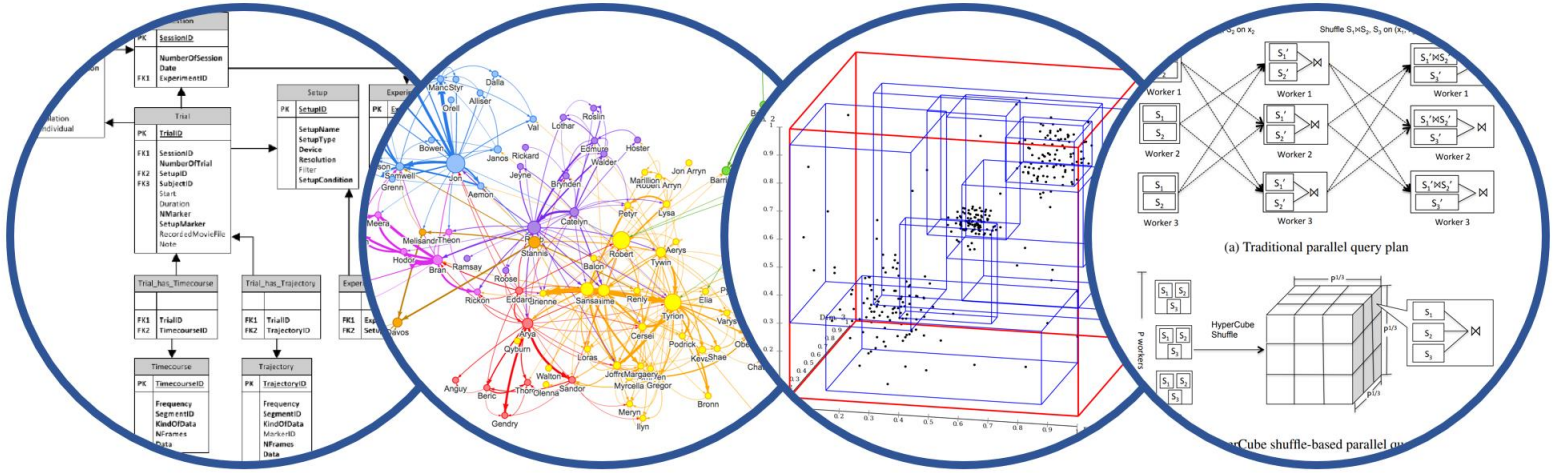


CSE 344: Intro to Data Management

Introduction

Paul G. Allen School of Computer Science and Engineering
University of Washington, Seattle

Outline

1. Administrivia
2. Databases, DBMS
3. The Relational Data Model

344 Staff

Instructor:

