Principles of Database Systems (CS307)

Zhong-Qiu Wang

Department of Computer Science and Engineering
Southern University of Science and Technology

- Most contents are from slides made by Stéphane Faroult and the authors of Database System Concepts (7th Edition).
- Their original slides have been modified to adapt to the schedule of CS307 at SUSTech.
- The slides are largely based on the slides provided by Dr. Yuxin Ma

Instructor

- **Dr. Zhong-Qiu Wang** (王中秋)
  - Associate Professor in CSE
  - Office: Room 517, South Tower, Engineering Building
  - Homepage: https://zqwang7.github.io/
  - Email: wangzq3@sustech.edu.cn
- Background
  - Postdoc, Carnegie Mellon University, 2021/09 ~ 2024/07
  - Research Scientist, Mitsubishi Electric Research Lab, 2020/06 ~ 2021/08
  - Ph.D. student, Ohio State University, 2013/08 ~ 2020/05
  - Undergrad, Harbin Institute of Technology, 2009/08 ~ 2013/07
- Research interests: AI + Speech and audio processing
  - Perception, understanding, and generation of speech and audio signals
- Office Hour: Tuesdays 3pm-5pm @ my office

Lecture Notes, Blackboard and QQ Group

- There is a Blackboard site for this course
  - Lecture slides
  - Lab sheets
  - Assignments
  - Project requirements
  - Announcements
- Lab sessions
  - Group 1: Mon 9-10 (Zhong-Qiu Wang)
  - Group 2: Mon 9-10 (Weiyu Wang)
  - Group 3: Tue 1-2 (Weiyu Wang)
  - Group 4: Wed 1-2 (Weiyu Wang)

Lecture Notes, Blackboard and QQ Group

- QR code for the QQ group
  - Teaching assistants (TA), Weiyu, and I will be there to help
- One TA for each session

Textbooks

- Reference book:
  - A. Silberschatz, H. Korth, and S. Sudarshan. **Database System Concepts**. McGraw-Hill, New York, 7th Edition, (2019).

Grading Policy

- Lecture and Lab Attendance (10%)
  - 5% for each part
- Assignments (20%)
  - 4 – 5 assignments
- Projects (25%)
  - 2 projects (week 5 - 9, and week 10 - 14)
- Final exam (45%)

Grading Policy

- Late Submission
  - We do not accept late submissions. All assigments, quizzes, and projects, etc. will receive a score of zero if you miss the deadline.
- Groups for Projects
  - Groups across different lab sessions are not allowed
  - Please try to find your teammate in the same lab session
- Grades
  - The teachers and TAs guarantee that your assignments and projects will be evaluated carefully and unbiasedly
  - We do not accept argueing with teachers over a certain grade once the decision has been made

Plagiarism

- Please read the regulations on academic misconduct on the Blackboard site
  - Blackboard – “Declaration Form Submission”
  - All documents are in the attachment of the assignment

- Upload the signed *Undergraduate Students Declaration Form* to Blackboard with the following file name format:
  - `SID_name.pdf`
  - * The submission entry will be announced later on Blackboard

- Do NOT plagiarize / copy others’ homework, code etc.
  - Severe consequences

Some Other Stuff

- Computing technologies advance very fast
  - Search online to learn more by yourself
    - Search engines (Google, Bing, Baidu, etc.), StackOverflow, GitHub.
  - The lecture notes can guide your self study

- You are encouraged to ask questions

- Practice makes perfect

Principles of Database Systems (CS307)

Lecture 1: Introduction to Databases

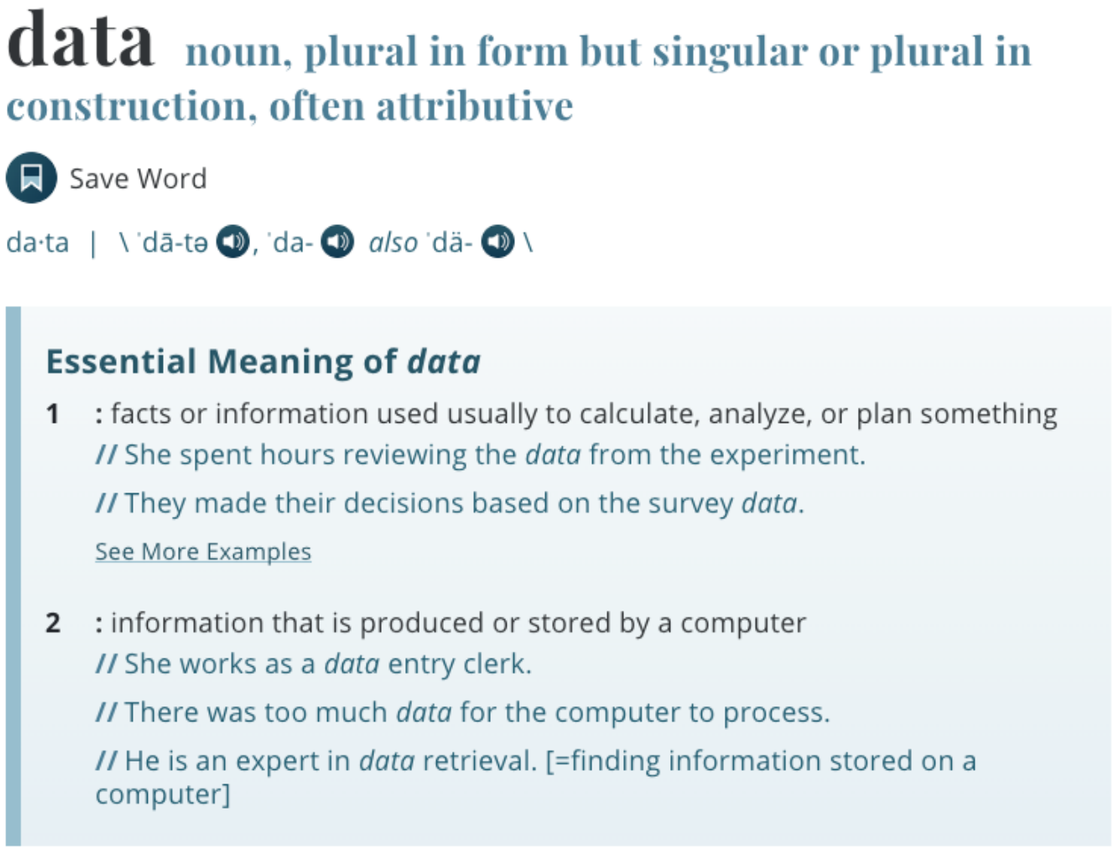
Zhong-Qiu Wang

Department of Computer Science and Engineering
Southern University of Science and Technology

- Most contents are from slides made by Stéphane Faroult and the authors of Database System Concepts (7th Edition).
- Their original slides have been modified to adapt to the schedule of CS307 at SUSTech.
- The slides are largely based on the slides provided by Dr. Yuxin Ma

What is a Database?

- Well, first, let's take a step back: What is data?



data noun, plural in form but singular or plural in construction, often attributive

Save Word

da·ta | \ ˈdā-tə, ˈda- *also* ˈdä- \

Essential Meaning of *data*

1 : facts or information used usually to calculate, analyze, or plan something
// She spent hours reviewing the *data* from the experiment.
// They made their decisions based on the survey *data*.
See More Examples

2 : information that is produced or stored by a computer
// She works as a *data* entry clerk.
// There was too much *data* for the computer to process.
// He is an expert in *data* retrieval. [=finding information stored on a computer]

What is a Database?

- A modern database system is a complex software system whose task is to manage a large, complex collection of data.
  - Maintains a collection of interrelated data
  - Provides a set of programs to access and modify the data
  - Provides an environment that is both *convenient* and *efficient* to use

- Databases touch all aspects of our lives

Applications of Database

- 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.
- Universities
  - Registration, grades

Applications of Database

- Databases are everywhere today
  - … but the concept is old
  - The idea was to have one system doing once and for all the boring data storage/retrieval part
    - Hide how the data are stored and maintained
    - What boring parts?

