Relational Algebra, Part III, and Other Operators

Hugh Darwen

hugh@dcs.warwick.ac.uk
www.dcs.warwick.ac.uk/~hugh

CS252.HACD: Fundamentals of Relational Databases
Section 6: Relational Algebra, Part III, and Other Operators

1

The Running Example …

IS_CALLED

StudentId	Name
S1	Anne
S2	Boris
S3	Cindy
S4	Devinder
S5	Boris

StudentId is called Name

IS_ENROLLED_ON

StudentId	CourseId
S1	C1
S1	C2
S2	C1
S3	C3
S4	C1

StudentId is enrolled on CourseId

2

… and these

COURSE

CourseId	Title
C1	Database
C2	HCI
C3	Op Systems
C4	Programming

CourseId is entitled Title

EXAM_MARK

StudentId	CourseId	Mark
S1	C1	85
S1	C2	49
S2	C1	49
S3	C3	66
S4	C1	93

StudentId scored Mark in the exam for course CourseId

3

Some Useful Shorthands

Relational operators:

- semijoin
- composition
- GROUP/UNGROUP
- UPDATE (not an update operator!)

Other operators:

- relation comparisons
- tuple extraction (from a relation)
- attribute value extraction (from a tuple)

4

Semijoin

StudentId is called Name **AND** is enrolled on some course.

StudentId	Name
S1	Anne
S2	Boris
S3	Cindy
S4	Devinder

IS_CALLED **MATCHING** IS_ENROLLED_ON

5

Definition of MATCHING

r1 **MATCHING** *r2* ≡ *r1* **JOIN** (*r2* { *common-attrs* })
where *common-attrs* is the attributes in common to *r1* and *r2*.

So, let *s* = *r1* **MATCHING** *r2*. Then:

The heading of *s* is the heading of *r1*.

The body of *s* consists of each tuple of *r1* that matches at least one tuple of *r2* on their common attributes.

It follows that in the case where there are no common attributes, *s* is empty if *r2* is empty, and is otherwise equal to *r1*.

6

Composition

StudentId is enrolled on a course entitled Title.

StudentId	Title
S1	Database
S1	HCI
S2	Database
S3	Op Systems
S4	Database

IS_ENROLLED_ON **COMPOSE** COURSE

7

Definition of COMPOSE

r1 **COMPOSE** *r2* ≡ (*r1* **JOIN** *r2*) { ALL BUT *common-attrs* }
where *common-attrs* is the attributes in common to *r1* and *r2*.

Exercise (see Notes):
Is COMPOSE commutative?
I.e., is *r1* COMPOSE *r2* equivalent to *r2* COMPOSE *r1*?
Is COMPOSE associative?
I.e., are (*r1* COMPOSE *r2*) COMPOSE *r3* and
r1 COMPOSE (*r2* COMPOSE *r3*) equivalent?

8

Read-only Counterparts of Update Operators

E.g. UPDATE (IS_CALLED WHERE StudentId = ‘S1’)
(Name := ‘Ann’)
Note lack of semicolon – this is an expression, not an imperative.
And the “target” is a relation, not a relvar

INSERT *rv1 r2*? Counterpart is *r1* UNION *r2*.
DELETE *rv* WHERE *c*? Counterpart is *r* WHERE NOT(*c*).

9

GROUP/UNGROUP

A

StudentId	Name
S1	Anne
S2	Boris
S3	Cindy
S4	Devinder
S5	Boris

B

Name	StudentIds
Anne	StudentId: S1
Boris	StudentId: S2, S5
Cindy	StudentId: S3
Devinder	StudentId: S4

UNGROUP

GROUP

10

From A to B and Back Again

B = A GROUP ({ StudentId } AS StudentIds)

A = B UNGROUP (StudentIds)

11

Other Operators

Operators on relations that do not yield relations:
- aggregate operators (already seen)
- relation comparison
- tuple extraction

Operators on tuples that do not yield tuples:
- relation “selection” (already seen)
- attribute value extraction

Operators that yield tuples:
- tuple “selection” (already seen)
- tuple counterparts of relational operators

12

Relation Comparison Case Study

B

Name	N_StudentIds
Anne	StudentId S1
Boris	StudentId S2 S5
Cindy	StudentId S3
Devinder	StudentId S4

C

CourseId	C_StudentIds
C1	StudentId S1 S2 S4
C2	StudentId S1
C3	StudentId S3

13

Relation Comparison (1)

Every student enrolled on <u>CourseId</u> is called <u>Name</u>
(and at least one student is enrolled on <u>CourseId</u>).

CourseId	Name
C2	Anne
C3	Cindy

((B JOIN C) WHERE N_StudentIds ⊇ C_StudentIds)
{ Name, CourseId }

14

Relation Comparison (2)

Every student called <u>Name</u> is enrolled on <u>CourseId</u>
(and at least one student is called <u>Name</u>).

Name	CourseId
Anne	C1
Anne	C2
Cindy	C3
Devinder	C1

((B JOIN C) WHERE N_StudentIds ⊆ C_StudentIds)
{ Name, CourseId }

15

Tuple Extraction

Given a relation *r* of cardinality 1 (no more, no less):

TUPLE FROM *r*

yields the single tuple contained in the body of *r*.

E.g.: TUPLE FROM (IS_CALLED
WHERE StudentId = 'S1')

gives TUPLE { StudentId 'S1', Name 'Anne' }

16

Attribute Value Extraction

Given a tuple *t* with an attribute *a*:

a FROM *t*

yields the value of the *a* attribute in *t*.

E.g.: Name FROM TUPLE FROM (IS_CALLED
WHERE StudentId = 'S1')
gives 'Anne'

17

A Relational View of Arithmetic

Recall the imagined relation PLUS:

a	b	c
1	2	3
2	3	5
2	1	3

Now, to compute, e.g., 2+3 …

18

Adding 2 and 3

PLUS

a	b	c
1	2	3
2	3	5
2	1	3

c FROM TUPLE FROM
(PLUS COMPOSE
{ RELATION { TUPLE { a 2, b 3 } })

(okay, 2+3 is perhaps a little easier!)

19

Tuple Counterparts of Relational Operators

Let *t1*, *t2,* ... be tuples. Then we have:

- tuple rename: *t1* RENAME (*a* AS *b*, …)
- tuple projection: *t1* { [ALL BUT] *attribute-name-list* }
- tuple extension: EXTEND *t1* ADD (*exp* AS *attribute-name*)
- tuple "update": UPDATE *t1* (*attribute-name* := *exp*)
- tuple join: *t1* JOIN *t2,* and JOIN{*t1*,*t2*,...}
- tuple compose: *t1* COMPOSE *t2*

20

EXERCISES
(see Notes)

21