CSc 484
Database Management Systems

Ken Gamradt

Spring 2024

Relational Model

Structure of Relational Databases

- A relational database consists of a collection of tables
- Each table is assigned a unique name
  - E.g., instructor
- The term relation is used to refer of a table
- The term attribute refers to a column of a table
- The term tuple refers to a row of a table

Attributes (Columns)

Tuples (Rows)

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

Structure of Relational Databases

Relation instance refers to a specific instance of a relation – specific set of rows

The order tuples appear in a relation is irrelevant – these are the same

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

Attributes

- The domain of an attribute
  - A set of permitted values
- The domain of all attributes must be atomic:
  - The elements of the domain must be indivisible units
    - E.g., a person may have a set of phone numbers
- The null value is a special value that signifies that the value is unknown or doesn't exist
  - Null is a member of every domain
  - Remember that NULL is not the same as 0, or '' (empty string)
  - Integer plus NULL equals NULL

Database Schema

- Database schema: logical design of the database
- Database instance: snapshot of the data in the database at a given time
- Relation schema: list of attributes and their corresponding domains
  - department (dept_name, building, budget)
- Relation instance:

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

Database Schema

The *section* relation

course_id	*sec_id*	*semester*	*year*	*building*	*room_number*	*time_slot_id*
BIO-101	1	Summer	2017	Painter	514	B
BIO-301	1	Summer	2018	Painter	514	A
CS-101	1	Fall	2017	Packard	101	H
CS-101	1	Spring	2018	Packard	101	F
CS-190	1	Spring	2017	Taylor	3128	E
CS-190	2	Spring	2017	Taylor	3128	A
CS-315	1	Spring	2018	Watson	120	D
CS-319	1	Spring	2018	Watson	100	B
CS-319	2	Spring	2018	Taylor	3128	C
CS-347	1	Fall	2017	Taylor	3128	A
EE-181	1	Spring	2017	Taylor	3128	C
FIN-201	1	Spring	2018	Packard	101	B
HIS-351	1	Spring	2018	Painter	514	C
MU-199	1	Spring	2018	Packard	101	D
PHY-101	1	Fall	2017	Watson	100	A

Keys

- Tuples in a relation must be unique and distinguished
  - No two tuples in a relation are allowed to have the same value for all attributes
- A **superkey** is a set of one or more attributes that, taken collectively, allow us to identify uniquely a tuple in the relation
  - A superkey may contain extraneous attributes
- A minimal set of a superkey is a **candidate key**
  - No proper subset is a superkey
  - A proper subset of a set A is **a subset of A that is not equal to A**
- Choose one candidate key to be **primary key**
  - Principle means of identifying tuples in a relation
  - Primary key should be chosen such that its attributes values are never, or very rarely changed

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

Keys

Please choose the primary key for each relation

The *teaches* 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

The *department* relation

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

Keys

- **Foreign key constraint**: value in one relation must appear in another
- dept_name is a **foreign key** from instructor, referencing department
  - instructor: referencing relation
  - department: referenced relation
  - dept_name must be a primary key in department relation
- The foreign key constraint is used to:
  - prevent actions that would destroy links between tables
  - prevents invalid data from being inserted into the foreign key column

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

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

Keys

- classroom (building, room_number, capacity)
- section (course_id, sec_id, semester, year, building, room_number, time_slot_id)

course_id	*sec_id*	*semester*	*year*	*building*	*room_number*	*time_slot_id*
BIO-101	1	Summer	2017	Painter	514	B
BIO-301	1	Summer	2018	Painter	514	A
CS-101	1	Fall	2017	Packard	101	H
CS-101	1	Spring	2018	Packard	101	F
CS-190	1	Spring	2017	Taylor	3128	E
CS-190	2	Spring	2017	Taylor	3128	A
CS-315	1	Spring	2018	Watson	120	D
CS-319	1	Spring	2018	Watson	100	B
CS-319	2	Spring	2018	Taylor	3128	C
CS-347	1	Fall	2017	Taylor	3128	A
EE-181	1	Spring	2017	Taylor	3128	C
FIN-201	1	Spring	2018	Packard	101	B
HIS-351	1	Spring	2018	Painter	514	C
MU-199	1	Spring	2018	Packard	101	D
PHY-101	1	Fall	2017	Watson	100	A

