Chapter 2: Intro to Relational Model

Edited by Radhika Sukapuram. Original slides by

Database System Concepts, 6th Ed.

Keys

- Let R: set of attributes in the schema of relation *r*
- 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)*
  - Example: {*ID*} and {ID,name} are both superkeys of *instructor.*
- Superkey *K* is a **candidate key** if *K* is minimal.
  - No proper subset of K is a superkey
  - Example: {*ID*} is a candidate key for *Instructor*
- One of the candidate keys is selected to be the **primary key**.
  - The candidate key that is least likely to change is selected
- Keys are a property of a relation, not a tuple

Instructor and department relations

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

Entity Integrity

- **Relational Database Schema**: A set S of relation schemas that belong to the same database. S is the *name* of the **database**.

$$S = \{R_1, R_2, ..., R_n\}$$

- **Entity Integrity**: The *primary key attributes* PK of each relation schema R in S cannot have null values in any tuple of r(R). This is because primary key values are used to *identify* the individual tuples.

$$t[PK] \neq \text{null for any tuple t in r(R)}$$

- <u>Note:</u> Other attributes of R may be similarly constrained to disallow null values, even though they are not members of the primary key.

Referential Integrity

- A constraint involving *two* relations (the previous constraints involved a *single* relation).
- Used to specify a *relationship* among tuples in two relations: the **referencing relation** and the **referenced relation**.
- Tuples in the *referencing relation* R_1 have attributes FK (called **foreign key** attributes) that reference the primary key attributes PK of the *referenced relation* R_2. A tuple t_1 in R_1 is said to **reference** a tuple t_2 in R_2 if $t_1[FK] = t_2[PK]$.
- A referential integrity constraint can be displayed in a relational database schema as a directed arc from $R_1.FK$ to R_2.

Referential Integrity Constraint

<u>Statement of the constraint</u>

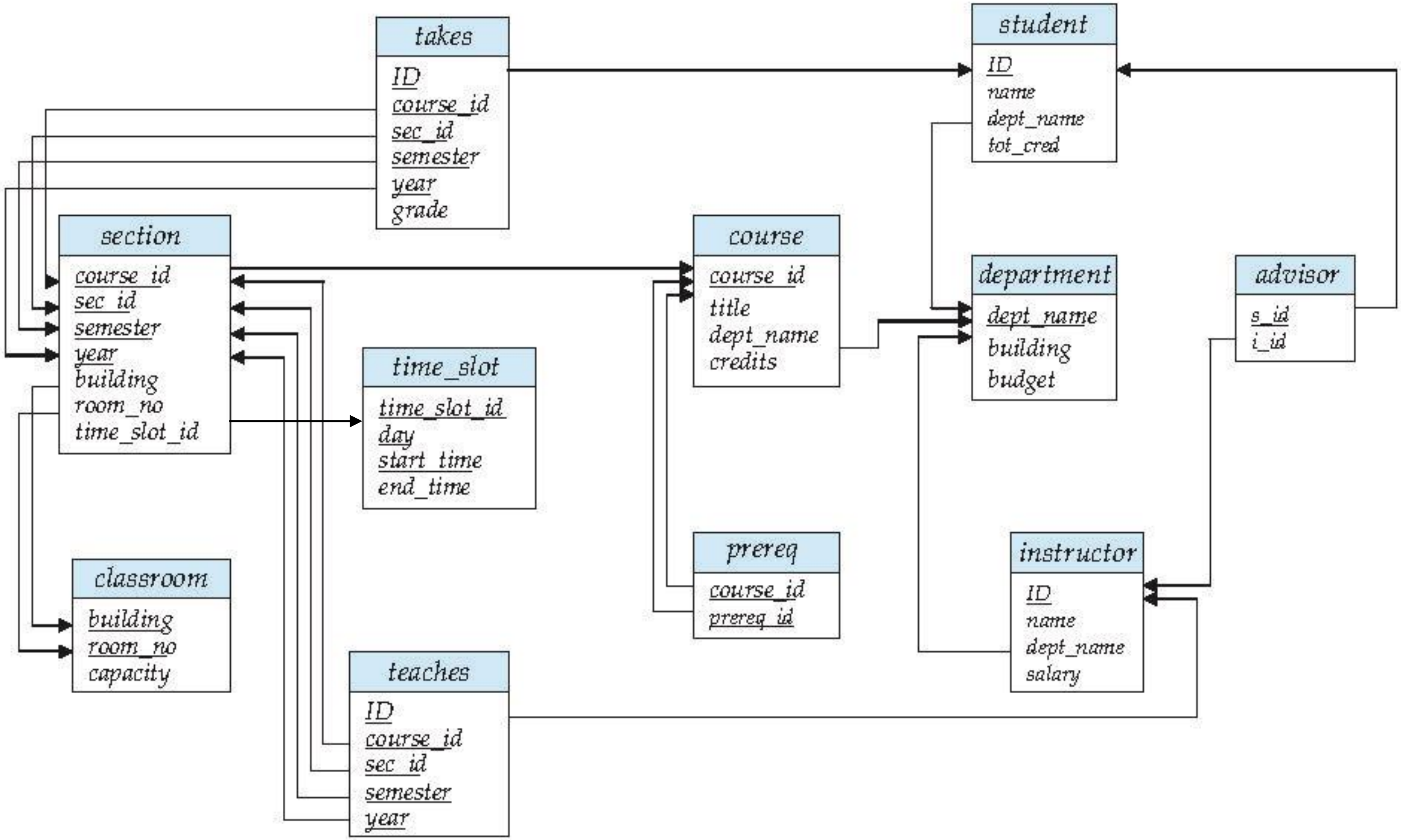
The value in the foreign key column (or columns) FK of the **referencing relation** R_1 can be <u>either</u>:

(1) a value of an existing primary key value of the corresponding primary key PK in the **referenced relation** $R_{2,}$, or..

(2) a null.

In case (2), the FK in R_1 should <u>not</u> be a part of its own primary key**.**

Schema Diagram for University Database



Relational Query Languages

- Procedural vs non-procedural, or declarative
- “Pure” languages:
  - Relational algebra - procedural
  - Tuple relational calculus - declarative
  - Domain relational calculus - declarative
- The above 3 pure languages are equivalent in computing power
- Not Turing complete

Introduction to relational algebra

Why is it required ?

- SQL is loosely based on relational algebra
- Many relational databases use relational algebra operations for representing execution plans
  - Simple, clean, effective mathematical abstraction for representing how results will be generated
  - Equivalent efficient queries can be generated during query optimization

Relational Algebra

- The basic set of operations for the relational model is known as the relational algebra. These operations enable a user to specify basic retrieval requests.

- The result of a retrieval is a new relation, which may have been formed from one or more relations. The **algebra operations** thus produce new relations, which can be further manipulated using operations of the same algebra.

- A sequence of relational algebra operations forms a **relational algebra expression**, whose result will also be a relation that represents the result of a database query (or retrieval request).

Relational Algebra

- Procedural language
- Six basic operators
  - select: σ
  - project: Π
  - union: $\cup$
  - set difference: –
  - Cartesian product: x
  - rename: ρ
- The operators take one or two relations as inputs and produce a new relation as a result. Hence, it is closed.

Select Operation – selection of rows (tuples)

SELECT operation is used to select a *subset* of the tuples from a relation that satisfy a **selection condition**. It is a filter that keeps only those tuples that satisfy a qualifying condition – those satisfying the condition are selected while others are discarded.

- Relation r

A	B	C	D
α	α	1	7
α	β	5	7
β	β	12	3
β	β	23	10

- $\sigma_{\underbrace{A=B \wedge D > 5}_{}}(r)$

predicate

A	B	C	D
α	α	1	7
β	β	23	10

Select Operation

- Notation: $\sigma_p(r)$
- p is called the **selection predicate**
- Defined as:

$$\sigma_p(\boldsymbol{r}) = \{t \mid t \in r \textbf{ and } p(t)\}$$

Where p is a formula in propositional calculus consisting of **terms** connected by : $\wedge$ (**and**), $\vee$ (**or**), $\neg$ (**not**)
Each **term** is one of:

<attribute> *op* <attribute> or <constant>

where *op* is one of: $=, \neq, >, \geq. <. \leq$

Select Operation

Example: To select the EMPLOYEE tuples whose department number is four or those whose salary is greater than \$30,000 the following notation is used:

$$\sigma_{\text{DNO} = 4 \vee \text{SALARY} > 30{,}000}\ (\textbf{EMPLOYEE})$$

In general, the select operation is denoted by $\sigma_{\text{<selection condition>}}(r)$ where the

symbol σ (sigma) is used to denote the select operator, and the selection condition is a Boolean expression specified on the attributes of relation r

Unary Relational Operations

SELECT Operation Properties

- The SELECT operation $\sigma_{<\text{selection condition}>}(r)$ produces a relation S that has the same schema as R of r

- The SELECT operation σ is **commutative;** i.e.,

  $\sigma_{<\text{condition1}>}(\sigma_{<\text{condition2}>}(r)) = \sigma_{<\text{condition2}>}(\sigma_{<\text{condition1}>}(r))$

- A cascaded SELECT operation **may be applied in any order;** i.e.,

  $\sigma_{<\text{condition1}>}(\sigma_{<\text{condition2}>}(\sigma_{<\text{condition3}>}(r))$

  $= \sigma_{<\text{condition2}>}(\sigma_{<\text{condition3}>}(\sigma_{<\text{condition1}>}(r)))$

- A cascaded SELECT operation may be replaced by a single selection with a conjunction of all the conditions; i.e.,

  $\sigma_{<\text{condition1}>}(\sigma_{<\text{condition2}>}(\sigma_{<\text{condition3}>}(r))$

  $= \sigma_{<\text{condition1}>\ \text{AND}\ <\text{condition2}>\ \text{AND}\ <\text{condition3}>}(r)))$

Figure 7.5 Schema diagram for the COMPANY relational database schema; the primary keys are underlined.