- Dan Suciu, suciu@cs

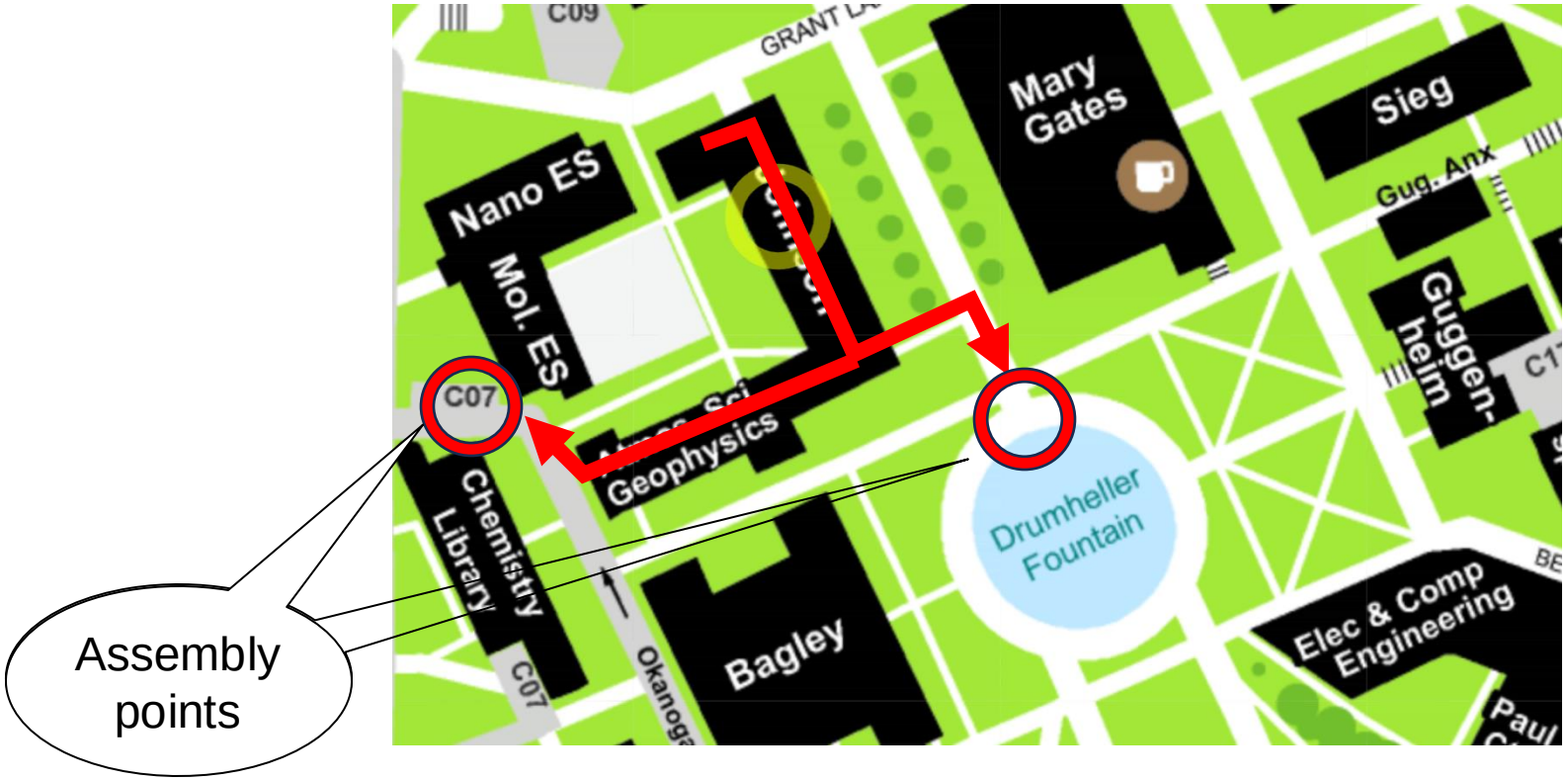
TAs:

- Rohini Arangam
- Cindy Fu
- Elvin Fu
- Christopher Erik Hunt
- Amal Jacob
- Sumedh Panatula
- Sai Likhitha Sunku
- Jay Yu

Emergency Procedures

Before we start:

