UMD DATA605 - Big Data Systems

Lesson 4.1: Relational DBs

Instructor: Dr. GP Saggese, gsaggese@umd.edu

Relational Model: Overview

- Introduced by Ted Codd (late 60's, early 70's)
- **First prototypes**
  - Ingres Project at Berkeley (1970-1985)
    - Ingres (INteractive Graphics REtrieval System)
    - $\rightarrow$ PostgreSQL (=Post Ingres)
  - IBM System R (1970) $\rightarrow$ Oracle, IBM DB2
- **Contributions from relational data model**
  - Formal semantics for data operations
  - Data independence: separation of logical and physical data models
  - Declarative query languages (e.g., SQL)
  - Query optimization
- **Key to commercial success**

Relational Model: Key Definitions

- Relational DB is a collection of **tables / relations**
  - Unique name and schema for each table
  - E.g., `instructor` and `course` relations
- **Row / tuple / record**: Represents a relationship among values
- **Element**: Corresponds to a **column / field / attribute**
  - Atomic elements (e.g., phone number as a single object)
  - `NULL` for unknown or non-existent values
- **Schema of a relation**
  - List of attributes and their domains
  - Like type definition in programming languages
  - E.g., domain of `salary` is integers $>= 0$
- **Instance of relation**
  - Specific instantiation with actual values
  - Changes over time

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

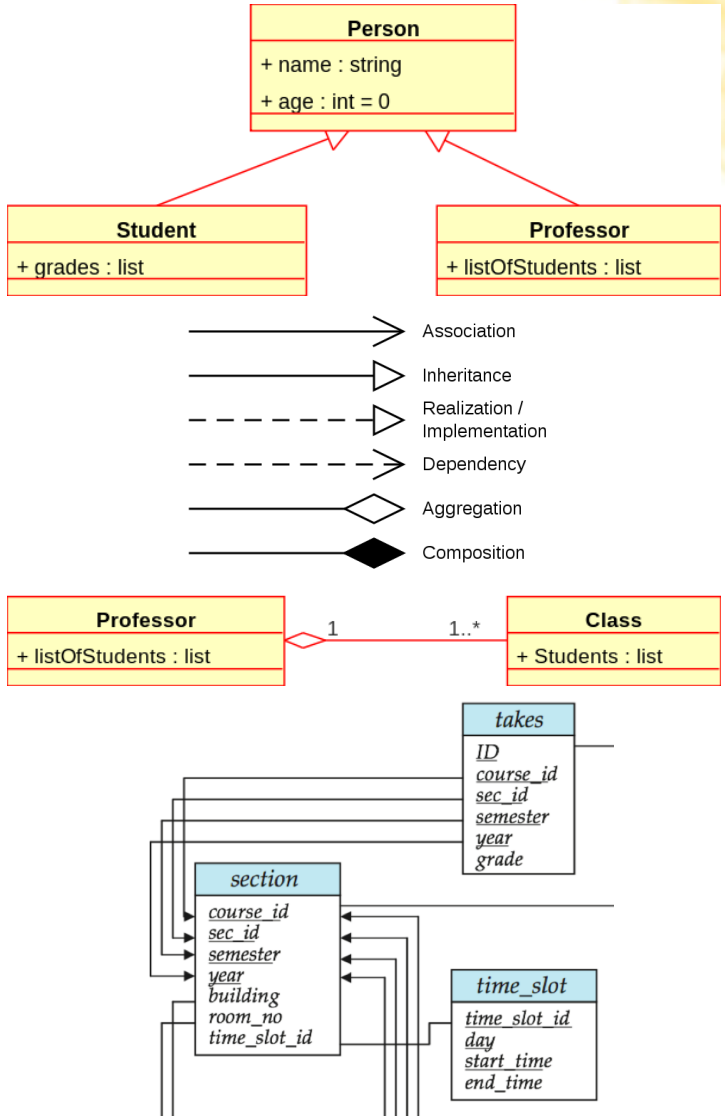
`instructor` relation

course_id	*title*	*dept_name*	*credits*
BIO-101	Intro. to Biology	Biology	4
BIO-301	Genetics	Biology	4
BIO-399	Computational Biology	Biology	3
CS-101	Intro. to Computer Science	Comp. Sci.	4
CS-190	Game Design	Comp. Sci.	4
CS-315	Robotics	Comp. Sci.	3
CS-319	Image Processing	Comp. Sci.	3
CS-347	Database System Concepts	Comp. Sci.	3
EE-181	Intro. to Digital Systems	Elec. Eng.	3
FIN-201	Investment Banking	Finance	3
HIS-351	World History	History	3
MU-199	Music Video Production	Music	3
PHY-101	Physical Principles	Physics	4