EMPLOYEE

FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO

DEPARTMENT

DNAME	DNUMBER	MGRSSN	MGRSTARTDATE

DEPT_LOCATIONS

DNUMBER	DLOCATION

PROJECT

PNAME	PNUMBER	PLOCATION	DNUM

WORKS_ON

ESSN	PNO	HOURS

DEPENDENT

ESSN	DEPENDENT_NAME	SEX	BDATE	RELATIONSHIP

Unary Relational Operations

Figure 7.8 Results of SELECT and PROJECT operations.

(a) $\sigma_{(\text{DNO}=4 \text{ AND SALARY}>25000) \text{ OR } (\text{DNO}=5 \text{ AND SALARY}>30000)}(\text{EMPLOYEE})$.

(b) $\pi_{\text{LNAME, FNAME, SALARY}}(\text{EMPLOYEE})$. (c) $\pi_{\text{SEX, SALARY}}(\text{EMPLOYEE})$

(a)

FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
Franklin	T	Wong	333445555	1955-12-08	638 Voss,Houston,TX	M	40000	888665555	5
Jennifer		Wallace	987654321	1941-06-20	291 Berry,Bellaire,TX	F	43000	888665555	4
Ramesh		Narayan	666884444	1962-09-15	975 FireOak,Humble,TX	M	38000	333445555	5

(b)

LNAME	FNAME	SALARY
Smith	John	30000
Wong	Franklin	40000
Zelaya	Alicia	25000
Wallace	Jennifer	43000
Narayan	Ramesh	38000
English	Joyce	25000
Jabbar	Ahmad	25000
Borg	James	55000

(c)

SEX	SALARY
M	30000
M	40000
F	25000
F	43000
M	38000
M	25000
M	55000

Project Operation – selection of columns (Attributes)

PROJECT Operation: This operation selects certain *columns* from the table and discards the other columns. The PROJECT creates a vertical partitioning – one with the needed columns (attributes) containing results of the operation and other containing the discarded Columns.

- Relation *r*:

A	B	C
α	10	1
α	20	1
β	30	1
β	40	2

- $\Pi_{A,C}\,(r)$

A	C
α	1
α	1
β	1
β	2

=

A	C
α	1
β	1
β	2

Project Operation

- The general form of the project operation is π<attribute list>(r) where π (pi) is the symbol used to represent the project operation and <attribute list> is the desired list of attributes from the attributes of relation r.

- Notation: $\prod_{A_1, A_2, \ldots, A_k} (r)$ where A_1, A_2 are attribute names and *r* is a relation name.
- The result is defined as the relation of *k* columns obtained by erasing the columns that are not listed
- The project operation *removes any duplicate tuples*, so the result of the project operation is a set of tuples and hence a valid relation.

Unary Relational Operations

PROJECT Operation Properties

- The number of tuples in the result of projection $\pi_{\text{<list>}}(r)$ is always less or equal to the number of tuples in r.

- If the list of attributes includes a key of r, then the number of tuples is equal to the number of tuples in r.

- $\pi_{\text{<list1>}}(\pi_{\text{<list2>}}(r)) = \pi_{\text{<list1>}}(r)$ as long as <list2> contains the attributes in <list1>

Assignment Operation

- The assignment operation ($\leftarrow$) provides a convenient way to express complex queries.
  - Write query as a sequential program consisting of
    - a series of assignments
    - followed by an expression whose value is displayed as a result of the query.
  - Assignment must always be made to a temporary relation variable.
  - The result to the right of the $\leftarrow$ is assigned to the relation variable on the left of the $\leftarrow$.
  - May use variable in subsequent expressions.

Assignment Operation

Example: To retrieve the first name, last name, and salary of all employees who work in department number 5, we must apply a select and a project operation. We can write a single relational algebra expression as follows:

$$\pi_{\text{FNAME, LNAME, SALARY}}(\sigma_{\text{DNO=5}}(\text{EMPLOYEE}))$$

OR We can explicitly show the sequence of operations, giving a name to each intermediate relation:

$$\text{DEP5\_EMPS} \leftarrow \sigma_{\text{DNO=5}}(\text{EMPLOYEE})$$

$$\text{RESULT} \leftarrow \pi_{\text{FNAME, LNAME, SALARY}}(\text{DEP5\_EMPS})$$