Purpose of Database Systems

- In the early days …
  - Database applications were built directly on top of file systems
    - Operating-system files + some applications processing the files
  - However, it suffers from many issues, including (but not limited to):
    - Data redundancy and inconsistency
    - Difficulty in accessing data
    - Data isolation
    - Integrity problems
    - Atomicity of updates
    - Concurrent access by multiple users
    - Security problems

Purpose of Database Systems

- In the early days …
  - Database applications were built directly on top of file systems
    - Operating-system files + some applications processing the files
  - However, it suffers from many issues, including (but not limited to):
    - Data redundancy and inconsistency
      - Data is stored in multiple file formats, resulting in duplication of information in different files
        - » E.g., a student with a double major
      - Only changing a subset of the files leads to inconsistency

Purpose of Database Systems

- In the early days …
  - Database applications were built directly on top of file systems
    - Operating-system files + some applications processing the files
  - However, it suffers from many issues, including (but not limited to):
    - Difficulty in accessing data
      - Need to write a new program to carry out each new task
      - Convnetional systems do not allow needed data to be retrieved in a convenient and efficient manner

Purpose of Database Systems

- In the early days …
  - Database applications were built directly on top of file systems
    - Operating-system files + some applications processing the files
  - However, it suffers from many issues, including (but not limited to):
    - Data isolation
      - Data are scattered in various files
      - Files may be in different formats
      - Writing new application programs to retrieve the appropriate data is difficult

Purpose of Database Systems

- In the early days …
  - Database applications were built directly on top of file systems
    - Operating-system files + some applications processing the files
  - However, it suffers from many issues, including (but not limited to):
    - Integrity problems
      - Integrity constraints (e.g., account balance > 0) become “buried” in program code rather than being stated explicitly
      - Hard to add new constraints or change existing ones

Purpose of Database Systems

- In the early days …
  - Database applications were built directly on top of file systems
    - Operating-system files + some applications processing the files
  - However, it suffers from many issues, including (but not limited to):
    - Atomicity of updates
      - Failures could happen anytime
      - Failures may leave database in an inconsistent state with partial updates carried out
      - Example: Transfer of funds from one account to another should either complete or not happen at all

Purpose of Database Systems

- In the early days …
  - Database applications were built directly on top of file systems
    - Operating-system files + some applications processing the files
  - However, it suffers from many issues, including (but not limited to):
    - Concurrent access by multiple users
      - Concurrent access needed for performance
      - Uncontrolled concurrent accesses can lead to inconsistencies
      - Example: Two people reading a balance (say 100) and updating it by withdrawing money (say 50 each) at the same time

Purpose of Database Systems

- In the early days …
  - Database applications were built directly on top of file systems
    - Operating-system files + some applications processing the files
  - However, it suffers from many issues, including (but not limited to):
    - Security problems
      - Not every user of the database system should be able to access all the data
      - Example: payroll personnel can only see financial information. No access to information about academic records
      - Enforcing such security constraints is difficult

Purpose of Database Systems

- In the early days …
  - Database applications were built directly on top of file systems

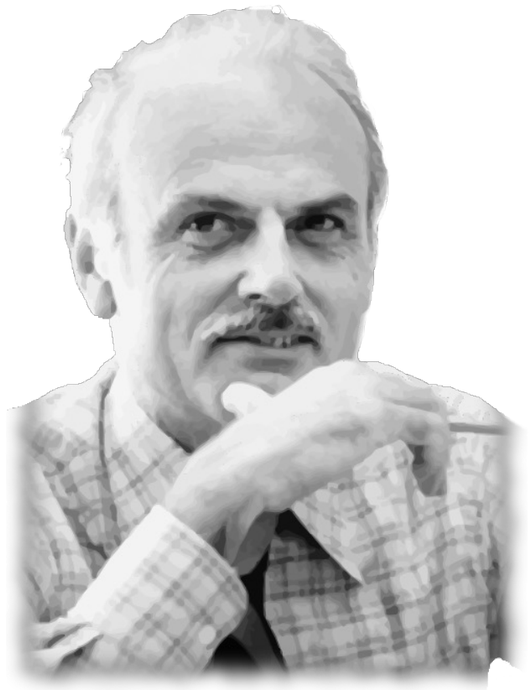
  - However, it suffers from many issues, including (but not limited to):
    - Data redundancy and inconsistency
    - Difficulty in accessing data
    - Data isolation
    - Integrity problems
    - Atomicity of updates
    - Concurrent access by multiple users
    - Security problems

Database systems offer solutions to all the problems mentioned above

A Bit of History

- 1950s and early 1960s:
  - Data processing using magnetic tapes for storage
    - Tapes provided only sequential access
  - Punched cards for input
- Late 1960s and 1970s:
  - Hard disks allowed direct access to data
  - Network and hierarchical data models in widespread use
  - **Ted Codd** defines the relational data model
    - Would win the ACM Turing Award for this work
    - IBM Research begins System R prototype
    - UC Berkeley (Michael Stonebraker) begins Ingres prototype
    - Oracle releases first commercial relational database
  - High-performance (for the era) transaction processing

A Bit of History



Edgar F. “Ted” Codd
1923 – 2003
Turing Award 1981

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 unaffected when the internal representation of data is changed and even when some aspects of the external representation are changed. Changes in data representation will often be needed as a result of changes in query, update, and report traffic and natural growth in the types of stored information.

Existing noninferential, formatted data systems provide users with tree-structured files or slightly more general network models of the data. In Section 1, inadequacies of these models are discussed. A model based on *n*-ary relations, a normal form for data base relations, and the concept of a universal

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 2. The network model, on the other hand, has spawned a number of confusions, not the least of which is mistaking the derivation of connections for the derivation of relations (see remarks in Section 2 on the “connection trap”).

Finally, the relational view permits a clearer evaluation of the scope and logical limitations of present formatted data systems, and also the relative merits (from a logical standpoint) of competing representations of data within a single system. Examples of this clearer perspective are cited in various parts of this paper. Implementations of systems to support the relational model are not discussed.

E. F. Codd, A Relational Model of Data for Large Shared Data Banks, Information Retrieval, June, 1970

A Bit of History

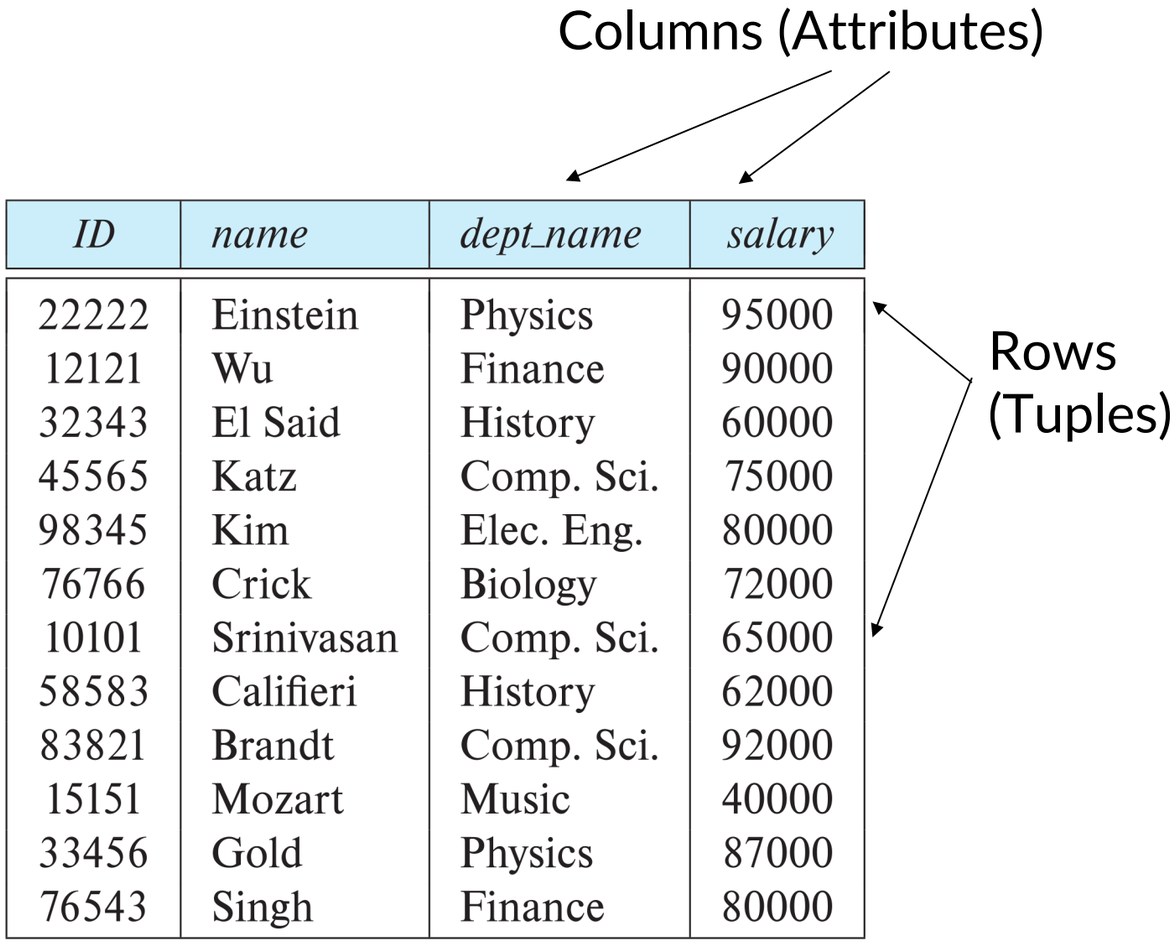
- 1980s:
  - Research relational prototypes evolve into commercial systems
    - SQL becomes industrial standard
  - Parallel and distributed database systems
    - Wisconsin, IBM, Teradata
  - Object-oriented database systems
- 1990s:
  - Large decision support and data-mining applications
  - Large multi-terabyte data warehouses
  - Emergence of Web commerce

