STUDENT

snum	sname	major	level	age
51135593	Maria White	CS	SR	21
60839453	Charles Harris	CS	So	22
99354543	Susan Martin	Architecture	JR	20
112348546	Joseph Thompson	CS	SO	19

CLASS

name	meets_at	room	fid
Data Structures	MWF 10	R128	489456522
Database Systems	MWF 12:30-1	1320DCL	142519864
Operating System	TuTh 12- 1:20	20AVW	489456522

ENROLLED

snum	cname
51135593	Data Structures
60839453	Data Structures
51135593	Database Systems
99354543	Database Systems
51135593	Operating System

FIND THE NAMES OF STUDENTS WHO HAVE ENROLLED IN ALL COURSES ?

```
SELECT sname
FROM Student
WHERE snum IN (
    SELECT snum
    FROM Enrolled
    GROUP BY snum
    HAVING COUNT(*) = SELECT COUNT(*) FROM Class
);
```

snum	Count
51135593	3
60839453	1
99354543	1

FIND THE NAMES OF ALL CLASSES THAT EITHER MEET IN ROOM R128 OR HAVE FIVE OR MORE STUDENTS ENROLLED.

```
SELECT cname
FROM Class
WHERE room="R128" OR cname IN (
        SELECT cname
        FROM Enrolled
GROUP BY cname HAVING COUNT(*) >= 5
);
```

STUDENT

snum	sname	major	level	age
51135593	Maria White	CS	SR	21
60839453	Charles Harris	CS	So	22
99354543	Susan Martin	Architecture	JR	20
112348546	Joseph Thompson	CS	SO	19

CLASS

name	meets_at	room	fid
Data Structures	MWF 10	R128	489456522
Database Systems	MWF 12:30-1	1320DCL	142519864
Operating System	TuTh 12- 1:20	20AVW	489456522

ENROLLED

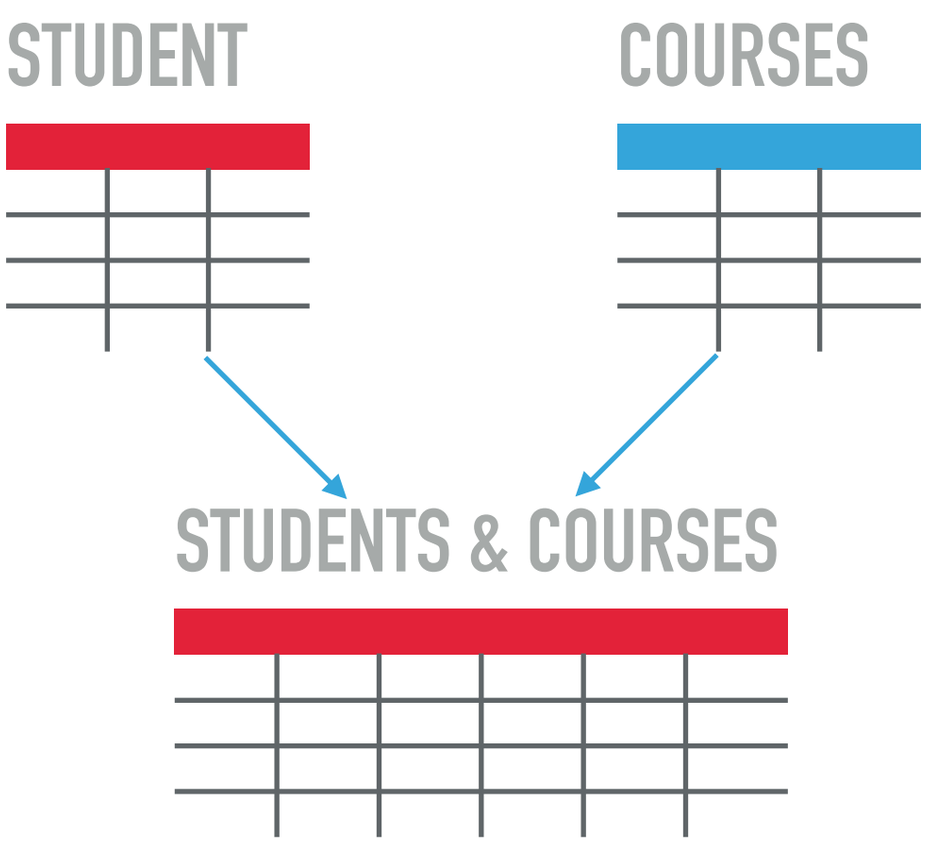
snum	cname
51135593	Data Structures
60839453	Data Structures
51135593	Database Systems
99354543	Database Systems
51135593	Operating System

DBMS

JOINS AND ITS TYPES

WHAT EXACTLY IS A JOIN?

