COMP1140: Database and Information Management

Lecture Note – Week 6

Dr Suhuai Luo

School of EEC

University of Newcastle

Notice

- Assignment 1 marks will be available this Friday. Details of getting back your hardcopy (with comments) will be given in an email.
- Assignment 2 marking scheme is on BB
- In your assignment 2, make sure consider the feedbacks from marker and work on a working EER.

Last lecture

- Normalization
- Q?
  - what’s the difference between 3NF and BCNF?

This lecture

- Relational algebra
  - Selection
  - Projection
  - Cartesian Product
  - Union
  - Set Difference
  - Intersection
  - Division
  - Joins (Theta, Equi, Natural, Outer)
  - Aggregate and Grouping
- About A2
- Ref: chapter 5

Relational Algebra

- A formal query language
- Relational algebra is a set of *relational algebra operators*
- A relational algebra operator takes one (unary operator) or two (binary operator) relations as input and produces a relation as output.

Relational Algebra (contd.)

- Five basic operations in relational algebra:
  - Selection (σ),
  - Projection (Π),
  - Cartesian product ($\times$),
  - Union ($\cup$), and
  - Set Difference (-).

- These perform most of the data retrieval operations needed.

- Also have Join, Intersection, and Division operations, which can be expressed in terms of 5 basic operations.

Sample Database Instance

Branch

branchNo	street	city	postcode
B005	22 Deer Rd	London	SW1 4EH
B007	16 Argyll St	Aberdeen	AB2 3SU
B003	163 Main St	Glasgow	G11 9QX
B004	32 Manse Rd	Bristol	BS99 1NZ
B002	56 Clover Dr	London	NW10 6EU

Staff

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000	B003
SL41	Julie	Lee	Assistant	F	13-Jun-65	9000	B005

PropertyForRent

propertyNo	street	city	postcode	type	rooms	rent	ownerNo	staffNo	branchNo
PA14	16 Holhead	Aberdeen	AB7 5SU	House	6	650	CO46	SA9	B007
PL94	6 Argyll St	London	NW2	Flat	4	400	CO87	SL41	B005
PG4	6 Lawrence St	Glasgow	G11 9QX	Flat	3	350	CO40		B003
PG36	2 Manor Rd	Glasgow	G32 4QX	Flat	3	375	CO93	SG37	B003
PG21	18 Dale Rd	Glasgow	G12	House	5	600	CO87	SG37	B003
PG16	5 Novar Dr	Glasgow	G12 9AX	Flat	4	450	CO93	SG14	B003

Sample Database Instance

Client

clientNo	fName	lName	telNo	prefType	maxRent
CR76	John	Kay	0207-774-5632	Flat	425
CR56	Aline	Stewart	0141-848-1825	Flat	350
CR74	Mike	Ritchie	01475-392178	House	750
CR62	Mary	Tregear	01224-196720	Flat	600

PrivateOwner

ownerNo	fName	lName	address	telNo
CO46	Joe	Keogh	2 Fergus Dr, Aberdeen AB2 7SX	01224-861212
CO87	Carol	Farrel	6 Achray St, Glasgow G32 9DX	0141-357-7419
CO40	Tina	Murphy	63 Well St, Glasgow G42	0141-943-1728
CO93	Tony	Shaw	12 Park Pl, Glasgow G4 0QR	0141-225-7025

Viewing

clientNo	propertyNo	viewDate	comment
CR56	PA14	24-May-04	too small
CR76	PG4	20-Apr-04	too remote
CR56	PG4	26-May-04	
CR62	PA14	14-May-04	no dining room
CR56	PG36	28-Apr-04	

Registration

clientNo	branchNo	staffNo	dateJoined
CR76	B005	SL41	2-Jan-04
CR56	B003	SG37	11-Apr-03
CR74	B003	SG37	16-Nov-02
CR62	B007	SA9	7-Mar-03

Selection

- $\sigma_{predicate}$ (R)

  - Defines a relation that has the schema of R, and

  - Tuples of R that satisfy the specified condition (*predicate*).