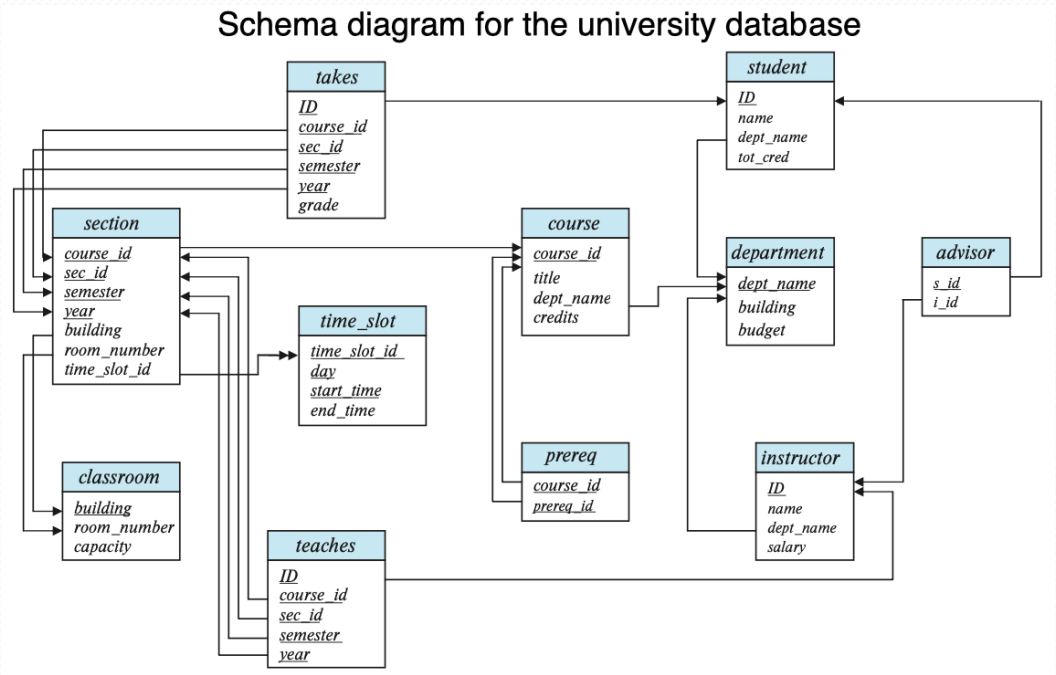
Keys

- Schema of the university database

classroom(*building*, *room_number*, *capacity*)
department(*dept_name*, *building*, *budget*)
course(*course_id*, *title*, *dept_name*, *credits*)
instructor(*ID*, *name*, *dept_name*, *salary*)
section(*course_id*, *sec_id*, *semester*, *year*, *building*, *room_number*, *time_slot_id*)
teaches(*ID*, *course_id*, *sec_id*, *semester*, *year*)
student(*ID*, *name*, *dept_name*, *tot_cred*)
takes(*ID*, *course_id*, *sec_id*, *semester*, *year*, *grade*)
advisor(*s_ID*, *i_ID*)
time_slot(*time_slot_id*, *day*, *start_time*, *end_time*)
prereq(*course_id*, *prereq_id*)

Schema Diagram

- Primary key attributes are shown underlined
- Foreign key constraints appear as arrows from the foreign key attributes of the referencing relation to the primary key of the referenced relation



Relational Query Languages

- A **query language** is a language in which a user requests information from the database
- Query languages **!=** standard programming languages!
  - Query languages don't include constructs for all computing needs
- "pure" query languages
  - Illustrate the fundamental techniques for extracting data from the database
  - Formal, non-user-friendly
- Three categories of "pure" query language
  - **Relational algebra**
    - Forms the theoretical basis of the SQL query language
  - Tuple relational calculus
  - Domain relational calculus

Relational Algebra

- Is a theoretical language with operations that work on one or more relations to define another relation without changing the original relation(s)
- Consists a set of operations that take one or two relations as input and produce a new relation as their result
- **unary** operation:
  - operate on one relation
  - such as: select, project, rename
- **binary** operation:
  - operate on pairs of relations
  - such as: union, set difference
- Database systems do not allow users to write queries in relational algebra

Acknowledgements