Relational Algebra Examples

Consider the 2 relations below:

A

Num	Square
1	1
2	4

B

Num	Square
1	1
3	9

1. Selection

a) $\sigma_{Num>1}(A)$ will give us:

Num	Square
2	4

b) $\sigma_{num=3}(B)$ will give us:

Num	Square
3	9

- Denoted by $\sigma_p(x)$ where
  - σ is the selection symbol.
  - P is the propositional logic.
  - X is the name of the relation.

- Selection gets all rows from the given relation that satisfies the propositional logic.

2. Projection

- Denoted by $\pi_{col1, col2, \ldots, coln}(x)$ where
  - π is the projection symbol.
  - col1, col2, ..., coln are the names of columns.
  - x is the name of the relation.

- Gets all the columns listed from the given relation.

a) $\pi_{Num}(A)$ will give us:

Num
1
2

3. Natural Join

- Denoted by $r \bowtie s$ where r and s are relations.
- It combines 2 relations into 1. It can only be used if there is at least 1 column in both relations that have the same name and same domain. It only gets rows that are in both relations.
- Also called inner join.

a) $A \bowtie B$ will give us

Num	A. square	B. square
1	1	1

4. Left Join
   - Denoted by $R \bowtie_L S$ where R and S are relations.
   - keeps all tuples in the left relation and tries to find matching tuples in the right relation. If there is no matching tuple, a null value is used.

   a) $A \bowtie_L B$ will give us:

Num	A. square	B. square
1	1	1
2	4	NULL

5. Right Join
   - Denoted by $R \bowtie_R S$ where R and S are relations.
   - keeps all tuples in the right relation and tries to find matching tuples in the left relation. If there is no matching tuple, a null value is used.

   a) $A \bowtie_R B$ will give us

Num	A. square	B. square
1	1	1
3	NULL	3

6. Full Join:
   - Denoted by $R \bowtie_o S$ where R and S are relations.
   - All tuples from both relations are included in the result.

   a) $A \bowtie_o B$ will give us:

Num	A. square	B. square
1	1	1
2	4	NULL
3	NULL	9

7. Cartesian Join:
   - Denoted by $R \times S$ where R and S are relations.
   - Produces all pairs of rows of the 2 relations that are possible.
   - If relation R has X rows and relation S has Y rows, then $R \times S$ will have $X \cdot Y$ rows.
   - Also called cross join.

   a) $A \times B$ will give us:

A. n	A. s	B. n	B. s
1	1	1	1
1	1	3	9
2	4	1	1
2	4	3	9

8. Theta Join:

- Denoted by $R \bowtie_c S$ where R and S are relations and c is a propositional logic or condition.
- with $R \bowtie_c S$, you first find $R \times S$ and then do σ_c on that.

a) $A \bowtie_{(B.s > A.s)} B$

We got $A \times B$ in the last example. Filtering out the rows where B. square > A. square, we get

A.n	A.s	B.n	B.s
1	1	3	9
2	4	3	9

9. Union:

- Denoted by $R \cup S$ where R and S are relations.
- It will give a relation with tuples which are either in R or S. It will also eliminate all duplicate tuples.
- For a union operation to be valid, the following conditions must hold
  - The 2 relations must have the same number of columns.
  - The domain of the columns must be compatible.

a) A ∪ B will give us:

Num	Square
1	1
2	4
3	9

10. Difference:
- Denoted by R−S where R and S are relations.
- It returns a relation consisting of all tuples in R and not in S.

a) A − B will give us:

Num	Square
2	4

b) B − A will give us:

Num	Square
3	9

11. Intersection:
- Denoted by R∩S where R and S are relations.
- It returns a table consisting of all tuples in both R and S.
- R ∩ S is equalivent to R−(R−S)

a) A ∩ B will give us:

Num	Square
1	1

12. Finding "every" using relational algebra:

Consider the 2 relational schemas below:

Student(id, name)
Marks (id, class, mark)

Find the id of the students taking every class.

Soln:

1. $R_1 = \Pi_{id}(S) \times \Pi_{class}(M)$
2. $R_2 = R_1 - \Pi_{id,\, class}(M)$
3. $R_3 = \Pi_{id}(R_1) - \Pi_{id}(R_2)$

R_1 is a relation with every possible permutation of id and classes. Therefore, it is a table of every student taking every class.

When you do $R_1 - \Pi_{id,\, class}(M)$, you are removing all instances of a student actually taking a class from R_1. Therefore, R_2 is a table of the students not taking class(es). If a student is taking every class, their id wouldn't be in R_2.