Example - Selection

- List all staff with a salary greater than £10,000.

$$\sigma_{\text{salary} > 10000} (\text{Staff})$$

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000	B003
SG14	David	Ford	Supervisor	M	24- Mar-58	18000	B003
SG5	Susan	Brand	Manager	F	3-Jun-40	24000	B003

Projection

- $\Pi_{col1, \ldots, coln}(R)$
  - Defines a relation that has the columns specified,
  - Extracts the values of columns from tuples of R, and
  - Eliminates any duplicate tuples in the resultant relation

Example - Projection

- Produce a list of salaries for all staff, showing only staffNo, fName, lName, and salary details.

$$\Pi_{\text{staffNo, fName, lName, salary}}(\text{Staff})$$

staffNo	fName	lName	salary
SL21	John	White	30000
SG37	Ann	Beech	12000
SG14	David	Ford	18000
SA9	Mary	Howe	9000
SG5	Susan	Brand	24000
SL41	Julie	Lee	9000

Relational algebra expressions and Rename operator

- You can combine multiple relational algebra operators to create a complex relational algebra expression.

- $\rho_S(E)$ or $\rho_{S(a1,\ a2,\ \ldots,\ an)}(E)$
  - The rename operator provides a new name S for the expression E, and optionally names the attributes a1, a2, ..., an.

Union-compatibility

- Two relations $R(A_1, A_2, \cdots, A_n)$ & $S(B_1, B_2, \cdots, B_n)$ are said to be **union–compatible** if the following holds:
  - degree of R & S are equal, and
  - if $dom(A_i) = dom(B_i)$ for $1 \leq i \leq n$.
    - (That is, corresponding fields of R & S, taken from left to right, have the same *domains*).

Union, Intersection and Set-Difference

- Assuming that R and S are union-compatible, we can define three relational algebra operators:
  - Union ($R \cup S$)
  - Intersection ($R \cap S$), and
  - Set-Difference ($R - S$)

Union

- R ∪ S
  - Union of two relations R and S defines a relation that contains all the tuples of R, or S, or both R and S
  - Duplicate tuples are eliminated.

Example - Union

- List all cities where there is *either* a branch office, *or* a property for rent.

$$\Pi_{city}(\text{Branch}) \cup \Pi_{city}(\text{PropertyForRent})$$

city
London
Aberdeen
Glasgow
Bristol

Set Difference

- R – S
  - Defines a relation consisting of the tuples that are in relation R, but not in S.

- For example,
  - List all cities where there is a branch office but no properties for rent.

Π_{city}(Branch) – Π_{city}(PropertyForRent)

city
Bristol

Intersection

- $R \cap S$
  - Defines a relation consisting of the set of all tuples that are in both R and S.

- Expressed using basic operations:

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

Example - Intersection

- List all cities where there is both a branch office and at least one property for rent.

$$\Pi_{city}(\text{Branch}) \cap \Pi_{city}(\text{PropertyForRent})$$

city
Aberdeen
London
Glasgow

Cartesian product

- R X S
  - Defines a relation that is the concatenation of every tuple of relation R with every tuple of relation S.
- Formally, the Cartesian product, denoted by $R \times S$, results in a relation V such that
  - V has the following relation schema $V(R_1, \ldots, R_n, S_1, \ldots, S_m)$ and
  - set of tuples: $\{r_1, \ldots, r_n, s_1, \ldots s_m\}$

where R has relation schema $(R_1, \ldots, R_n)$
S has relation schema $(S_1, \ldots, S_m)$
$<r_1, \ldots, r_n>$ is a tuple in R and
$<s_1, \ldots, s_m>$ is a tuple in S

Example - Cartesian product

List the names and comments of all clients who have viewed a property for rent.