- Quick, mandatory review of emergency procedure



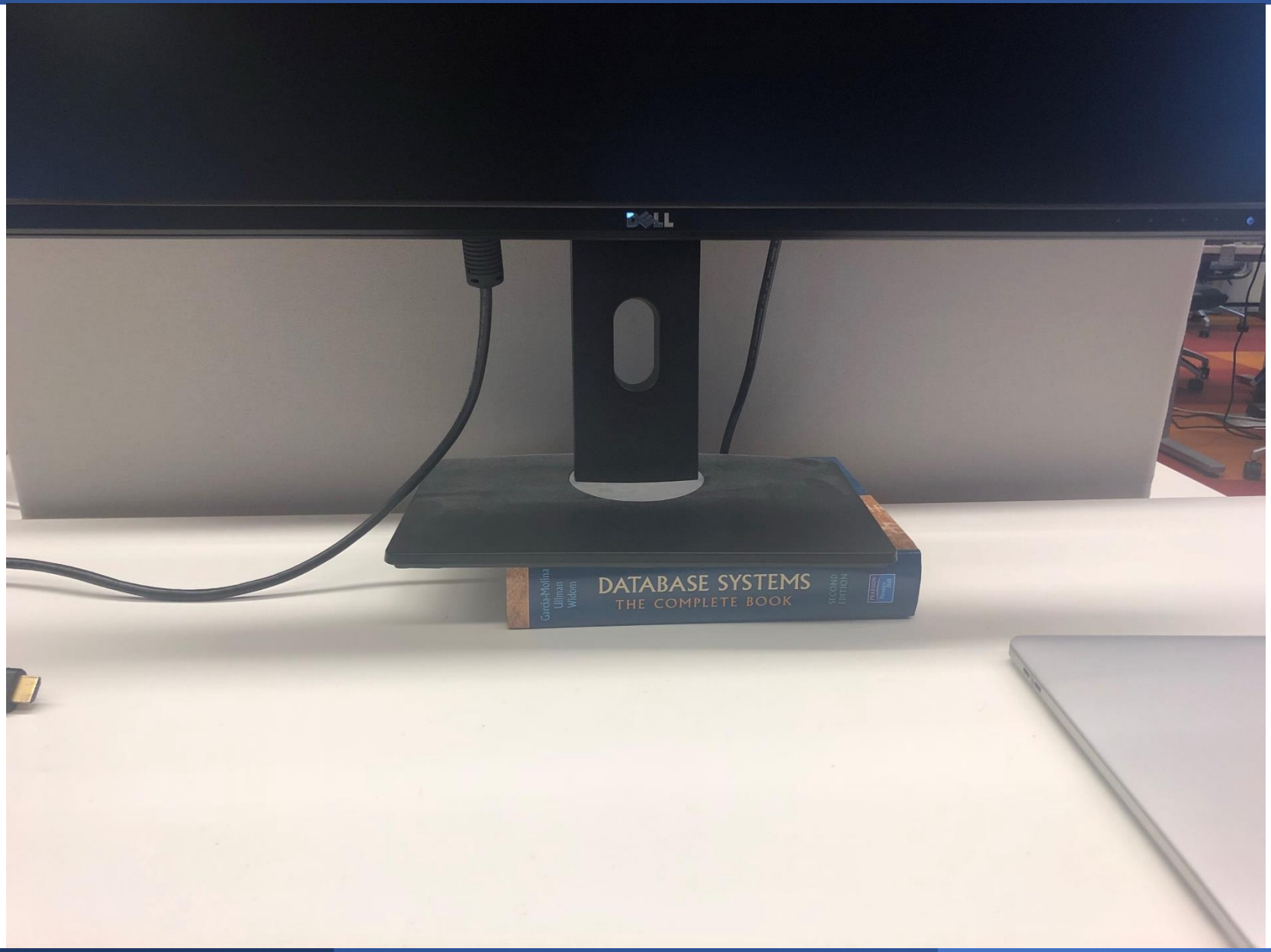
Course Format

- Lectures: in person, in this room
  - Attend. Arrive on time. Pay attention.
- Sections: in person, see locations at my.uw.edu
  - **Bring your laptop!**
- Several homework assignments
  - First assignment published on gradescope, due 10/2
- Two exams:
  - Midterm: Friday, Oct 25, 9:30-10:20 this room
  - Final: Wednesday, Dec 11, 8:30-10:20 this room

Communication

- Website:
  - www.cs.uw.edu/344 same as:
  - https://sites.google.com/cs.washington.edu/cse344-24au/home
  - Start here: it contains LOTs of information
- Canvas: zoom link+recordings
- Gradescope: homework submissions
- Ed message board: ask your question here
- Class mailing list: very low traffic

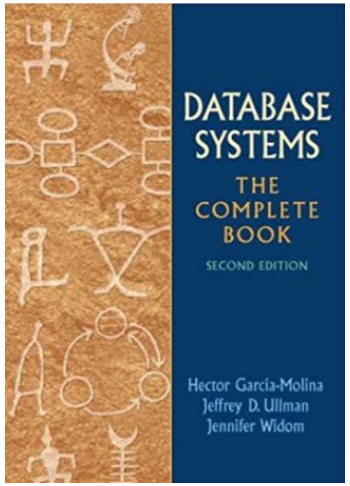
Textbook


DELL
DATABASE SYSTEMS
THE COMPLETE BOOK
Garcia-Molina
Ullman
Widom
SECOND EDITION

Textbook

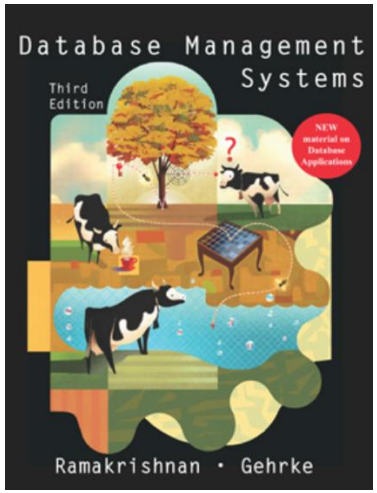
Main textbook, available at the bookstore or pdf:

- Database Systems: The Complete Book, 2nd edition



Also useful:

- Database Management Systems 3rd edition



Grading

- Grading:
  - Homeworks 50%, Exams 20%+30%
- Late days:
  - 6 in total, max 2/assignment in 24 hours chunks
  - **No late submissions accepted after that**
- Collaboration:
  - Do complete homeworks individually
  - Do discuss concepts, but see previous item
  - Don’t show your work
  - Don’t post it on the Web
  - Don’t look at other peoples’ work
  - Don’t use AI tools to produce your work

Questions?

