UNIVERSITÀ DEGLI STUDI DI PADOVA

Relational Algebra

Basi di Dati

Bachelor's Degree in Computer Engineering
Academic Year 2024/2025

Ornella Irrera

Intelligent Interactive Information Access (IIIA) Hub
Department of Information Engineering
University of Padua

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Theta-join

Given two relations $R_1(X)$ and $R_2(Y)$and a proposition Θ, the theta-join is the relation

$$R_1 \bowtie_\Theta R_2 = \{t \mid \text{exists } x \in R_1, y \in R_2 \text{ such that } t[X] = x \wedge t[Y] = y \wedge \Theta\}$$

- Given two relations $R_1(X_1, X_2, \ldots, X_n)$ and $R_2(Y_1, Y_2, \ldots, Y_m)$, the theta-join
$$Q = R_1 \bowtie_\Theta R_2 = (X_1, X_2, \ldots, X_n, Y_1, Y_2, \ldots, Y_m)$$
has degree $q = n + m$
- Let $|R_1| = n_{R_1}$ and $|R_2| = n_{R_2}$ be the cardinalities of the two relations, the cardinality of the theta-join, also called **selectivity**, is
$$0 \leq |Q| = n_Q \leq n_{R_1} * n_{R_2}$$
- The proposition Θ can be defined as follows
  - $X_i \; \theta \; Y_i$ with X_i and Y_i attributes of R_1 and R_2 on the same domain and $\theta \in \{<, >, =, \neq, \leq, \geq\}$ comparison operator
  - if ϕ and ψ are propositions, then also $\phi \wedge \psi$ is a proposition

Theta-join

Given two relations $R_1(X)$ and $R_2(Y)$ and a proposition Θ, the theta-join is the relation

$$R_1 \bowtie_\Theta R_2 = \{t \mid \text{exists } x \in R_1, y \in R_2 \text{ such that } t[X] = x \wedge t[Y] = y \wedge \Theta\}$$

Equivalence

$$R_1 \bowtie_\Theta R_2 = \sigma_\Theta(R_1 \times R_2)$$

(non procedural)

- Given two relations $R_1(X_1, X_2, \ldots, X_n)$ and $R_2(Y_1, \ldots, Y_m)$, the theta-join $Q = R_1 \bowtie_\Theta R_2$ has degree $q = n + m$
- Let $|R_1| = n_{R_1}$ and $|R_2| = n_{R_2}$ be the cardinalities of the two relations, the cardinality of the theta-join, also called **selectivity**, is $0 \leq |Q| = n_Q \leq n_{R_1} * n_{R_2}$
- The proposition Θ can be defined as follows
  - $X_i \; \theta \; Y_i$ with X_i and Y_i attributes of R_1 and R_2 on the same domain and $\theta \in \{<, >, =, \neq, \leq, \geq\}$ comparison operator
  - if ϕ and ψ are propositions, then also $\phi \wedge \psi$ is a proposition

Theta-join

Graduated

GBadge	GSurname	GAge
7274	Rossi	42
7432	Neri	54
9824	Verdi	45

Manager

MBadge	MSurname	MAge
9297	Neri	33
7432	Neri	54
9824	Verdi	45

GBadge	GSurname	GAge	MBadge	MSurname	MAge

Product (cartesian): example

Graduated

GBadge	GSurname	GAge
7274	Rossi	42
7432	Neri	54
9824	Verdi	45

Manager

MBadge	MSurname	MAge
9297	Neri	33
7432	Neri	54
9824	Verdi	45

Graduated × Manager

GBadge	GSurname	GAge	MBadge	MSurname	MAge
7274	Rossi	42	9297	Neri	33
7274	Rossi	42	7432	Neri	54
7274	Rossi	42	9824	Verdi	45
7432	Neri	54	9297	Neri	33
7432	Neri	54	7432	Neri	54
7432	Neri	54	9824	Verdi	45
9824	Verdi	45	9297	Neri	33
9824	Verdi	45	7432	Neri	54
9824	Verdi	45	9824	Verdi	45