Step 1: to combine the 2 tables to get all possible combinations:

$(\Pi_{\text{clientNo, fName,lName}}(\text{Client}))$ X $(\Pi_{\text{clientNo, propertyNo, comment}}(\text{Viewing}))$

client.clientNo	fName	lName	Viewing.clientNo	propertyNo	comment
CR76	John	Kay	CR56	PA14	too small
CR76	John	Kay	CR76	PG4	too remote
CR76	John	Kay	CR56	PG4	
CR76	John	Kay	CR62	PA14	no dining room
CR76	John	Kay	CR56	PG36	
CR56	Aline	Stewart	CR56	PA14	too small
CR56	Aline	Stewart	CR76	PG4	too remote
CR56	Aline	Stewart	CR56	PG4	
CR56	Aline	Stewart	CR62	PA14	no dining room
CR56	Aline	Stewart	CR56	PG36	
CR74	Mike	Ritchie	CR56	PA14	too small
CR74	Mike	Ritchie	CR76	PG4	too remote
CR74	Mike	Ritchie	CR56	PG4	
CR74	Mike	Ritchie	CR62	PA14	no dining room
CR74	Mike	Ritchie	CR56	PG36	
CR62	Mary	Tregear	CR56	PA14	too small
CR62	Mary	Tregear	CR76	PG4	too remote
CR62	Mary	Tregear	CR56	PG4	
CR62	Mary	Tregear	CR62	PA14	no dining room
CR62	Mary	Tregear	CR56	PG36	

Example - Cartesian product (contd.)

Step 2: Use selection operation to extract those tuples where Client.clientNo = Viewing.clientNo.

$$\sigma_{\text{Client.clientNo} = \text{Viewing.clientNo}}$$
$$((\Pi_{\text{clientNo, fName, lName}}(\text{Client})) \text{ X } (\Pi_{\text{clientNo, propertyNo, comment}}(\text{Viewing})))$$

client.clientNo	fName	lName	Viewing.clientNo	propertyNo	comment
CR76	John	Kay	CR76	PG4	too remote
CR56	Aline	Stewart	CR56	PA14	too small
CR56	Aline	Stewart	CR56	PG4	
CR56	Aline	Stewart	CR56	PG36	
CR62	Mary	Tregear	CR62	PA14	no dining room

Join Operations

- Join is a derivative of Cartesian product.
- Equivalent to performing a Selection (using join predicate as selection formula) over Cartesian product of the two operand relations.

Join Operations

- Various forms of join operation
  - Theta join
  - Equijoin (a particular type of Theta join)
  - Natural join
  - Outer joins

Theta join (θ-join)

- $R \bowtie_F S$
  - Defines a relation that contains tuples satisfying the predicate F from the Cartesian product of R and S.
  - The predicate F is of the form $R.a_i \ \theta \ S.b_i$ where θ may be one of the comparison operators ($<, \leq, >, \geq, =, \neq$).
  - Can rewrite theta join using cartesian product and selection operations
    - $R \bowtie_F S \equiv \sigma_F (R \times S)$

- If predicate F contains only equality (=), the term *Equijoin* is used.

Example - Equijoin

- List the names and comments of all clients who have viewed a property for rent.

$(\Pi_{\text{clientNo, fName, lName}}(\text{Client})) \bowtie_{\text{Client.clientNo = Viewing.clientNo}} (\Pi_{\text{clientNo, propertyNo, comment}}(\text{Viewing}))$

client.clientNo	fName	lName	Viewing.clientNo	propertyNo	comment
CR76	John	Kay	CR76	PG4	too remote
CR56	Aline	Stewart	CR56	PA14	too small
CR56	Aline	Stewart	CR56	PG4	
CR56	Aline	Stewart	CR56	PG36	
CR62	Mary	Tregear	CR62	PA14	no dining room

Natural join

- R $\bowtie$ S
  - An Equijoin of the two relations R and S over all common attributes *x*.
  - The joining attributes (i.e. common attributes) have the same name
  - One occurrence of each common attribute is eliminated from the result.