`course` relation

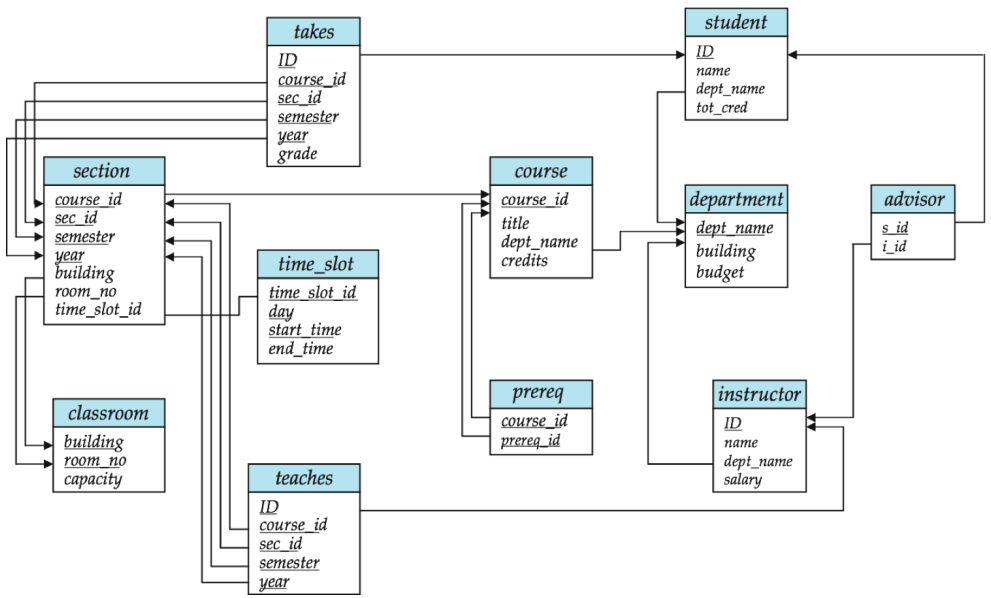
UML Class Diagram

- **UML class diagram**
  - UML = Unified Modeling Language
  - Used in OOP and DB design
- **In OOP design**
  - Diagram showing classes, attributes, methods, and relationships
- **In DB design**
  - Each box is a table / relation
  - Columns / fields / attributes are listed inside the box
  - Primary keys underlined
  - Foreign key constraints are arrows



Example: University DB

- **UML diagram of a DB and schemas representing a University**
  - Each box is a table / relation
  - Column / fields / attributes are listed inside the box
  - Primary keys are underlined fields
  - Foreign key constraints are arrows between boxes
- **Analysis of the diagram**
  - ER model
  - Entities
    - `student`
    - `department`
    - `...`
  - Relationships
    - `takes`
    - `teaches`
    - `...`

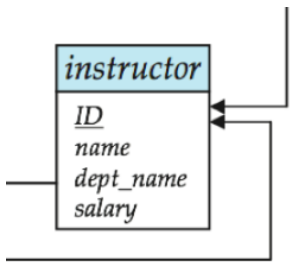


Primary Key

- R is set of attributes of a relation r
  - E.g., `ID`, `name`, `dept_name`, `salary` are attributes of `instructor`
- K is superkey of R if values for K identify a unique tuple of each relation $r(R)$
  - E.g., `(ID)` and `(ID, name)` are superkeys of `instructor`
  - `(name)` is not a superkey of `instructor`
- **Primary key**: minimal set of attributes that uniquely identify each row
  - Typically small and immutable
  - Would `SSN` be a primary key? Yes and no
- **Primary key constraint**: rows can't have the same primary key

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

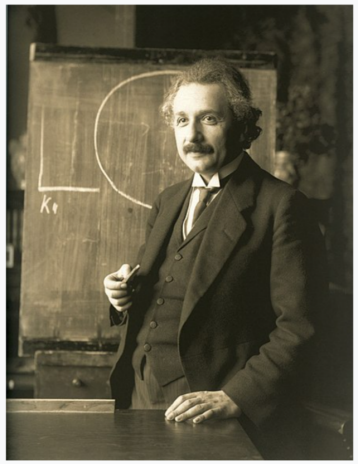
`instructor` relation



Question: What Are Primary Keys?