Theta-join

Graduated

GBadge	GSurname	GAge
7274	Rossi	42
7432	Neri	54
9824	Verdi	45

Manager

MBadge	MSurname	MAge
9297	Neri	33
7432	Neri	54
9824	Verdi	45

Graduated $\bowtie_{\text{GAge} > \text{MAge}}$ Manager

GBadge	GSurname	GAge	MBadge	MSurname	MAge
7274	Rossi	42	9297	Neri	33
7432	Neri	54	9297	Neri	33
7432	Neri	54	9824	Verdi	45
9824	Verdi	45	9297	Neri	33

Theta-join

Project

Location	Pnumber	Hours	Pname
Padova	1	32.5	Product X
Padova	2	7.5	Product Y
Roma	2	20.0	Product Y

Employee

SSNEmp	EnumberPrj	ELocation	Ename
123456	1	Padova	Smith John
236711	2	Padova	Doe John
453453	2	Roma	English Joyce

Theta-join

Project

Location	Pnumber	Hours	Pname
Padova	1	32.5	Product X
Padova	2	7.5	Product Y
Roma	2	20.0	Product Y

Employee

SSNEmp	EnumberPrj	ELocation	Ename
123456	1	Padova	Smith John
236711	2	Padova	Doe John
453453	2	Roma	English Joyce

Project x Employee

Location	Pnumber	Hours	Pname	SSNEmp	ELocation	EnumberPro	Ename
Padova	1	32.5	Product X	123456	Padova	1	Smith John
Padova	1	32.5	Product X	236711	Padova	2	Doe John
Padova	1	32.5	Product X	453453	Roma	2	English Joyce
Padova	2	7.5	Product Y	123456	Padova	1	Smith John
Padova	2	7.5	Product Y	236711	Padova	2	Doe John
Padova	2	7.5	Product Y	453453	Roma	2	English Joyce
Roma	2	20.0	Product Y	123456	Padova	1	Smith John
Roma	2	20.0	Product Y	236711	Padova	2	Doe John
Roma	2	20.0	Product Y	453453	Roma	2	English Joyce

Theta-join

Project

Location	Pnumber	Hours	Pname
Padova	1	32.5	Product X
Padova	2	7.5	Product Y
Roma	2	20.0	Product Y

Employee

SSNEmp	EnumberPrj	ELocation	Ename
123456	1	Padova	Smith John
236711	2	Padova	Doe John
453453	2	Roma	English Joyce

Project ⋈ Location = ELocation AND PNumber = EnumberProj Employee

Location	Pnumber	Hours	Pname	SSNEmp	ELocation	EnumberProj	Ename
Padova	1	32.5	Product X	123456	Padova	1	Smith John
Padova	2	7.5	Product Y	236711	Padova	2	Doe John
Roma	2	20.0	Product Y	453453	Roma	2	English Joyce

Theta-join

Project

Location	Pnumber	Hours	Pname
Padova	1	32.5	Product X
Padova	2	7.5	Product Y
Roma	2	20.0	Product Y

Employee

SSNEmp	EnumberPrj	ELocation	Ename
123456	1	Padova	Smith John
236711	2	Padova	Doe John
453453	2	Roma	English Joyce

Theta-join

Project

Location	Pnumber	Hours	Pname
Padova	1	32.5	Product X
Padova	2	7.5	Product Y
Roma	2	20.0	Product Y

Employee

SSNEmp	EnumberPrj	ELocation	Ename
123456	1	Padova	Smith John
236711	2	Padova	Doe John
453453	2	Roma	English Joyce

Project x Employee

Location	Pnumber	Hours	Pname	SSNEmp	ELocation	EnumberPro	Ename
Padova	1	32.5	Product X	123456	Padova	1	Smith John
Padova	1	32.5	Product X	236711	Padova	2	Doe John
Padova	1	32.5	Product X	453453	Roma	2	English Joyce
Padova	2	7.5	Product Y	123456	Padova	1	Smith John
Padova	2	7.5	Product Y	236711	Padova	2	Doe John
Padova	2	7.5	Product Y	453453	Roma	2	English Joyce
Roma	2	20.0	Product Y	123456	Padova	1	Smith John
Roma	2	20.0	Product Y	236711	Padova	2	Doe John
Roma	2	20.0	Product Y	453453	Roma	2	English Joyce