Questions?

Let’s get started!

Database

What is a database ?

Give examples of databases

Database

What is a database ?

- A collection of files storing related data

Give examples of databases

Database

What is a database ?

- A collection of files storing related data

Give examples of databases

- Accounts database
- Payroll database
- UW’s student database
- Amazon’s products database
- Airline reservation database

Database Management System

What is a DBMS ?

Database Management System

What is a DBMS ?

- *“A big program written by someone else that allows us to manage efficiently a large database and allows it to persist over long periods of time”*

Database Management System

What is a DBMS ?

- *“A big program written by someone else that allows us to manage efficiently a large database and allows it to persist over long periods of time”*

Examples of DBMSs

- Oracle, IBM DB2, Microsoft SQL Server, Vertica, Teradata
- Cloud: Snowflake, Redshift, BigQuery, SQL Azure
- Open source: MySQL (Sun/Oracle), PostgreSQL, DuckDB
- Open source library: **SQLite**

Database Management System

What is a DBMS ?

- *“A big program written by someone else that allows us to manage efficiently a large database and allows it to persist over long periods of time”*

Examples of DBMSs

- Oracle, IBM DB2, Microsoft SQL Server, Vertica, Teradata
- Cloud: Snowflake, Redshift, BigQuery, SQL Azure
- Open source: MySQL (Sun/Oracle), PostgreSQL, DuckDB
- Open source library: **SQLite**

A DBMS needs a Data Model

Relational Model

Data Models

Data Model = mathematical definition of data

There are lots of models out there!

- Relational
- Semi-structured
- Key-value pairs
- Graph
- OO
- …

Data Models

Data Model = mathematical definition of data

There are lots of models out there!

- **Relational**
- Semi-structured
- Key-value pairs
- Graph
- OO
- …

The Relational Model

A Relational Model of Data for Large Shared Data Banks

E. F. CODD
IBM Research Laboratory, San Jose, California

Future users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation). A prompting service which supplies such information is not a satisfactory solution. Activities of users at terminals and most application programs should remain

The relational view (or model) of data described in Section 1 appears to be superior in several respects to the graph or network model [3, 4] presently in vogue for non-inferential systems. It provides a means of describing data with its natural structure only—that is, without superimposing any additional structure for machine representation purposes. Accordingly, it provides a basis for a high level data language which will yield maximal independence between programs on the one hand and machine representation and organization of data on the other.

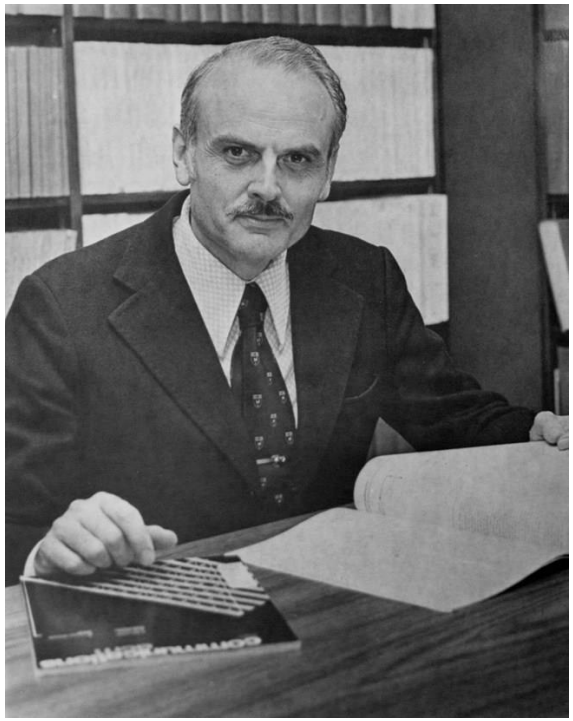
A further advantage of the relational view is that it forms a sound basis for treating derivability, redundancy, and consistency of relations—these are discussed in Section

systems has been to deductive question-answering systems. Levein and Maron [2] provide numerous references to work in this area.

