



Intermediate SQL

Joined Relations

- **Join operations** take two relations and return as a result another relation
- A join operation is a Cartesian product which requires that tuples in the two relations match (under some condition)
- It also specifies the attributes that are present in the result of the join
- The join operations are typically used as subquery expressions in the **from** clause

Types of Joins between Relations

- Cross join
- Inner join
 - Equi-join
 - Natural join
- Outer join
 - Left outer join
 - Right outer join
 - Full outer join
- Self-join

Cross Join in SQL

- **Cross join** returns the Cartesian product of rows from tables in the join

- Explicit

```
select *  
from employee cross join department
```

- Implicit

```
select *  
from employee, department
```

Join Operations Examples

- Relation *course*

| <i>course_id</i> | <i>title</i> | <i>dept_name</i> | <i>credits</i> |
|------------------|--------------|------------------|----------------|
| BIO-301 | Genetics | Biology | 4 |
| CS-190 | Game Design | Comp. Sci. | 4 |
| CS-315 | Robotics | Comp. Sci. | 3 |

- Relation *prereq*

| <i>course_id</i> | <i>prereq_id</i> |
|------------------|------------------|
| BIO-301 | BIO-101 |
| CS-190 | CS-101 |
| CS-347 | CS-101 |

- Observe that

course information is missing CS-437

prereq information is missing CS-315

Inner Join

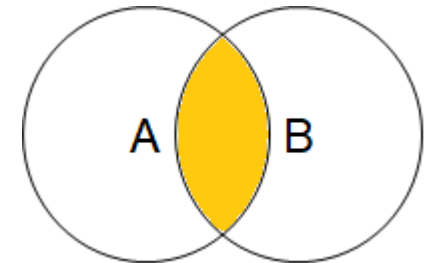
- course inner join prereq*

| <i>course_id</i> | <i>title</i> | <i>dept_name</i> | <i>credits</i> | <i>prereq_id</i> | <i>course_id</i> |
|------------------|--------------|------------------|----------------|------------------|------------------|
| BIO-301 | Genetics | Biology | 4 | BIO-101 | BIO-301 |
| CS-190 | Game Design | Comp. Sci. | 4 | CS-101 | CS-190 |

- If specified as **natural** (**natural inner join**), the second *course_id* field is skipped

| <i>course_id</i> | <i>title</i> | <i>dept_name</i> | <i>credits</i> |
|------------------|--------------|------------------|----------------|
| BIO-301 | Genetics | Biology | 4 |
| CS-190 | Game Design | Comp. Sci. | 4 |
| CS-315 | Robotics | Comp. Sci. | 3 |

| <i>course_id</i> | <i>prereq_id</i> |
|------------------|------------------|
| BIO-301 | BIO-101 |
| CS-190 | CS-101 |
| CS-347 | CS-101 |



Natural Join in SQL

- Natural join matches tuples with the same values for all common attributes, and retains only one copy of each common column
- List the names of instructors along with the course ID of the courses that they taught

```
select name, course_id  
from students, takes  
where student.ID = takes.ID;
```

- Same query in SQL with “natural join” construct

```
select name, course_id  
from student natural join takes;
```

Natural Join in SQL

- The **from** clause can have multiple relations combined using natural join:

```
select A1, A2, ... An
from r1 natural join r2 natural join .. natural join rn
where P;
```

student relation

| ID | name | dept_name | tot_cred |
|-------|----------|------------|----------|
| 00128 | Zhang | Comp. Sci. | 102 |
| 12345 | Shankar | Comp. Sci. | 32 |
| 19991 | Brandt | History | 80 |
| 23121 | Chavez | Finance | 110 |
| 44553 | Peltier | Physics | 56 |
| 45678 | Levy | Physics | 46 |
| 54321 | Williams | Comp. Sci. | 54 |
| 55739 | Sanchez | Music | 38 |
| 70557 | Snow | Physics | 0 |
| 76543 | Brown | Comp. Sci. | 58 |
| 76653 | Aoi | Elec. Eng. | 60 |
| 98765 | Bourikas | Elec. Eng. | 98 |
| 98988 | Tanaka | Biology | 120 |