Theta-join

Project

Location	Pnumber	Hours	Pname
Padova	1	32.5	Product X
Padova	2	7.5	Product Y
Roma	2	20.0	Product Y

Employee

SSNEmp	EnumberPrj	ELocation	Ename
123456	1	Padova	Smith John
236711	2	Padova	Doe John
453453	2	Roma	English Joyce

Project ⋈ Location = ELocation OR PNumber = EnumberProj Employee

Location	Pnumber	Hours	Pname	SSNEmp	ELocation	EnumberPro	Ename
Padova	1	32.5	Product X	123456	Padova	1	Smith John
Padova	1	32.5	Product X	236711	Padova	2	Doe John
Padova	2	7.5	Product Y	123456	Padova	1	Smith John
Padova	2	7.5	Product Y	236711	Padova	2	Doe John
Padova	2	7.5	Product Y	453453	Roma	2	English Joyce
Roma	2	20.0	Product Y	236711	Padova	2	Doe John
Roma	2	20.0	Product Y	453453	Roma	2	English Joyce

Theta-join

Project

Location	Pnumber	Hours	Pname
Padova	1	32.5	Product X
Padova	2	7.5	Product Y
Roma	2	20.0	Product Y

Employee

SSNEmp	EnumberPrj	ELocation	Ename
123456	1	Padova	Smith John
236711	2	Padova	Doe John
453453	2	Roma	English Joyce

Project ⋈ Location = ELocation OR PNumber = EnumberProj Employee

Location	Pnumber	Hours	Pname	SSNEmp	ELocation	EnumberPro	Ename
Padova	1	32.5	Product X	123456	Padova	1	Smith John
Padova	1	32.5	Product X	236711	Padova	2	Doe John
Padova	2	7.5	Product Y	123456	Padova	1	Smith John
Padova	2	7.5	Product Y	236711	Padova	2	Doe John
Padova	2	7.5	Product Y	453453	Roma	2	English Joyce
Roma	2	20.0	Product Y	236711	Padova	2	Doe John
Roma	2	20.0	Product Y	453453	Roma	2	English Joyce

OR produces spurious tuples!!

Theta-join

Project

Location	Pnumber	Hours	Pname
Padova	1	32.5	Product X
Padova	2	7.5	Product Y
Roma	2	20.0	Product Y

Employee

SSNEmp	EnumberPrj	ELocation	Ename
123456	1	Padova	Smith John
236711	2	Padova	Doe John
453453	2	Roma	English Joyce

Project ⋈ Location = ELocation OR PNumber = EnumberProj Employee

Location	Pnumber	Hours	Pname	SSNEmp	ELocation	EnumberPro	Ename
Padova	1	32.5	Product X	123456	Padova	1	Smith John
Padova	1	32.5	Product X	236711	Padova	2	Doe John
Padova	2	7.5	Product Y	123456	Padova	1	Smith John
Padova	2	7.5	Product Y	236711	Padova	2	Doe John
Padova	2	7.5	Product Y	453453	Roma	2	English Joyce
Roma	2	20.0	Product Y	236711	Padova	2	Doe John
Roma	2	20.0	Product Y	453453	Roma	2	English Joyce

OR produces spurious tuples!!

Equi-Join

The **equi-join** is a **theta-join** where we can use only the **=** comparison operator

Equi-Join

The **equi-join** is a **theta-join** where we can use only the **=** comparison operator

Employee

Surname	Dep
Rossi	A
Neri	B
Bianchi	B

Department

Code	Manager
A	Mori
B	Bruni

Employee $\bowtie_{\text{Dep = Code}}$ **Department**

Surname	Dep	Code	Manager

Equi-Join

The **equi-join** is a **theta-join** where we can use only the **=** comparison operator

Employee

Surname	Dep
Rossi	A
Neri	B
Bianchi	B

Department

Code	Manager
A	Mori
B	Bruni

Employee $\bowtie_{Dep = Code}$ Department

Surname	Dep	Code	Manager
Rossi	A	A	Mori
Neri	B	B	Bruni
Bianchi	B	B	Bruni

Natural Join

The **natural join** is an **equi-join** where the **attributes** of the two relations in the join condition have the **same names** and the **duplicated attributes** are **removed** from the schema of the output relation

