Introduction

Miao Qiao

The University of Auckland

Database System Topics

- Relational Model, Relational Algebra
- SQL (Very Important!)
- Storage and Indexing
- ER Model and DB Design
- Relational DB Design and Normalization
- Query Processing
- Query Optimization
- Transaction

Teaching Team

- Lecturers
  - **Miao Qiao (Course Coordinator)**
    - Room 303-524, miao.qiao@auckland.ac.nz
    - Office hour: Wednesday 2:30pm - 3:30pm
    - Zoom Meeting ID: https://auckland.zoom.us/j/3892847086
  - **Gerald Weber**
    - Room 303-527, g.weber@auckland.ac.nz
    - Office hour: Fri 2pm.
- Tutors
  - Yizhou Dai Email: ydai992@aucklanduni.ac.nz
  - Hongyu Li Email: cli776@aucklanduni.ac.nz
  - Xizhe Zhang Email: xzha593@aucklanduni.ac.nz

Assessment

- Note: Students must obtain a pass in both the Practical (labs + assignments) and Theory (test + exam) work to pass the course as a whole.
- Assignments 35%
  - A1 due Mar 28 11:59 pm 10%
  - A2 due Apr 18 11:59 pm 10%
  - A3 due May 23 11:59 pm 15%
- Projects 15%
  - P1 due May 9 11:59 pm 10%
  - P2 due May 30 11:59 pm 5%
- Late policy:
  - Each assignment/project can be submitted up to 2 days late, with a 10% penalty per day.
- Term test 20%
  - Friday, 2/05/2025, 18:00 (end of week 7)
- Exam 30%

Expectations

- **Time Management**
  - This course is a standard 15 point course and students are expected to spend 10 hours per week on this course.
  - Learning Activities: There is a 1-2 hours of preparation required before attending the 1-hour laboratories every week.
  - Self-Directed Time: The remaining 4 to 5 hours should be devoted to self-directed research and completing assignments / exam prep.
- **Attendance**
  - Learning success highly depend on engagement, so please try and attend most lectures and tutorials.
  - Please do not come to campus if you are unwell. When you are well again, watch the recordings and view the online content to catchup.

Resources

- Lecture slides, recordings can be found on Canvas
- Reference Textbooks
  - (main) Database System Concepts, 7th edition, by Abraham Silberschatz, Henry F. Korth, S. Sudarshan, Mc Graw Hill.
    - If you want to keep an ebook we have a 20% discount code COMPSCI751 under link https://www.mheducation.com.au/database-system-concepts-ise-9781260569568-aus
  - (ref) Designing Data-Intensive Applications, by Martin Kleppmann, O'Reilly Media, Inc. Publisher.

Class Representative

- Select a class rep
- Attends 2 staff student meetings
- Pass on students’ feedback to lecturers

Database Applications Examples

- Enterprise
- Manufacturing
- Banking and finance
  - Customer information, accounts, loans, and banking transactions.
  - Credit card transactions
- Universities
  - registration, grades
- Airlines
- Telecommunication
- Web-based services
- Navigation systems

Purpose of Database Systems

- Massive
- Persistent
- Safe
- Multi-user
- Convenience
- Efficiency
- Reliable

University Database Running Example

- In this text we will be using a university database to illustrate all the concepts
- Data consists of information about:
  - Students
  - Instructors
  - Classes
- Application program examples:
  - Add new students, instructors, and courses
  - Register students for courses, and generate class rosters
  - Assign grades to students, compute grade point averages (GPA) and generate transcripts

Data Models

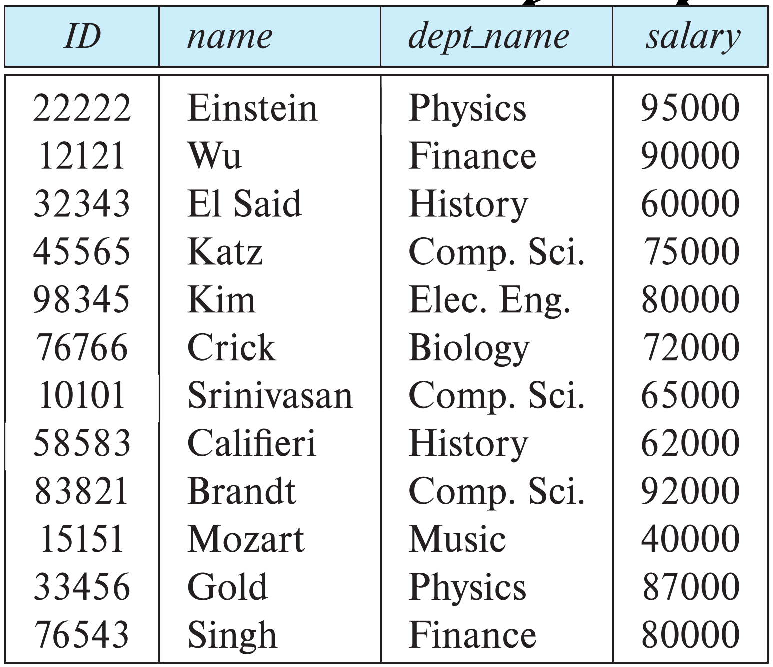
- A collection of tools for describing
  - Data
  - Data relationships
  - Data semantics
  - Data constraints
- Relational model
- Entity-Relationship data model (mainly for database design)
- Graph model (many to many relationships)
- Document (one to many relationships) XML JSON
- Key value (one to one relationships)