takes relation

| ID | course_id | sec_id | semester | year | grade |
|-------|-----------|--------|----------|------|-------|
| 00128 | CS-101 | 1 | Fall | 2017 | A |
| 00128 | CS-347 | 1 | Fall | 2017 | A- |
| 12345 | CS-101 | 1 | Fall | 2017 | C |
| 12345 | CS-190 | 2 | Spring | 2017 | A |
| 12345 | CS-315 | 1 | Spring | 2018 | A |
| 12345 | CS-347 | 1 | Fall | 2017 | A |
| 19991 | HIS-351 | 1 | Spring | 2018 | B |
| 23121 | FIN-201 | 1 | Spring | 2018 | C+ |
| 44553 | PHY-101 | 1 | Fall | 2017 | B- |
| 45678 | CS-101 | 1 | Fall | 2017 | F |
| 45678 | CS-101 | 1 | Spring | 2018 | B+ |
| 45678 | CS-319 | 1 | Spring | 2018 | B |
| 54321 | CS-101 | 1 | Fall | 2017 | A- |
| 54321 | CS-190 | 2 | Spring | 2017 | B+ |
| 55739 | MU-199 | 1 | Spring | 2018 | A- |
| 76543 | CS-101 | 1 | Fall | 2017 | A |
| 76543 | CS-319 | 2 | Spring | 2018 | A |
| 76653 | EE-181 | 1 | Spring | 2017 | C |
| 98765 | CS-101 | 1 | Fall | 2017 | C- |
| 98765 | CS-315 | 1 | Spring | 2018 | B |
| 98988 | BIO-101 | 1 | Summer | 2017 | A |
| 98988 | BIO-301 | 1 | Summer | 2018 | null |

student natural join takes

| ID | name | dept_name | tot_cred | course_id | sec_id | semester | year | grade |
|-------|----------|------------|----------|-----------|--------|----------|------|-------|
| 00128 | Zhang | Comp. Sci. | 102 | CS-101 | 1 | Fall | 2017 | A |
| 00128 | Zhang | Comp. Sci. | 102 | CS-347 | 1 | Fall | 2017 | A- |
| 12345 | Shankar | Comp. Sci. | 32 | CS-101 | 1 | Fall | 2017 | C |
| 12345 | Shankar | Comp. Sci. | 32 | CS-190 | 2 | Spring | 2017 | A |
| 12345 | Shankar | Comp. Sci. | 32 | CS-315 | 1 | Spring | 2018 | A |
| 12345 | Shankar | Comp. Sci. | 32 | CS-347 | 1 | Fall | 2017 | A |
| 19991 | Brandt | History | 80 | HIS-351 | 1 | Spring | 2018 | B |
| 23121 | Chavez | Finance | 110 | FIN-201 | 1 | Spring | 2018 | C+ |
| 44553 | Peltier | Physics | 56 | PHY-101 | 1 | Fall | 2017 | B- |
| 45678 | Levy | Physics | 46 | CS-101 | 1 | Fall | 2017 | F |
| 45678 | Levy | Physics | 46 | CS-101 | 1 | Spring | 2018 | B+ |
| 45678 | Levy | Physics | 46 | CS-319 | 1 | Spring | 2018 | B |
| 54321 | Williams | Comp. Sci. | 54 | CS-101 | 1 | Fall | 2017 | A- |
| 54321 | Williams | Comp. Sci. | 54 | CS-190 | 2 | Spring | 2017 | B+ |
| 55739 | Sanchez | Music | 38 | MU-199 | 1 | Spring | 2018 | A- |
| 76543 | Brown | Comp. Sci. | 58 | CS-101 | 1 | Fall | 2017 | A |
| 76543 | Brown | Comp. Sci. | 58 | CS-319 | 2 | Spring | 2018 | A |
| 76653 | Aoi | Elec. Eng. | 60 | EE-181 | 1 | Spring | 2017 | C |
| 98765 | Bourikas | Elec. Eng. | 98 | CS-101 | 1 | Fall | 2017 | C- |
| 98765 | Bourikas | Elec. Eng. | 98 | CS-315 | 1 | Spring | 2018 | B |
| 98988 | Tanaka | Biology | 120 | BIO-101 | 1 | Summer | 2017 | A |
| 98988 | Tanaka | Biology | 120 | BIO-301 | 1 | Summer | 2018 | null |

