Relational Model and Algebra

P.J. McBrien

Imperial College London

Relations are sets of typed tuples

Relations

Relations take the form $R(A, B, \ldots)$ where

- R is the name of the relation
- $A, B, \ldots$ is the set of attributes of the relation
  - Often write the set without commas: $A, B, \ldots \equiv AB \ldots$, and can refer to a set of attributes as A
  - The number of attributes n is the **arity** of the relation
    Can call $R(A_1, \ldots, A_n)$ an n-ary relation
  - $Domain(A)$ is the set of values (type) that the attribute can have
  - Will use $Atts(R)$ to find $A, B, \ldots$
- The **extent** of $R(A, B, \ldots)$ is the set of **tuples**
  $\{\langle v_1^A, v_1^B, \ldots \rangle, \langle v_2^A, v_2^B, \ldots \rangle, \langle v_3^A, v_3^B, \ldots \rangle, \ldots\}$
  - $\forall x. v_x^A \in Domain(A)$
  - No duplicate tuples
  - Not ordered
  - All tuples have the same arity

Relation=Table

R		
A	B	$\ldots$
v_1^A	v_1^B	$\ldots$
v_2^A	v_2^B	$\ldots$
v_3^A	v_3^B	$\ldots$
	$\vdots$	

Set Semantics

- Order of columns not significant
- Order of rows not significant
- No duplicate rows

- Attribute=Column
- Tuple=Row

Quiz 1: Equivalent Relations

Which is the odd one out?

A

branch		
sortcode	bname	cash
56	'Wimbledon'	94340.45
34	'Goodge St'	8900.67
67	'Strand'	34005.00

B

branch		
bname	sortcode	cash
'Wimbledon'	56	94340.45
'Goodge St'	34	8900.67
'Strand'	67	34005.00

C

branch		
sortcode	bname	cash
34	'Goodge St'	8900.67
56	'Wimbledon'	94340.45
67	'Strand'	34005.00

D

branch		
sortcode	bname	cash
56	'Wimbledon'	94340.45
56	'Wimbledon'	94340.45
34	'Goodge St'	8900.67
67	'Strand'	34005.00

Handling 'missing' attribute values

Suppose we want to have a relation account(no,type,cname,rate,sortcode), but not all accounts have a rate.

Solution 1: Separate relations

account			
no	type	cname	sortcode
100	'current'	'McBrien, P.'	67
101	'deposit'	'McBrien, P.'	67
103	'current'	'Boyd, M.'	34
107	'current'	'Poulovassilis, A.'	56
119	'deposit'	'Poulovassilis, A.'	56
125	'current'	'Bailey, J.'	56

account_rate	
no	rate
101	5.25
119	5.50

Solution 2: NULL values

account				
no	type	cname	rate	sortcode
100	'current'	'McBrien, P.'	NULL	67
101	'deposit'	'McBrien, P.'	5.25	67
103	'current'	'Boyd, M.'	NULL	34
107	'current'	'Poulovassilis, A.'	NULL	56
119	'deposit'	'Poulovassilis, A.'	5.50	56
125	'current'	'Bailey, J.'	NULL	56

Relational Keys

Key

A **key** of a relation $R(AB\ldots)$ is a subset of the attributes for which the values in any extent are unique across all tuples

- Every relation has at least one key, which is the entire set of attributes
- A key is **violated** by there being two tuples in the extent which have the same values for the attributes of the key
- If A is a key, then so must AB be a key
- A **minimal key** is a set of attributes $AB\ldots$ for which no subset of the attributes is also a key
- The **primary key** is one of the keys of the relation: serves as the default key when no key explicitly stated

Quiz 2: Violation of Relational Keys

movement			
mid	no	amount	tdate
1000	100	2300.00	5/1/1999
1001	101	4000.00	5/1/1999
1002	100	-223.45	5/1/1999
1004	107	-100.00	11/1/1999
1005	103	145.50	12/1/1999
1006	100	10.23	15/1/1999
1007	107	345.56	15/1/1999
1008	101	1230.00	15/1/1999
1009	119	5600.00	18/1/1999

Which key is violated?

A

movement(mid)

B

movement(no,amount)

C

movement(no,tdate)

D

movement(amount,tdate)

Quiz 3: Correct Keys for Relations

movement			
mid	no	amount	tdate
1000	100	2300.00	5/1/1999
1001	101	4000.00	5/1/1999
1002	100	-223.45	5/1/1999
1004	107	-100.00	11/1/1999
1005	103	145.50	12/1/1999
1006	100	10.23	15/1/1999
1007	107	345.56	15/1/1999
1008	101	1230.00	15/1/1999
1009	119	5600.00	18/1/1999

Which key makes most sense in a bank UoD?