Relational Model

- Database = a set of relations (names) = a set of tables
- A relation has a set of attributes (names, types/domains)
- A relation has an indefinite set of tuples
- Example of tabular data in the relational model
- Schema & instances

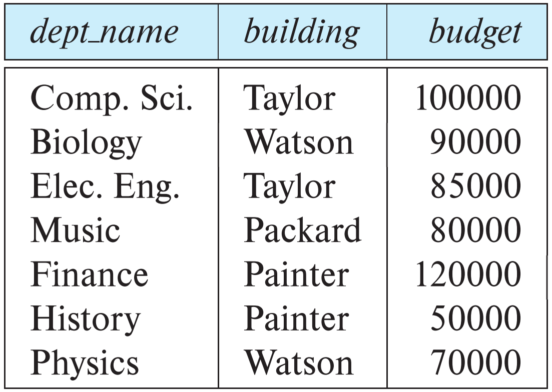
Columns / attributes

Rows / tuples



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

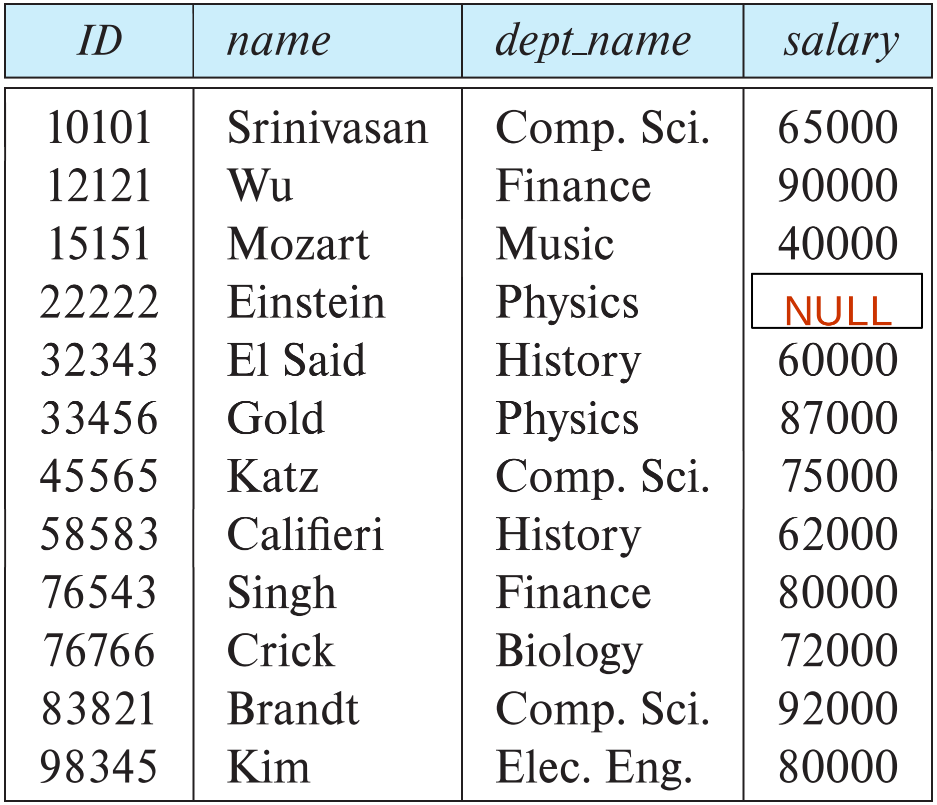


dept_name	*building*	*budget*
Comp. Sci.	Taylor	100000
Biology	Watson	90000
Elec. Eng.	Taylor	85000
Music	Packard	80000
Finance	Painter	120000
History	Painter	50000
Physics	Watson	70000