- Marital status
  - Married(person1_ssn, person2_ssn, date_married, date_divorced)
- Bank account
  - Account(cust_ssn, account_number, cust_name, balance, cust_address)
- Research assistantship at UMD
  - RA(student_id, project_id, supervisor_id, appt_time, appt_start_date, appt_end_date)
- Information typically found on Wikipedia
  - Person(Name, Born, Died, Citizenship, Education, ...)
- Info about US President on Wikipedia
  - President(name, start_date, end_date, vice_president, preceded_by, succeeded_by)
- Tour de France: historical rider participation information
  - Rider(Name, Born, Team-name, Coach, Sponsor, Year)

Albert Einstein



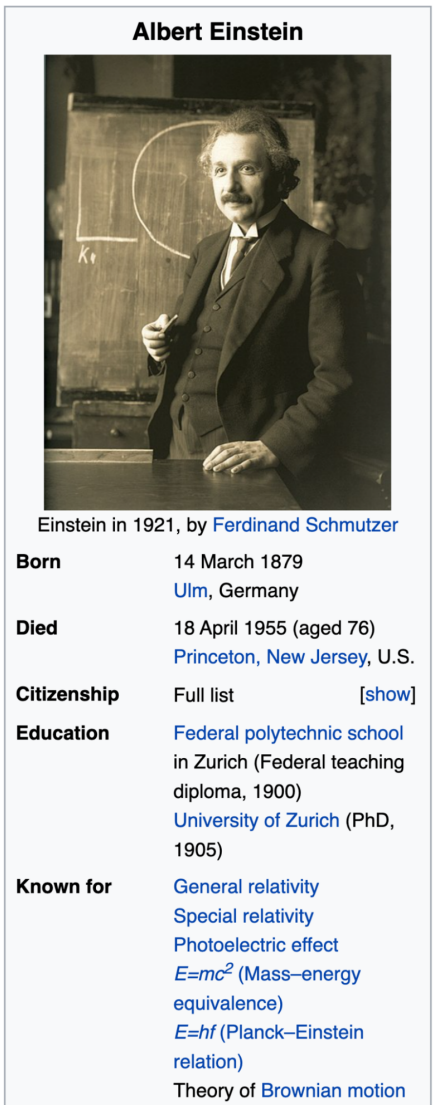
Einstein in 1921, by Ferdinand Schmutzer

Born	14 March 1879 Ulm, Germany
Died	18 April 1955 (aged 76) Princeton, New Jersey, U.S.
Citizenship	Full list [show]
Education	Federal polytechnic school in Zurich (Federal teaching diploma, 1900) University of Zurich (PhD, 1905)
Known for	General relativity Special relativity Photoelectric effect $E=mc^2$ (Mass–energy equivalence) $E=hf$ (Planck–Einstein relation) Theory of Brownian motion

SCIENCE ACADEMY

Answer: What Are Primary Keys?

- Marital status
  - Married(**person1_ssn, person2_ssn, date_married**, date_divorced)
- Bank account
  - Account(cust_ssn, **account_number**, cust_name, balance, cust_address)
- Research assistantship at UMD
  - RA(**student_id, project_id**, supervisor_id, appt_time, **appt_start_date**, appt_end_date)
- Information typically found on Wikipedia
  - Person(**Name, Born, Died, Citizenship, Education, ...**)
- Info about US President on Wikipedia
  - President(**name, start_date**, end_date, vice_president, preceded_by, succeeded_by)
- Tour de France: historical rider participation information
  - Rider(**Name, Born, Team-name,** Coach, Sponsor, **Year**)



