Chapter 6: Formal Relational Query Languages – Part 1

Revision by Gun-Woo Kim

Dept. of Computer Science and Engineering

Hanyang University

Database System Concepts, 6th Ed.

©Silberschatz, Korth and Sudarshan
See www.db-book.com for conditions on re-use

Contents

6.1 Relational Algebra

Relational Algebra (관계대수)

릴레이션에서 원하는 결과를 얻기 위해 수학의 대수와 같은 연산을 이용하여 질의하는 방법을 기술하는 언어

관계대수식은 대상이 되는 릴레이션과 연산자로 구성됨, 결과는 릴레이션으로 반환됨

Six basic operators **하위의 6개 만으로 고급 SQL을 구현가능**

Selection: σ **where랑 같음**

Projection: $\prod$

Union: $\cup$

Set Difference: –

Cartesian Product: x

Rename: ρ **as같은거**

The operators take one or two relations as inputs and produce a new relation as a result.

Urinary Operator (단항 연산자)
: 연산자$_{<p(조건)>}$ 릴레이션

Binary Operator (이항 연산자)
: 릴레이션1 연산자$_{<p(조건)>}$ 릴레이션2

Select Operation – Example

Relation r

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

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

A	B	C	D
α	α	1	7
β	β	23	10

r(A, B, C, D) : { (α, α, 1, 7), (β, β, 23, 10) }

관계대수 연산 결과는 집합형태임

① Select Operation

Notation: $\sigma_p(r)$

p is called the **selection predicate**

선택 술어

Defined as:

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

p(t): 해당 조건안에 t가 만족하면

명제

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: =, $\neq$, >, $\geq$. <. $\leq$

Example of selection:

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

Question

* Find the books that have a cost less than $800 and bookID greater than 3 (from Book table)?

select cost<800 and bookID>3

Project Operation – Example

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

r(A,C) : { (α,1), (β,1), (β,2) }

multi set이라고도함

② Project Operation

Notation:

$$\Pi_{A_1, A_2, \ldots, A_k}(r)$$

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

Duplicate rows removed from result, since relations are sets

Example: To eliminate the *dept_name* attribute of *instructor*

$$\Pi_{ID,\ name,\ salary}(instructor)$$

Question

r(A, B) : { (0,1), (2,3), (0,1), (2,4), (3,4) }

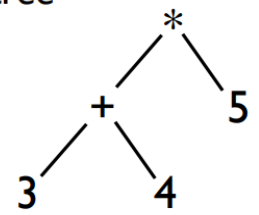
$\Pi_{A+B,\ A^2,\ B^2}(r)$ = ? **{(1,0,1),(5,4,9), … , (A+B,A^2,B^2)}**

Expression Tree for Relational Algebra

Expression Tree (수식 트리)

- Syntax Tree와 유사한 형태

The syntax tree



represents the expression (3+4)*5

$$\Pi_{name,\ dept\_name}$$
$$|$$
$$\sigma_{salary>90000}$$
$$|$$
$$instructor$$

- represents

$$\Pi_{name,\ dept\_name}(\sigma_{salary>90000}(instructor))$$

Union Operation – Example

Relations *r, s:*

A	B
α	1
α	2
β	1

r

A	B
α	2
β	3

s

r ∪ s:

A	B
α	1
α	2
β	1
β	3

r(A,B) : { (α,1), (α,2), (β,1), (β,3) }

③ Union Operation

Notation: $r \cup s$

Defined as:

$$r \cup s = \{t \mid t \in r \text{ or } t \in s\}$$

For $r \cup s$ to be valid.

1. *r, s* must have the *same* **arity** (same number of attributes)
2. The attribute domains must be **compatible** (example: 2nd column of *r* deals with the same type of values as does the 2nd column of *s*)

인자의 개수

즉, 도메인이 같아야 (속성갯수가 다른게 존재하면 집합연산못함) 한다

Example: to find all courses taught in the Fall 2009 semester, or in the Spring 2010 semester, or in both

