COLUMBIA | ENGINEERING
EXECUTIVE EDUCATION

# Week 3

## Relational Databases: Where Big Data is Typically Stored

**Applied Data Science**

**Columbia University - Columbia Engineering**

# Course Agenda

COLUMBIA | ENGINEERING
EXECUTIVE EDUCATION

- Week 1: Python Basics: How to Translate Procedures into Codes
- Week 2: Intermediate Python — Data Structures for Your Analysis
- **Week 3: Relational Databases — Where Big Data is Typically Stored**
- Week 4: SQL — Ubiquitous Database Format/Language
- Week 5: Statistical Distributions — The Shape of Data
- Week 6: Sampling — When You Can't or Won't Have ALL the Data
- Week 7: Hypothesis Testing — Answering Questions About Your Data
- Week 8: Data Analysis and Visualization — Using Python's NumPy for Analysis
- Week 9: Data Analysis and Visualization — Using Python's Pandas for Data Wrangling
- Week 10: Text Mining — Automatic Understanding of Text
- Week 11: Machine Learning — Basic Regression and Classification
- Week 12: Machine Learning — Decision Trees and Clustering

## Transient vs. Persistent data

- Program data is transient
- When the program ends, data is lost
- If we rerun the program, the data will need to regenerated

Organized collections of data
(that reside on a computer)

Digital organization methods:
Relational databases
NoSQL databases

## Who uses databases?

Almost every application today uses some sort of database!

If you go on the web, you're almost certainly interacting with a database

## Relational databases

- Data is stored in 2-dimensional tables
- Tables (relations) are logically connected sets of data
- Table rows (records/tuples) are information about one entity
- Table columns are attribute values
- Uses SQL for information retrieval
- Goal: Minimize redundancy and maximize consistency

## NoSQL Databases

- Low latency
- Scalability
- Redundancy

- Typically stored on the cloud
- Does not (necessarily) use SQL (hence NoSQL)
- Examples: MongoDB, Google BigTable, Sparksee, Amazon DynamoDB

# Relational databases

**Data Model**: the abstract structure of the database. entities and their relationships

**Relational model**: the database represented as a set of tables (relations)

**Normalization**: the process of reorganizing a relational database to decrease data redundancy and increase data consistency

## Components of the Entity Relationship Model

- Entities: Real world objects
  - student, course, professor, room
- Relationships: Association between entities
  - student enrolled-in course
  - professor teaches course
  - professor advises student
  - professor has-office room
- Attributes: Properties of entities or relationships
  - student: name, id_number, major
  - professor: name, office, department
  - professor teaches course: rating

- Conceptual data model
- Models entities and relationships in the data
- Captures semantic information about the world being modeled

- Types of relationships:
  - one-one: professor has-office room
  - one-many/many-one: professor (1) advises student (many)
  - many-many: student enrolls-in course
    - one student enrolls in many courses
    - one course has many students enrolled in it

## The Entity Relationship Diagram

- Schematic view of the data model
- Diagrams a connected network of entities and relationships

# Relational model

- ER Model: An abstract representation of the data
- Relational model: The database schema
  - the physical structure of the database
- ER Model to Relational model:
  - Create tables for each entity and each relationship
  - Keys: Keys link records across tables

## Student

| ssn | f name | l name | emails | phone | phone2 | city | zip |
|---|---|---|---|---|---|---|---|
| 111-22-3333 | John | Childs | john@gmail.com, jc123@columbia.edu | 646.123.1212 | | New York | 10025 |
| 123-12-1234 | Mary | Arias | m.a@columbia.edu, ma222@hotmail.com | | 347.766.7689 | New York | 10011 |
| 555-111-7777 | Roberto | Perez | perezr@gmail.com, rp1@columbia.edu | 917.333.5479 | 415.348.4789 | San Francisco | 94110 |
| 222-33-4455 | Lila | Pennington | lilap@gmail.com, lila@mail.com | 425.123.1212 | | Seattle | 98105 |

## Course

| numb | name | room | seats |
|---|---|---|---|
| c1 | Data | 1127 | 60 |
| c2 | Python | 303 | 100 |
| c3 | corp fin | 331 | 55 |
| c4 | prod. mgmt | 1127 | 60 |
| c5 | Ethics | 303 | 100 |
| c6 | leadership | 303 | 100 |
| c7 | bus analytics | 1127 | 60 |