Example - Natural join

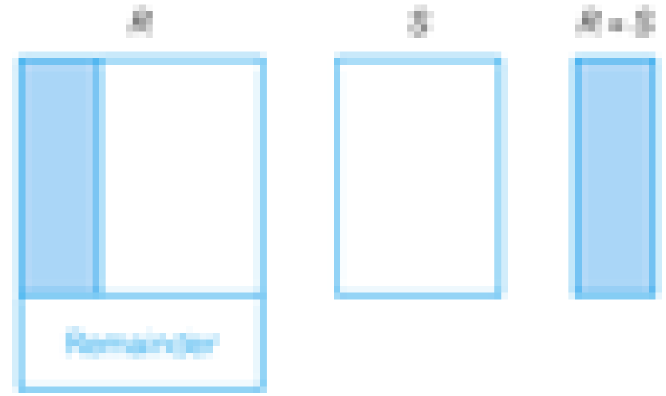
- List the names and comments of all clients who have viewed a property for rent.

$$(\Pi_{\text{clientNo, fName, lName}}(\text{Client})) \bowtie (\Pi_{\text{clientNo, propertyNo, comment}}(\text{Viewing}))$$

NOTE: Joined on attribute clientNo and only a single occurrence of it in the result

clientNo	fName	lName	propertyNo	comment
CR76	John	Kay	PG4	too remote
CR56	Aline	Stewart	PA14	too small
CR56	Aline	Stewart	PG4	
CR56	Aline	Stewart	PG36	
CR62	Mary	Tregear	PA14	no dining room

Division



- $R \div S$
  - Defines a relation (over the attributes C) that consists of set of tuples (from R) that match combination of *every* tuple in S. Where C is the set of attributes that are in R but not in S.

- Expressed using basic operations:

  $T_1 \leftarrow \Pi_C(R)$

  $T_2 \leftarrow \Pi_C((S \times T_1) - R)$

  $T \leftarrow T_1 - T_2$

- Arrow ($\leftarrow$) assigns resultant of expression in right-hand side to a relation on the left-hand side

Example - Division

- Identify all clients who have viewed all properties with three rooms.

$$(\Pi_{\text{clientNo, propertyNo}}(\text{Viewing})) \div (\Pi_{\text{propertyNo}}(\sigma_{\text{rooms = 3}}(\text{PropertyForRent})))$$

$\Pi_{\text{clientNo,propertyNo}}(\text{Viewing})$

clientNo	propertyNo
CR56	PA14
CR76	PG4
CR56	PG4
CR62	PA14
CR56	PG36

$\Pi_{\text{propertyNo}}(\sigma_{\text{rooms}=3}(\text{PropertyForRent}))$

propertyNo
PG4
PG36

RESULT

clientNo
CR56

Limitations of RA operators

- There are some common requests that cannot be expressed using **relational algebra** discussed so far**.**
  - For example: How many staff in a branch?

- Therefore, important extensions have been proposed:
  - Outer joins
  - Aggregation and Grouping

Outer joins

- ALSO display rows in the result that do not have matching values in the join column.

- Three types of outer joins
  - **Left Outer Join (⟕):** result also includes non-matching tuples on left-hand side relation
  - **Right Outer Join (⟖):** result also includes non-matching tuples on right-hand side relation
  - **Full Outer Join (⟗):** result also includes non-matching tuples on both left-hand and right-hand side relations

Example - Left Outer join

- Produce a status report on property viewings.

$$\Pi_{\text{propertyNo, street, city}}(\text{PropertyForRent}) ⟕ \text{Viewing}$$

propertyNo	street	city	clientNo	viewDate	comment
PA14	16 Holhead	Aberdeen	CR56	24-May-01	too small
PA14	16 Holhead	Aberdeen	CR62	14-May-01	no dining room
PL94	6 Argyll St	London	null	null	null
PG4	6 Lawrence St	Glasgow	CR76	20-Apr-01	too remote
PG4	6 Lawrence St	Glasgow	CR56	26-May-01	
PG36	2 Manor Rd	Glasgow	CR56	28-Apr-01	
PG21	18 Dale Rd	Glasgow	null	null	null
PG16	5 Novar Dr	Glasgow	null	null	null

Aggregate Operations

- $\Im_{AL}(R)$
  - Applies aggregate function list, AL, to R to define a relation over the aggregate list.
  - AL contains one or more (<aggregate_function>, <attribute>) pairs .
- Main aggregate functions are:
  - COUNT: return no. of values in the associated attribute
  - SUM: return sum of values
  - AVG: return average of values
  - MIN: return minimum
  - MAX: return maximum

Example – Aggregate Operations

- How many properties cost more than £350 per month to rent?