A

movement(amount)

B

movement(no,amount)

C

movement(no,tdate)

D

movement(amount,tdate)

Relational Foreign Keys

Foreign Key

A **foreign key** $R(\vec{X}) \overset{fk}{\Rightarrow} S(\vec{Y})$ of a relation $R(AB\ldots)$ is a subset $\vec{X} \subseteq AB\ldots$ of the attributes for which the values in the extent of R also appear as values of attributes $\vec{Y}$ in the extent of S, and $\vec{Y}$ is a key of S.

account(sortcode) $\overset{rk}{\Rightarrow}$ branch(sortcode)

account

no	type	cname	rate	sortcode
100	'current'	'McBrien, P.'	NULL	67
101	'deposit'	'McBrien, P.'	5.25	67
103	'current'	'Boyd, M.'	NULL	34
107	'current'	'Poulovassilis, A.'	NULL	56
119	'deposit'	'Poulovassilis, A.'	5.50	56
125	'current'	'Bailey, J.'	NULL	56

key branch(sortcode)

branch

sortcode	bname	cash
56	'Wimbledon'	94340.45
34	'Goodge St'	8900.67
67	'Strand'	34005.00

Quiz 4: Foreign Key Violation

account(sortcode) $\stackrel{fk}{\Rightarrow}$ branch(sortcode)

account

no	type	cname	rate	sortcode
100	'current'	'McBrien, P.'	NULL	67
101	'deposit'	'McBrien, P.'	5.25	67
103	'current'	'Boyd, M.'	NULL	34
107	'current'	'Poulovassilis, A.'	NULL	56
119	'deposit'	'Poulovassilis, A.'	5.50	56
125	'current'	'Bailey, J.'	NULL	56

key branch(sortcode)

branch

sortcode	bname	cash
56	'Wimbledon'	94340.45
34	'Goodge St'	8900.67
67	'Strand'	34005.00

Which update violates the foreign key?

A

insert into account
(126,'business','McBrien, P.',1.00,67)

B

insert into branch
(78,'Ealing',1000.00)

C

delete from branch
(67,'Strand',34005.00)

D

delete from account
(103,'current','Boyd, M.',NULL,34)

Example Relational Schema

branch

sortcode	bname	cash
56	'Wimbledon'	94340.45
34	'Goodge St'	8900.67
67	'Strand'	34005.00

movement

mid	no	amount	tdate
1000	100	2300.00	5/1/1999
1001	101	4000.00	5/1/1999
1002	100	-223.45	8/1/1999
1004	107	-100.00	11/1/1999
1005	103	145.50	12/1/1999
1006	100	10.23	15/1/1999
1007	107	345.56	15/1/1999
1008	101	1230.00	15/1/1999
1009	119	5600.00	18/1/1999

account

no	type	cname	rate	sortcode
100	'current'	'McBrien, P.'	NULL	67
101	'deposit'	'McBrien, P.'	5.25	67
103	'current'	'Boyd, M.'	NULL	34
107	'current'	'Poulovassilis, A.'	NULL	56
119	'deposit'	'Poulovassilis, A.'	5.50	56
125	'current'	'Bailey, J.'	NULL	56

key branch(sortcode)
key branch(bname)
key movement(mid)
key account(no)

movement(no) $\stackrel{fk}{\Rightarrow}$ account(no)

account(sortcode) $\stackrel{fk}{\Rightarrow}$ branch(sortcode)

Relational Algebra: A Query Language for the Relational Model

Primitive operators of the Relational Algebra

Symbol	Name	Type
π	Project	Unary
σ	Select	Unary
$\times$	Cartesian Product	Binary
$\cup$	Union	Binary
$-$	Difference	Binary

- All operators take relations as input
- All operators produce one relation as their output
- Other (useful) operators may be defined in terms of the five primitive operators

Relational Algebra: Project π

account				
no	type	cname	rate	sortcode
100	'current'	'McBrien, P.'	NULL	67
101	'deposit'	'McBrien, P.'	5.25	67
103	'current'	'Boyd, M.'	NULL	34
107	'current'	'Poulovassilis, A.'	NULL	56
119	'deposit'	'Poulovassilis, A.'	5.50	56
125	'current'	'Bailey, J.'	NULL	56

Project Operator

$\pi_{no,type}$ account

no	type
100	'current'
101	'deposit'
103	'current'
107	'current'
119	'deposit'
125	'current'

