Database Management Systems (CSE-251)

Presented by

Md. Atiqul Islam Rizvi

Assistant Professor, Dept. of CSE, CUET

Outer Join

- *course* **natural left outer join** *prereq* (Rel. algebra - *course* ⟕ *prereq)*

course_id	*title*	*dept_name*	*credits*	*prereq_id*
BIO-301	Genetics	Biology	4	BIO-101
CS-190	Game Design	Comp. Sci.	4	CS-101
CS-315	Robotics	Comp. Sci.	3	*null*

- *course* **natural right outer join** *prereq* (Rel. algebra - *course* ⟖ *prereq*)

course_id	*title*	*dept_name*	*credits*	*prereq_id*
BIO-301	Genetics	Biology	4	BIO-101
CS-190	Game Design	Comp. Sci.	4	CS-101
CS-347	*null*	*null*	*null*	CS-101

- *course* **natural full outer join** *prereq* (Rel. algebra - *course* ⟗ *prereq*)

course_id	*title*	*dept_name*	*credits*	*prereq_id*
BIO-301	Genetics	Biology	4	BIO-101
CS-190	Game Design	Comp. Sci.	4	CS-101
CS-315	Robotics	Comp. Sci.	3	*null*
CS-347	*null*	*null*	*null*	CS-101

Aggregate Functions

- These functions operate on the multiset of values of a column of a relation, and return a value

  avg: average value
  min: minimum value
  max: maximum value
  sum: sum of values
  count: number of values

- Relational Algebra syntax –

$$G_1, G_2, \ldots, G_n \mathcal{G} F_1(A_1), F_2(A_2), \ldots, F_n(A_n) (r)$$

Where

- r is any relational-algebra expression
- Each G_i is an attribute on which to group (list can be empty)
- Each F_i is an aggregate function
- Each A_i is an attribute name

Aggregate Functions (Cont.)

- **Find the average salary of instructors in each department**
  - $dept\_name\ \mathcal{G}\ \mathbf{avg}(salary)\ (instructor)$

ID	*name*	*dept_name*	*salary*
76766	Crick	Biology	72000
45565	Katz	Comp. Sci.	75000
10101	Srinivasan	Comp. Sci.	65000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000
12121	Wu	Finance	90000
76543	Singh	Finance	80000
32343	El Said	History	60000
58583	Califieri	History	62000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
22222	Einstein	Physics	95000

dept_name	*avg_salary*
Biology	72000
Comp. Sci.	77333
Elec. Eng.	80000
Finance	85000
History	61000
Music	40000
Physics	91000

Data Modification

- We can use assignment operator for insertion, deletion and update operations.

- **Deletion** – $r \leftarrow r - s$

  $student \leftarrow student - \sigma_{dept\_name=\text{“CSE”}}(student)$

- **Insertion** – $r \leftarrow r \cup s$

  $student \leftarrow student \cup \{(1234, \text{‘M. X’}, \text{‘CSE’}, null)\}$

- **Update** – $r \leftarrow \prod_{A1, A2, \ldots, An} (r^*) \cup (\sim r^*)$

$instructor \leftarrow \prod_{ID, name, dept\_name, salary*1.03} (\sigma_{salary > 100{,}000} (instructor)) \cup (\sigma_{salary <= 100{,}000} (instructor)$

Equivalent Queries

- There is more than one way to write a query in relational algebra.
- Example: **Find information about instructors in the Physics department with salary greater than 90,000**
- Query 1

$$\sigma_{dept\_name=\text{“Physics”} \wedge salary > 90{,}000}(instructor)$$

- Query 2

$$\sigma_{dept\_name=\text{“Physics”}}(\sigma_{salary > 90.000}(instructor))$$

- The two queries are not identical; they are, however, equivalent -- they give the same result on any database.

Equivalent Queries

- There is more than one way to write a query in relational algebra.
- Example: **Find information about courses taught by instructors in the Physics department**
- Query 1

