

3. Relational Model & Basic SQL

▼ Relational Query Languages

- Relational algebra - tuple relational calculus - domain telational calculus

SQL(structured query languages)과 같은 표현식을 가지는 pure language

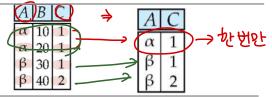


Relational Algebra > tuple 322 AIH >> default

- Selection of tuples (σ): 6 A=B and D>5 (r) tuple है ड्रेंकार्स्ट गर्ह

- =, ≠, 부등호 사용 가능
- 。 tuple 보여줌
- Projection of attributes : TALCCH L) Attrovie GCT.

(34 tuple Her) $\Pi(\Delta Z)(relation)$



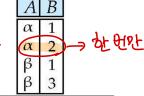
Union of two relations → 기본적으로 attribute가 같아야 함

LIZE tuple 24191

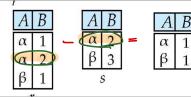
 $r \mid \mid_{S}$







- 。 중복된 건 제거하고 올림
- Set difference of two relations (-)
 - o r-s
 - \circ → r에서 s와 공통으로 가지고 있는 tuple 세거한 r



Set Intersection of two relations









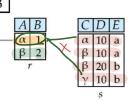


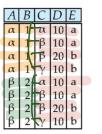
o r과 s의 공통된 tuple만!

• joining two relations - Cartesian Product(x)

िक्षिति वर्ष

 $r \times x$





जात

JCH3

- → 만들 수 있는 모든 pair, 두 relation에 들어 있는 모든 tuple
- 。 불필요한 연산, 데이터 ⇒ 손해
- 。 같은 이름을 갖는 attribute : Felotion 이름 . Attribute 이끌 .
 - \rightarrow 서로 다른 pair라고 인식, 같은 attribute라도 다른 값으로 구분 \star

-					ID	cours	e id se	c id	seme	ster	year	Teache	_
					10101	CS-10			Fall		2017	· (eache	7
TO CHEN AFOR	ID	name	dept_name	salarv	10101	CS-31			Spring	a	2018		
inattactor	22222			95000	10101	CS-34			Fall	9	2017		
	12121			90000									
	32343 45565				12121	FIN-2			Spring		2018		
	98345		Comp. Sci. Elec. Eng.	80000 \	15151	MU-1			Spring	g	2018		
	76766		Biology	72000	22222	PHY-1	101 1		Fall		2017		
	10101				32343	HIS-3	51 1		Spring	g	2018		
	58583		-	62000	45565	CS-10	1 1		Spring	g	2018		
	83821 15151		Comp. Sci. Music	92000 40000	45565	CS-31	9 1		Spring	a a	2018		
	33456		Physics		76766	BIO-1	01 1		Sumn	_	2017		
	76543			20000	76766	BIO-3			Sumn		2018		
'		(1	\									
	\		\	\	83821	CS-19			Spring	_	2017		
	`	\ \	\	\	83821	CS-19			Spring	g	2017		
		\ \	\	\	83821	CS-31	9 2		Spring	g	2018		
_		\ \	\	\ \	98345	EE-18	1 1	ſ	Spring	g	2017		
		\ .	\	\.\		١.	Α,	- 1					
		\sim	V	7	\mathcal{Y}	V	V	W	1	- 1/			
		instructor.II	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-101	1		0	·	BOE	
		22222	Einstein	Physics	95000	10101	CS-347	A					
		22222	Einstein	Physics	95000	12121	FIN-201	α	-	a	7	$1 \mid a \mid \alpha \mid$	
		22222	Einstein	Physics	95000	15151	MU-199	β		a	\times .	3 a β	
		22222	Einstein	Physics	95000	22222	PHY-101	. Υ		b	/	1 a γ	
		1	Database -	Lecture 3			lel & Ba			a		2 b δ	
								<u></u> δ	2 β	b -		3 b ε	

• Joining two relations - Natural Join

 $r\bowtie s$

Tuple (86)

A B C D E

A 1 A a A

A 1 A a A

A 1 A a A

A 1 A a A

A 1 A A A

A 1 A A A

A 2 A

B C D E

。 같은 이름을 갖는 attribute가 있는지 확인

 \rightarrow 서로 같은 값을 가지는 tuple을 하나로 합쳐줌(모든 경우의 수에 대해 작성) $RMS \Rightarrow \tau f$, 七간는 $tuple add \rightarrow t : t_{42} Roll 가지고 있는 <math>t : t_{42} Goll 가지고 있는$

the rename operations → といれる

$$\rho x(E)$$

- expression E를 x로 치환
- 복잡한 친구를 간단한 친구로 대체하는 operation $\rho(A1,A2,..An)(E)$

▼ Example Queries in Relational Algebra

example 1

loan (loan-number, branch-name, amount)

• Find the loan number for each loan of an amount greater than \$1200

example 2

• Find the names of all customers who have a loan, an account, both, from the bank

Find the names of all customers who have a loan and an account at bank.

