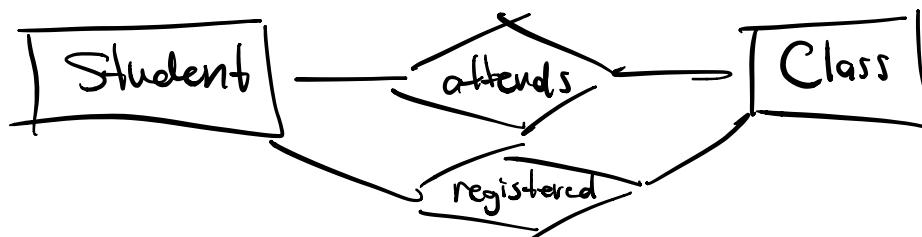
- speeding up queries (efficiency)

## ORM (Object-relational models)

- libraries or packages that abstract DB interaction away from the DBMS languages (SQL) into a more familiar language
- lets you continue coding in your language of choice
- there are inevitably many ORMs for any given DBMS ↔ programming language pair
- in this class we will look at:
  - MySQL ↔ Python
  - MySQL → Ruby

## ER Model (Entity-Relationship Model)

- Entities: things that exist and are distinguishable  
ex: Bob, Alice, book, place, tree, job, coupon code
- Relationships: two or more entities participate in a relationship  
ex: owes money, wrote book, lives at, friend



Entities are represented by rectangles



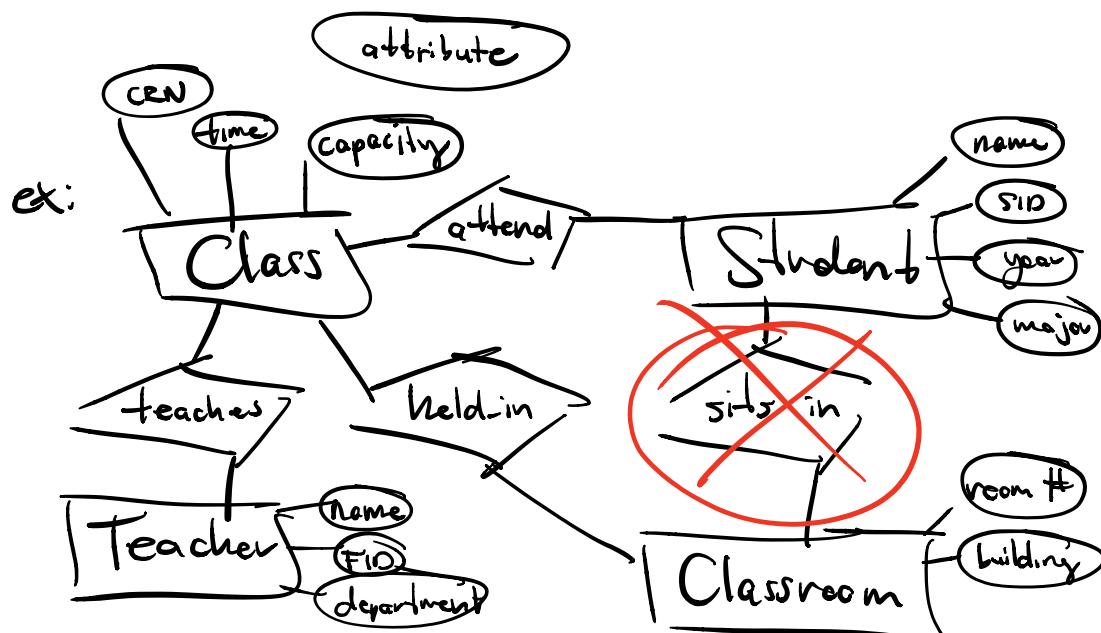
Relationships are represented by diamonds



Entities participate in relationships via lines

—

Attributes of entities represented by ellipses



- for any real world domain, we could model a potentially infinite number of relationships

but we really don't want to  
-ideally we model the minimum relationships  
necessary to answer whatever problems we're  
trying to solve