$\pi_{sortcode}$ account

sortcode
67
34
56

Relational Algebra: Select σ

account

no	type	cname	rate	sortcode
100	'current'	'McBrien, P.'	NULL	67
101	'deposit'	'McBrien, P.'	5.25	67
103	'current'	'Boyd, M.'	NULL	34
107	'current'	'Poulovassilis, A.'	NULL	56
119	'deposit'	'Poulovassilis, A.'	5.50	56
125	'current'	'Bailey, J.'	NULL	56

Select Operator

$\sigma_{rate>0}$ account

no	type	cname	rate	sortcode
101	'deposit'	'McBrien, P.'	5.25	67
119	'deposit'	'Poulovassilis, A.'	5.50	56

Relational Algebra: Product ×

branch

sortcode	bname	cash
56	'Wimbledon'	94340.45
34	'Goodge St'	8900.67
67	'Strand'	34005.00

$\sigma_{rate>0}$account

no	type	cname	rate	sortcode
101	'deposit'	'McBrien, P.'	5.25	67
119	'deposit'	'Poulovassilis, A.'	5.50	56

Product Operator

branch × $\sigma_{rate>0}$account

sortcode	bname	cash	no	type	cname	rate	sortcode
56	'Wimbledon'	94340.45	101	'deposit'	'McBrien, P.'	5.25	67
56	'Wimbledon'	94340.45	119	'deposit'	'Poulovassilis, A.'	5.50	56
34	'Goodge St'	8900.67	101	'deposit'	'McBrien, P.'	5.25	67
34	'Goodge St'	8900.67	119	'deposit'	'Poulovassilis, A.'	5.50	56
67	'Strand'	34005.00	101	'deposit'	'McBrien, P.'	5.25	67
67	'Strand'	34005.00	119	'deposit'	'Poulovassilis, A.'	5.50	56

Quiz 5: RA Queries

branch

sortcode	bname	cash
56	'Wimbledon'	94340.45
34	'Goodge St'	8900.67
67	'Strand'	34005.00

account

no	type	cname	rate	sortcode
100	'current'	'McBrien, P.'	NULL	67
101	'deposit'	'McBrien, P.'	5.25	67
103	'current'	'Boyd, M.'	NULL	34
107	'current'	'Poulovassilis, A.'	NULL	56
119	'deposit'	'Poulovassilis, A.'	5.50	56
125	'current'	'Bailey, J.'	NULL	56

Which RA query lists the name of branches that have deposit accounts?

A

$\pi_{\text{sortcode}}\, \sigma_{\text{type}=\text{'deposit'}}\, \text{account}$

B

$\pi_{\text{bname}}\, \sigma_{\text{account.sortcode}=\text{branch.sortcode} \wedge \text{type}=\text{'deposit'}}\, (\text{account} \times \text{branch})$

C

$\pi_{\text{bname}}(\text{branch} \times \sigma_{\text{type}=\text{'deposit'}}\, \text{account})$

D

$\pi_{\text{bname}}\, \sigma_{\text{type}=\text{'deposit'}}(\text{account} \times \text{branch})$

SPJ Queries

Select Project Join (SPJ) queries

If a product of tables is formed, where a selection is then done that compares the attributes of those tables, we say that a **join** has been performed.
Normally not all columns of the product are returned, and therefore a project is also required.

Branches with current accounts

$\pi_{\text{bname,no}}\, \sigma_{\text{branch.sortcode}=\text{account.sortcode} \wedge \text{account.type}=\text{'current'}} (\text{branch} \times \text{account})$

bname	no
'Goodge St'	103
'Wimbledon'	107
'Wimbledon'	125
'Strand'	100

Relational Algebra: Union ∪

$\pi_{\text{sortcode as id}}$account

id
67
34
56

$\pi_{\text{no as id}}$account

id
100
101
103
107
119
125

Union Operator

$\pi_{\text{sortcode as id}}$account ∪ $\pi_{\text{no as id}}$account

id
67
34
56
100
101
103
107
119
125

- relations must be **union compatible**

Relational Algebra: Difference –

π_{no}account

no
100
101
103
107
119
126

π_{no}movement

no
100
101
103
107
119

Difference Operator

π_{no}account – π_{no}movement

no
125

- relations must be **union compatible**

Rules for Combining Operators

Since all operators produce a relation as output, *any* operator may produce one of the inputs to any other operator.

well formed RA query

- the output of the nested operator must contain the attributes required by an outer π or σ
- the two inputs to a $\cup$ or $-$ must contain the same number of attributes