▼ Introduction to SQL

SQL: data-definition to language > 나람에게 전화한 뉡에는 맛

```
    domain : attribute 범위
```

```
type
```

```
■ char(n) → 고정된 길이
```

- varchar(n) → 境外 小男
- int
- smallint (z+p (n+))
 numeric(p,d) → +++>
 decimal ⇒ d digits
-] flooting point That => byte > to) float(n)

integrity constraints

- primary key (A1, ..., An)
- o foreign key(Am, ..., An) references → += primary key
- not null

▼ Create table construct

```
create table (r) relation of (telation tel schema)
               (A1 D1, A2 D2, ..., An Dn, -) attendate of / domain of .
               (integrity-constraint1),
                                         ) - integrity constant.
               (integrity-constraintk))
       //example
       create table student (
                                    HONNIED OFUZES > 3>+
               ID varchar(5),
               name varchar(20) not null,
               dept_name varchar(20),
               tot_cred numeric(3,0),
THERETY Frimary key (ID),

GOISHOTH: foreign key (dept_name) references department);
```

➤ Updates to tables

- Insert : realtion 안에 tuple value 추가
- **Delete**: 모든 tuple 다 삭제(table은 남아 있음)

- **Drop Table**: table 자체를 삭제(table도 사라짐)
- Alter: 수정하고 싶을 때 사용 ⇒ 0++17 bute 수첫
 - o alter table r add A D: attribute A를 domain D에 null 값으로 추가
 - o alter table r drop A: attribute A를 삭제

▼ Basic Query Structure

· select = 대부분의 guery, 내가 콧다가 하는 tuple 찾아걸

```
GOL Hatement
select A1, A2, ..., An → Mattibute
from r1, r2, ..., rm → relation
where P → predicate
//대소문자 구분 안 함
```

- 。 select distinct ... : 중복된 건 하나만 출력 (상육 temove)
- select all ... : 중복되어도 다 출력 (경박선 적과에서 다)
- o select*...: 모든 tuple의 모든 attribute 출력
- select (존재하지 않는 table 이름): 하나의 열, 행이 있는 table이 임시적으로 생김
 김 → table 만든기.
- o select (") as foo: foo의 attribute의 value가 (기 table이 임시적으로 생김
- ∘ select('A').. : tuple의 value가 'A'로 가득 찬 table 생성
- where : 22th (resultat the RIST blue of)

```
* 경영 보고 전쟁 select name from instructor //table 이름이 오는게 가장 심플하지만 표현이 올 수 있는 where dept_name = 'Comp. Sci.' //내가 원하는 조건에 대해 작성 는 어떻게 구단이 성문 성용이 들은 교수들 select name from instructor
```

3. Relational Model & Basic SQL 5

```
where dept_name = 'Comp. Sci.' and salary > 70000
//여러 가지 조건에 대해 괄호로 구분 지어서 작성하면 good
```

- o and, or, not 사용 가능
- 。 비교연산차 , 부정연산자 사용 가능
- 。 결과에 arithmetic expression 적용할 수 있음
- · from: guety etail Italian telation = littlet

```
(telation algebrae) Cartesian productet (13-25)
select
from instructor teaches
//re/tion 여러 개 작성할 때 ,로 구분 => pair로 만들어줌.
```

- realtional algebra □ cartesian product SQL ver.
- 가능한 모든 pair를 만들어 줌 → 두 re(afton(TNGHRUCHER, teaches)인
 같은 attribute에 대해 → instructor.ID와 같이 설정됨
- o where문과 함께 잘 쓰임. table watchwite
- example [Thetherence CID, name, dept)
 teaches (ID, course td, section)

//1. 주어진 course를 가르친 모든 instructor의 이름과 course_id을 가져운 select name, course id a conteston product > attitute cfexit? from instructor , teaches - telation where instructor.ID = teaches.ID //열린 과목의 ID = 교수의 ID //비교하는 attribute가 겹치는 attribute가 아니라면 table 이름 명시기 //2. 주어진 course를 가르친 미대의 모든 instructor의 이름을 찾음 select name, course_id from instructor , teaches where instructor.ID = teaches.ID (and)instructor. dept_name = 'Art' 1> 77601 bf4 cf 1

FOHERING ZE

6 3. Relational Model & Basic SQL