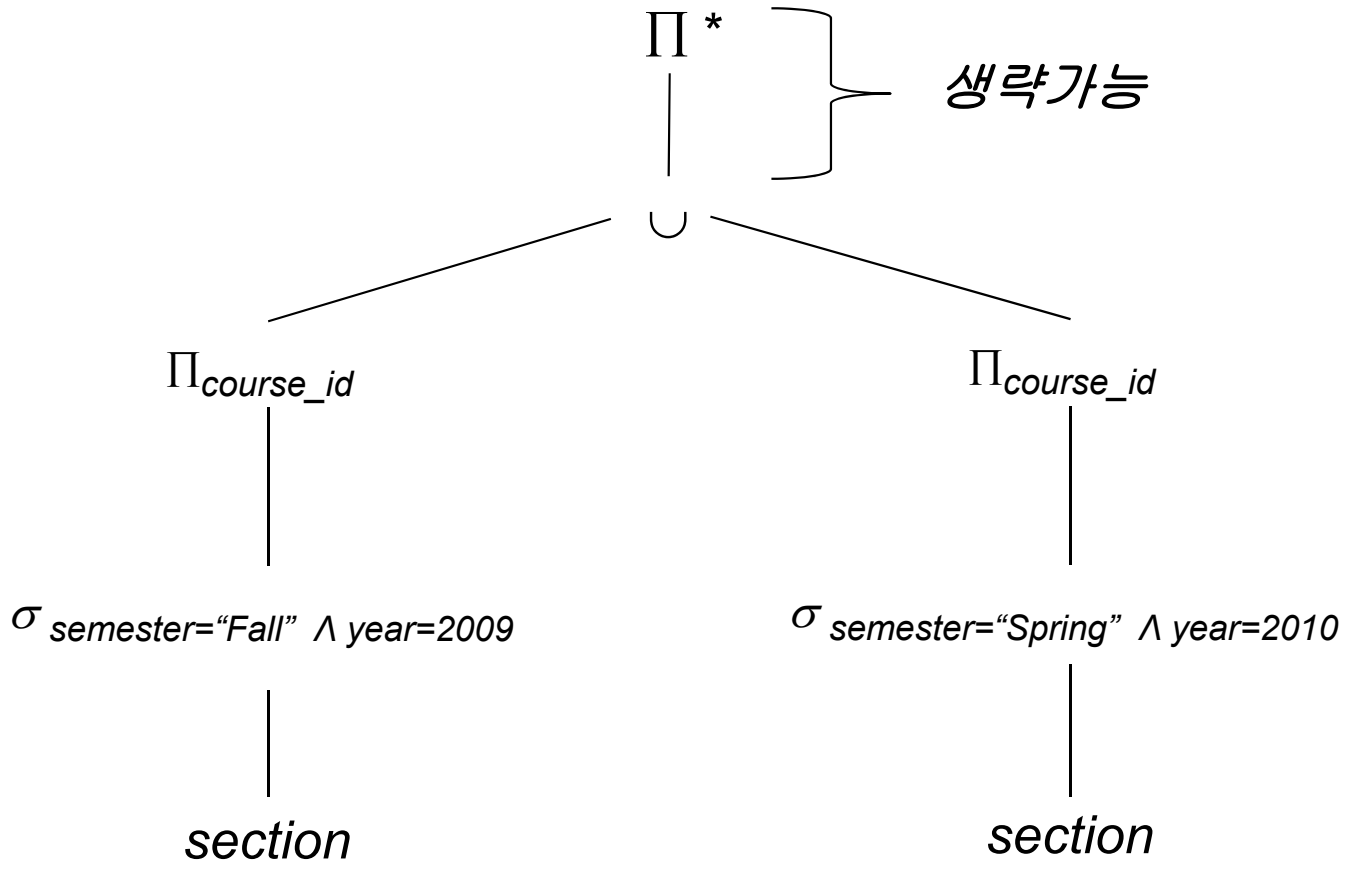
$$\Pi_{course\_id}(\sigma_{semester=\text{“Fall”} \wedge year=2009}(section)) \cup$$
$$\Pi_{course\_id}(\sigma_{semester=\text{“Spring”} \wedge year=2010}(section))$$

section(course_id, sec_id, semester, year, building, room_number, time_slot_id)

③ Union Operation

Expression Tree

$$\Pi_{course\_id}(\sigma_{semester=\text{“Fall”} \wedge year=2009}(section)) \cup \Pi_{course\_id}(\sigma_{semester=\text{“Spring”} \wedge year=2010}(section))$$



Set difference of two relations

Relations *r*, *s*:

A	B
α	1
α	2
β	1

r

A	B
α	2
β	3

s

r – s:

A	B
α	1
β	1

r(A,B) : { (α,1), (β,1) }

④ Set Difference Operation

Notation $r - s$

Defined as:

$$r - s = \{t \mid t \in r \textbf{ and } t \notin s\}$$

Set differences must be taken between **compatible** relations.