Quiz 6: Well formed queries

account

no	type	cname	rate	sortcode
100	'current'	'McBrien, P.'	NULL	67
101	'deposit'	'McBrien, P.'	5.25	67
103	'current'	'Boyd, M.'	NULL	34
107	'current'	'Poulovassilis, A.'	NULL	56
119	'deposit'	'Poulovassilis, A.'	5.50	56
125	'current'	'Bailey, J.'	NULL	56

movement

mid	no	amount	tdate
1000	100	2300.00	5/1/1999
1001	101	4000.00	5/1/1999
1002	100	-223.45	8/1/1999
1004	107	-100.00	11/1/1999
1005	103	145.50	12/1/1999
1006	100	10.23	15/1/1999
1007	107	345.56	15/1/1999
1008	101	1230.00	15/1/1999
1009	119	5600.00	18/1/1999

Which RA query is well formed?

A

$\sigma_{\text{type}=\text{'current'}}\ \pi_{\text{no}}$ account

B

π_{no} account $-\ \pi_{\text{no,mid}}$ movement

C

$\pi_{\text{no}}\ \sigma_{\text{type}=\text{'current'}}$ account

D

$\pi_{\text{no}}\ \pi_{\text{type}}$ account

Worksheet: Primitive Relational Algebra Operators

branch

sortcode	bname	cash
56	'Wimbledon'	94340.45
34	'Goodge St'	8900.67
67	'Strand'	34005.00

movement

mid	no	amount	tdate
1000	100	2300.00	5/1/1999
1001	101	4000.00	5/1/1999
1002	100	-223.45	8/1/1999
1004	107	-100.00	11/1/1999
1005	103	145.50	12/1/1999
1006	100	10.23	15/1/1999
1007	107	345.56	15/1/1999
1008	101	1230.00	15/1/1999
1009	119	5600.00	18/1/1999

account

no	type	cname	rate	sortcode
100	'current'	'McBrien, P.'	NULL	67
101	'deposit'	'McBrien, P.'	5.25	67
103	'current'	'Boyd, M.'	NULL	34
107	'current'	'Poulovassilis, A.'	NULL	56
119	'deposit'	'Poulovassilis, A.'	5.50	56
125	'current'	'Bailey, J.'	NULL	56

key branch(sortcode)
key branch(bname)
key movement(mid)
key account(no)

movement(no) $\stackrel{fk}{\Rightarrow}$ account(no)

account(sortcode) $\stackrel{fk}{\Rightarrow}$ branch(sortcode)

Derived Relational Algebra: Natural Join ⋈

Natural Join

$$R \bowtie S = \sigma_{R.A_1=S.A_1 \wedge \ldots \wedge R.A_m=S.A_m} R \times S$$

Natural Join

branch ⋈ account ≡ $\pi_{\text{branch.sortcode=account.sortcode}}$ branch × account

branch ⋈ account

sortcode	bname	cash	no	type	cname	rate
34	'Goodge St'	8900.67	103	'current'	'Boyd, M.'	NULL
56	'Wimbledon'	94340.45	107	'current'	'Poulovassilis, A.'	NULL
56	'Wimbledon'	94340.45	119	'deposit'	'Poulovassilis, A.'	5.50
56	'Wimbledon'	94340.45	125	'current'	'Bailey, J.'	NULL
67	'Strand'	34005.00	100	'current'	'McBrien, P.'	NULL
67	'Strand'	34005.00	101	'deposit'	'McBrien, P.'	5.25

Quiz 7: Natural Join

What is the result of $\pi_{no}(account \bowtie movement)$?

A

$\pi_{no}(account \bowtie movement)$
no
100
101
103
107
119
125

B

$\pi_{no}(account \bowtie movement)$
no
100
101
103
107
119

C

$\pi_{no}(account \bowtie movement)$
no
125

D

$\pi_{no}(account \bowtie movement)$
no

Derived Relational Algebra: Semi Join $\ltimes$

Semi Join

$$R \ltimes S = R \bowtie \pi_{Attr(R) \cap Attr(S)}(S)$$

Semi Join

account $\ltimes$ movement = account $\bowtie$ π_{no}(movement)

account $\ltimes$ movement

no	type	cname	rate	sortcode
100	'current'	'McBrien, P.'	NULL	67
101	'deposit'	'McBrien, P.'	5.25	67
103	'current'	'Boyd, M.'	NULL	34
107	'current'	'Poulovassilis, A.'	NULL	56
119	'deposit'	'Poulovassilis, A.'	5.50	56

