Introduction to Databases

MSCI346
Lukasz Golab

Acknowledgement: slides derived from material provided by
Sliberschatz, Korth and Sudarshan, copyright 2019, www.db-book.com

Learning Outcomes

- Introduction to database management systems
- Introduction to MSCI346
- Textbook reading (6th ed.): Chapter 1

What is a Database?

- A collection of data
- How do you store “simple” data?
  - In a file
  - In a spreadsheet
  - In an address book
- Database Management System
  - A system to manage large, complex datasets (many tables), accessed by multiple users
  - Built-in language to insert/retrieve data (SQL)
  - Built-in functionality to enforce business rules
    - E.g., can’t insert two students with the same student IDs

Database Application Examples

- Enterprise Information
  - Sales: customers, products, purchases
  - Accounting: payments, receipts, assets
  - Human Resources: Information about employees, salaries, payroll taxes.
- Manufacturing: management of production, inventory, orders, supply chain.
- Banking and finance
  - customer information, accounts, loans, and banking transactions.
  - Credit card transactions
  - Finance: sales and purchases of financial instruments (e.g., stocks and bonds; storing real-time market data
- Universities: registration, grades

Database Application Examples (Cont.)

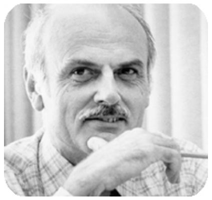
- Airlines: reservations, schedules
- Telecommunication: records of calls, texts, and data usage, generating monthly bills, maintaining balances on prepaid calling cards
- Web-based services
  - Online retailers: order tracking, customized recommendations
  - Online advertisements
- Document databases

Data Model

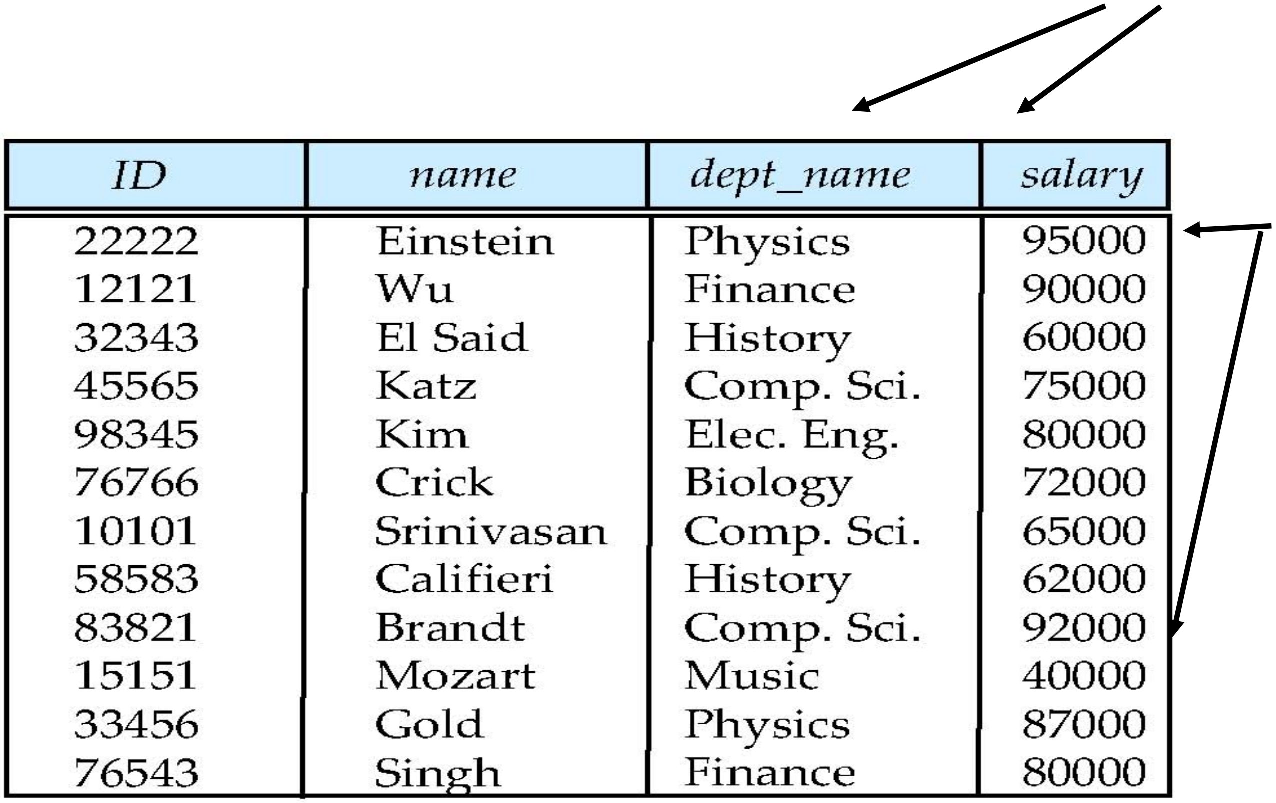
- A collection of tools for describing
  - Data
  - Data relationships
  - Data semantics
  - Data constraints
- Entity-Relationship data model (mainly for database design)
- Relational model