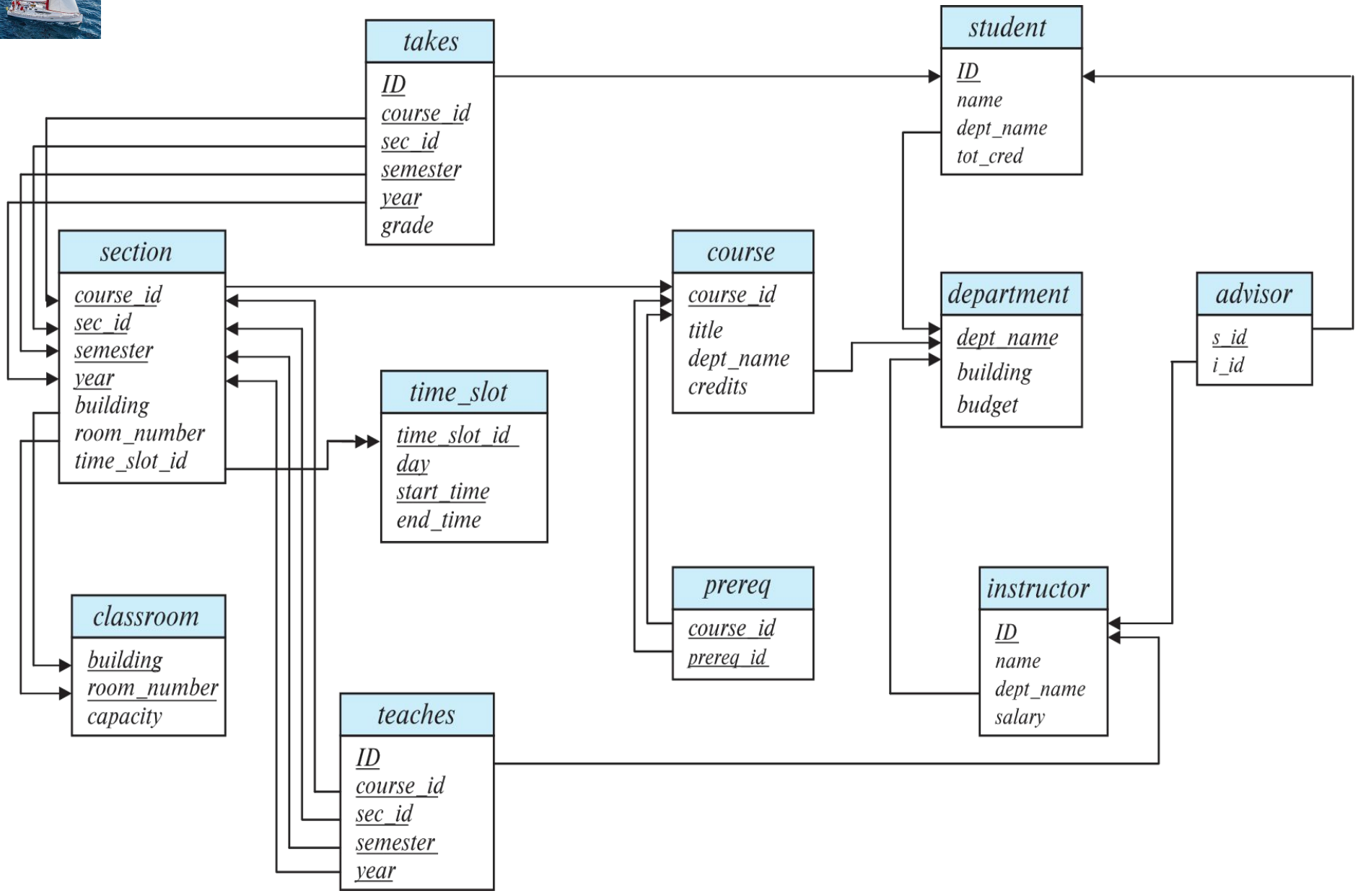
$$\sigma_{dept\_name=\text{“Physics”}}(instructor \bowtie_{instructor.ID = teaches.ID} teaches)$$

- Query 2

$$(\sigma_{dept\_name=\text{“Physics”}}(instructor)) \bowtie_{instructor.ID = teaches.ID} teaches$$

- The two queries are not identical; they are, however, equivalent -- they give the same result on any database.

Schema Diagram for University Database



Exercises

- **Find the ID and name of each instructor in the Physics department.**

$$\Pi_{ID,\ name}\ (\sigma_{dept\_name\ =\ 'Physics'}\ (instructor))$$

- **Find the ID and name of each active instructor in a department located in the building "Watson".**

$$\Pi_{ID,\ name}\ (\sigma_{building\ =\ 'Watson'}\ ((instructor \bowtie_{instructor.ID\ =\ teaches.ID} teaches) \bowtie_{teaches.course\_id\ =\ course.course\_id} course) \bowtie_{course.dept\_name\ =\ department.dept\_name} department)$$

Practice Exercises

- *employee (<u>person_name</u>, street, city)*

  works (<u>person_name</u>, company_name, salary)

  company (<u>company_name</u>, city)

- **Find the name of each employee who lives in city "Miami".**

$$\Pi_{\text{person\_name}} (\sigma_{\text{city = 'Miami'}} (\text{employee}))$$

- **Find the name of each employee whose salary is greater than $100000.**

$$\Pi_{\text{person\_name}} (\sigma_{\text{salary > 100000}} (\text{works}))$$

- **Find the name of the employee who earns more than every employee in the database.**

$$\text{temp} \leftarrow \Pi_{\text{A.person\_name}} (\rho_{\text{A}} (\text{works}) \bowtie_{\text{A.salary < B.salary}} \rho_{\text{B}} (\text{works}))$$

$$\Pi_{\text{person\_name}} (\text{works}) - \text{temp}$$

End of Chapter 2