Dangerous in Natural Join

- Beware of unrelated attributes with same name which get equated incorrectly
- Example: List the names of students instructors along with the titles of courses that they have taken
- Correct version

```
select name, title  
from student natural join takes, course  
where takes.course_id = course.course_id;
```

- Incorrect version

```
select name, title  
from student natural join takes natural join course;
```

- This query omits all (student name, course title) pairs where the student takes a course in a department other than the student's own department
- The correct version (above), correctly outputs such pairs

Natural Join with Using Clause

- To avoid the danger of equating attributes erroneously, we can use the “**using**” construct that allows us to specify exactly which columns should be equated
- Query example

```
select name, title  
from (student natural join takes) join course using (course_id)
```

Outer Join

- An extension of the join operation that avoids loss of information
- Computes the join and then adds tuples from one relation that does not match tuples in the other relation to the result of the join
- Uses *null* values

Left Outer Join

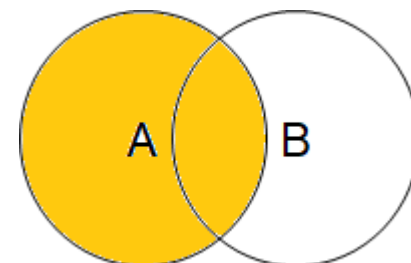
- course* natural left outer join *prereq*

| <i>course_id</i> | <i>title</i> | <i>dept_name</i> | <i>credits</i> | <i>prereq_id</i> |
|------------------|--------------|------------------|----------------|------------------|
| BIO-301 | Genetics | Biology | 4 | BIO-101 |
| CS-190 | Game Design | Comp. Sci. | 4 | CS-101 |
| CS-315 | Robotics | Comp. Sci. | 3 | <i>null</i> |

- In relational algebra: *course* \bowtie *prereq*

| <i>course_id</i> | <i>title</i> | <i>dept_name</i> | <i>credits</i> |
|------------------|--------------|------------------|----------------|
| BIO-301 | Genetics | Biology | 4 |
| CS-190 | Game Design | Comp. Sci. | 4 |
| CS-315 | Robotics | Comp. Sci. | 3 |

| <i>course_id</i> | <i>prereq_id</i> |
|------------------|------------------|
| BIO-301 | BIO-101 |
| CS-190 | CS-101 |
| CS-347 | CS-101 |



Right Outer Join

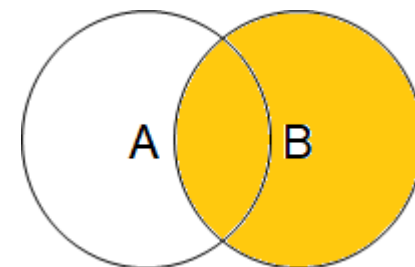
- course* **natural right outer join** *prereq*

| <i>course_id</i> | <i>title</i> | <i>dept_name</i> | <i>credits</i> | <i>prereq_id</i> |
|------------------|--------------|------------------|----------------|------------------|
| BIO-301 | Genetics | Biology | 4 | BIO-101 |
| CS-190 | Game Design | Comp. Sci. | 4 | CS-101 |
| CS-347 | <i>null</i> | <i>null</i> | <i>null</i> | CS-101 |

- In relational algebra: *course* \bowtie *prereq*

| <i>course_id</i> | <i>title</i> | <i>dept_name</i> | <i>credits</i> |
|------------------|--------------|------------------|----------------|
| BIO-301 | Genetics | Biology | 4 |
| CS-190 | Game Design | Comp. Sci. | 4 |
| CS-315 | Robotics | Comp. Sci. | 3 |

| <i>course_id</i> | <i>prereq_id</i> |
|------------------|------------------|
| BIO-301 | BIO-101 |
| CS-190 | CS-101 |
| CS-347 | CS-101 |