SCIENCE ACADEMY

Foreign Key

- **Foreign key** = primary key of another relation
  - E.g., `(ID)` from `student` in `takes`, `advisor`
  - `takes` is the "referencing relation", has the foreign key
  - `student` is the "referenced relation", has the primary key
  - Shown by an arrow from referencing → referenced

- **Foreign key constraint**: for each row, the primary key tuple must exist
  - Aka referential integrity constraint
  - If `(student101, DATA605)` in `takes`, there must be `student101` in `student`
- The key referenced as foreign key must exist as primary key

Relational Algebra: 1/4

- **Relation**: set of tuples
- **Relational algebra**: operations on relations producing a new relation
  - Unary: selection, projection, rename
  - Binary: union, set difference, intersection, Cartesian product, join
- **Selection** Σ: select tuples satisfying a predicate
  - E.g., select `instructor` tuples where `dept_name = "Physics"`
- **Projection** π: return tuples with subset of attributes
  - E.g., project `instructor` tuples with `(name, salary)`
- **Set operations**: union, intersection, set difference
  - Must be compatible (same attributes)

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
33456	Gold	Physics	87000

$\sigma_{dept\_name = \text{"Physics"}}(instructor)$

ID	*name*	*salary*
10101	Srinivasan	65000
12121	Wu	90000
15151	Mozart	40000
22222	Einstein	95000
32343	El Said	60000
33456	Gold	87000
45565	Katz	75000
58583	Califieri	62000
76543	Singh	80000
76766	Crick	72000
83821	Brandt	92000
98345	Kim	80000

$\Pi_{ID,\ name,\ salary}(instructor)$

Relational Algebra: 2/4

- **Cartesian product**: combine two relations into a new one
  - `instructor = (ID, name, dept_name, salary)`
  - `teaches = (ID, course_id, sec_id, semester, year)`
- E.g., `instructor x teaches` gives `(instructor.ID, instructor.name, instructor.dept_name, teaches.ID, ...)`

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` relation

ID	*course_id*	*sec_id*	*semester*	*year*
10101	CS-101	1	Fall	2017
10101	CS-315	1	Spring	2018
10101	CS-347	1	Fall	2017
12121	FIN-201	1	Spring	2018
15151	MU-199	1	Spring	2018
22222	PHY-101	1	Fall	2017
32343	HIS-351	1	Spring	2018
45565	CS-101	1	Spring	2018
45565	CS-319	1	Spring	2018
76766	BIO-101	1	Summer	2017
76766	BIO-301	1	Summer	2018
83821	CS-190	1	Spring	2017
83821	CS-190	2	Spring	2017
83821	CS-319	2	Spring	2018
98345	EE-181	1	Spring	2017

`teaches` relation

instructor.ID	*name*	*dept_name*	*salary*	*teaches.ID*	*course_id*	*sec_id*	*semester*	*year*
10101	Srinivasan	Comp. Sci.	65000	10101	CS-101	1	Fall	2017
10101	Srinivasan	Comp. Sci.	65000	10101	CS-315	1	Spring	2018
10101	Srinivasan	Comp. Sci.	65000	10101	CS-347	1	Fall	2017
10101	Srinivasan	Comp. Sci.	65000	12121	FIN-201	1	Spring	2018
10101	Srinivasan	Comp. Sci.	65000	15151	MU-199	1	Spring	2018
10101	Srinivasan	Comp. Sci.	65000	22222	PHY-101	1	Fall	2017
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...
12121	Wu	Finance	90000	10101	CS-101	1	Fall	2017
12121	Wu	Finance	90000	10101	CS-315	1	Spring	2018
12121	Wu	Finance	90000	10101	CS-347	1	Fall	2017
12121	Wu	Finance	90000	12121	FIN-201	1	Spring	2018
12121	Wu	Finance	90000	15151	MU-199	1	Spring	2018
12121	Wu	Finance	90000	22222	PHY-101	1	Fall	2017
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...
15151	Mozart	Music	40000	10101	CS-101	1	Fall	2017
15151	Mozart	Music	40000	10101	CS-315	1	Spring	2018
15151	Mozart	Music	40000	10101	CS-347	1	Fall	2017
15151	Mozart	Music	40000	12121	FIN-201	1	Spring	2018
15151	Mozart	Music	40000	15151	MU-199	1	Spring	2018
15151	Mozart	Music	40000	22222	PHY-101	1	Fall	2017
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...
22222	Einstein	Physics	95000	10101	CS-101	1	Fall	2017
22222	Einstein	Physics	95000	10101	CS-315	1	Spring	2018
22222	Einstein	Physics	95000	10101	CS-347	1	Fall	2017
22222	Einstein	Physics	95000	12121	FIN-201	1	Spring	2018
22222	Einstein	Physics	95000	15151	MU-199	1	Spring	2018
22222	Einstein	Physics	95000	22222	PHY-101	1	Fall	2017
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...