## Enrolls-in

| ssn | class | f name | l name | grade |
|---|---|---|---|---|
| 111-22-3333 | c1 | John | Childs | A |
| 123-12-1234 | c1 | Mary | Arias | B |
| 111-22-3333 | c2 | John | Childs | A |
| 222-33-4455 | c3 | Lila | Pennington | F |

- The process of reorganizing a database to reduce redundancies and increase integrity in the data
- Normalization makes querying more efficient and consistent
- Normalization typically addresses three types of anomalies that give rise to redundancies and inconsistencies
  - insertion anomalies
  - update anomalies
  - deletion anomalies

# Insertion Anomalies: Definition and Example



An insertion anomaly occurs when something needs to be added to the database but there is no place to add it

**Professor**

| name | offi | dept |
|---|---|---|
| Prof. | A7 | al Engineering and Operations Re |
| Prof. | A3 | al Engineering and Operations Re |
| Prof. Lee | A6 | Computer Science |
| Prof. Wu | A2 | Mechanical Engineering |

- Example:
  - The university decides to add a physics department
  - Initially, there are no faculty members
  - There is no way to add the department

**Professor**

| name | office | dept |
|---|---|---|
| Prof. Micha | A702 | IEOR |
| Prof. Robyn | A301 | IEOR |
| Prof. Lee | A673 | CS |
| Prof. Wu | A244 | MECH |

**Department**

| Code | Name |
|---|---|
| IEOR | al Engineering and Operations Re |
| MECH | Mechanical Engineering |
| CS | Computer Science |
| PHYS | Physics |

Make Department into a separate entity and create a new Department —-> Professor relationship

# Update Anomalies: Definition and Example



- An update anomaly occurs when there is a change to the value of an attribute of an entity (or relationship) but that change needs to be made in multiple places
- A database with the potential for update anomalies can have redundant data and can therefore be inconsistent

**Course**

| numb | name | room | seats |
|---|---|---|---|
| c1 | Data | 1127 | 60 |
| c2 | Python | 303 | 100 |
| c3 | corp fin | 331 | 55 |
| c4 | prod. | 1127 | 60 |
| c5 | Ethics | 303 | 100 |
| c6 | leadershio | 303 | 100 |
| c7 | bus | 1127 | 60 |

- Example:
  - Room 1127 is restructured and the number of seats increased to 70
  - The change needs to be made for every course that is being taught in room 1127

**Course**

| number | name | room |
|---|---|---|
| c1 | Data analytics | 1127 |
| c2 | Python | 303 |
| c3 | corp fin | 331 |
| c4 | prod. mgmt | 1127 |
| c5 | Ethics | 303 |
| c6 | leadership | 303 |
| c7 | bus analytics | 1127 |

**Room**

| numb | seats |
|---|---|
| 1127 | ~~60~~ 70 |
| 303 | 100 |
| 331 | 55 |

- Solution:
  - Make Room into a separate table

# Deletion Anomalies: Definition and Example



- A deletion anomaly occurs when deleting something from the database results in some other, possibly important, fact being deleted as well
- A database with the potential for deletion anomalies can lose data

**Course**

| numbe | name | room | enrollment |
|---|---|---|---|
| c1 | Data | 1127 | 6 |
| c2 | Python | 303 | 30 |
| c3 | corp fin | 331 | 25 |
| c4 | prod. | 1127 | 45 |
| c5 | Ethics | 303 | 32 |
| c6 | leadership | 303 | 17 |
| c7 | bus | 1127 | 31 |

- Because of insufficient enrollment, course c1 - Data Analytics is dropped for the semester
- Our database no longer has a record of the course
- And we lose useful information like course description, teacher ratings, past enrollments, etc.

**Offered courses**

| number | room | enrollment |
|---|---|---|
| c1 | 1127 | 6 |
| c2 | 303 | 30 |
| c3 | 331 | 25 |
| c4 | 1127 | 45 |
| c5 | 303 | 32 |
| c6 | 303 | 17 |
| c7 | 1127 | 31 |

**Courses**

| number | name |
|---|---|
| c1 | Data analytics |
| c2 | Python |
| c3 | corp fin |
| c4 | prod. mgmt |
| c5 | Ethics |
| c6 | leadership |
| c7 | bus analytics |

- Solution:
  - Remove course number and title and put them into a separate table

# Normalized Database



- Database normalization is the process of removing anomalies
- Any relational database can be said to be in one of several normal forms: 1NF, 2NF, 3NF, Boyce-Codd NF and 4NF
- A database in 3NF is generally free of insertion, update, and deletion anomalies
- We won't study these normal forms but you should be aware they exist


www.emeritus.org