In contrast, the problems treated here are those of *data independence*—the independence of application programs and terminal activities from growth in data types and changes in data representation—and certain kinds of *data inconsistency* which are expected to become troublesome even in nondeductive systems.

element to participate in several orderings. Let us consider those existing systems which either require or permit data elements to be stored in at least one total ordering which is closely associated with the hardware-determined ordering of addresses. For example, the records of a file concerning parts might be stored in ascending order by part serial number. Such systems normally permit application programs to assume that the order of presentation of records from such a file is identical to (or is a subordering of) the

The Relational Model



Ted Codd

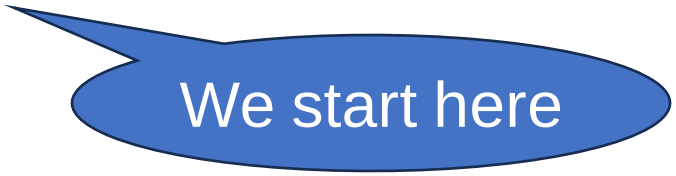
Turing Award 1981

The Relational Model

- Data is stored in simple, flat relations
- Is retrieved via a set-at-a-time query language
- No prescription for the physical representation

The Relational Model

- Data is stored in simple, flat relations



- Is retrieved via a set-at-a-time query language

- No prescription for the physical representation

Relational Model: an Example

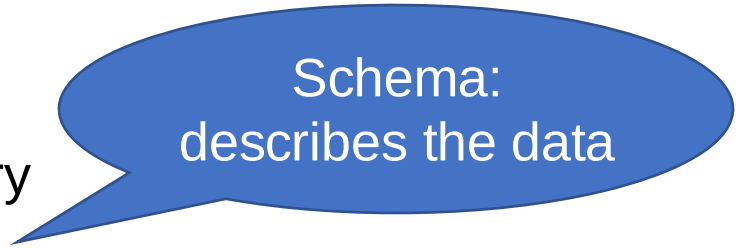
The Payroll department needs to store data about employees:

their IDs, names, job titles, and salary

Relational Model: an Example

The Payroll department needs to store data about employees:

their IDs, names, job titles, and salary



Payroll (UserId, Name, Job, Salary)

Relational Model: an Example

The Payroll department needs to store data about employees:

their IDs, names, job titles, and salary

Schema: describes the data

Payroll (UserId, Name, Job, Salary)

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

Instance: the actual data

Terminology

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

Terminology

Table/ Relation

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

Terminology

Table/ Relation

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

Rows/ Tuples/ Records

Terminology

Table/ Relation

Columns/Attributes/Fields

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

Rows/ Tuples/ Records

Every Data Model has Three Parts

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

Every Data Model has Three Parts

- Instance: the actual data

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

Every Data Model has Three Parts

- Instance: the actual data
- Schema: its type

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

Every Data Model has Three Parts

- Instance: the actual data
- Schema: its type
- Query Language: how to get the data

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

```
SELECT Name
FROM Payroll
WHERE Salary > 70000;
```

Discussion of the Relational Model

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

Discussion of the Relational Model

- Set semantics: order doesn't matter

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

Discussion of the Relational Model

- Set semantics: order doesn't matter
- Duplicates not allowed

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000
789	Dan	Prof	100000

Violates set semantics!

Discussion of the Relational Model

- Set semantics: order doesn't matter
- Duplicates not allowed

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000
789	Dan	Prof	100000

Allowed by systems, but bad idea

…then we call this collection a **bag** instead of a **set**

Discussion of the Relational Model

- Set semantics: order doesn't matter
- Duplicates not allowed
- Attrs have types

Payroll

UserID	Name	Job	Salary
123	Jack	TA	banana
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

Violates attribute type assuming INT

Discussion of the Relational Model

- Set semantics: order doesn't matter
- Duplicates not allowed
- Attrs have types
- Tables are **flat**

No sub-tables allowed!

UserID	Name	Job	Salary
123	Jack	JobName: TA, DaysPerWeek: 3 JobName: Lecturer, DaysPerWeek: 2	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

Recap: The Relational Model

- Data is stored in simple, flat relations

- Is retrieved via a set-at-a-time query language

- No prescription for the physical representation

Recap: The Relational Model

- Data is stored in simple, flat relations

- Is retrieved via a set-at-a-time query language

- No prescription for the physical representation

Recap: The Relational Model

- Data is stored in simple, flat relations

- Is retrieved via a set-at-a-time query language

- No prescription for the physical representation