Derived Relational Algebra: Joins

Natural Join

$$R \bowtie S = \sigma_{R.A_1=S.A_1 \wedge \ldots \wedge R.A_m=S.A_m} R \times S$$

Equi Join

$$R \overset{A=B}{\bowtie} S = \sigma_{R.A=S.B} R \times S$$

Semi Join

$$R \ltimes S = R \bowtie \Pi_{Attr(R)} (S)$$

Theta Join

$$R \overset{\theta}{\bowtie} S = \sigma_\theta R \times S$$

Quiz 8: Understanding join operators

branch

sortcode	bname	cash
56	'Wimbledon'	94340.45
34	'Goodge St'	8900.67
67	'Strand'	34005.00

account

no	type	cname	rate	sortcode
100	'current'	'McBrien, P.'	NULL	67
101	'deposit'	'McBrien, P.'	5.25	67
103	'current'	'Boyd, M.'	NULL	34
107	'current'	'Poulovassilis, A.'	NULL	56
119	'deposit'	'Poulovassilis, A.'	5.50	56
125	'current'	'Bailey, J.'	NULL	56

Which RA query produces the most tuples?

A

$$\text{branch} \overset{\text{branch.sortcode}<\text{account.sortcode}}{\bowtie} \text{account}$$

B

$\text{branch} \bowtie \text{account}$

C

$\text{branch} \ltimes \text{account}$

D

$$\text{branch} \overset{\text{branch.sortcode}=\text{account.sortcode}}{\bowtie} \text{account}$$

Quiz 9: Foreign Keys and Natural Joins (1)

Suppose R and S only share attribute A, and there is a foreign key $S(A) \xrightarrow{fk} R(A)$.

If $|R| = 100$ and $|S| = 1,000$, what is $|R \bowtie S|$?

A

100

B

1,000

C

100,000

D

900

Note that $|R|$ returns the number of tuples in the current extent of R

Quiz 10: Foreign Keys and Natural Joins (2)

Suppose R and S only share attribute A, and there is a foreign key $R(A) \overset{fk}{\Rightarrow} S(A)$.

If $|R| = 100$ and $|S| = 1,000$, what is $|R \bowtie S|$?

A

100

B

1,000

C

100,000

D

900

Derived Relational Algebra: Intersection ∩

Intersection

$$R \cap S = R - (R - S)$$

π_{no}account ∩ π_{no}movement

π_{no}account
no
100
101
103
107
119
125

π_{no}account − π_{no}movement
no
125

π_{no}account ∩ π_{no}movement
no
100
101
103
107
119

Quiz 11: Intersection

email	
name	address
'McBrien, P.'	p.mcbrien@imperial.ac.uk
'Poulovassilis, A.'	ap@dcs.bbk.ac.uk
'Pietzuch, P.'	prp@doc.ic.ac.uk

What is the result of π_{cname} account $\cap$ π_{name} email?

A

cname
'McBrien, P.'
'Boyd, M.'
'Poulovassilis, A.'
'Bailey, J.'
'Pietzuch, P.'

B

cname
'McBrien, P.'
'Boyd, M.'
'Poulovassilis, A.'
'Bailey, J.'

C

cname
'McBrien, P.'
'Poulovassilis, A.'
'Pietzuch, P.'

D

cname
'McBrien, P.'
'Poulovassilis, A.'

Derived Relational Algebra: Division ÷

Division

$$R \div S = \pi_{Atts(R)-Atts(S)} R - \pi_{Atts(R)-Atts(S)}((\pi_{Atts(R)-Atts(S)} R \times S) - R)$$

Division

$$\pi_{\text{cname,type}}\,\text{account} \div \pi_{\text{type}}\,\text{account} = \pi_{\text{cname}}\,\pi_{\text{cname,type}}\,\text{account} - \pi_{\text{cname}}((\pi_{\text{cname}}\,\pi_{\text{cname,type}}\,\text{account} \times \pi_{\text{type}}\,\text{account}) - \pi_{\text{cname,type}}\,\text{account})$$

$$\pi_{\text{cname,type}}\,\text{account} \div \pi_{\text{type}}\,\text{account} = \pi_{\text{cname}}\,\text{account} - \pi_{\text{cname}}((\pi_{\text{cname}}\,\text{account} \times \pi_{\text{type}}\,\text{account}) - \pi_{\text{cname,type}}\,\text{account})$$

$\pi_{\text{cname,type}}$ account	
cname	type
'McBrien, P.'	'current'
'McBrien, P.'	'deposit'
'Boyd, M.'	'current'
'Poulovassilis, A.'	'current'
'Poulovassilis, A.'	'deposit'
'Bailey, J.'	'current'

π_{type} account
type
'current'
'deposit'

$\pi_{\text{cname,type}}$ account ÷ π_{type} account
cname
'McBrien, P.'
'Poulovassilis, A.'