- Information from two or more relations is *joined* into one.



THE NATURAL JOIN

- Unlike the Cartesian Product, considers only those pairs of tuples with the same value on those attributes that appear in the schemas of both relations.

INSTRUCTOR

ID	Name	Dept.
101	Kim	Physics
102	Adolf	CS
103	Donald	History

TEACHES

ID	Course	Sec.	Dept.
102	ECE-101	A	ECE
102	CS-193	B	CS
103	HIS-101	B	History

INSTRUCTOR – TEACHES

ID	Name	Dept.	Course	Sec
102	Adolf	CS	CS-193	B
103	Donald	History	HIS-101	B

THE NATURAL JOIN

Select *name, course*
From *instructor, teaches*
Where *instructor.ID= teaches.ID*
AND *instructor.Dept=teaches.Dept;*

→

Select *name, course*
From *instructor* natural join *teaches;*

INSTRUCTOR

ID	Name	Dept.
101	Kim	Physics
102	Adolf	CS
103	Donald	History

TEACHES

ID	Course	Sec.	Dept.
102	ECE-101	A	ECE
102	CS-193	B	CS
103	HIS-101	B	History

INSTRUCTOR – TEACHES

ID	Name	Dept.	Course	Sec
102	Adolf	CS	CS-193	B
103	Donald	History	HIS-101	B

THE NATURAL JOIN

Select *name, course*
From *instructor* natural join *teaches* natural join *classroom … ;*

Select *name, room_no*
From *instructor* natural join *teaches, classroom*
Where *teaches.sec = classroom.sec;*

Select *name, course*
From *instructor* join *teaches* using *(ID, …);*

Select *name, course*
From *instructor* join *teaches* on *instructor.ID = teaches.ID;*

Natural, on and using are Join Conditions.

OUTER JOINS

- What if we wanted to preserve all tuples?

INSTRUCTOR

ID	Name	Dept.
101	Kim	Physics
102	Adolf	CS
103	Donald	History

TEACHES

ID	Course	Sec.	Dept.
102	ECE-101	A	ECE
102	CS-193	B	CS
103	HIS-101	B	History

INSTRUCTOR – TEACHES

ID	Name	Dept.	Course	Sec
102	Adolf	CS	CS-193	B
103	Donald	History	HIS-101	B
101	Kim	Physics	*Null*	*Null*

Left Outer Join

instructor natural left outer join *teaches;*

OUTER JOINS

- What if we wanted to preserve all tuples?

INSTRUCTOR

ID	Name	Dept.
101	Kim	Physics
102	Adolf	CS
103	Donald	History

TEACHES

ID	Course	Sec.	Dept.
102	ECE-101	A	ECE
102	CS-193	B	CS
103	HIS-101	B	History

INSTRUCTOR – TEACHES

ID	Name	Dept.	Course	Sec
102	Adolf	CS	CS-193	B
103	Donald	History	HIS-101	B
102	*Null*	ECE	ECE-101	A

Right Outer Join

instructor natural right outer join *teaches;*

OUTER JOINS

- What if we wanted to preserve all tuples?

INSTRUCTOR

ID	Name	Dept.
101	Kim	Physics
102	Adolf	CS
103	Donald	History

TEACHES

ID	Course	Sec.	Dept.
102	ECE-101	A	ECE
102	CS-193	B	CS
103	HIS-101	B	History

INSTRUCTOR – TEACHES

ID	Name	Dept.	Course	Sec
102	Adolf	CS	CS-193	B
103	Donald	History	HIS-101	B
102	*Null*	ECE	ECE-101	A
101	Kim	Physics	*Null*	*Null*

Full Outer Join

instructor natural full outer join *teaches;*

THE NATURAL JOIN – INNER JOIN

Select *name, course*
From *instructor, teaches*
Where *instructor.ID= teaches.ID;*

Select *name, course*
From *instructor* natural join *teaches;*

INSTRUCTOR

ID	Name	Dept.
101	Kim	Physics
102	Adolf	CS
103	Donald	History

TEACHES

ID	Course	Sec.	Dept.
102	ECE-101	A	ECE
102	CS-193	B	CS
103	HIS-101	B	History

INSTRUCTOR – TEACHES

ID	Name	Dept.	Course	Sec
102	Adolf	CS	CS-193	B
103	Donald	History	HIS-101	B

End