A Bit of History

- 2000s
  - Big data storage systems
    - Google BigTable, Yahoo PNuts, Amazon,
    - “NoSQL” systems.
  - Big data analysis: beyond SQL
    - Map reduce and friends
- 2010s
  - SQL reloaded
    - SQL front end to Map Reduce systems
    - Massively parallel database systems
    - Multi-core in-memory databases

Relational Database

- Based on the relational model of data
  - Organizes data into one or more tables
  - Rows are also called records or tuples
  - Columns are also called attributes



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

Relational Database

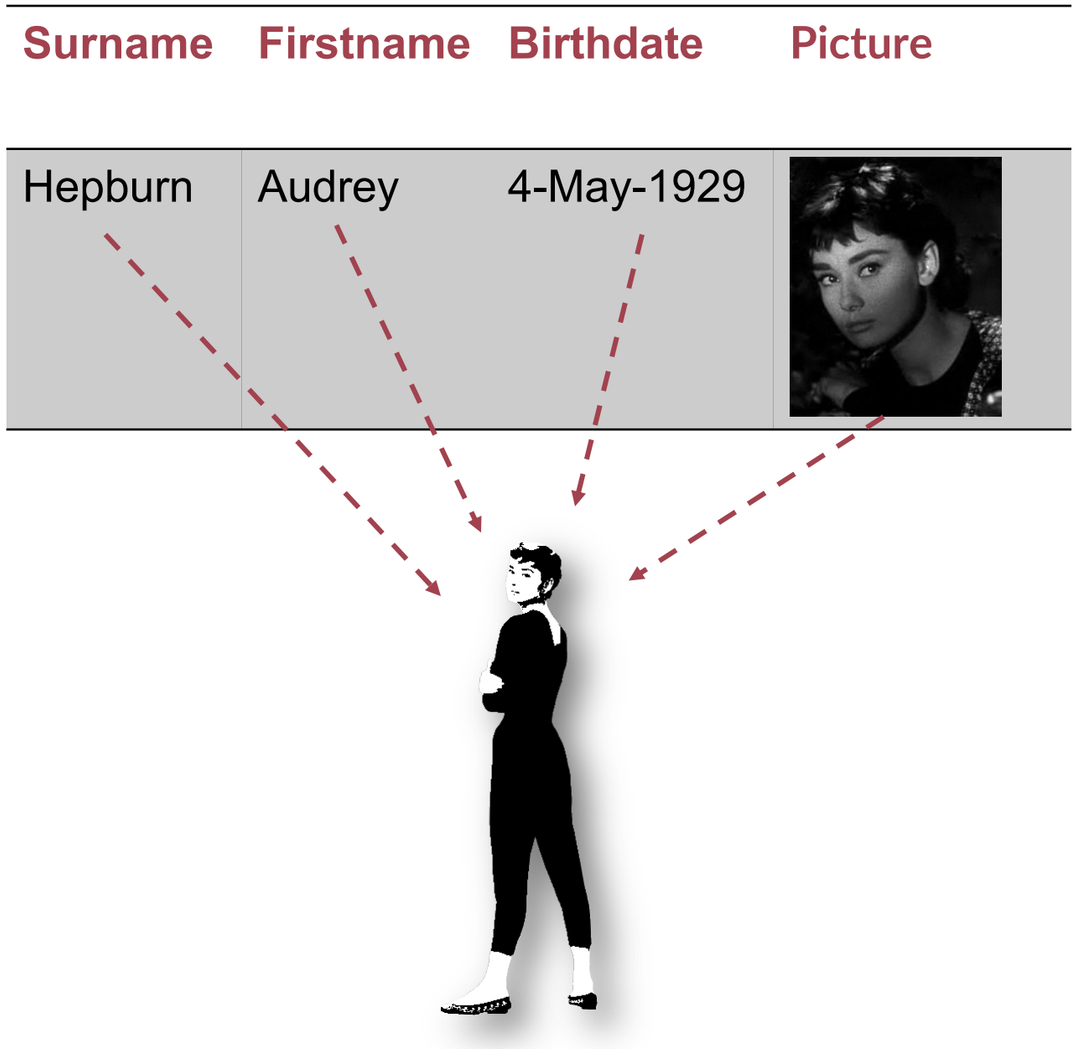
- Each column stores a piece of data

- One row represents a “known fact”:
  - “Audrey Hepburn was born on 1929/05/04 and looked like this.”

Surname	Firstname	Birthdate	Picture
Hepburn	Audrey	4-May-1929	

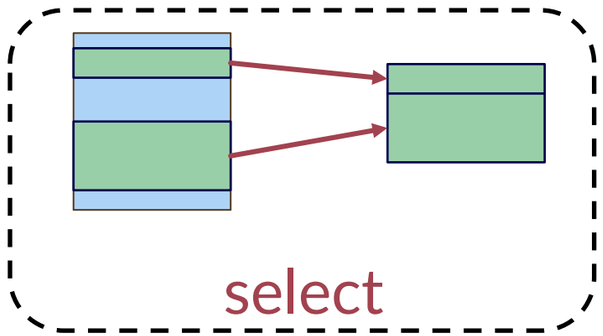
Relational Database

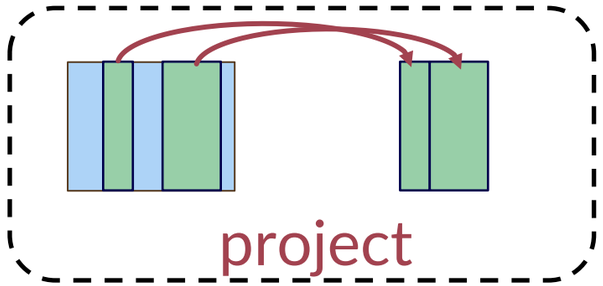
- Each column stores a piece of data
- One row represents a “known fact”:
  - “Audrey Hepburn was born on 1929/05/04 and looked like this.”
- All the pieces of data in a row are related, hence "**relational**"

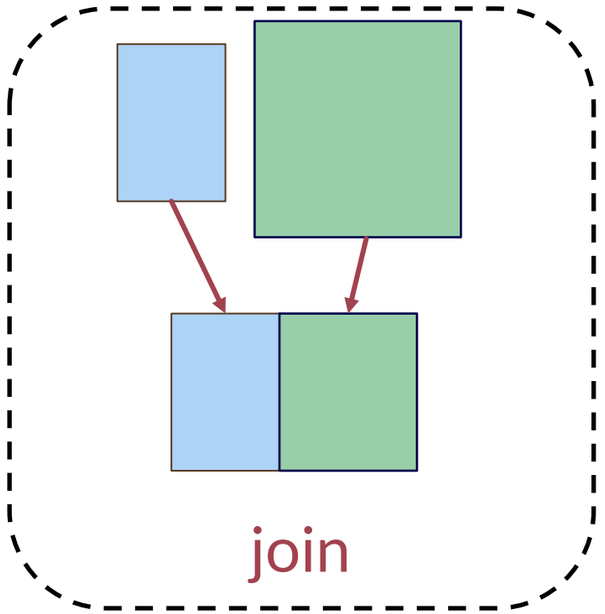


Relational Database

- But Codd's "big idea"
  - You could operate on the relations and get new sets
- Relational Algebra
  - Theoretical foundation for relational databases







Key (键)

- Example: A Film Database
  - Easy to find such as “The 100 greatest films ever”
  - Sometimes as a `.csv` file that you can load into a spreadsheet