- Let $R_1(XW)$ and $R_2(YZ)$ the two relations with W and Z sets of attributes with the same name, then the natural join can be expressed as

$$R_1 \bowtie R_2 = \pi_{X,W_1,W_2,\ldots,W_m,Y}\left(R_1 \bowtie_{W_1=Z_1 \wedge W_2=Z_2 \wedge \ldots \wedge W_m=Z_m} R_2\right)$$

Natural Join: Example

Employee

Surname	Dep
Rossi	A
Neri	B
Bianchi	B

Department

Dep	Manager
A	Mori
B	Bruni

Employee ⋈ Department

Surname	Dep	Manager

Natural Join: Example

Employee

Surname	Dep
Rossi	A
Neri	B
Bianchi	B

Department

Dep	Manager
A	Mori
B	Bruni

Employee ⋈ Department

Surname	Dep	Manager
Rossi	A	Mori
Neri	B	Bruni
Bianchi	B	Bruni

Cardinality of the Natural Join: Example (1/3)

Employee (R_1)

Surname	Dep
Rossi	A
Neri	B
Bianchi	B
Neri	C

Department (R_2)

Dep	Manager
A	Mori
B	Bruni
B	Verdi

Employee ⋈ Department

Surname	Dep	Manager
Rossi	A	Mori
Neri	B	Bruni
Neri	B	Verdi
Bianchi	B	Bruni
Bianchi	B	Verdi

Cardinality of the Natural Join: Example (1/3)

Employee (R_1)

Surname	Dep
Rossi	A
Neri	B
Bianchi	B
Neri	C

Department (R_2)

Dep	Manager
A	Mori
B	Bruni
B	Verdi

Dep is not a key for R_2

$$|R_1 \bowtie R_2| = 5 \leq |R_1| * |R_2| = 12$$

Employee ⋈ Department

Surname	Dep	Manager
Rossi	A	Mori
Neri	B	Bruni
Neri	B	Verdi
Bianchi	B	Bruni
Bianchi	B	Verdi

Cardinality of the Natural Join: Example (2/3)

Employee (R_1)

Surname	Dep
Rossi	A
Neri	B
Bianchi	B
Neri	C

Department (R_2)

Dep	Manager
A	Mori
B	Bruni
D	Verdi

Employee ⋈ Department

Surname	Dep	Manager
Rossi	A	Mori
Neri	B	Bruni
Bianchi	B	Bruni

Cardinality of the Natural Join: Example (2/3)

Employee (R_1)

Surname	Dep
Rossi	A
Neri	B
Bianchi	B
Neri	C

Department (R_2)

Dep	Manager
A	Mori
B	Bruni
D	Verdi

Dep is a key (primary) for R_2 but it is not a foreign key for R_1

$$|R_1 \bowtie R_2| = 3 \leq |R_1| = 4$$

Employee ⋈ Department

Surname	Dep	Manager
Rossi	A	Mori
Neri	B	Bruni
Bianchi	B	Bruni

Cardinality of the Natural Join: Example (3/3)

Employee (R_1)

Surname	Dep
Rossi	A
Neri	B
Bianchi	B
Neri	D

Department (R_2)

Dep	Manager
A	Mori
B	Bruni
D	Verdi

Employee ⋈ Department

Surname	Rep	Manager
Rossi	A	Mori
Neri	B	Bruni
Bianchi	B	Bruni
Neri	D	Verdi

Cardinality of the Natural Join: Example (3/3)

Employee (R_1)

Surname	Dep
Rossi	A
Neri	B
Bianchi	B
Neri	D

Department (R_2)

Dep	Manager
A	Mori
B	Bruni
D	Verdi

Dep is a key (primary) for R_2 and a foreign key for R_1

$$|R_1 \bowtie R_2| = 4 = |R_1| = 4$$

Employee ⋈ Department

Surname	Dep	Manager
Rossi	A	Mori
Neri	B	Bruni
Bianchi	B	Bruni
Neri	D	Verdi

Cardinality of the Natural Join (Equi-Join)

