

SQL and RA

Gary KL Tam

Department of Computer Science
Swansea University

Past Muddiest Points

- RA \Rightarrow SQL Natural Join
- SQL: Natural Join
- SQL: Division

PROF

pid	name	dept	rank	sal
<i>p1</i>	Adam	CS	asst	6000
<i>p2</i>	Bob	EE	asso	8000
<i>p3</i>	Calvin	CS	full	10000
<i>p4</i>	Dorothy	EE	asst	5000
<i>p5</i>	Emily	EE	asso	8500

TEACH

pid	cid	year
<i>p1</i>	<i>c1</i>	2011
<i>p2</i>	<i>c2</i>	2012
<i>p1</i>	<i>c2</i>	2012

PROF \bowtie TEACH returns:

pid	name	dept	rank	sal	cid	year
<i>p1</i>	Adam	CS	asst	6000	<i>c₁</i>	2011
<i>p2</i>	Bob	EE	asso	8000	<i>c₂</i>	2012
<i>p1</i>	Adam	CS	asst	6000	<i>c₂</i>	2012

How many attributes / columns are there?

In general:

Natural Join

$$T_1 \bowtie T_2 = \Pi_S \left(\sigma_{T_1.A_1 = T_2.A_2 \wedge \dots \wedge T_1.A_d = T_2.A_d} (T_1 \times T_2) \right)$$

where

$$S = (S_1 - S_2) \cup \{T_1.A_1, \dots, T_1.A_d\} \cup (S_2 - S_1)$$

where S_1 and S_2 are the schemas of T_1 and T_2 respectively, and A_1, \dots, A_d are the common attributes of T_1 and T_2 .

PROF				
pid	name	dept	rank	sal

TEACH		
pid	cid	year

PROF \bowtie TEACH

A: pid S1-S2: name, dept, rank, sal S2-S1: cid, year

pid	name	dept	rank	sal	cid	year
p1	Adam	CS	asst	6000	c ₁	2011
p2	Bob	EE	asso	8000	c ₂	2012
p1	Adam	CS	asst	6000	c ₂	2012

PROF

pid	name	dept	rank	sal
<i>p1</i>	Adam	CS	asst	6000
<i>p2</i>	Bob	EE	asso	8000
<i>p3</i>	Calvin	CS	full	10000
<i>p4</i>	Dorothy	EE	asst	5000
<i>p5</i>	Emily	EE	asso	8500

TEACH

pid	cid	year
<i>p1</i>	<i>c1</i>	2011
<i>p2</i>	<i>c2</i>	2012
<i>p1</i>	<i>c2</i>	2012

select distinct PROF.pid, name, dept, rank, sal, cid, year
 from PROF, TEACH
 where PROF.pid = TEACH.pid

$\Pi_{\text{PROF.pid, name, dept, rank, sal, cid, year}}(\sigma_{\text{PROF.pid}=\text{TEACH.pid}}(\text{PROF} \times \text{TEACH}))$

PROF \bowtie TEACH

pid	name	dept	rank	sal	cid	year
<i>p1</i>	Adam	CS	asst	6000	<i>c₁</i>	2011
<i>p2</i>	Bob	EE	asso	8000	<i>c₂</i>	2012
<i>p1</i>	Adam	CS	asst	6000	<i>c₂</i>	2012

Past Muddiest Points

- RA \Rightarrow SQL Natural Join
- SQL: Join Condition, Inner Join, Natural Join
- SQL: Division

Past Muddiest Points

- Phone codes

areaCode	cityCode	cityName
40	01792	Swansea
86	10	Beijing
1	416	Toronto
1	212	NewYork

City

areaCode	areaName
40	United Kingdom
852	Hong Kong
1	North America
86	China

Area

```
select areaCode, cityCode, cityName, areaName  
from City, Area
```

```
Where City.areaCode = Area.areaCode
```

Explicit Join Condition

areaCode	cityCode	cityName	areaName
40	01792	Swansea	United Kingdom
86	10	Beijing	China
1	416	Toronto	North America
1	212	NewYork	North America

Past Muddiest Points

● Phone codes

areaCode	cityCode	cityName
40	01792	Swansea
86	10	Beijing
1	416	Toronto
1	212	NewYork

City

areaCode	areaName
40	United Kingdom
852	Hong Kong
1	North America
86	China

Area

```
select areaCode, cityCode, cityName, areaName
from City inner join Area
on City.areaCode = Area.areaCode
```

areaCode	cityCode	cityName	areaName
40	01792	Swansea	United Kingdom
86	10	Beijing	China
1	416	Toronto	North America
1	212	NewYork	North America

Past Muddiest Points

● Phone codes

areaCode	cityCode	cityName
40	01792	Swansea
86	10	Beijing
1	416	Toronto
1	212	NewYork

City

areaCode	areaName
40	United Kingdom
852	Hong Kong
1	North America
86	China

Area

In the past some students gave me this, which I did not teach before.

```
select *  
from City natural join Area
```

areaCode	cityCode	cityName	areaName
40	01792	Swansea	United Kingdom
86	10	Beijing	China
1	416	Toronto	North America
1	212	NewYork	North America

Past Muddiest Points

- One day, a guy added a column

areaCode	cityCode	cityName	lastupdate
40	01792	Swansea	1990
86	10	Beijing	1988
1	416	Toronto	1975
1	212	NewYork	1971

City

areaCode	areaName	lastupdate
40	United Kingdom	1980
852	Hong Kong	1978
1	North America	1979
86	China	1987

Area

```
select areaCode, cityCode, cityName, areaName
from City, Area
Where City.areaCode = Area.areaCode
```

```
select areaCode, cityCode, cityName, areaName
from City inner join Area
on City.areaCode = Area.areaCode
```

areaCode	cityCode	cityName	areaName
40	01792	Swansea	United Kingdom
86	10	Beijing	China
1	416	Toronto	North America
1	212	NewYork	North America

Past Muddiest Points

- One day, a guy added a column

areaCode	cityCode	cityName	lastupdate
40	01792	Swansea	1990
86	10	Beijing	1988
1	416	Toronto	1975
1	212	NewYork	1971

City

areaCode	areaName	lastupdate
40	United Kingdom	1980
852	Hong Kong	1978
1	North America	1979
86	China	1987

Area

```
select *  
from City natural join Area
```

areaCode	cityCode	cityName	areaName	lastupdate
empty				

Natural Join breaks the whole system!