1	Movie Title	Country	Year	Director	Starring
2	Citizen Kane	US	1941	welles, o.	Orson Welles, joseph Cotten
3	La règle du jeu	FR	1939	Renoir, J.	Roland Toutain, Nora Grégor, Marcel Dalio, Jean Renoir
4	North by Northwest	US	1959	HITCHCOCK, A.	Cary GRANT, Eva Marie SAINT, James MASON
5	Singin' In the Rain	US	1952	Donen/Kelly	Gene Kelly, Debbie Reynolds, Donald O'Connor
6	Rear Window	US	1954	HITCHCOCK, A.	James STEWART, Grace KELLY
7	City Lights	US	1931	CHAPLIN, C.	Charlie CHAPLIN, Virginia CHERRILL
8	The Third Man	GB	1949	Reed, C.	Joseph Cotten, Alida Valli, Orson Welles
9	The Searchers	US	1956	Ford, J.	John Wayne, Jeffrey Hunter, Natalie Wood
10	Ladri di biciclette	IT	1949	DeSica, V.	Lamberto Maggiorani, Enzo Staiola
11	Annie Hall	US	1977	Allen, W.	Woody Allen, Diane Keaton
12	On the Waterfront	US	1954	Kazan, E.	Marlon Brando, Eva Marie Saint, Karl Malden
13	All about Eve	US	1950	Mankiewicz, J.	Bette Davis, Anne Baxter, George Sanders
14	Casablanca	US	1942	Curtiz, M.	Humphrey Bogart,Ingrid Bergman, Claude Rains
15	The Treasure of the Sierra Madre	US	1948	HUSTON, J.	Humphrey BOGART, Walter HUSTON, Tim HOLT
16	High Noon	US	1952	Zinnemann, F.	Gary Cooper, Grace Kelly
17	Some Like It Hot	US	1959	Wilder, B.	Tony Curtis, Jack Lemmon, Marilyn Monroe
18					

Key

- Duplicates are forbidden in relational tables
  - Introduces potential errors such as when counting the number of movies

Movie Title	Country	Year	Director	Starring
Citizen Kane	US	1941	welles, o.	Orson Welles, Joseph Cotten
La règle du jeu	FR	1939	Renoir, J.	Roland Toutain, Nora Grégor, Marcel Dalio, Jean Renoir
North By Northwest	US	1959	HITCHCOCK, A.	Cary Grant, Eva Marie Saint, James Mason
Singin' in the Rain	US	1952	Donen/Kelly	Gene Kelly, Debbie Reynolds, Donald O'Connor
Rear Window	US	1954	HITCHCOCK, A.	James Stewart, Grace Kelly
North By Northwest	US	1959	HITCHCOCK, A.	Cary Grant, Eva Marie Saint, James Mason

Key

- How to identify “different rows” in a table?
  - A column (or a set of columns) to differentiate one row from another
    - In the film data … How about the movie titles?

Movie Title	Country	Year	Director	Starring
Citizen Kane	US	1941	welles, o.	Orson Welles, Joseph Cotten
La règle du jeu	FR	1939	Renoir, J.	Roland Toutain, Nora Grégor, Marcel Dalio, Jean Renoir
North By Northwest	US	1959	HITCHCOCK, A.	Cary Grant, Eva Marie Saint, James Mason
Singin' in the Rain	US	1952	Donen/Kelly	Gene Kelly, Debbie Reynolds, Donald O'Connor
Rear Window	US	1954	HITCHCOCK, A.	James Stewart, Grace Kelly

豆瓣电影 神雕侠侣

影讯&购票 选电影 电视剧 排行榜 分类 影评 2021年度榜单 2021书影音报告

搜索 神雕侠侣

神雕侠侣 神雕侠侣 (1995) [剧集] [可播放]

★★★★½ 9.2 (176564人评价)

中国香港 / 爱情 / 武侠 / 古装 / 新神雕侠侣 / Return Of The Condor Heroes / 45分钟

李添胜 / 古天乐 / 李若彤 / 白彪 / 魏秋桦 / 傅明宪 / 李绮虹 / 雪梨 / 简佩玛

话题 《神雕侠侣》哪个角色最让你惊喜?

10030人浏览 · 22篇文章

神雕侠侣 (2006) [剧集] [可播放]

★★★★☆ 7.5 (70883人评价)

中国大陆 / 武侠 / 古装 / The Return of the Condor Heroes / 40分钟

于敏 / 刘亦菲 / 黄晓明 / 陈紫函 / 杨幂 / 叮当 / 王洛勇 / 赵鸿飞 / 钱博

神雕侠侣 (2014) [剧集] [可播放]

★★½☆☆ 4.9 (26245人评价)

中国大陆 / 剧情 / 武侠 / 新神雕侠侣 / The Condor Heroes / 45分钟

李慧珠 / 邓伟恩 / 李达超 / 陈晓 / 陈妍希 / 张馨予 / 张雪迎 / 郑国霖 / 杨明娜 / 陈翔 / 毛晓彤

新神雕侠侣 (2022) [剧集]

☆☆☆☆☆ (尚未播出)

中国大陆 / 爱情 / 武侠 / 古装

林峰 / 佟梦实 / 毛晓慧 / 文淇 / 涂冰 / 邵兵 / 龚蓓苾 / 毛林林 / 宗峰岩

神雕侠侣 (2023)

☆☆☆☆☆ (尚未上映)

中国大陆 / 剧情 / 爱情 / 武侠 / 古装 / The Romance Of The Condor Heros

徐克

神雕侠侣 神雕侠侣 (1983) [剧集] [可播放]

★★★★☆ 7.9 (6328人评价)

中国香港 / 剧情 / 动作 / 爱情 / The Return of the Condor Heroes / 42分钟

范秀明 / 鞠觉亮 / 萧显辉 / 司徒立光 / 谭锐铭 / 刘德华 / 陈玉莲 / 梁家仁 / 欧阳佩珊 / 廖安丽 / 吕有慧 / 曾江

神雕侠侣 (1998) [剧集] [可播放]

★★★½☆ 7.1 (5767人评价)

新加坡 / 爱情 / 武侠 / 古装 / 神雕侠侣 98版 / Return of the Condor Heroes / 47分钟

马玉辉 / 谢敏洋 / 蔡晶盛 / 张龙敏 / 卢燕金 / 李铭顺 / 范文芳 / 朱厚任 / 何咏芳 / 林湘萍 / 丁岚 / 李南星

神雕侠侣 (1998) [剧集] [可播放]

★★½☆☆ 5.2 (10379人评价)

中国台湾 / 中国大陆 / 武侠 / 古装 / 杨过与小龙女 / 45分钟

李惠民 / 赖水清 / 任贤齐 / 吴倩莲 / 孙兴 / 季芹 / 李立群 / 夏文汐 / 蔡君茹 / 高捷

神雕侠侣 (2001) [剧集] [可播放]

★★★★☆ 7.5 (1772人评价)

日本 / 中国香港 / 动画 / 24分钟

案纳正美 / 高木淳 / 浪川大辅 / 园崎未惠 / 中田让治 / 唐泽润 / 木村亚希子 / 小村哲生 / 广田行生 / 高户靖广

神雕侠侣 (1984) [剧集]

★★★★☆ 7.5 (926人评价)

中国台湾 / 剧情 / 爱情 / 武侠 / 古装 / 95分钟

何东兴 / 孟飞 / 潘迎紫 / 傅娟

九一神雕侠侣 九一神鵰侠侶 (1991) [可播放]

★★★☆☆ 6.4 (8789人评价)

中国香港 / 动作 / 剧情 / 奇幻 / 爱情 / 科幻 / 神秘英豪 / 新神雕侠侣 / 92分钟

元奎 / 黎大炜 / 刘德华 / 梅艳芳 / 郭富城 / 叶蕴仪 / 刘嘉玲

神雕侠侣 神鵰侠侶 (1982) [可播放]

★★½☆☆ 5.2 (1402人评价)

中国香港 / 动作 / 爱情 / 武侠 / 古装 / 射雕英雄传4 / Brave Archer 4 / 100分钟

张彻 / 江生 / 郭追 / 傅声 / 龙天翔 / 黄淑仪

九二神雕之痴心情长剑 九二神鵰之痴心情長劍 (1992) [可播放]

★★★☆☆ 6.0 (6152人评价)

中国香港 / 喜剧 / 爱情 / 奇幻 / 武侠 / 神秘情侠 / 新神雕侠侣2(台) / 92分钟

元奎 / 黎大炜 / 刘德华 / 关之琳 / 吴耀汉 / 关淑怡

神雕侠侣 (1976) [剧集]

☆☆☆☆☆ (暂无评分)

中国香港 / 古装

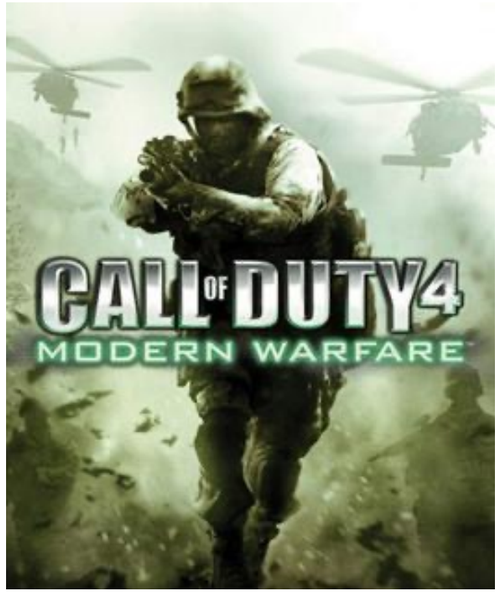
萧笙 / 罗乐林 / 李通明 / 白彪 / 米雪 / 曾江 Kenneth Tsang / 秦煌 / 郑裕玲 / 冯淬帆

Key

- How to identify “different rows” in a table?
  - A column (or a set of columns) to differentiate one row from another
    - In the film data … ~~How about the movie titles?~~ Title + Director?