- As in the general case, the cardinality of the natural join (equi-join) is

$$0 \leq |R_1 \bowtie R_2| \leq |R_1| * |R_2|$$

- If the natural join (equi-join) involves attributes K which are a key (maybe primary, PK) of R_2 then

$$0 \leq |R_1 \bowtie R_2| \leq |R_1|$$

- If the natural join (equi-join) involves attributes K which are a key (maybe primary, PK) of R_2 and the same attributes are a foreign key FK of R_1 with a referential integrity constraint on R_2 then

$$|R_1 \bowtie R_2| = |R_1|$$

Issues with NULL Values in Relational Algebra

Employee

Badge	Surname	Age	Branch
7309	Rossi	34	Roma
5998	Bianchi	42	Milano
9553	Neri	NULL	Milano

Issues with NULL Values in Relational Algebra

Employee

Badge	Surname	Age	Branch
7309	Rossi	34	Roma
5998	Bianchi	42	Milano
9553	Neri	NULL	Milano

$$\sigma_{\text{Age}>40}(\text{Employee})$$

Badge	Surname	Age	Branch
5998	Bianchi	42	Milano

Issues with NULL Values in Relational Algebra

Employee

Badge	Surname	Age	Branch
7309	Rossi	34	Roma
5998	Bianchi	42	Milano
9553	Neri	NULL	Milano

$$\sigma_{\text{Age}>40}(\text{Employee}) \cup \sigma_{\text{Age}\leq 40}(\text{Employee}) \neq \text{Employee}$$

Selections are evaluated separately

Issues with NULL Values in Relational Algebra

Employee

Badge	Surname	Age	Branch
7309	Rossi	34	Roma
5998	Bianchi	42	Milano
9553	Neri	NULL	Milano

$$\sigma_{\text{Age}>40 \ \vee \ \text{Age}\leq 40}(\text{Employee}) \neq \text{Impiegato}$$

Atomic conditions are evaluated separately

Managing the NULL Values

- To manage NULL values we introduce specific conditions:

  IS NULL

  IS NOT NULL

Employee

Badge	Surname	Age	Branch
7309	Rossi	34	Roma
5998	Bianchi	42	Milano
9553	Neri	NULL	Milano

σ_{Age} **IS NULL** (Employee)

Badge	Surname	Age	Branch
9553	Neri	NULL	Milano

σ_{Age} **IS NOT NULL** (Employee)

Badge	Surname	Age	Branch
7309	Rossi	34	Roma
5998	Bianchi	42	Milano

Managing the NULL Values: Example

Employee

Badge	Surname	Age	Branch
7309	Rossi	34	Roma
5998	Bianchi	42	Milano
9553	Neri	NULL	Milano

$$\sigma_{\text{Age}>40 \,\vee\, \text{Age}\leq 40 \,\vee\, \text{Age } \textbf{IS NULL}}(\text{Employee}) = \text{Employee}$$

$$\left.\begin{array}{l} \sigma_{\text{Age}>40}(\text{Employee}) \\ \cup \\ \sigma_{\text{Age}\leq 40}(\text{Employee}) \\ \cup \\ \sigma_{\text{Age } \textbf{IS NULL}}(\text{Employee}) \end{array}\right\} = \text{Employee}$$

Join: Dangling Tuples

Employee (R_1)

Surname	Dep
Rossi	A
Neri	B
Bianchi	B
Neri	C

Department (R_2)

Dep	Manage
A	Mori
B	Bruni
D	Verdi

Employee ⋈ Department

Surname	Dep	Manager
Rossi	A	Mori
Neri	B	Bruni
Bianchi	B	Bruni

- Some tuples are dangling and do not contribute to the results

Full Outer Join

Given two relations $R_1(XW)$ and $R_2(YZ)$ with W and Z sets of attributes on the same domain, the full outer join is

$$R_1 ⟗_{W\ \theta\ Z} R_2 = R_1 ⋈_{W\ \theta\ Z} R_2$$
$$\bigcup$$
$$\left(R_1 - \pi_{X,W}(R_1 ⋈_{W\ \theta\ Z} R_2)\right) \times \{Y = \text{NULL}, Z = \text{NULL}\}$$
$$\bigcup$$
$$\left(R_2 - \pi_{Y,Z}(R_1 ⋈_{W\ \theta\ Z} R_2)\right) \times \{X = \text{NULL}, W = \text{NULL}\}$$