$\rho_R(\text{myCount})\ \Im_{\text{COUNT propertyNo}}\ (\sigma_{\text{rent} > 350}(\text{PropertyForRent}))$

PropertyForRent

propertyNo	street	city	postcode	type	rooms	rent	ownerNo	staffNo	branchNo
PA14	16 Holhead	Aberdeen	AB7 5SU	House	6	650	CO46	SA9	B007
PL94	6 Argyll St	London	NW2	Flat	4	400	CO87	SL41	B005
PG4	6 Lawrence St	Glasgow	G11 9QX	Flat	3	350	CO40		B003
PG36	2 Manor Rd	Glasgow	G32 4QX	Flat	3	375	CO93	SG37	B003
PG21	18 Dale Rd	Glasgow	G12	House	5	600	CO87	SG37	B003
PG16	5 Novar Dr	Glasgow	G12 9AX	Flat	4	450	CO93	SG14	B003

myCount
5

(a)

Grouping Operation

- $_{GA}\Im_{AL}(R)$
  - Groups tuples of R by grouping attributes, GA, and then applies aggregate function list, AL, to each grouping to define a new relation.
  - AL contains one or more (<aggregate_function>, <attribute>) pairs.
  - Resulting relation contains the grouping attributes, GA, along with results of each of the aggregate functions.

Example – Grouping Operation

- Find the number of staff working in each branch and the sum of their salaries.

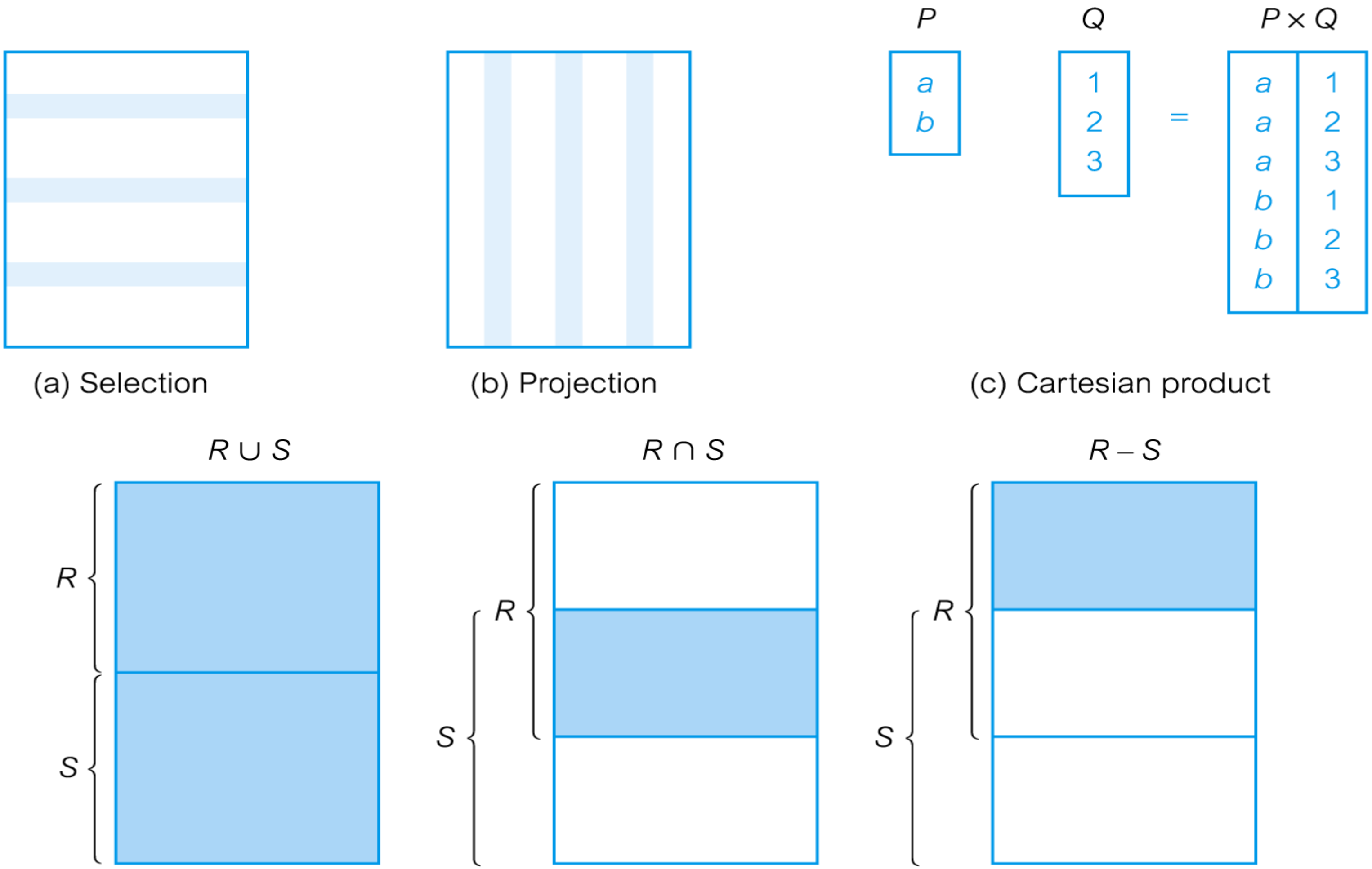
$$\rho_R(\text{branchNo, myCount, mySum})\ {}_{\text{branchNo}}\mathfrak{I}_{\text{COUNT staffNo, SUM salary}}\ (\text{Staff})$$