Movie Title	Country	Year	Director	Starring
Citizen Kane	US	1941	welles, o.	Orson Welles, Joseph Cotten
La règle du jeu	FR	1939	Renoir, J.	Roland Toutain, Nora Grégor, Marcel Dalio, Jean Renoir
North By Northwest	US	1959	HITCHCOCK, A.	Cary Grant, Eva Marie Saint, James Mason
Singin' in the Rain	US	1952	Donen/Kelly	Gene Kelly, Debbie Reynolds, Donald O'Connor
Rear Window	US	1954	HITCHCOCK, A.	James Stewart, Grace Kelly

But, Hitchcock did It (So it was with the COD game series)



Same developer (Infinite Ward)
Same publisher (Activision)
Two different years (2007, 2019)

Key

- How to identify “different rows” in a table?
  - A column (or a set of columns) to differentiate one row from another
    - In the film data … ~~How about the movie titles? Title + Director?~~
    - Title + Director + Year?

Movie Title	Country	Year	Director	Starring
Citizen Kane	US	1941	welles, o.	Orson Welles, Joseph Cotten
La règle du jeu	FR	1939	Renoir, J.	Roland Toutain, Nora Grégor, Marcel Dalio, Jean Renoir
North By Northwest	US	1959	HITCHCOCK, A.	Cary Grant, Eva Marie Saint, James Mason
Singin' in the Rain	US	1952	Donen/Kelly	Gene Kelly, Debbie Reynolds, Donald O'Connor
Rear Window	US	1954	HITCHCOCK, A.	James Stewart, Grace Kelly

Key

- Ok, you have made it this time, but –
  - The combination is too difficult for either remembering or computing
    - Need to compare multiple times on different columns
    - Think about deduplication
  - What if there are multiple items in a single column?
    - For example, more than one directors in a movie?

Primary Key (主键)

Additional material about creating a unique ID for each row:
`https://en.wikipedia.org/wiki/Universally_unique_identifier`

- Some of the keys may be unique for every row
  - Student ID, Email address, 18-digit ID number, etc.
- Usually, it is a good practice to choose the simplest one
  - (Or, create one)

Movie ID	Movie Title	Country	Year	Director	Starring
0	Citizen Kane	US	1941	welles, o.	Orson Welles, Joseph Cotten
1	La règle du jeu	FR	1939	Renoir, J.	Roland Toutain, Nora Grégor, Marcel Dalio, Jean Renoir
2	North By Northwest	US	1959	HITCHCOCK, A.	Cary Grant, Eva Marie Saint, James Mason
3	Singin' in the Rain	US	1952	Donen/Kelly	Gene Kelly, Debbie Reynolds, Donald O'Connor
4	Rear Window	US	1954	HITCHCOCK, A.	James Stewart, Grace Kelly

Normalization

- A way of standardize your data

Movie ID	Movie Title	Country	Year	Director	Starring
0	Citizen Kane	US	1941	welles, o.	Orson Welles, Joseph Cotten
1	La règle du jeu	FR	1939	Renoir, J.	Roland Toutain, Nora Grégor, Marcel Dalio, Jean Renoir
2	North By Northwest	US	1959	HITCHCOCK, A.	Cary Grant, Eva Marie Saint, James Mason
3	Singin' in the Rain	US	1952	Donen/Kelly	Gene Kelly, Debbie Reynolds, Donald O'Connor
4	Rear Window	US	1954	Alfred Hitchcock	James Stewart, Grace Kelly

Too many different ways of spelling

Normalization

- “First Norm Rule” (1NF, 第一范式)
  - Each column should only contain ONE piece of information

Director	Starring
welles, o.	Orson Welles, Joseph Cotten
Renoir, J.	Roland Toutain, Nora Grégor, Marcel Dalio, Jean Renoir
HITCHCOCK, A.	Cary Grant, Eva Marie Saint, James Mason
Donen/Kelly	Gene Kelly, Debbie Reynolds, Donald O'Connor
HITCHCOCK, A.	James Stewart, Grace Kelly

Director_Firstname	Director_Lastname
Alfred	Hitchcock
Orson	Welles

Starring_Firstname	Starring_Lastname
Orson	Welles
Joseph	Cotten

Normalization

- “First Norm Rule” (1NF, 第一范式)
  - Each column should only contain ONE piece of information

Director	Starring
welles, o.	Orson Welles, Joseph Cotten
Renoir, J.	Roland Toutain, Nora Grégor, Marcel Dalio, Jean Renoir
HITCHCOCK, A.	Cary Grant, Eva Marie Saint, James Mason
Donen/Kelly	Gene Kelly, Debbie Reynolds, Donald O'Connor
HITCHCOCK, A.	James Stewart, Grace Kelly

Director_ Firstname	Director_ Lastname	Born	Died
Alfred	Hitchcock	1899	1980
Orson	Welles	1915	1985

Extend the table to represent all directors

Normalization

- “First Norm Rule” (1NF, 第一范式)
  - Each column should only contain ONE piece of information

Movie ID	Movie Title	Country	Year	Director ID	Starring
0	Citizen Kane	US	1941	2	Orson Welles, Joseph Cotten
1	La règle du jeu	FR	1939	5	Roland Toutain, Nora Grégor, Marcel Dalio, Jean Renoir
2	North By Northwest	US	1959	1	Cary Grant, Eva Marie Saint, James Mason
3	Singin' in the Rain	US	1952	6	Gene Kelly, Debbie Reynolds, Donald O'Connor
4	Rear Window	US	1954	1	James Stewart, Grace Kelly

Director ID	Director_ Firstname	Director_L astname	Born	Died
1	Alfred	Hitchcock	1899	1980
2	Orson	Welles	1915	1985
3	….	…	…	…

Link the director information between tables

Normalization

- Normal Form (NF)
  - 1NF: Simple attributes
  - 2NF: Attributes depend on the full key
  - 3NF: Non-key attributes do not depend on each other
  - And many others

	UNF (1970)	1NF (1970)	2NF (1971)	3NF (1971)	EKNF (1982)	BCNF (1974)	4NF (1977)	ETNF (2012)	5NF (1979)	DKNF (1981)	6NF (2003)
Primary key (no duplicate tuples)[4]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Atomic columns (cells cannot have tables as values)[5]	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Every non-trivial functional dependency either does not begin with a proper subset of a candidate key or ends with a prime attribute (no partial functional dependencies of non-prime attributes on candidate keys)[5]	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓
Every non-trivial functional dependency either begins with a superkey or ends with a prime attribute (no transitive functional dependencies of non-prime attributes on candidate keys)[5]	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓
Every non-trivial functional dependency either begins with a superkey or ends with an elementary prime attribute	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	N/A
Every non-trivial functional dependency begins with a superkey	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	N/A
Every non-trivial multivalued dependency begins with a superkey	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	N/A
Every join dependency has a superkey component[8]	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	N/A
Every join dependency has only superkey components	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	N/A
Every constraint is a consequence of domain constraints and key constraints	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗
Every join dependency is trivial	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓

https://en.wikipedia.org/wiki/Database_normalization

Normalization

Every non key attribute must provide a fact about the key, the whole key, and nothing but the key.

William Kent (1936 – 2005)

William Kent. "A Simple Guide to Five Normal Forms in Relational Database Theory", Communications of the ACM 26 (2), Feb. 1983, pp. 120–125.

Normalization

Question:
What if there are multiple director?