- r and s must have the same arity
- attribute domains of r and s must be compatible

Example: to find all courses taught in the Fall 2009 semester, but not in the Spring 2010 semester

$$\Pi_{course\_id}(\sigma_{semester=\text{“Fall”} \wedge year=2009}(section)) -$$
$$\Pi_{course\_id}(\sigma_{semester=\text{“Spring”} \wedge year=2010}(section))$$

Cartesian-Product Operation – Example

Relations *r, s*:

A	*B*
α	1
β	2

r

C	*D*	*E*
α	10	a
β	10	a
β	20	b
γ	10	b

s

r x *s*:

A	*B*	*C*	*D*	*E*
α	1	α	10	a
α	1	β	10	a
α	1	β	20	b
α	1	γ	10	b
β	2	α	10	a
β	2	β	10	a
β	2	β	20	b
β	2	γ	10	b

⑤ Cartesian-Product Operation

Notation r x s

Defined as:

$$r \times s = \{t\, q \mid t \in r \textbf{ and } q \in s\}$$

Assume that attributes of r(R) and s(S) are disjoint. (That is, $R \cap S = \varnothing$).

e.g.) R(name, dept_name, salary)

S(course_id, sec_id, semester, year)

If attributes of *r(R)* and *s(S)* are not disjoint, then renaming must be used.

e.g.) R(id, name, dept_name, salary)

S(id, course_id, sec_id, semester, year)

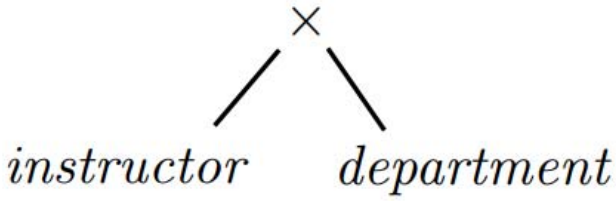
이때 R.id , S.id같은 형식이 필요함

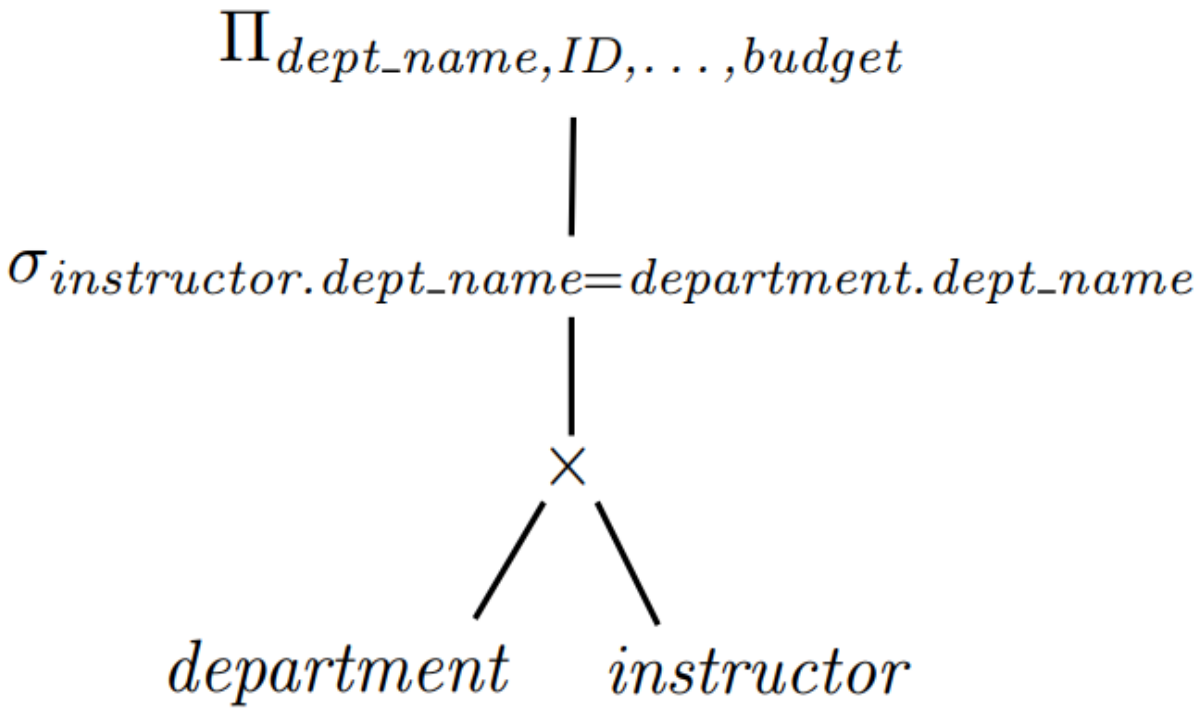
⑤ Cartesian-Product Operation

```
select * from instructor, department;
```

