Relational Algebra Examples

Consider the 2 relations below:

A

Num	Square
1	1
2	4

B

Num	Cube
1	1
3	27

1. Select:
   - Denoted by $\sigma_c(R)$ where
     - σ is the select symbol.
     - c is the condition/propositional logic.
     - R is the name of the relation.
   - Select gets all tuples from the given relation that satisfies the condition.
   - E.g. 1 $\sigma_{Num>1}(A)$ will give us

Num	Square
2	4

   - E.g. 2 $\sigma_{Num=3}(B)$ will give us

Num	Cube
3	27

   - E.g. 3 $\sigma_{Num<2}(B)$ will give us

Num	Cube
1	1

2. Project:
- Denoted by $\pi_{col1, col2, \ldots, coln}(R)$ where
  - π is the project symbol.
  - col1, ..., coln are the names of attributes of the given relation.
  - R is the name of the relation.
- Project gets all the attributes listed.
- E.g. 4 $\pi_{Num}(A)$ will give us

Num
1
2

3. Natural Join:
- Denoted by $R \bowtie S$ where $\bowtie$ is the natural join symbol and R and S are relations.
- Also called inner join.
- It joins 2 relations based on same column name(s), and same values in those columns.

Note: If the 2 relations have no common attribute name, the result is the cartesian product of the 2 relations.
- E.g. 5 $A \bowtie B$ gives us

Num	Square	Cube
1	1	1

Here, both A and B have an attribute called Num. We join the tables based on that col. Next, we try to find all values that are in both relation's Num col. The only value is 1. Hence, we have a row with 1 under the Num col in the result relation.

Furthermore, since 2 is only in A and 3 is only in B, we omit those.

- E.g. 6 Consider these 2 new relations below

C

A	B
1	2
2	3

D

x	y
4	6
3	5

$C \bowtie D$ gives us

A	B	x	y
1	2	4	6
1	2	3	5
2	3	4	6
2	3	3	5

Here, since C and D do not share any cols, the result is $C \times D$ or the cartesian product of C and D.

- E.g. 7 Consider these 2 new relations below

E

A	B
1	2
2	3

F

A	B
1	3
2	3

$E \bowtie F$ gives us

A	B
2	3

Here, since A and B have the same 2 cols, we need to use row(s) s.t. both values of those row(s) are the same for E and F. While both E and F have a 1 in their A col of their first row, that row's B value is different for E and F, so we can't use it. E and F's second row have the same values, so we use those values.

4. Cartesian Product:

- Denoted by $R \times S$ where R and S are relations.
- Also called cross join.
- Produces all possible pairs of rows of the 2 relations.
- If R has m rows and S has n rows, then $R \times S$ has $m \cdot n$ rows.
- E.g. 8 $A \times B$ gives us

A.num	Square	B.num	Cube
1	1	1	1
1	1	3	27
2	4	1	1
2	4	3	27

5. Left Join:

- Denoted by $R \bowtie_L S$ where R and S are relations.
- Keeps all the tuples from the first relation and tries to find matching tuples from the second relation. If there is no matching tuple, a null value is used.
- E.g. 9 $A \bowtie_L B$ gives us

Num	Square	Cube
1	1	1
2	4	Null

Since B doesn't have a 2 in its Num col, its value for col Cube when Num =2 is Null.

6. Right Join:
- Denoted as $R \bowtie_R S$ where R and S are relations.
- Like left join, but this time, we keep all the tuples from the right relation and try to match tuples from the left relation.
- E.g. 10 $A \bowtie_R B$ gives us

Num	Cube	Square
1	1	1
3	27	Null

7. Full Join:
- Denoted by $R \bowtie_O S$ where R and S are relations.
- All tuples from both relations are included in the result.
- E.g. 11 $A \bowtie_O B$ gives us

Num	Square	Cube
1	1	1
2	4	Null
3	Null	27

8. Theta Join:
- Denoted by $R \bowtie_c S$ where R and S are relations and c is a condition.
- With $R \bowtie_c S$, you first do $R \times S$ and then σ_c on that.
- E.g. 12 $A \bowtie_{A.Num>1} B$ gives us

A.num	Square	B.Num	Cube
2	4	1	1
2	4	3	27

9. Union:

- Denoted by $R \cup S$ where R and S are relations.
- Gives a relation with the tuples that are in R or S.
- Will eliminate duplicate tuples.
- E.g. 13 Consider R and S below:

R

A	B
1	2

S

A	B
3	4

$R \cup S$

A	B
1	2
3	4

Note: For union, diff and intersection, the relations must have the same num of cols, and cols with the same names and order.

10. Difference:

- Denoted by $R - S$ where R and S are relations.
- Gives a relation with all the tuples only in R and not in S or both R and S.
- E.g. 14 Consider R and S below

R

A	B
1	2
3	4

S

A	B
3	4
4	5

$R - S$

A	B
1	2

11. Intersection:

- Denoted by $R \cap S$ where R and S are relations.
- Gives a relation with all tuples in both R and S.
- $R \cap S$ is equivalent to $R - (R - S)$.
- E.g. 15 Consider R and S below

R

A	B
1	1
2	3

S

A	B
1	1
3	4

$R \cap S$

A	B
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. $R_4 = R_2 \bowtie_{(R2.mark < R3.mark)} R_3$
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 R_2.mark < R_3.mark. That means R_2.id in R_4 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 R4, R2.id is a relation that does not contain the id of the students with the highest mark. R5 is a table consisting of the R2.id and R2.mark cols from R4.

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 $R_1.id$ equals to $R_2.id$ but $R_1.class$ doesn't equal to $R_2.class$.

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

Soln:

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

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

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

$R_4 = R_1 \times R_2 \times R_3$

$R_5 = \sigma_{(R1.id = R2.id \text{ and } R1.id = R3.id \text{ and } R1.class \,!= R2.class \text{ and } R1.class \,!= R3.class \text{ and } R2.class \,!= R3.class)}(R_4)$

16. Finding "exactly" using relational algebra.

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

Soln:

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

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

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

$R4 = R_1 \bowtie_{(R1.id = R2.id \text{ and } R1.class != R2.class)} R_2$

$R5 = R_1 \times R_2 \times R_3$

$R6 = \sigma_{(R1.id = R2.id \text{ and } R1.id = R3.id \text{ and } R1.class != R2.class \text{ and } R1.class != R3.class \text{ and } R2.class != R3.class)} (R5)$

$R7 = \pi_{R1.id}(R4) - \pi_{R1.id}(R6)$

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.