Since $\pi_{id}(R_1)$ contains the id of every student and $\pi_{id}(R_2)$ contains the id of the students who are not taking some classes, $\pi_{id}(R_1) - \pi_{id}(R_2)$ will give us a table of the ids of the students taking every class.

13. Finding the biggest or highest:

Using the schemas in 12, find the id of the students who has the highest mark.

Soln:

1. $R_1 = \pi_{id, mark}(M)$
2. $R_2 = \rho_{R2}(R1)$
3. $R_3 = \rho_{R3}(R1)$
4. $R_4 = R_2 \bowtie_{(R2.mark < R3.mark)} R3$
5. $R_5 = \pi_{id}(R_1) - \pi_{R2.id}(R_4)$

R_1 is just a relation with only the id and mark columns from Marks. R_2 and R_3 are just renamed instances of R_1. R_4 is a relation of all possible permutations between R_2 and R_3 where R2. mark < R3. mark. That means R2.id in R4 does not contain the id of the students with the highest mark.

However, R2.id in R4 contains the id of all other students. Therefore, $\pi_{id}(R_1) - \pi_{R2.id}(R_4)$ gets back a table with the id of the students with the highest mark.

14 Finding the second biggest or second highest:

Using the schemas in 12, find the id of the students who have the second highest mark.

Soln:

$R_1 = \pi_{id,\ mark}(M)$

$R_2 = \rho_{R2}(R_1)$

$R_3 = \rho_{R3}(R_1)$

$R_4 = R_2 \bowtie_{(R2.mark < R3.mark)} R_3$

$R_5 = \pi_{R2.id,\ R2.mark}(R_4)$

$R_6 = \rho_{R6}(R_5)$

$R_7 = R_5 \bowtie_{(R5.mark < R6.mark)} R_6$

$R_8 = \pi_{id}(R_5) - \pi_{R5.id}(R_7)$

The steps R_1 to R_4 are used to remove the id of the students with the highest mark. In R_4, R2.id is a relation that does not contain the id of the students with the highest mark. R_5 is a table consisting of the R2.id and R2.mark cols from R_4.

As such, it doesn't have the id of the students with the highest mark. R6 is a renamed instance of R5. R7 is a relation where R5.id doesn't contain the id of the student with the current highest mark. They used to be the students with the second highest marks. R8 is a table with the id of the students with the second highest mark. $\pi_{id}(R5)$ contains all the id of the students that do not have the highest mark. $\pi_{R5.id}(R7)$ contains the id of the students who do not have the highest or second highest marks. Therefore, $\pi_{id}(R5) - \pi_{R5.id}(R7)$ gets you the id of the students who have the second highest mark.

15. Finding "at least" using relational algebra.

Using the 2 schemas from 12, find the id of the students taking at least 2 courses.

Soln:

$R_1 = \pi_{id, class}(M)$

$R2 = \rho_{R2}(R_1)$

$R3 = R_1 \bowtie_{(R1.id = R2.id \text{ and } R1.class \,!= R2.class)} R2$

You're finding all the rows of $R_1 \times R_2$ where R1.id equals to R2.id but R1.class doesn't equal to R2.class.

Using the schemas from 12, find the id of the students taking at least 3 classes.

Soln:

$R_1 = \pi_{id,\, class}(M)$

$R2 = \rho_{R2}(R_1)$

$R3 = \rho_{R3}(R_1)$

$R4 = R_1 \times R2 \times R3$

R5 = σ(R1.id = R2.id and R1.id = R3.id and R1.class != R2.class and R1.class != R3.class and R2.class != R3.class) (R4)

16. Finding "exactly" using relational algebra.

Using the schemas from 12, find the id of the students taking exactly 2 courses.

Soln:

$R_1 = \pi_{id,\, class}(M)$

$R2 = \rho_{R2}(R_1)$

$R3 = \rho_{R3}(R_1)$

$R_4 = R_1 \bowtie_{(R1.id = R2.id \text{ and } R1.class\ != R2.class)} R_2$

$R_5 = R_1 \times R_2 \times R_3$

$R_6 = \sigma_{(R1.id = R2.id \text{ and } R1.id = R3.id \text{ and } R1.class\ != R2.class \text{ and } R1.class\ != R3.class \text{ and } R2.class\ != R3.class)} (R_5)$

$R_7 = \pi_{R1.id}(R_4) - \pi_{R1.id}(R_6)$

To find exactly n of something, find at least n and at least n+1 and subtract at least n from at least n+1. In this example, to find the ids of the students taking exactly 2 courses, we found the id of the students taking at least 2 courses and subtracted it from the id of the students taking at least 3 courses.