$instructor \times department$

시험에서 수식트리를 그릴줄 알아야함





Composition of Operations

Can build expressions using multiple operations

Example: $\sigma_{A=C}(r \times s)$

r x s를 하나의 테이블로 인식해서 함

r x s

A	*B*	*C*	*D*	*E*
α	1	α	10	a
α	1	β	10	a
α	1	β	20	b
α	1	γ	10	b
β	2	α	10	a
β	2	β	10	a
β	2	β	20	b
β	2	γ	10	b

$\sigma_{A=C}(r \times s)$

A	*B*	*C*	*D*	*E*
α	1	α	10	a
β	2	β	10	a
β	2	β	20	b

⑥ Rename Operation

Allows us to name, and therefore to refer to, the results of relational-algebra expressions.

Allows us to refer to a relation by more than one name.

Example:

$\rho_x(E)$ → 표현식 *E*의 결과를 *x*로 재명명한다.

returns the expression *E* under the name *X*

If a relational-algebra expression *E* has arity *n*, then

$$\rho_{x(A_1, A_2, \ldots, A_n)}(E)$$

Px(projection(course_id{ selection(semester=fall{section})}))같은 형식으로도 사용가능

returns the result of expression *E* under the name *X*, and with the attributes renamed to $A_1, A_2, \ldots, A_n$.

→ "표현식 *E*가 n개의 항으로 구성되었다면, *E*의 결과를 *x*로 재명명하고, $A_1, A_2, .., A_n$으로 재명명된 속성을 갖는다"는 의미

Rename Operation - Example

Find the largest salary in the university

Step 1: find instructor salaries that are less than some other instructor salary (i.e. not maximum)

- using a copy of *instructor* under a new name *d*
- $\Pi_{instructor.salary} (\sigma_{instructor.salary < d.salary} (instructor \times \rho_d (instructor)))$

가장 큰 salary보다 작은 것들만 return

Step 2: Find the largest salary

- $\Pi_{salary} (instructor)$ –
- $\Pi_{instructor.salary} (\sigma_{instructor.salary < d.salary} (instructor \times \rho_d (instructor)))$

결국 최대 salary를 return 함

instructor

name	salary
Lee	5000
Kim	10000
Park	7000

d

name	salary
Lee	5000
Kim	10000
Park	7000

Formal Definition of Relational Algebra

A basic expression in the relational algebra consists of either one of the following:

A relation in the database

A constant relation

Let E_1 and E_2 be relational-algebra expressions; the following are all relational-algebra expressions:

$\sigma_p (E_1)$, P is a predicate on attributes in E_1

$\Pi_s(E_1)$, S is a list consisting of some of the attributes in E_1

$E_1 \cup E_2$

$E_1 - E_2$

$E_1 \times E_2$

$\rho_x (E_1)$, x is the new name for the result of E_1

Additional Relational-Algebra Operations

We define additional operations that do not add any power to the relational algebra, but that simplify common queries.

6개의 기본 연산자만으로 추가적인 관계를 표현하기에 충분하기 때문임.