Evaluation of Division

π_{cname} account

cname
'McBrien, P.'
'Boyd, M.'
'Poulovassilis, A.'
'Bailey, J.'

π_{type} account

type
'current'
'deposit'

π_{cname} account $\times$ π_{type} account

cname	type
'McBrien, P.'	'current'
'McBrien, P.'	'deposit'
'Boyd, M.'	'current'
'Boyd, M.'	'deposit'
'Poulovassilis, A.'	'current'
'Poulovassilis, A.'	'deposit'
'Bailey, J.'	'current'
'Bailey, J.'	'deposit'

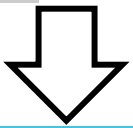
Evaluation of Division

π_{cname} account $\times$ π_{type} account

cname	type
'McBrien, P.'	'current'
'McBrien, P.'	'deposit'
'Boyd, M.'	'current'
'Boyd, M.'	'deposit'
'Poulovassilis, A.'	'current'
'Poulovassilis, A.'	'deposit'
'Bailey, J.'	'current'
'Bailey, J.'	'deposit'

$\pi_{\text{cname,type}}$ account

cname	type
'McBrien, P.'	'current'
'McBrien, P.'	'deposit'
'Boyd, M.'	'current'
'Poulovassilis, A.'	'current'
'Poulovassilis, A.'	'deposit'
'Bailey, J.'	'current'



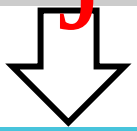
(π_{cname} account $\times$ π_{type} account) $-$ $\pi_{\text{cname,type}}$ account

cname	type
'Boyd, M.'	'deposit'
'Bailey, J.'	'deposit'

Evaluation of Division

$(\pi_{\text{cname}}\ \text{account} \times \pi_{\text{type}}\ \text{account}) - \pi_{\text{cname,type}}\ \text{account}$

cname	type
'Boyd, M.'	'deposit'
'Bailey, J.'	'deposit'



$\pi_{\text{cname}}((\pi_{\text{cname}}\ \text{account} \times \pi_{\text{type}}\ \text{account}) - \pi_{\text{cname,type}}\ \text{account})$

cname
'Boyd, M.'
'Bailey, J.'

Evaluation of Division

π_{cname} account
cname
'McBrien, P.'
'Boyd, M.'
'Poulovassilis, A.'
'Bailey, J.'

$\pi_{\text{cname}}((\pi_{\text{cname}}$ account $\times \pi_{\text{type}}$ account$) - \pi_{\text{cname,type}}$ account$)$
cname
'Boyd, M.'
'Bailey, J.'

π_{cname} account $- \pi_{\text{cname}}((\pi_{\text{cname}}$ account $\times \pi_{\text{type}}$ account$) - \pi_{\text{cname,type}}$ account$)$
cname
'McBrien, P.'
'Poulovassilis, A.

Worksheet: Derived Relational Algebra Operators

branch

sortcode	bname	cash
56	'Wimbledon'	94340.45
34	'Goodge St'	8900.67
67	'Strand'	34005.00

movement

mid	no	amount	tdate
1000	100	2300.00	5/1/1999
1001	101	4000.00	5/1/1999
1002	100	-223.45	8/1/1999
1004	107	-100.00	11/1/1999
1005	103	145.50	12/1/1999
1006	100	10.23	15/1/1999
1007	107	345.56	15/1/1999
1008	101	1230.00	15/1/1999
1009	119	5600.00	18/1/1999

account

no	type	cname	rate	sortcode
100	'current'	'McBrien, P.'	NULL	67
101	'deposit'	'McBrien, P.'	5.25	67
103	'current'	'Boyd, M.'	NULL	34
107	'current'	'Poulovassilis, A.'	NULL	56
119	'deposit'	'Poulovassilis, A.'	5.50	56
125	'current'	'Bailey, J.'	NULL	56

key branch(sortcode)
key branch(bname)
key movement(mid)
key account(no)

movement(no) $\stackrel{fk}{\Rightarrow}$ account(no)

account(sortcode) $\stackrel{fk}{\Rightarrow}$ branch(sortcode)

Equivalences Involving Project

Project and Project

$\pi_{\vec{X}}\, \pi_{\vec{Y}}\, R \equiv \pi_{\vec{X}}\, R$

You can eliminate any inner project (note that to be well formed $\vec{X} \subseteq \vec{Y}$)