(b) The *department* table

Relational Model: Attributes

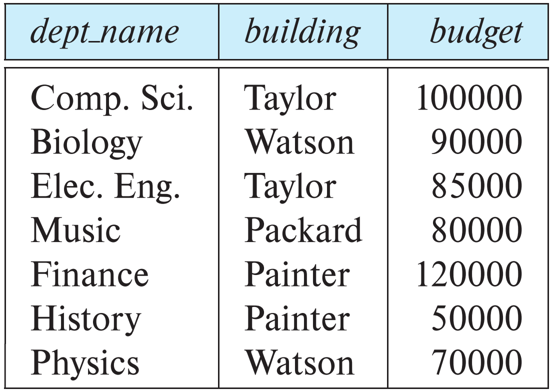
- Domain
- Atomic types / structured types
- NULL: a special value, indicating that the particular value is unknown.
- The NULL value causes complications of many operations
- Example:



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

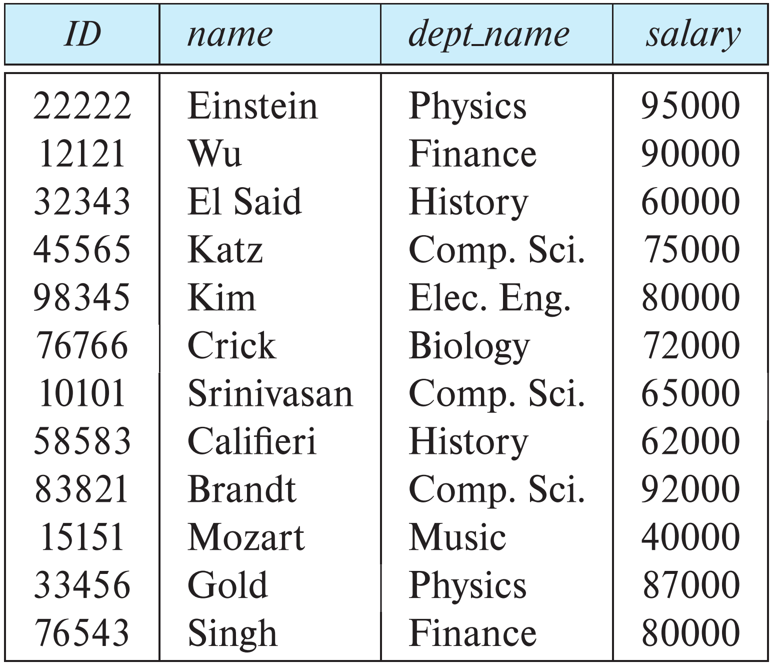
Relational Model: Keys

- Attribute of a relation where every attribute value of the relation is **unique**
- **A set of attributes** that are unique: {*ID*} and {ID,name} are both keys
- Candidate key (minimal)
- **Primary key**
- Why key is important
  - Identify specific tuples
  - Query/index efficiency
  - Reference (**Foreign keys**)



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

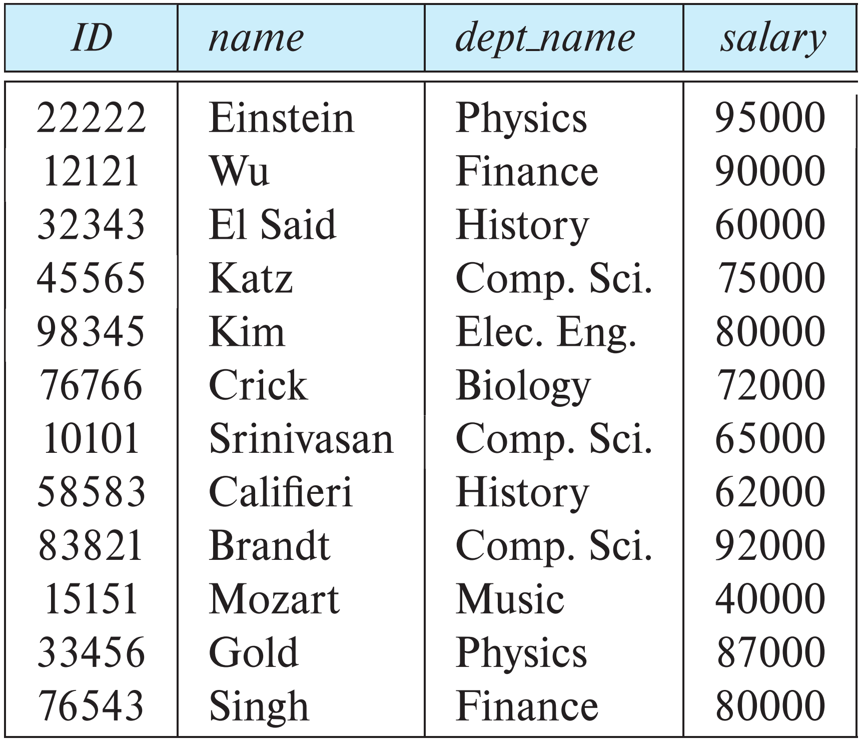


ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music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

