3.4 Additional Basic Operations + 3.5 Set Operations

Judy Wang

Geoffrey Towell

CMSC 383: Topics in Computer Science: Databases in Practice

Overview

3.4

- Rename Operator
- String Operations
- Attribute Specification in the Select Clause
- Ordering the Display of Tuples
- Where Clause Predicates

3.5

- Union Operation
- Intersect Operation
- Except Operation

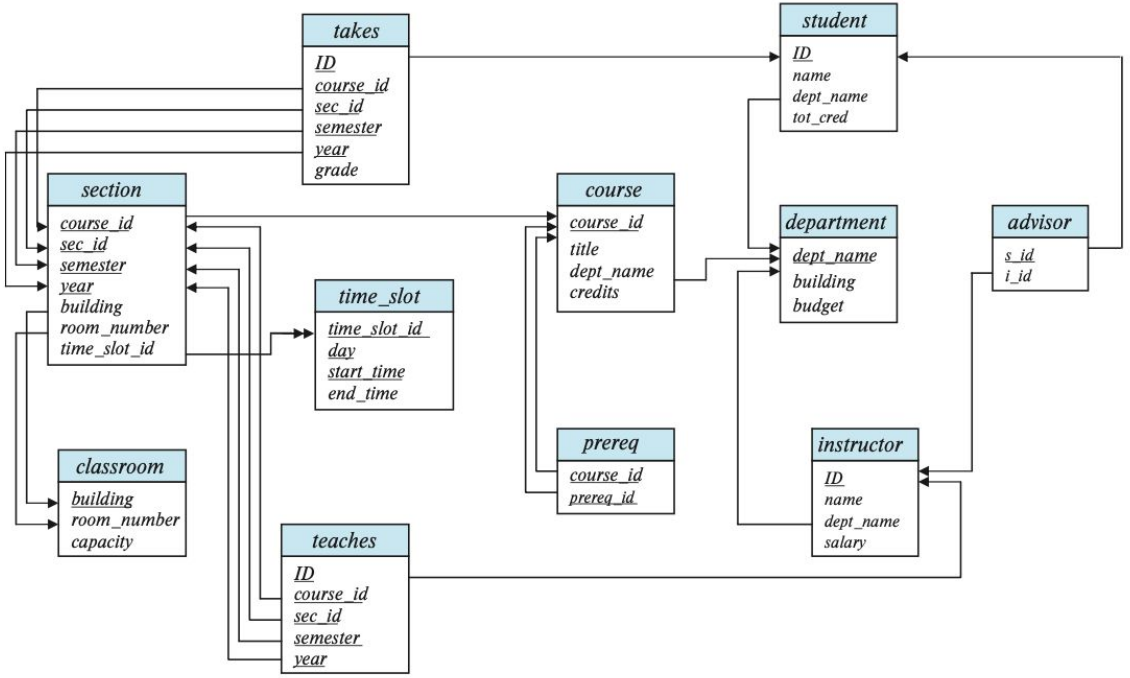

Figure 2.9 Schema diagram for the university database.

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
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...

Rename Operator

The SQL allows renaming relations and attributes using the as clause:

old-name **as** new-name

"Find the names of all instructors whose salary is greater than at least one instructor in the Comp. Sci. department."

```
select distinct T .name

from instructor as T , instructor as S

where T.salary > S.salary and S.dept_name = 'Comp. Sci.';
```

Keyword as is optional and may be omitted

instructor as T ≡ instructor T

Identifiers such as T and S often referred to as **correlation name**

String Operations

SQL includes a string-matching operator for comparisons on character strings. The operator like uses patterns that are described using two special characters:

percent (%). The % character matches any substring.

underscore (_). The _ character matches any character.

- 'Intro%' matches any string beginning with “Intro”.
- '%Comp%' matches any string containing “Comp” as a substring,

for example, 'Intro. to Computer Science', and 'Computational Biology'.

- '_ _ _ ' matches any string of exactly three characters.
- '_ _ _ %' matches any string of at least three characters.

String Operations

Example

“Find the names of all departments whose building name includes the substring 'Park'.” This query can be written as:

```
select dept_name
from department
where building like ‘%Park%’;
```

String Operations

SQL supports a variety of string operations such as

- concatenation (using “||”)
- converting from upper to lower case (and vice versa)
- finding string length, extracting substrings, etc.

Escape Character: ‘\’

Match the string “100%”

```
like '100 \%'  escape  '\'
```

in that above we use backslash (\) as the escape character.

Ordering the Display of Tuples

List in alphabetic order the names of all instructors

```
select distinct name
from    instructor
order by name
```

We may specify desc for descending order or asc for ascending order

for each attribute; **ascending order is the default.**

Example: `order by name desc`

Can sort on multiple attributes

Example: `order by  dept_name, name`

Where-Clause Predicates

SQL includes a **between** comparison operator

Example:

Find the names of all instructors with salary between $90,000 and $100,000 (that is, $90,000 and $100,000)

```
select name
from instructor
where salary between 90000 and 100000
```

Tuple comparison

```
select name, course_id
from instructor, teaches
where (instructor.ID, dept_name) = (teaches.ID, 'Biology');
```

Set Operations

The SQL operations union, intersect, and except correspond to the mathematical operations ∪, ∩, and -.

Union - finds all the sets that relate to **EITHER** of the select-from-where statements
→ union operation automatically eliminates duplicates
→ retain duplicates use **union all**

Find courses that ran in Fall 2021 **OR** in Spring 2022

```
(select course_id
from section
where semester = 'Fall' and year = 2021)
union
(select course_id
from section
where semester = 'Spring' and year = 2022)
```

Set Operations (cont.)

Intersect - finds all the sets that relate to **BOTH** ofthe select-from-where statements
→ Intersect operation automatically eliminates duplicates
→ retain duplicates use **intersect all**

Find courses that ran in Fall 2021 **AND** in Spring 2022

```
(select course_id
from section
where sem = 'Fall' and year = 2021)
intersect
(select course_id
from section
where sem = 'Spring' and year = 2022)
```

Set Operations (cont.)

Except - finds all the sets that relate to **one** the select-from-where statement but **not** the other select-from-where statement
→ Except operation automatically eliminates duplicates
→ retain duplicates use **except all**

Find courses that ran in Fall 2021 **BUT NOT** in Spring 2022

```
(select course_id
from section
where sem = 'Fall' and year = 2021)
except
(select course_id
from section
where sem = 'Spring' and year = 2022)
```

Work Cited

Silberschatz, Abraham, Henry F. Korth, and S Sudarshan. Database System Concepts. , 2020. Print.