- Set intersection
- Natural join
- Assignment
- Outer join

⑦ Set-Intersection Operation

Notation: $r \cap s$

Defined as:

$$r \cap s = \{ t \mid t \in r \textbf{ and } t \in s \}$$ <- 이것을 해석하고 쓸줄알아야함

Assume:

r, s have the *same arity*

attributes of r and s are compatible

Note: $r \cap s = r - (r - s)$ 차집합관계일때, r-s는 순수 r부분, 전체r에서 순수 r부분을 뺌

Set-Intersection Operation – Example

Relation *r, s*:

A	*B*
α	1
α	2
β	1

r

A	*B*
α	2
β	3

s

$r \cap s$

A	*B*
α	2

⑧ Natural-Join Operation

Notation: r $\bowtie$ s

Let *r* and *s* be relations on schemas *R* and *S* respectively.
Then, r $\bowtie$ s is a relation on schema $R \cup S$ obtained as follows:

Consider each pair of tuples t_r from *r* and t_s from *s*.

If t_r and t_s have the same value on each of the attributes in $R \cap S$, add a tuple *t* to the result, where

- *t* has the same value as t_r on *r*
- *t* has the same value as t_s on *s*

Example:

$R = (A, \underline{B}, C, \underline{D})$

$S = (E, \underline{B}, \underline{D})$

Result schema = (*A, B, C, D, E*)

r $\bowtie$ *s* is defined as:

$$\Pi_{r.A,\ r.B,\ r.C,\ r.D,\ s.E}\ (\sigma_{r.B = s.B \wedge r.D = s.D}\ (r \times s))$$

Natural Join Example

Relations r, s:

A	B	C	D
α	1	α	a
β	2	γ	a
γ	4	β	b
α	1	γ	a
δ	2	β	b

r

B	D	E
1	a	α
3	a	β
1	a	γ
2	b	δ
3	b	ε

s

r ⋈ s

A	B	C	D	E
α	1	α	a	α
α	1	α	a	γ
α	1	γ	a	α
α	1	γ	a	γ
δ	2	β	b	δ

Natural Join and Theta Join

instructor가 가르치는 모든 course의 title을 이용해서
comp.sci. dept에 있는 모든 instructor의 이름을 찾아라

Find the names of all instructors in the Comp. Sci. department together with the course titles of all the courses that the instructors teach

$$\Pi_{name,\ title}(\sigma_{dept\_name=\text{“Comp. Sci.”}}(instructor \bowtie teaches \bowtie course))$$

Natural join is associative

$(instructor \bowtie teaches) \bowtie course$ is equivalent to
$instructor \bowtie (teaches \bowtie course)$

Natural join is commutative

$instruct \bowtie teaches$ is equivalent to
$teaches \bowtie instructor$

The **theta join** operation $r \bowtie_\theta s$ is defined as

natural조인의 변형된 형태,즉 natural join에서 더많은 조건을 추가해서 적용할수있음

$$r \bowtie_\theta s = \sigma_\theta(r \times s)$$

[세타 조인]

- 세타 (θ) 는 {=, <>, <=, <, >=, >} 중에서 하나가 될 수 있다.
- 세타조인 중에서 “=“인 경우를 동등조인이라 하고, 동등조인에서 중복된 속성을 하나 제거 한 것이 자연조인이다.
- 좀 더 일반적인 형태의 조인이라 할 수 있다.

‘=‘ 인경우 일반적인 natrual join

Natural Join and Theta Join

세타(theta) 조인: $\bowtie_C$

임의의 조건을 만족하는 두 릴레이션의 튜플들을 결합.

1. R 과 S 의 프로덕트를 구한다.
2. 프로덕트 결과로부터, 조건 C 를 만족하는 튜플만을 선택.

A	B	C
1	2	3
6	7	8
9	7	8

U

B	C	D
2	3	4
2	3	5
7	8	10

V

A	U.B	U.C	V.B	V.C	D
1	2	3	2	3	4
1	2	3	2	3	5
1	2	3	7	8	10
6	7	8	7	8	10
9	7	8	7	8	10

$U \bowtie_{A<D} V$

중복된 열이 제거되지 않는다.

⑨ Assignment Operation

배정연산?

The assignment operation (←) provides a convenient way to express complex queries. 쿼리문의 결과값을 일시적인 변수에 저장하는 연산

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.

A	B
α	1
β	2

r

C	D	E
α	10	a
β	10	a
β	20	b
γ	10	b

s

r x s

A	B	C	D	E
α	1	α	10	a
α	1	β	10	a
α	1	β	20	b
α	1	γ	10	b
β	2	α	10	a
β	2	β	10	a
β	2	β	20	b
β	2	γ	10	b

temp1 ← R x S

temp2 ← $\sigma_{A=C}(temp1)$

Result = $\Pi_{A, B}$ (temp2)

$$\Pi_{A, B}(\sigma_{A=C}(R \times S))$$

⑩ Outer Join

An extension of the join operation that avoids loss of information.