branchNo	myCount	mySum
B003	3	54000
B005	2	39000
B007	1	9000

Summary

- Selection
- Projection
- Cartesian Product
- Union, Intersection, Set Difference
- Division
- Joins (Theta, Equi, Natural, Outer)
- Aggregate and Grouping

Summary (contd.)



(a) Selection (b) Projection (c) Cartesian product

(d) Union (e) Intersection (f) Set difference

Summary (contd.)

T

A	B
a	1
b	2

U

B	C
1	x
1	y
3	z

$T \bowtie U$

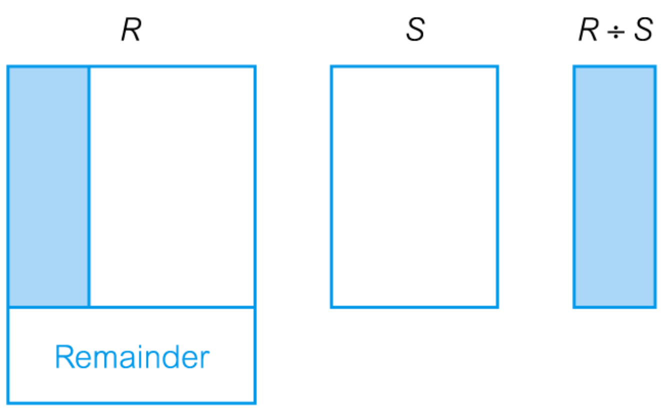
A	B	C
a	1	x
a	1	y

(g) Natural join

$T ⟕_C U$

A	B	C
a	1	x
a	1	y
b	2	

(h) Left Outer join



(i) Divis on (shaded area)

V

A	B
a	1
a	2
b	1
b	2
c	1

W

B
1
2

$V \div W$

A
a
b

(j) Example of division

Lab This Week

- Query in relational algebra
- Query in SQL

On A2

- About A2 –
  - specification
  - A2 Marking Rules