- "First Norm Rule" (1NF, 第一范式)
  - Each column should only contain ONE piece of information

Movie ID	Movie Title	Country	Year	Director ID	Starring
0	Citizen Kane	US	1941	2	Orson Welles, Joseph Cotten
1	La règle du jeu	FR	1939	5	Roland Toutain, Nora Grégor, Marcel Dalio, Jean Renoir
2	North By Northwest	US	1959	1	Cary Grant, Eva Marie Saint, James Mason
3	Singin' in the Rain	US	1952	6	Gene Kelly, Debbie Reynolds, Donald O'Connor
4	Rear Window	US	1954	1	James Stewart, Grace Kelly

Director ID	Director_ Firstname	Director_Lastname	Born	Died
1	Alfred	Hitchcock	1899	1980
2	Orson	Welles	1915	1985
3	....	...	...	...

Link the director information between tables

Entity and Relationship

- A bad idea: Add more columns in the Movie table

Movie ID	Movie Title	Country	Year	Director ID	Director 2 ID	Director 3 ID	Starring

  - Waste of space (not too many movies have 3 directors, let alone 6)

Entity and Relationship

- A bad idea: Add more columns in the Movie table

Movie ID	Movie Title	Country	Year	Director ID	Director 2 ID	Director 3 ID	Starring

  - Waste of space (not too many movies have 3 directors, let alone 6)
  - How about starring? 10+ more columns?

Entity and Relationship

- Further refactoring of the tables …

Movie ID	Movie Title	Country	Year
0	Citizen Kane	US	1941
1	La règle du jeu	FR	1939
2	North By Northwest	US	1959
3	Singin' in the Rain	US	1952
4	Rear Window	US	1954

Movie Entities

Relationship?

Director ID	Director_ Firstname	Director_ Lastname	Born	Died
1	Alfred	Hitchcock	1899	1980
2	Orson	Welles	1915	1985
3	….	…	…	…

Director Entities

Entity and Relationship

- Further refactoring of the tables …

Movie ID	Movie Title	Country	Year
0	Citizen Kane	US	1941
1	La règle du jeu	FR	1939
2	North By Northwest	US	1959
3	Singin' in the Rain	US	1952
4	Rear Window	US	1954

Movie Entities

Directed By

Movie ID	Director ID
0	2
1	5
2	1

Relationship!

Director ID	Director_ Firstname	Director_ Lastname	Born	Died
1	Alfred	Hitchcock	1899	1980
2	Orson	Welles	1915	1985
3	….	…	…	…

Director Entities

Entity and Relationship

- Further refactoring of the tables …

Movie ID	Movie Title	Country	Year
0	Citizen Kane	US	1941
1	La règle du jeu	FR	1939
2	North By Northwest	US	1959
3	Singin' in the Rain	US	1952
4	Rear Window	US	1954

Movie Entities

Directed By

Movie ID	Director ID
0	2
1	5
2	1
…	…
16	8
16	9
16	10

Director ID	Director_ Firstname	Director_ Lastname	Born	Died
1	Alfred	Hitchcock	1899	1980
2	Orson	Welles	1915	1985
3	….	…	…	…

Director Entities

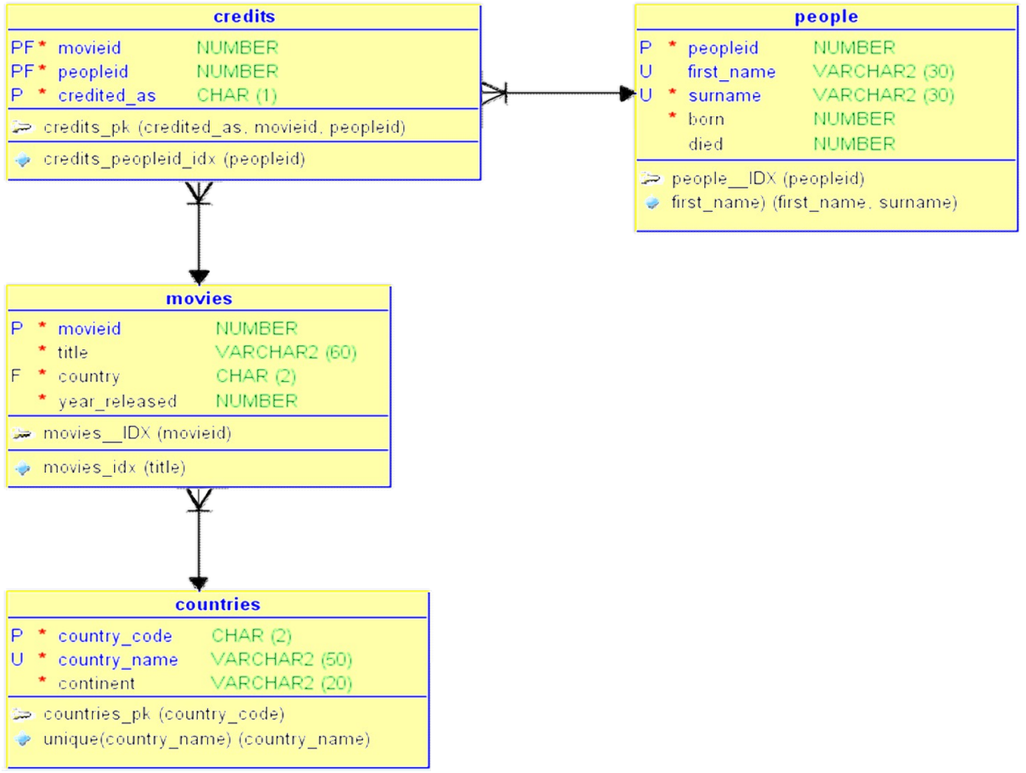
Question:
What if there are multiple director?

Answer:
Multiple rows in the relationship!

Entity and Relationship

- Starring -> Actor table
- Country -> Country and Region table
  - You can also link the movies with corresponding actors, countries/regions, etc.

- Entity Relationship Diagram (E/R Diagram, ER Diagram, ERD)
  - A way of representing entity tables and their relationships (relationship tables)



Outline

Part 1: Relational Language

Basic SQL
- Create Tables and Insert Data
- Basic Queries

Intermediate SQL
- Advanced Join
- Subquery
- Window Function
- *Transactions*

Advanced SQL
- Function
- Procedure
- Trigger
- View

Lab Only

Part 2: DB Design and Application

Relational DB Design
- The Entity-Relationship Model
- E-R Diagram Design

Relational Algebra

Normalization
- Functional Dependency
- Normal Forms
- FD Theory
- Decomposition w/ FD

Application Design
- Use Databases in Programs
- Complex Data Types (JSON, XML, etc.)

Part 3: Internal Mechanism Of Relational DB Systems

Storage Management
- Physical and Logical Storage Structure
- Indexing

Query Management
- Query Processing
- Query Optimization

Transaction Management
- Transaction
- Concurrency

Part 4: Miscellaneous Topics

Big Data
- Big Data Storage
- Query and Process Big Data

Advanced Topics
- Beyond Relational DB
- Data Analytics

Outline

- What we may not (fully) cover
  - Programming Language Support
    - We will focus on one or two languages
  - Parallel and Distributed Databases
  - Object-based Databases
  - Blockchain
  - Advanced Query Processing and Optimization Techniques
  - Advanced Relational Algebra and Calculus
  - Advanced Data Mining and Analytics
  - Implement a Relational DB from Scratch

Data Definition and Manipulation

- Data Definition Language (DDL)
  - DDL compiler generates a set of table templates stored in a data dictionary
    - Database schema
    - Integrity constraints (primary key, etc.)
    - Authorization (who can access it)

```
create table lab(
    id serial primary key,
    address varchar(20) not null,
    time varchar(20) not null,
    capacity int,
    teacher varchar(20),
    unique (address,time)
);
```