- The full outer join extends, with NULL values, the tuples which are excluded from the theta-join, also called **inner join**, keeping tuples from both operands

Left Outer Join

Given two relations $R_1(XW)$ and $R_2(YZ)$ with W and Z sets of attributes on the same domain, the left outer join is

$$R_1 ⟕_{W\ \theta\ Z} R_2 = R_1 \bowtie_{W\ \theta\ Z} R_2$$
$$\bigcup$$
$$\left(R_1 - \pi_{X,W}(R_1 \bowtie_{W\ \theta\ Z} R_2)\right) \times \{Y = \text{NULL}, Z = \text{NULL}\}$$

Right Outer Join

Given two relations $R_1(XW)$ and $R_2(YZ)$ with W and Z sets of attributes on the same domain, the right outer join is

$$R_1 \rightouterjoin_{W\ \theta\ Z} R_2 = R_1 \bowtie_{W\ \theta\ Z} R_2 \cup \left(R_2 - \pi_{Y,Z}(R_1 \bowtie_{W\ \theta\ Z} R_2)\right) \times \{X = \text{NULL}, W = \text{NULL}\}$$

Full Outer Join: Example

Employee (R_1)

Surname	Dep
Rossi	A
Neri	B
Bianchi	B
Neri	C

Department (R_2)

Code	Manager
A	Mori
B	Bruni
D	Verdi

Employee $\bowtie_{Dep=Code}$ Department

Surname	Dep	Code	Manager
Rossi	A	A	Mori
Neri	B	B	Bruni
Bianchi	B	B	Bruni
Neri	C	NULL	NULL
NULL	NULL	D	Verdi

Full Outer Join: Example

Employee (R_1)

Surname	Dep
Rossi	A
Neri	B
Bianchi	B
Neri	C

Department (R_2)

Code	Manager
A	Mori
B	Bruni
D	Verdi

Employee $\bowtie_{\text{Dep}=\text{Code}}$ **Department**

Employee $\bowtie_{\text{Dep}=\text{Code}}$ **Department**

Surname	Dep	Code	Manager
Rossi	A	A	Mori
Neri	B	B	Bruni
Bianchi	B	B	Bruni
Neri	C	NULL	NULL
NULL	NULL	D	Verdi

Full Outer Join: Example

Employee (R_1)

Surname	Dep
Rossi	A
Neri	B
Bianchi	B
Neri	C

Department (R_2)

Code	Manager
A	Mori
B	Bruni
D	Verdi

$$(\textbf{Employee} - \pi_{\text{Surname,Dep}}(\textbf{Employee} \bowtie_{\text{Dep}=\text{Code}} \textbf{Department})) \times \{\text{Code} = \text{NULL}, \text{Manager} = \text{NULL}\}$$

Employee $\bowtie_{\text{Dep}=\text{Code}}$ Department

Surname	Dep	Code	Manager
Rossi	A	A	Mori
Neri	B	B	Bruni
Bianchi	B	B	Bruni
Neri	C	NULL	NULL
NULL	NULL	D	Verdi

Full Outer Join: Example

Employee (R_1)

Surname	Dep
Rossi	A
Neri	B
Bianchi	B
Neri	C

Department (R_2)

Code	Manager
A	Mori
B	Bruni
D	Verdi

$$\left(\textbf{Department} - \pi_{\text{Code,Manager}}\left(\textbf{Employee} \bowtie_{\text{Rep}=\text{Code}} \textbf{Department}\right)\right) \times \{\text{Surname} = \text{NULL}, \text{Dep} = \text{NULL}\}$$

Employee $\bowtie_{\text{Dep}=\text{Code}}$ Department

Surname	Dep	Code	Manager
Rossi	A	A	Mori
Neri	B	B	Bruni
Bianchi	B	B	Bruni
Neri	C	NULL	NULL
NULL	NULL	D	Verdi

Left Outer Join: Example

Employee (R_1)

Surname	Dep
Rossi	A
Neri	B
Bianchi	B
Neri	C

Department (R_2)

Code	Manager
A	Mori
B	Bruni
D	Verdi

Employee $\bowtie_{Dep=Code}$ Department

Surname	Dep	Code	Manager
Rossi	A	A	Mori
Neri	B	B	Bruni
Bianchi	B	B	Bruni
Neri	C	NULL	NULL