Project and Select

$\pi_{\vec{X}}\, \sigma_{P(\vec{Y})}\, R \equiv \sigma_{P(\vec{Y})}\, \pi_{\vec{X}}\, R$

You can move a project of attributes $\vec{X}$ inside a select, provided the select predicate can be answered from those attributes, *i.e.* $\vec{Y} \subseteq \vec{X}$

Project and Product

$\pi_{\vec{X}}(R \times S) \equiv \pi_{\vec{X} \cap Atts(R)}\, R \times \pi_{\vec{X} \cap Atts(S)}\, S$

Project and Union

$\pi_{\vec{X}}(R \cup S) \equiv \pi_{\vec{X}}\, R \cup \pi_{\vec{X}}\, S$

Project and Difference

$\pi_{\vec{X}}(R - S) \supseteq \pi_{\vec{X}}\, R - \pi_{\vec{X}}\, S$

Equivalences Involving Select

Select and Project

$\sigma_{P(\vec{X})}\, \pi_{\vec{X}}\, R \equiv \pi_{\vec{X}}\, \sigma_{P(\vec{X})}\, R$

Select and Select

$\sigma_{P_x(\vec{X})}\, \sigma_{P_y(\vec{Y})}\, R \equiv \sigma_{P_x(\vec{X}) \wedge P_y(\vec{Y})}\, R$

Select and Product

$\sigma_{P(\vec{X})}(R \times S) \equiv \sigma_{P(\vec{X})}\, R \times S \iff \vec{X} \subseteq \mathsf{Atts}(R)$

You can move a select predicate $P(\vec{X})$ onto one of the relations inside a product provided $\vec{X} \subseteq \mathsf{Atts}(R)$.

Select and Union

$\sigma_{P(\vec{X})}(R \cup S) \equiv \sigma_{P(\vec{X})}\, R \cup \sigma_{P(\vec{X})}\, S$

Select and Difference

$\sigma_{P(\vec{X})}(R - S) \equiv \sigma_{P(\vec{X})}\, R - S$

Quiz 12: Equivalent RA Expressions (Unary Operators)

Which RA expression is not equivalent to the other three?

A

$\pi_{\text{no}}\ \sigma_{\text{type}=\text{'current'}}\ \text{account}$

B

$\pi_{\text{no}}\ \sigma_{\text{type}=\text{'current'}}\ \pi_{\text{no,type,cname}}\ \text{account}$

C

$\pi_{\text{no}}\ \sigma_{\text{type}<>\text{'deposit'}}\ \pi_{\text{no,type,cname}}\ \text{account}$

D

$\pi_{\text{no}}\ \sigma_{\text{type}=\text{'current'}}\ \sigma_{\text{type}<>\text{'deposit'}}\ \text{account}$

Quiz 13: Query Evaluation

Which RA means that the $\times$ operator handles fewer tuples?

A

$\sigma_{\text{account.no}=\text{movement.no}}$ ($\sigma_{\text{sortcode}=67}$ account $\times$ $\sigma_{\text{amount}<0}$ movement)

B

$\sigma_{\text{account.no}=\text{movement.no} \wedge \text{sortcode}=67}$ (account $\times$ $\sigma_{\text{amount}<0}$ movement)

C

$\sigma_{\text{account.no}=\text{movement.no} \wedge \text{amount}<0}$ ($\sigma_{\text{sortcode}=67}$ account $\times$ movement)

D

$\sigma_{\text{account.no}=\text{movement.no} \wedge \text{sortcode}=67 \wedge \text{amount}<0}$ (account $\times$ movement)

Equivalences Involving Binary Operators

Product and Union

$R \times (S \cup T) \equiv (R \times S) \cup (R \times T)$

Product and Difference

$R \times (S - T) \equiv (R \times S) - (R \times T)$

Union and Product

$R \cup (S \times T)$ unable to move $\cup$ inside $\times$

Union and Difference

$R \cup (S - T)$ unable to move $\cup$ inside $-$

Difference and Product

$R - (S \times T)$ unable to move $-$ inside $\times$

Difference and Union

$R - (S \cup T) \equiv (R - S) - T$

Quiz 14: Equivalent RA Expressions (Binary Operators)

Which equivalence does not hold?