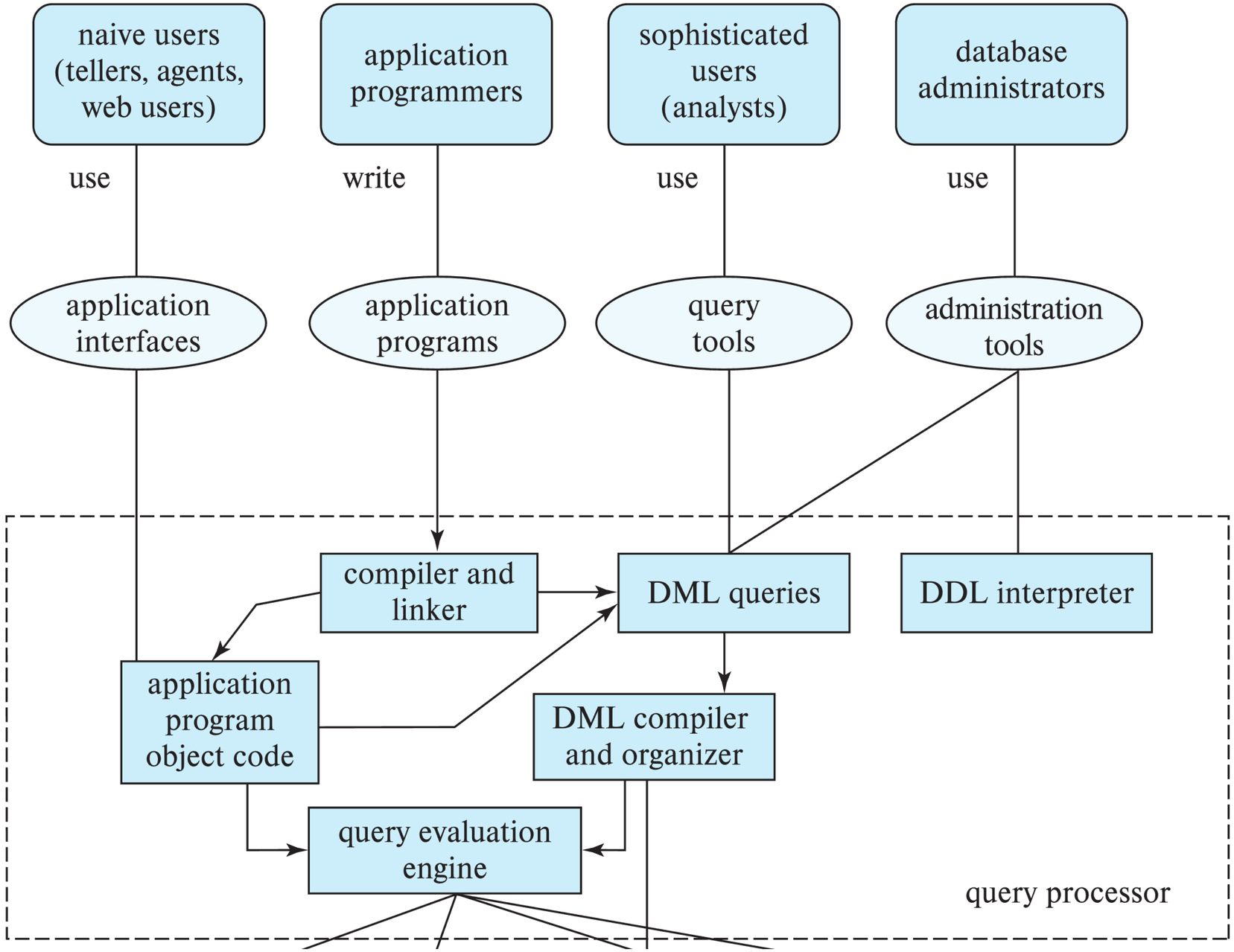
Data Definition and Manipulation

- Data Manipulation Language (DML)
  - **Language** for accessing and updating the data organized by the appropriate data model (also known as query language)
- SQL (Structured Query Language)
  - Takes several tables as input (possibly only one) and always returns a single table

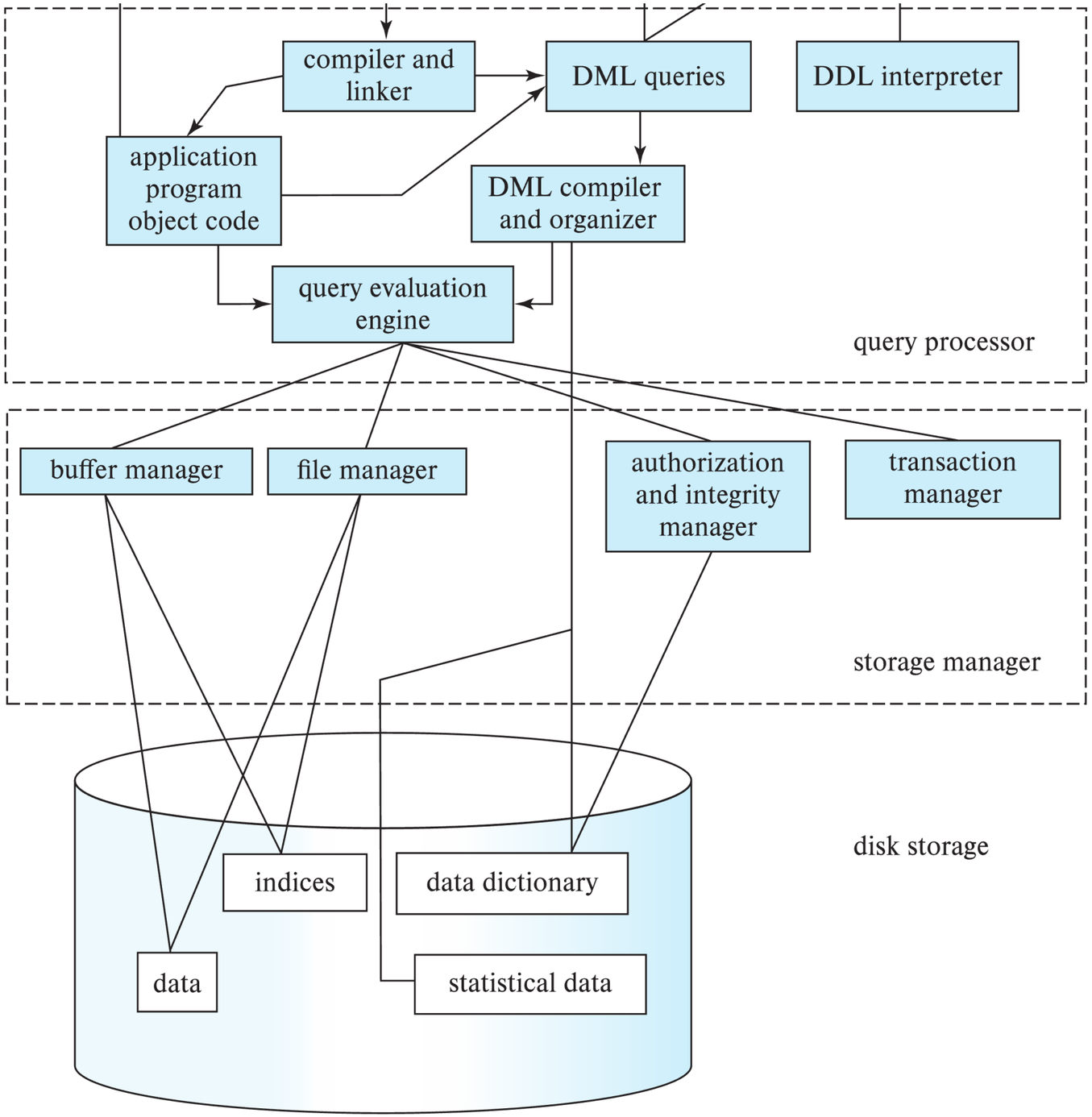
```sql
select * from lab;

select * from lab where time = '3-34';
```