Computes the join and then adds tuples form one relation that does not match tuples in the other relation to the result of the join.

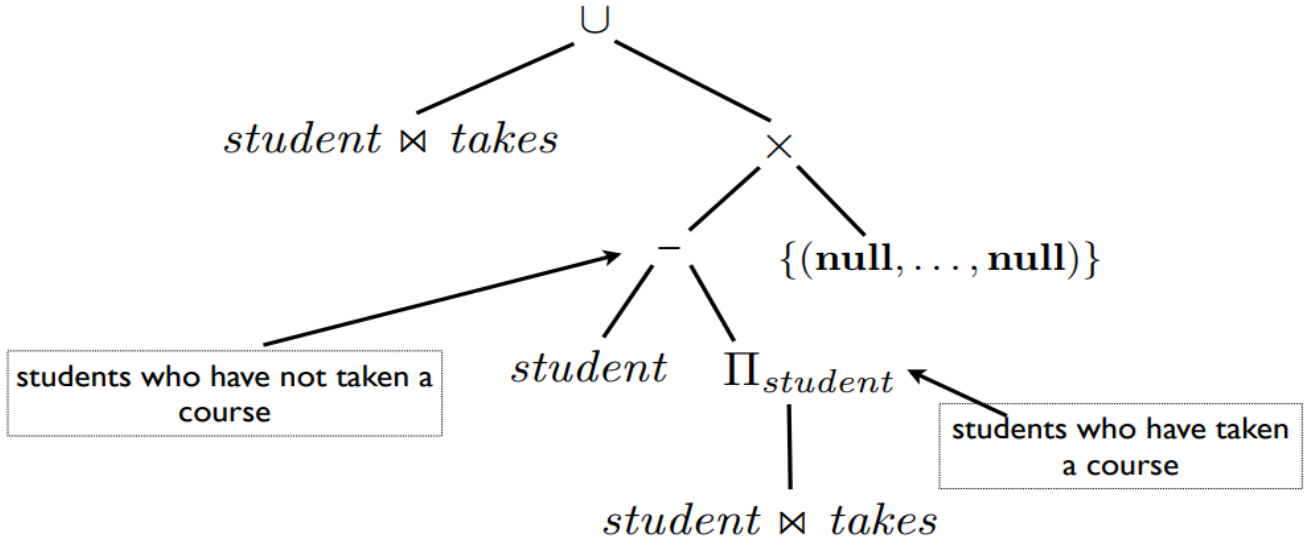
Uses *null* values:

null signifies that the value is unknown or does not exist

All comparisons involving *null* are (roughly speaking) **false** by definition.

- We shall study precise meaning of comparisons with nulls later

- Left outer join defined as



Outer join을 기본 basic operator로만 표현하는 경우

Outer Join – Example

Relation *instructor1*

ID	*name*	*dept_name*
10101	Srinivasan	Comp. Sci.
12121	Wu	Finance
15151	Mozart	Music

Relation *teaches1*

ID	*course_id*
10101	CS-101
12121	FIN-201
76766	BIO-101

Outer Join – Example

Join

instructor ⋈ *teaches*

ID	*name*	*dept_name*	*course_id*
10101	Srinivasan	Comp. Sci.	CS-101
12121	Wu	Finance	FIN-201

Left Outer Join

instructor ⟕ *teaches*

ID	*name*	*dept_name*	*course_id*
10101	Srinivasan	Comp. Sci.	CS-101
12121	Wu	Finance	FIN-201
15151	Mozart	Music	*null*

Outer Join – Example (Cont.)

Right Outer Join

instructor ⟖ *teaches*

ID	*name*	*dept_name*	*course_id*
10101	Srinivasan	Comp. Sci.	CS-101
12121	Wu	Finance	FIN-201
76766	null	null	BIO-101

Full Outer Join

instructor ⟗ *teaches*

ID	*name*	*dept_name*	*course_id*
10101	Srinivasan	Comp. Sci.	CS-101
12121	Wu	Finance	FIN-201
15151	Mozart	Music	*null*
76766	null	null	BIO-101