`instructor x teaches`

Relational Algebra: 3/4

- **Join**: composition of two operations
  - Cartesian-product
  - Selection based on equality between two fields
  - E.g., `instructor` x `teaches` when `instructor.ID = teaches.ID`

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` relation

ID	*course_id*	*sec_id*	*semester*	*year*
10101	CS-101	1	Fall	2017
10101	CS-315	1	Spring	2018
10101	CS-347	1	Fall	2017
12121	FIN-201	1	Spring	2018
15151	MU-199	1	Spring	2018
22222	PHY-101	1	Fall	2017
32343	HIS-351	1	Spring	2018
45565	CS-101	1	Spring	2018
45565	CS-319	1	Spring	2018
76766	BIO-101	1	Summer	2017
76766	BIO-301	1	Summer	2018
83821	CS-190	1	Spring	2017
83821	CS-190	2	Spring	2017
83821	CS-319	2	Spring	2018
98345	EE-181	1	Spring	2017

`teaches` relation

instructor.ID	*name*	*dept_name*	*salary*	*teaches.ID*	*course_id*	*sec_id*	*semester*	*year*
10101	Srinivasan	Comp. Sci.	65000	10101	CS-101	1	Fall	2017
10101	Srinivasan	Comp. Sci.	65000	10101	CS-315	1	Spring	2018
10101	Srinivasan	Comp. Sci.	65000	10101	CS-347	1	Fall	2017
12121	Wu	Finance	90000	12121	FIN-201	1	Spring	2018
15151	Mozart	Music	40000	15151	MU-199	1	Spring	2018
22222	Einstein	Physics	95000	22222	PHY-101	1	Fall	2017
32343	El Said	History	60000	32343	HIS-351	1	Spring	2018
45565	Katz	Comp. Sci.	75000	45565	CS-101	1	Spring	2018
45565	Katz	Comp. Sci.	75000	45565	CS-319	1	Spring	2018
76766	Crick	Biology	72000	76766	BIO-101	1	Summer	2017
76766	Crick	Biology	72000	76766	BIO-301	1	Summer	2018
83821	Brandt	Comp. Sci.	92000	83821	CS-190	1	Spring	2017
83821	Brandt	Comp. Sci.	92000	83821	CS-190	2	Spring	2017
83821	Brandt	Comp. Sci.	92000	83821	CS-319	2	Spring	2018
98345	Kim	Elec. Eng.	80000	98345	EE-181	1	Spring	2017

$\sigma_{instructor.ID = teaches.ID}(instructor \times teaches)$

Relational Algebra: 4/4

- **Query**: combination of relational algebra operations
  - E.g., *"find `course_id` from table section for fall 2017"*
- **Assignment**: assign parts of relational algebra to temporary relation variables
  - Write a query as a sequential program
  - E.g., *"find `course_id` for classes in both fall 2017 and spring 2018"*
- **Equivalent queries**: two queries giving the same result on any DB instance
  - Some formulations are more efficient

$$\Pi_{course\_id}\ (\sigma_{semester = \text{“Fall”} \wedge year=2017}\ (section))$$

$$courses\_fall\_2017 \leftarrow \Pi_{course\_id}(\sigma_{semester = \text{“Fall”} \wedge year=2017}\ (section))$$
$$courses\_spring\_2018 \leftarrow \Pi_{course\_id}(\sigma_{semester = \text{“Spring”} \wedge year=2018}\ (section))$$
$$courses\_fall\_2017 \cap courses\_spring\_2018$$