A

$(R \times S) \times T \equiv R \times (S \times T)$

B

$(R - S) - T \equiv R - (S - T)$

C

$(R \cup S) \cup T \equiv R \cup (S \cup T)$

D

$(R \cap S) \cap T \equiv R \cap (S \cap T)$

Worksheet: Equivalences Between RA Expressions

1. $\pi_{\text{no,type}}\ \sigma_{\text{sortcode}=56}\ \pi_{\text{no,type,sortcode}}\ \sigma_{\text{type}=\text{'deposit'}}\ \text{account}$
2. $\sigma_{\text{account.no}=\text{movement.no}}(\pi_{\text{no,cname}}\ \text{account} \times \pi_{\text{mid,no}}\ \sigma_{\text{amount}>1000}\ \text{movement})$
3. $\sigma_{\text{account.no}=\text{movement.no}}(\pi_{\text{no,cname,rate}}\ \text{account} \times (\sigma_{\text{amount}>1000}\ \pi_{\text{mid,no}}\ \text{movement} \cup \sigma_{\text{amount}<100}\ \pi_{\text{mid,no}}\ \text{movement}))$
4. $\pi_{\text{no,cname,tdate}}\ \sigma_{\text{amount}<0 \wedge \text{account.no}=\text{movement.no}}\ \text{account} \times \text{movement}$

Quiz 15: Monotonic and non-monotonic operators

A monotonic operator has the property that an additional tuple put into any input relation which only cause additional tuples to be generated in the output relation.

A non-monotonic operator has the property that an additional tuple put into an input relation may remove tuples from the output relation

Which RA operator is non-monotonic?

A

π R

B

R × S

C

R ∪ S

D

R − S

Incremental Query Evaluation

Suppose we add rows Δ_R to extent of relation R so it becomes R'

If we represent Δ_R as a relation (with the same attributes as R) then

$R' = R \cup \Delta_R$

$\pi_{\vec{X}}\, R' \equiv \pi_{\vec{X}}\, R \cup \pi_{\vec{X}}\, \Delta_R$

$\sigma_{P(\vec{X})}\, R' \equiv \sigma_{P(\vec{X})}\, R \cup \sigma_{P(\vec{X})}\, \Delta_R$

$R' \times S \equiv (R \times S) \cup (\Delta_R \times S)$

$R' \cup S \equiv (R \cup S) \cup \Delta_R$

$R' - S \equiv (R - S) \cup (\Delta_R - S)$

$S - R' \equiv (S - R) - \Delta_R$

Example: Query result after update to account (1)

1. Suppose that we had already evaluated query Q

$\pi_{\text{bname,no}}\sigma_{\text{branch.sortcode=account.sortcode}\wedge\text{account.type='current'}}(\text{branch} \times \text{account})$

bname	no
'Goodge St'	103
'Wimbledon'	107
'Wimbledon'	125
'Strand'	100

2. If $\Delta_{account}$ is added to account to get account′:

$\pi_{\text{bname,no}}\sigma_{\text{branch.sortcode=account.sortcode}\wedge\text{account.type='current'}}(\text{branch} \times \text{account}')$

$\pi_{\text{bname,no}}\sigma_{\text{branch.sortcode=account.sortcode}\wedge\text{account.type='current'}}((\text{branch} \times \text{account}) \cup (\text{branch} \times \Delta_{\text{account}}))$

$\pi_{\text{bname,no}}\sigma_{\text{branch.sortcode=account.sortcode}\wedge\text{account.type='current'}}(\text{branch} \times \text{account}) \cup \pi_{\text{bname,no}}\sigma_{\text{branch.sortcode=account.sortcode}\wedge\text{account.type='current'}}(\text{branch} \times \Delta_{\text{account}})$

3. Thus if $\Delta_{account}$ is added to account, we only need evaluate

$\pi_{\text{bname,no}}\sigma_{\text{branch.sortcode=account.sortcode}\wedge\text{account.type='current'}}(\text{branch} \times \Delta_{\text{account}})$

Example: Query result after update to account (2)

4. Suppose we have

$\Delta_{account}$

126	'business'	'McBrien, P.'	1.00	67
127	'current'	'Pietzuch, P.'	NULL	34

Then

$\pi_{bname,no}\,\sigma_{branch.sortcode=account.sortcode \wedge account.type='current'}(branch \times \Delta_{account})$

bname	no
'Goodge St'	127

5. Thus since $Q' = Q \cup \Delta_Q$

$\pi_{bname,no}\,\sigma_{branch.sortcode=account.sortcode \wedge account.type='current'}(branch \times account')$

bname	no
'Goodge St'	103
'Wimbledon'	107
'Wimbledon'	125
'Strand'	100
'Goodge St'	127