Relational Model

- All the data is stored in various tables.
- Example of tabular data in the relational model



Ted Codd
Turing Award 1981



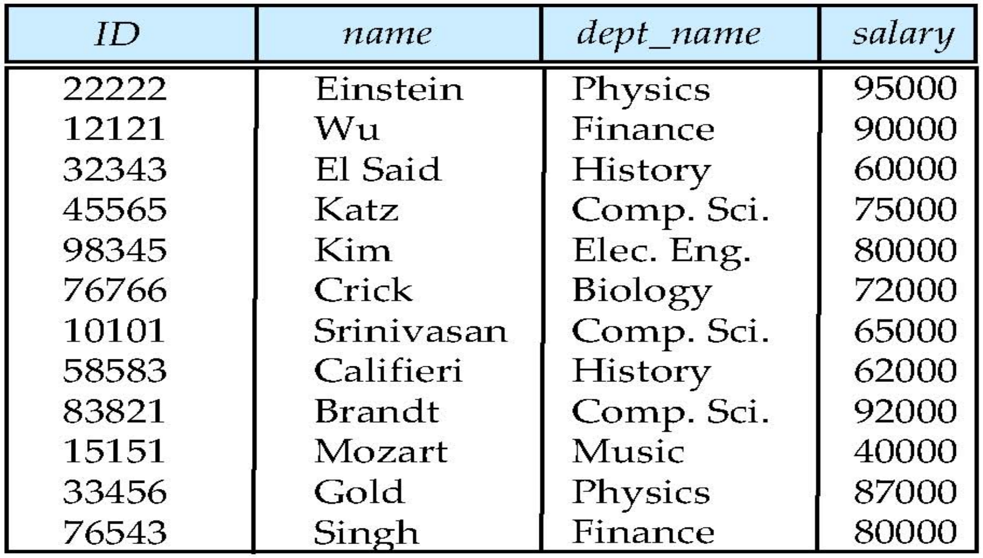
Columns

Rows

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

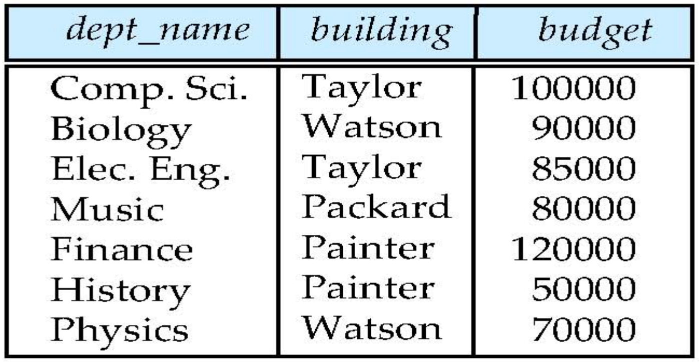
(a) The *instructor* table

A Sample Relational Database



ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

(a) The *instructor* table



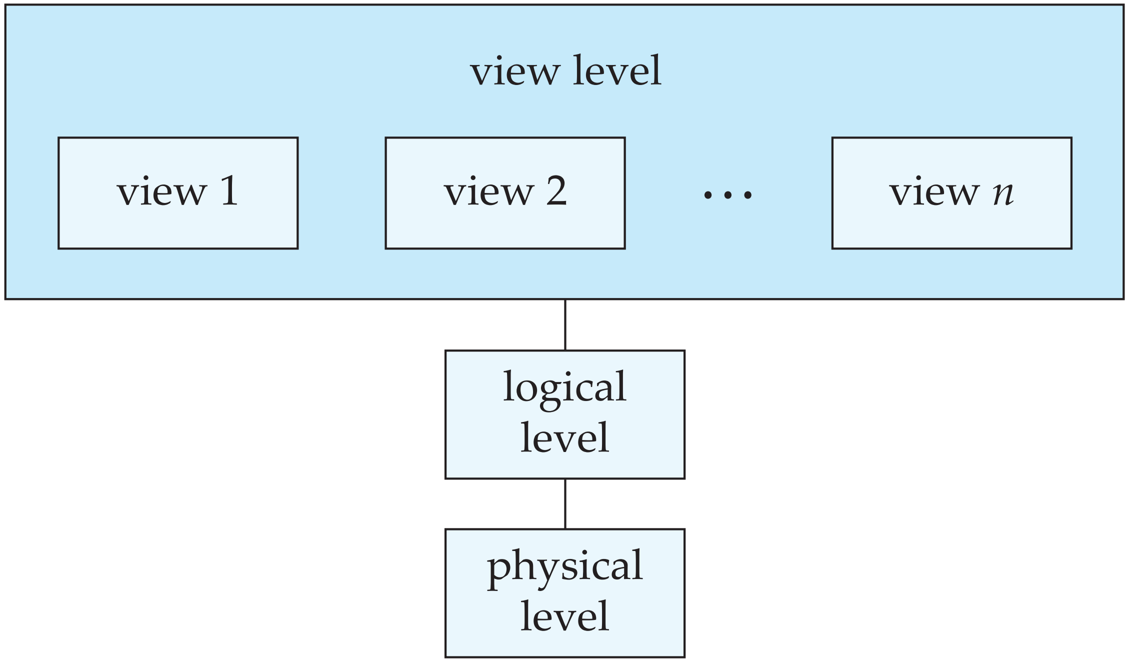
dept_name	*building*	*budget*
Comp. Sci.	Taylor	100000
Biology	Watson	90000
Elec. Eng.	Taylor	85000
Music	Packard	80000
Finance	Painter	120000
History	Painter	50000
Physics	Watson	70000

(b) The *department* table

Levels of Abstraction in a Database

- **Physical level:** describes how a record (e.g., customer) is stored.
- **Logical level:** describes the structure of the data stored in a database, and the relationships among the data.
  - e.g., a student has an ID and a name, and is enrolled in some academic program.
- **View level:** describes a virtual structure of the data imposed by the database designer on top of the logical level
  - e.g., a virtual student table with dates of birth removed

Levels of Abstraction



Data Definition Language (DDL)

- Notation for defining the database schema

Example:

```
create table instructor (
        ID          char(5),
        name        varchar(20),
        dept_name   varchar(20),
        salary      numeric(8,2))
```

SQL Query Language

- Example to find all instructors in Comp. Sci. dept

```
select name
from instructor
where dept_name = 'Comp. Sci.'
```

- Application programs generally access databases through language extensions that allow embedded SQL

In This Course

- Data modelling
  - Conceptual modelling using Entity-Relationship (ER) diagrams
  - Logical modelling using the relational model
    - Converting the entities and relationships in an ER diagram to a set of tables with corresponding business rules
      - e.g., that student IDs must be unique
- Querying the data
  - SQL
  - Embedding SQL within php for websites with a database back end
- How databases work
  - Relational algebra (the science behind SQL)
  - Concurrency control (the science behind allowing multiple users to access the same data)