Left Outer Join: Example

Employee (R_1)

Surname	Dep
Rossi	A
Neri	B
Bianchi	B
Neri	C

Department (R_2)

Code	Manager
A	Mori
B	Bruni
D	Verdi

Employee $\bowtie_{\text{Dep}=\text{Code}}$ **Department**

Employee $\bowtie_{\text{Dep}=\text{Code}}$ **Department**

Surname	Dep	Code	Manager
Rossi	A	A	Mori
Neri	B	B	Bruni
Bianchi	B	B	Bruni
Neri	C	NULL	NULL

Left Outer Join: Example

Employee (R_1)

Surname	Dep
Rossi	A
Neri	B
Bianchi	B
Neri	C

Department (R_2)

Code	Manager
A	Mori
B	Bruni
D	Verdi

$$\left(\textbf{Employee} - \pi_{\text{Surname,Dep}}\left(\textbf{Employee} \bowtie_{\text{Dep}=\text{Code}} \textbf{Department}\right)\right) \times \{\text{Code} = \text{NULL}, \text{Manager} = \text{NULL}\}$$

Employee $\bowtie_{\text{Dep}=\text{Code}}$ Department

Surname	Dep	Code	Manager
Rossi	A	A	Mori
Neri	B	B	Bruni
Bianchi	B	B	Bruni
Neri	C	NULL	NULL

Right Outer Join: Example

Employee (R_1)

Surname	Dep
Rossi	A
Neri	B
Bianchi	B
Neri	C

Department (R_2)

Code	Manager
A	Mori
B	Bruni
D	Verdi

Employee $\bowtie_{Dep=Code}$ Department

Surname	Dep	Code	Manager
Rossi	A	A	Mori
Neri	B	B	Bruni
Bianchi	B	B	Bruni
NULL	NULL	D	Verdi

Right Outer Join: Example

Employee (R_1)

Surname	Dep
Rossi	A
Neri	B
Bianchi	B
Neri	C

Department (R_2)

Code	Manager
A	Mori
B	Bruni
D	Verdi

Employee $\bowtie_{\text{Dep}=\text{Code}}$ **Department**

Employee $\overrightarrow{\bowtie}_{\text{Dep}=\text{Code}}$ **Department**

Surname	Dep	Code	Manager
Rossi	A	A	Mori
Neri	B	B	Bruni
Bianchi	B	B	Bruni
NULL	NULL	D	Verdi

Right Outer Join: Example

Employee (R_1)

Surname	Dep
Rossi	A
Neri	B
Bianchi	B
Neri	C

Department (R_2)

Code	Manager
A	Mori
B	Bruni
D	Verdi

$$\left(\textbf{Department} - \pi_{\text{Code,Manager}}\left(\textbf{Employee} \bowtie_{\text{Rep}=\text{Code}} \textbf{Department}\right)\right) \times \{\text{Surname} = \text{NULL}, \text{Dep} = \text{NULL}\}$$

Employee $\bowtie_{\text{Dep}=\text{Code}}$ Department

Surname	Dep	Code	Manager
Rossi	A	A	Mori
Neri	B	B	Bruni
Bianchi	B	B	Bruni
NULL	NULL	D	Verdi

Outer Join and NULL Values

Employee (R_1)

