CS W4111.001
Introduction to Databases
Fall 2020

Computer Science Department
Columbia University
Overview of Course

- **How to use a database management system (DBMS)** → bulk of course
  - database design (today and following lectures)
  - SQL + relational model + object-relational model
- **What happens under the covers of a DBMS** → just a few lectures
  - introduction to query processing and optimization
  - introduction to transaction processing
Steps for Database Design

1. Understand data characteristics and requirements in real-world organization
Steps for Database Design

1. Understand data characteristics and requirements in real-world organization

2. Produce conceptual design, ("pen-and-pencil") diagram using Entity-Relationship (ER) Model
   - Entities and relationships in organization?
   - Information about entities and relationships that we should store in database?
   - Integrity constraints or business rules that hold?
Steps for Database Design

1. Understand data characteristics and requirements in real-world organization
2. Produce conceptual design, (“pen-and-pencil”) diagram using Entity-Relationship (ER) Model
   - Entities and relationships in organization?
   - Information about entities and relationships that we should store in database?
   - Integrity constraints or business rules that hold?
3. Map ER diagram into a relational schema in SQL
Steps for Database Design (cont.)

4. Refine and normalize SQL schema
   ● Check relational schema for redundancies and related anomalies

5. Produce and tune physical database design
   ● Consider typical workloads and further refine the database design, with appropriate indexes, etc.
Conceptual Database Design Using the Entity-Relationship (ER) Model
Conceptual Database Design

**Goal:** Model data in a real-world organization

**Approach:** Use a *semantic data model*, which is a collection of concepts for describing data

We will focus on the **Entity-Relationship Model**

**ER diagrams** help us move from an informal knowledge of the data in a real-world organization to a *detailed, precise description of the data* that is much closer to DBMS
ER Model Basics

• **Entity**: Real-world object distinguishable from other objects; an entity is described using a set of attributes

Examples:
• Jane Smith, with ssn = …
• a branch of a bank
• the address of the branch
• an account in the bank
• …
ER Model Basics

- **Entity**: Real-world object distinguishable from other objects; an entity is described using a set of attributes

- **Entity set**: Collection of similar entities (e.g., all employees)
  - All entities in an entity set have the same set of attributes (until we consider ISA hierarchies!)

- Each attribute has a **domain**, which is a single, “simple” value (e.g., string, integer, real, date, ...)
- Each entity set has a **key** (more on this later)

<table>
<thead>
<tr>
<th>Employees</th>
</tr>
</thead>
<tbody>
<tr>
<td>ssn: string</td>
</tr>
<tr>
<td>name: string</td>
</tr>
<tr>
<td>lot: string</td>
</tr>
</tbody>
</table>
Entity Sets

- Entity sets are not necessarily disjoint
  - Example: Bank Employees, Tax Payers

- We should choose attributes to model each entity set carefully:
  - ssn, name, lot relevant for Employees
  - eye color not relevant for Employees
Each Entity Set Has a Key

**Key:** A minimal set of attributes whose values uniquely identify an entity in the entity set.

If an entity set has only one key, then that key is the **primary key**, and it is indicated in the ER diagram by underlining its attribute names.

<table>
<thead>
<tr>
<th>Employees</th>
</tr>
</thead>
<tbody>
<tr>
<td>ssn: string</td>
</tr>
<tr>
<td>name: string</td>
</tr>
<tr>
<td>lot: string</td>
</tr>
</tbody>
</table>
Candidate Keys

If an entity set has multiple keys, one is selected as the primary key, and the rest are candidate keys.

Candidate keys are noted in plain-English in the ER diagram (no special notation)
ER Model Basics

• **Relationship**: Association between 2 or more entities (e.g., Jane *works in* Pharmacy Department)

• **Relationship set**: Collection of similar relationships
  
  An n-ary relationship set R relates n entity sets $E_1 \ldots E_n$; each relationship in R involves entities $e_1, \ldots, e_n$

  Relationship sets can also have descriptive attributes (e.g., “since” attribute of Works_In)
ER Model Basics

Relationships are identified fully by the entities that participate in them, without using the descriptive attributes.
Example of Ternary Relationship Set

Locations
- address: string
- capacity: integer

Employees
- ssn: string
- name: string
- lot: string

Works_In2
- since: date

Departments
- did: string
- dname: string
- budget: real
ER Model Basics

Same entity set could participate in different relationship sets, or in different “roles” in the same relationship set → we add “role indicators”
Constraints in ER Diagrams

- ER diagrams should express all known real-world “constraints” that the data should satisfy.
- We’ll be able to express many of these constraints using ER model notation.
- Some constraints cannot be expressed using ER model notation: we should still include them as plain-English annotations in the ER diagram, so that we can then remember to model them in SQL when we map the ER diagram to SQL.
Key and Participation Constraints

- We have already discussed domain constraints (for attributes), and primary key and candidate key constraints (for entity sets).

- We will now discuss two kinds of constraints for relationship sets: **key constraints and participation constraints**
Key Constraints

Key constraint for Manages relationship set: each department can have at **most one** manager

```
<table>
<thead>
<tr>
<th>Employees</th>
<th>Departments</th>
</tr>
</thead>
<tbody>
<tr>
<td>ssn: string</td>
<td>did: string</td>
</tr>
<tr>
<td>name: string</td>
<td>dname: string</td>
</tr>
<tr>
<td>lot: string</td>
<td>budget: real</td>
</tr>
<tr>
<td></td>
<td>since: date</td>
</tr>
</tbody>
</table>
```
Key Constraints

Key constraint for Manages relationship set: each department can have at most one manager

Also now: Each employee can manage at most one department; how do we update diagram?
Key Constraints in n-ary Relationship Sets with n>2

Meaning of arrow here?

Restriction for n-ary relationship sets with n>2: can have at most one arrow (more on this later)
Key Constraints in n-ary Relationship Sets with n>2
Participation Constraints

Participation constraint for Works_In relationship set: each employee must work in at least one department

The participation of Employees in Works_In is total
Participation Constraints in n-ary Relationship with n>2

- Locations
  - address: string
  - capacity: integer

- Employees
  - ssn: string
  - name: string
  - lot: string

- Departments
  - did: string
  - dname: string
  - budget: real

- Works_In4
  - since: date

Diagram showing the relationship between Locations, Employees, Departments, and Works_In4, with participation constraints.
Multiple Participation Constraints Allowed
ER Diagrams

- We now know how to create:
  - Entity sets
  - Relationship sets
- We also know how to specify:
  - Domain constraints for attributes
  - Primary key and candidate key constraints for entity sets
  - Key and participation constraints for relationship sets
- Several advanced features still to come!