Past Muddiest Points

- One day, the guy change the schema, by mistake...

	cityCode	cityName
	01792	Swansea
	10	Beijing
	416	Toronto
	212	NewYork

City

areaCode	areaName
40	United Kingdom
852	Hong Kong
1	North America
86	China

Area

```
select areaCode, cityCode, cityName, areaName
from City, Area
where City.areaCode = Area.areaCode
```

```
select areaCode, cityCode, cityName, areaName
from City inner join Area
on City.areaCode = Area.areaCode
```

SQL Warning:
No areaCode

```
select *
from City natural join Area
```

Cartesian Product!

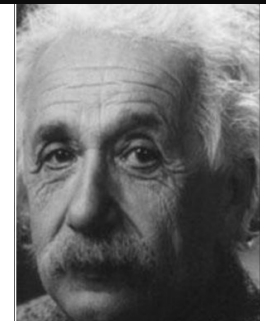
No warning.
But it breaks the system!

Past Muddiest Points

- Lesson learnt
 - Database related codes may not update for decades. aka, compile-time warning in Java / C / C++. Spot mistakes early.
 - If something wrong, you need some warnings reminders.
- Old School Wisdom
 - Use join with **explicit join condition in SQL**
 - Never use **Natural Join Keyword in SQL**
- Natural join in Relational Algebra is fine.
 - The **concept** is fine.
 - Part of the theory, designed for easy manipulation.
- But practically, treat **Natural Join keyword in SQL** as if it never exists, avoids it likes a plague.
- In CW2 & Exam, if you use **natural join keyword in SQL**, **no marks**.
 - Learn the **correct** way of translating concept of natural join from RA to SQL.

In theory, theory and practice are the same. In practice, they are not.

— Albert Einstein —



http://www.dba-oracle.com/oracle_news/2004_2_19_rittman.htm

<http://blog.mclaughlinsoftware.com/2008/05/24/unnatural-outcome-of-natural-joins>

<http://www.postgresqltutorial.com/postgresql-natural-join>

<https://stackoverflow.com/questions/1599050/ansi-vs-non-ansi-sql-join-syntax>

In CS-250: Natural join in SQL (I repeat)

PROF ⋈ TEACH

```
select distinct PROF.pid, name, dept, rank, sal, cid, year  
from PROF, TEACH  
where PROF.pid = TEACH.pid
```



Or using inner join + “on PROF.pid = TEACH.pid”

```
select *  
from PROF natural join TEACH
```



No marks in CS-250.

Gary does not want to see “natural join” keyword in SQL in CWs or exam.

Past Muddiest Points

- RA \Rightarrow SQL Natural Join
- SQL: Natural Join
- SQL: Division

Division

- RA: Division was a bit confusing and to understand

$$T_1 \div T_2$$

T_1	
pid	cid
p1	c1
p1	c2
p1	c3
p2	c2
p2	c3
p3	c1
p4	c1
p4	c2
p4	c3

T_2	
cid	
c1	
c2	
c3	

Results:

p1

p4

$$\Pi_{S_1-S_2}(T_1) - \Pi_{S_1-S_2}(\Pi_{S_1-S_2}(T_1) \times T_2 - T_1) = T_1 \div T_2$$

SQL and Division

```
(select pid from T1)  
minus  
select pid from (  
  (select * from (select pid from T1), T2)  
  minus  
  (select * from T1))
```

$$\Pi_{S_1-S_2}(T_1) - \Pi_{S_1-S_2}(\Pi_{S_1-S_2}(T_1) \times T_2 - T_1) = T_1 \div T_2$$

SQL and Division

(select pid from T_1)

$\Pi_{S_1-S_2}(T_1)$

SQL and Division

`(select * from (select pid from T_1), T_2)`

$$\Pi_{S_1 - S_2}(T_1) \times T_2$$

SQL and Division

```
(select * from (select pid from T1), T2)  
minus  
(select * from T1)
```

$$\left(\Pi_{S_1 - S_2}(T_1) \times T_2 - T_1 \right)$$

SQL and Natural Join

```
select pid from (  
  (select * from (select pid from T1), T2)  
  minus  
  (select * from T1))
```

$$\Pi_{S_1-S_2} \left(\Pi_{S_1-S_2}(T_1) \times T_2 - T_1 \right)$$

SQL and Natural Join

```
(select pid from  $T_1$ )
```

```
select pid from (  
  (select * from (select pid from  $T_1$ ),  $T_2$ )  
  minus  
  (select * from  $T_1$ ))
```

$$\Pi_{S_1-S_2}(T_1) \quad \Pi_{S_1-S_2}(\Pi_{S_1-S_2}(T_1) \times T_2 - T_1)$$

SQL and Division

```
(select pid from T1)  
minus  
select pid from (  
  (select * from (select pid from T1), T2)  
  minus  
  (select * from T1))
```

$$\Pi_{S_1-S_2}(T_1) - \Pi_{S_1-S_2}(\Pi_{S_1-S_2}(T_1) \times T_2 - T_1) = T_1 \div T_2$$