Surname	Dep
Rossi	A
Neri	B
Bianchi	B
Neri	NULL

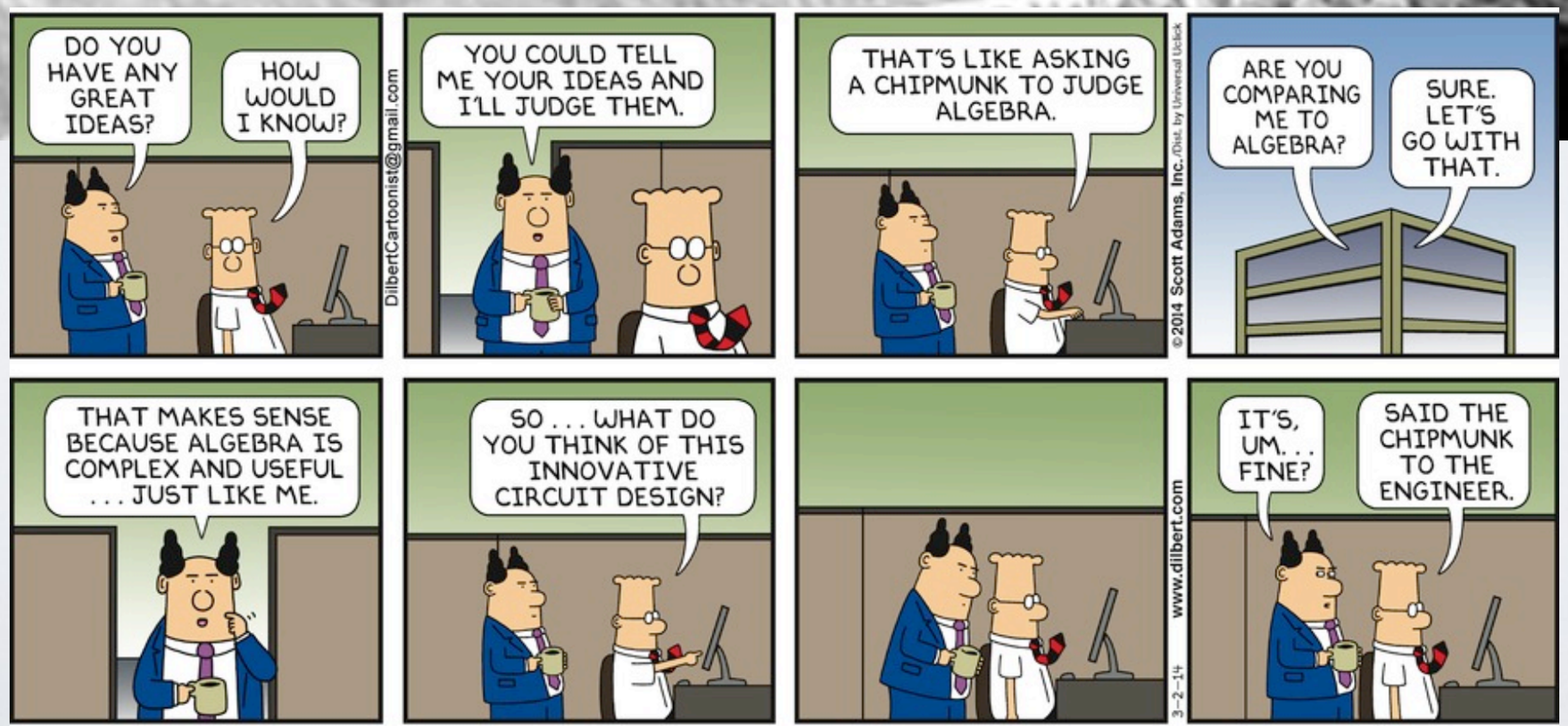
Department (R_2)

Code	Manager
A	Mori
B	Bruni
NULL	Verdi

Employee $\bowtie_{Dep=Code}$ Department

Surname	Dep	Code	Manager
Rossi	A	A	Mori
Neri	B	B	Bruni
Bianchi	B	B	Bruni
Neri	NULL	NULL	NULL
NULL	NULL	NULL	Verdi

Questions?


DO YOU HAVE ANY GREAT IDEAS?
HOW WOULD I KNOW?
DilbertCartoonist@gmail.com
YOU COULD TELL ME YOUR IDEAS AND I'LL JUDGE THEM.
THAT'S LIKE ASKING A CHIPMUNK TO JUDGE ALGEBRA.
ARE YOU COMPARING ME TO ALGEBRA?
SURE. LET'S GO WITH THAT.
© 2014 Scott Adams, Inc. /Dist. by Universal Uclick
THAT MAKES SENSE BECAUSE ALGEBRA IS COMPLEX AND USEFUL . . . JUST LIKE ME.
SO . . . WHAT DO YOU THINK OF THIS INNOVATIVE CIRCUIT DESIGN?
IT'S, UM. . . FINE?
SAID THE CHIPMUNK TO THE ENGINEER.
www.dilbert.com
3-2-14