(a) The *instructor* table

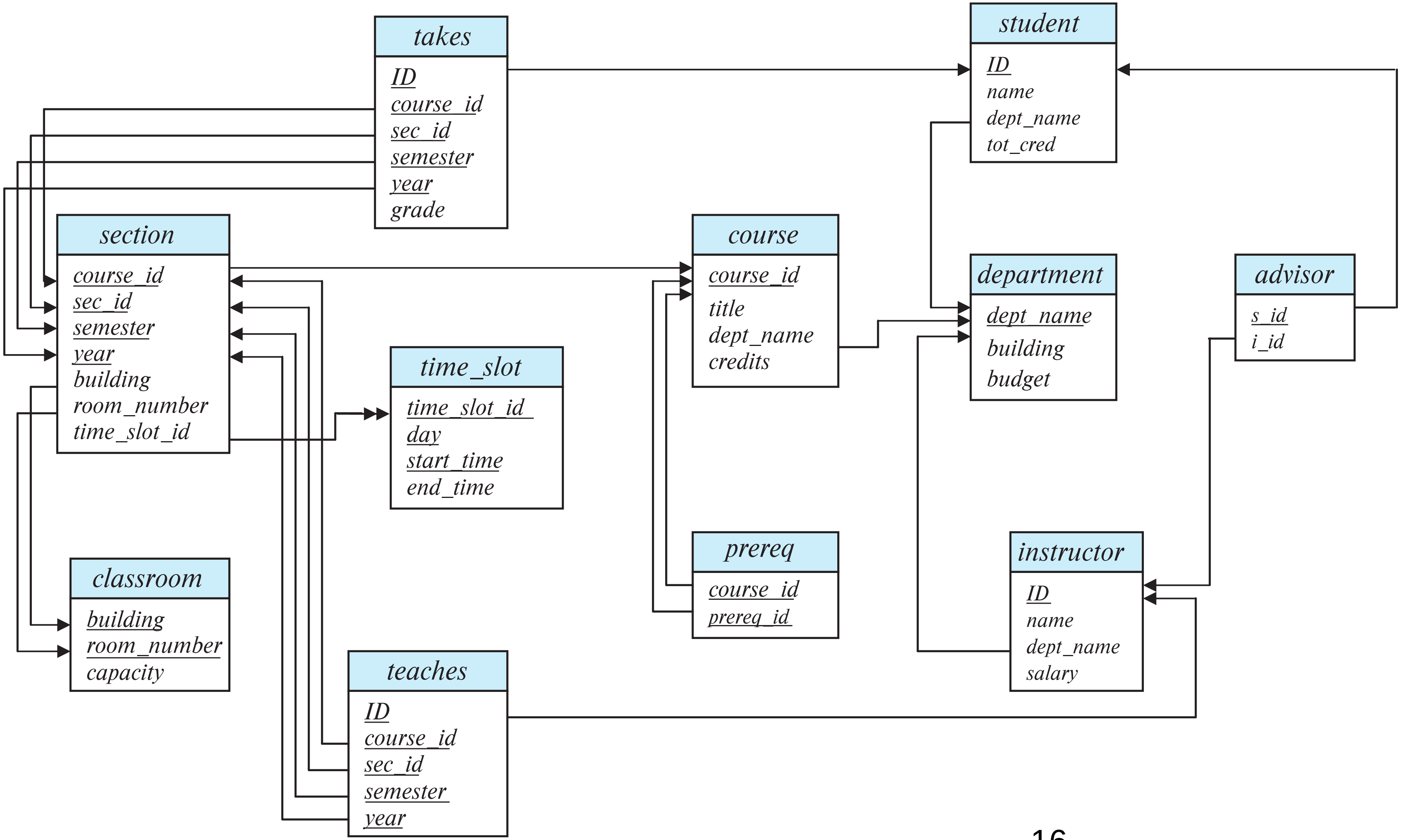
Relations are Unordered

- Order of tuples is irrelevant (tuples may be stored in an arbitrary order)
- Example: *instructor* relation with unordered tuples



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

Schema Diagram for University Database



Wrap up

- Database application examples
- Purpose of database systems
- Data models
- Relational model
  - NULL values
  - Keys
  - Schema Diagram

FIN

Any questions?