Database Users



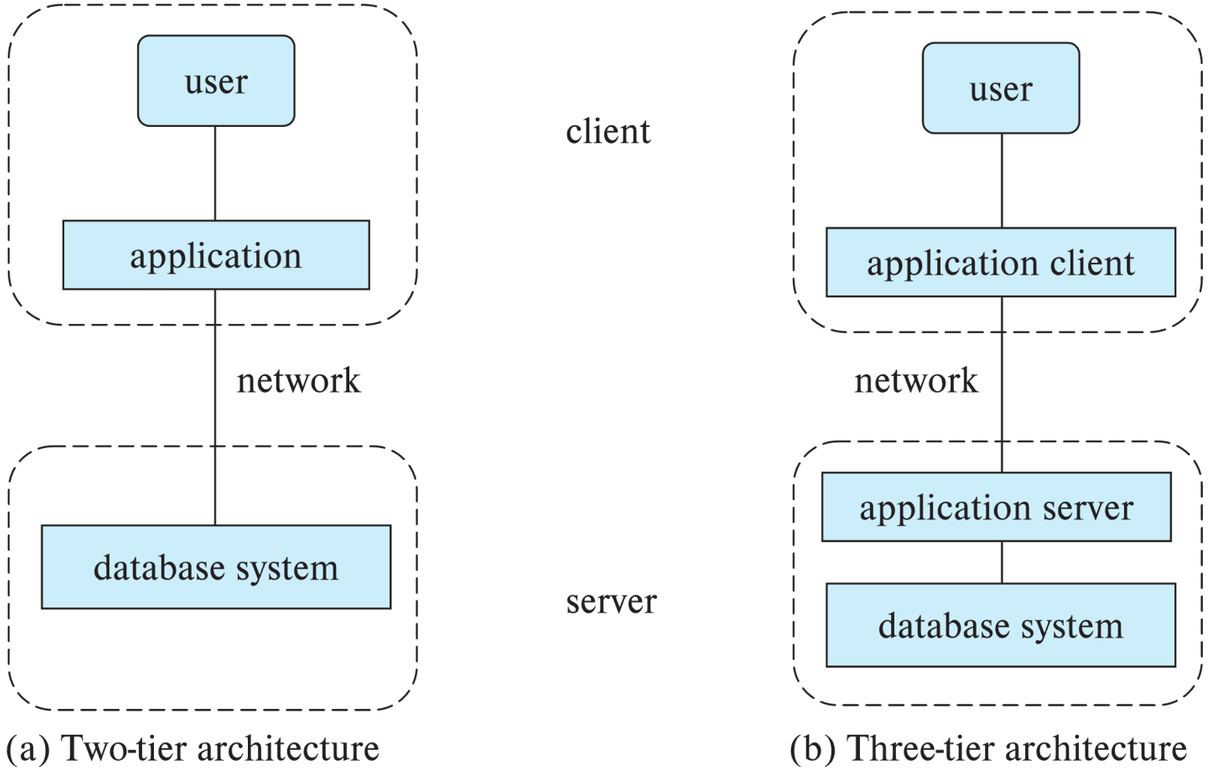
Database Architecture



Database Applications

- Database applications are usually partitioned into 2 or 3 parts

- Application programs generally access databases through one of
  - Language extensions to allow embedded SQL
  - Application Program Interface (e.g., ODBC/JDBC) which allow SQL queries to be sent to a database system



(a) Two-tier architecture

(b) Three-tier architecture

Principles of Database Systems (CS307)

Lecture 2: Introduction to Relation Model and SQL

Zhong-Qiu Wang

Department of Computer Science and Engineering
Southern University of Science and Technology

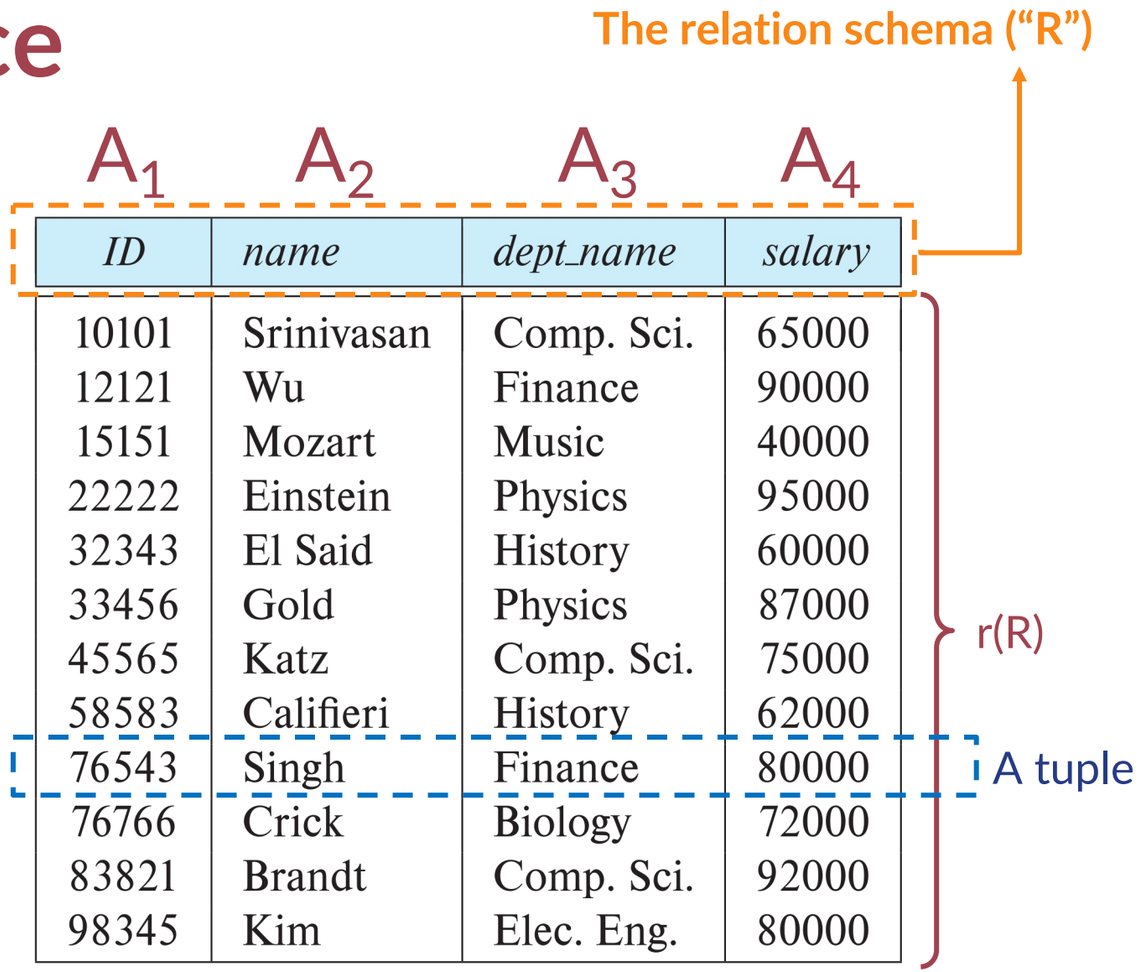
- Most contents are from slides made by Stéphane Faroult and the authors of Database System Concepts (7th Edition).
- Their original slides have been modified to adapt to the schedule of CS307 at SUSTech.
- The slides are largely based on the slides provided by Dr. Yuxin Ma

Relational Model

Relation Schema and Instance

- A_1, A_2, …, A_n are attributes
- $R = (A_1, A_2, \ldots, A_n)$ is a **relation schema**
  - Example on the right side:

    instructor = (*ID*, *name*, *dept_name*, *salary*)
- $r(R)$ denotes a relation instance r defined over schema R
  - Or to say, the entire table on the right side
- An element t of relation r is called a tuple
  - … and is represented by a row in a table



Relation Schema and Instance

- An analogy to programming languages:
  - Relation - Variables
  - Relation schema – Variable types
  - Relation instance – Value(s) stored in the variable

Something More about Attributes

- Domain (of an attribute): The set of allowed values for the attribute
  - E.g., the domain for *dept_name* is the set of all existing departments in the university

- Being atomic (or "atomicity"): indivisible
  - E.g., *salary* is considered indivisible, and full names of instructors may not always be indivisible (which can be further divided into "family name" and "surname name")

NULL in Attributes

- null: a special marker that represents an “unknown” status
  - null ≠ 0; null ≠ 0.0; null ≠ false
    - Null is null, a marker indicating that a data value is missing
  - In practice (SQL for example), it is not advised to treat null as a “value”
    - … however, you can always see the term “null value” in technical articles
  - Complications will be introduced when null appears in many operations
    - “three-valued logic”, “1+null=null”, etc., which will be introduced later

Relations are Unordered

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

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
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

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

Considered the same relation

Database Schema

- Database schema is the logical structure of the database
  - It contains a set of relation schemas and a set of integrity constraints
- Database instance is a snapshot of the data in the database at a given instant in time

Schema

instructor(*ID*, *name*, *dept_name*, *salary*)

An instance of the schema:

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

Keys

- Let K $\subseteq$ *R*
  - K is a **superkey** of *R* if values for K are sufficient to identify a unique tuple of each possible relation *r*(*R*)
    - E.g., {*ID*} and {*ID*, *name*} are both superkeys of instructor
    - If K is a superkey, any superset K' of K where K' $\subseteq$ *R* is a superkey as well
  - Superkey K is a **candidate key** if K is minimal, i.e., no proper subset of K is a superkey
    - E.g., {ID} is a candidate key for *instructor*

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
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

instructor

Keys

- Let K $\subseteq$ *R*
  - One of the candidate keys is selected to be the **primary key**
    - We mark the primary key with an underline:

      instructor = (*ID*, *name*, *dept_name*, *salary*)

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
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

instructor

Keys

- Let $K \subseteq R$
  - **Foreign key (外键)** constraint: Values in one relation must appear in another relation
    - E.g., *dept_name* in *instructor* is a foreign key from *instructor* referencing *department*

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
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

instructor

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

department

Notices

- Find your own lab session (4 different groups)
  - Attendance is required
- QQ Group
  - Teaching assistants will be there to help you
- (For international students) If you don’t know how to join the QQ group or Blackboard site, please email me or find TAs as soon as possible
  - My Email: wangzq3@sustech.edu.cn

Please remember to attend your lab class.
It is strongly recommended to install the required software before your lab session.