Full Outer Join

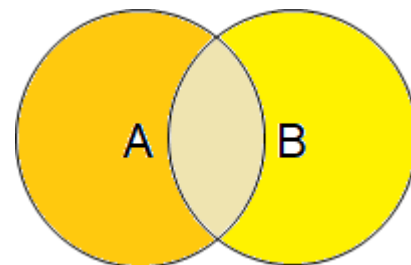
- course* **natural full outer join** *prereq*

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

- In relational algebra: *course* \bowtie *prereq*

| <i>course_id</i> | <i>title</i> | <i>dept_name</i> | <i>credits</i> |
|------------------|--------------|------------------|----------------|
| BIO-301 | Genetics | Biology | 4 |
| CS-190 | Game Design | Comp. Sci. | 4 |
| CS-315 | Robotics | Comp. Sci. | 3 |

| <i>course_id</i> | <i>prereq_id</i> |
|------------------|------------------|
| BIO-301 | BIO-101 |
| CS-190 | CS-101 |
| CS-347 | CS-101 |



Joined Types and Conditions

- **Join operations** take two relations and return as a result another relation
- These additional operations are typically used as subquery expressions in the **from** clause
- **Join condition:** Defines which tuples in the two relations match
- **Join type:** Defines how tuples in each relation that do not match any tuple in the other relation (based on the join condition) are treated

| <i>Join types</i> |
|-------------------------|
| inner join |
| left outer join |
| right outer join |
| full outer join |

| <i>Join conditions</i> |
|---------------------------------------|
| natural |
| on <predicate> |
| using (A_1, A_2, \dots, A_n) |

Join Condition

- The **on** condition allows a general predicate over the relations being joined
- This predicate is written like a **where** clause predicate except for the use of the keyword **on**
- Query example

```
select *  
from student join takes on student_ID = takes_ID
```

- The **on** condition above specifies that a tuple from *student* matches a tuple from *takes* if their *ID* values are equal
- Equivalent to:

```
select *  
from student, takes  
where student_ID = takes_ID
```


Joined Relations – Examples

- course* **inner join** *prereq* on
course.course_id = prereq.course_id

| <i>course_id</i> | <i>title</i> | <i>dept_name</i> | <i>credits</i> | <i>prereq_id</i> | <i>course_id</i> |
|------------------|--------------|------------------|----------------|------------------|------------------|
| BIO-301 | Genetics | Biology | 4 | BIO-101 | BIO-301 |
| CS-190 | Game Design | Comp. Sci. | 4 | CS-101 | CS-190 |

- What is the difference between the above(equi_join), and a natural join?
- course* **left outer join** *prereq* on
course.course_id = prereq.course_id

| <i>course_id</i> | <i>title</i> | <i>dept_name</i> | <i>credits</i> | <i>prereq_id</i> | <i>course_id</i> |
|------------------|--------------|------------------|----------------|------------------|------------------|
| BIO-301 | Genetics | Biology | 4 | BIO-101 | BIO-301 |
| CS-190 | Game Design | Comp. Sci. | 4 | CS-101 | CS-190 |
| CS-315 | Robotics | Comp. Sci. | 3 | <i>null</i> | <i>null</i> |

Joined Relations – Examples

- course* **natural right outer join** *prereq*

| <i>course_id</i> | <i>title</i> | <i>dept_name</i> | <i>credits</i> | <i>prereq_id</i> |
|------------------|--------------|------------------|----------------|------------------|
| BIO-301 | Genetics | Biology | 4 | BIO-101 |
| CS-190 | Game Design | Comp. Sci. | 4 | CS-101 |
| CS-347 | <i>null</i> | <i>null</i> | <i>null</i> | CS-101 |

- course* **full outer join** *prereq* **using** (*course_id*)

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

Next Lecture

Intermediate SQL

Thank you for your attention...

Any question?

Contact:

Department of Information Technology, NITK Surathkal, India
6th Floor, Room: 13

Phone: +91-9477678768

E-mail: shrutilipi@nitk.edu.in