

1 Relational Algebra (RA)

Relation (R): Db table

Attribute (A): Column of a table

Tuples (t): Row of a table

k: Name of the attribute

1.1 Projection (Π)

Select all t and some attributes A_1, A_2, \dots, A_n from a relation R . Then,

$$\Pi_{A_1, A_2, \dots, A_n}(R) = \{t[A_1, A_2, \dots, A_n] : t \in R\} \quad (1)$$

| RA | SQL |
|---------------------------------|--|
| $\Pi_{A_1, A_2, \dots, A_n}(R)$ | select A_1, A_2, \dots, A_n from R |

Table 1: Equivalence RA and SQL

1.2 Selection (σ)

Select all tuples t that satisfies the condition in the relation R . Then,

$$\sigma_{condition}(R) = \{t \in R : condition(t) \text{ is true}\} \quad (2)$$

| RA | SQL |
|-------------------------|-------------------------------------|
| $\sigma_{condition}(R)$ | select * from R where $condition$ |

Table 2: Equivalence RA and SQL

1.3 Composition (Π) and (σ)

Select attributes A_1, A_2, \dots, A_n and all tuples t that satisfies the condition from a relation R . Then,

$$\Pi_{A_1, A_2, \dots, A_n}(\sigma_{condition}(R)) = \{t \in R : t[A_1, A_2, \dots, A_n] \ \& \ condition(t) \text{ is true}\} \quad (3)$$

| RA | SQL |
|---|--|
| $\Pi_{A_1, A_2, \dots, A_n}(\sigma_{condition}(R))$ | select A_1, A_2, \dots, A_n from R where $condition$ |

Table 3: Equivalence RA and SQL

1.4 Tuples without duplicate information (δ)

Select all tuples t that satisfies the condition and $t_a \neq t_c$ from a relation R . Then,

| RA | SQL |
|-------------|----------------------------|
| $\delta(R)$ | select DISTINCT * from R |

Table 4: Equivalence RA and SQL

1.5 Cartesian Product (\times)

Set of tuples obtained when we combine two relation A, B where the tuples $a \in A$ and $b \in B$. Then,

$$A \times B = \{(a, b) : a \in A \ \& \ b \in B\} = (a_1, b_1), \dots, (a_m, b_1), \dots, (a_m, b_2), \dots, (a_m, b_n) \quad (4)$$

| RA | SQL |
|-------|---|
| AxB | select * from A,B select * from A cross join B |

Table 5: Equivalence RA and SQL. Take account that we go to obtain $m * n$ tuples

1.6 Inner Join or Join (\bowtie_k)

Combines two relations A, B by an attribute that has different name ($A.name1, B.name2$) and same value ($A_n = B_m$). If the value appears in only one table then the tuple is not taken account.

| RA | SQL |
|---------------------------------|---|
| $\sigma_{A.name1=B.name2}(AxB)$ | select * from A INNER JOIN B ON A.name1 = B.name2 |
| $A \bowtie_{A.name1=B.name2} B$ | select * from A JOIN B ON A.name1 = B.name2 select * from A, B where A.name1 = B.name2 |

Table 6: Equivalence RA and SQL.

1.7 Natural Join (\bowtie)

Combines two relations A, B by attributes that has same name ($A.name = B.name$) and same value ($A_n = B_m$). If the value appears in only one table then the tuple is not taken account.

| RA | SQL |
|-------------------------------|--------------------------------|
| $\sigma_{A.name=B.name}(AxB)$ | select * from A NATURAL JOIN B |
| $A \bowtie B$ | |

Table 7: Equivalence RA and SQL. Be careful if there are more than one attribute with the same name.

1.8 Left Join ($A \bowtie_k B$)

Combines two relations A, B by attributes that has different name ($A.name1 = B.name2$) and same value ($A_n = B_m$). If the value appears in only table A then the other values go to be null.

| RA | SQL |
|---------------------------------|--|
| $A \bowtie_{A.name1=B.name2} B$ | select * from A LEFT JOIN B on A.name1 = B.name2 select * from A LEFT OUTER JOIN B on A.name1 = B.name2 |

Table 8: Equivalence RA and SQL.

1.9 Right Join ($A \bowtie_k B$)

Combines two relations A, B by attributes that has different name ($A.name1 = B.name2$) and same value ($A_n = B_m$). If the value appears in only table B then the other values go to be null.

| RA | SQL |
|---------------------------------|--|
| $A \bowtie_{A.name1=B.name2} B$ | select * from A RIGHT JOIN B on A.name1 = B.name2 select * from A RIGHT OUTER JOIN B on A.name1 = B.name2 |

Table 9: Equivalence RA and SQL.

1.10 Full Join ($A \bowtie_k B$)

Combines two relations A, B by attributes that has different name ($A.name1 = B.name2$) and same value ($A_n = B_m$). If the value appears in only table A then the other values go to be null and if the value appears in only table B then the other values go to be null.

| RA | SQL |
|---------------------------------|--|
| $A \bowtie_{A.name1=B.name2} B$ | select * from A FULL OUTER JOIN B on A.name1 = B.name2 select * from A FULL JOIN B on A.name1 = B.name2 |

Table 10: Equivalence RA and SQL.

1.11 Rename (ρ)

Variable used to rename a relation $\rho_{new_name}(R)$ or rename an attribute $\rho_{new_name,(A_1,A_2,\dots,A_n)}(R)$ where A_1, A_2, \dots, A_n could be new names.

| RA | SQL |
|--------------------------------------|---|
| $\rho_{R1}(R)$ | select * from R AS R1 |
| $\rho_{R2(AA_1,AA_2,\dots,AA_n)}(R)$ | select A1 AS AA1, ..., An AS AAn from R AS R2 |

Table 11: Equivalence RA and SQL.

1.12 Union (\cup)

If we have two relations A, B with same *length* and *types* where $type_{A_m} = type_{B_m}$. Then, $R_{C_1,\dots,C_m} := A \cup B$.

| RA | SQL |
|------------|--|
| $A \cup B$ | select * from A UNION select * from B; |

Table 12: Equivalence RA and SQL.

1.13 Intersection (\cap)

If we have two relations A, B with same *length* and *types* where $type_{A_m} = type_{B_m}$. Then, $R_{C_1,\dots,C_m} := A \cap B$.

| RA | SQL |
|------------|--|
| $A \cap B$ | select * from A INTERSECT select * from B; |

Table 13: Equivalence RA and SQL.

1.14 Difference ($-$)

If we have two relations A, B with same *length* and *types* where $type_{A_m} = type_{B_m}$. Then, $R_{C_1,\dots,C_m} := A - B$.

| RA | SQL |
|------------|---|
| $A \cap B$ | select * from A EXCEPT select * from B; |

Table 14: Equivalence RA and SQL.

1.15 Division (\div)

If we have two relations A, B where $A \cap B$ and B have the attributes $A_t = [A_{l+1}, A_{l+2}, \dots, A_m]$. Then, $A \div B$ returns tuples with attributes $A_{sol} = A - A_t$ where for each tuple of B there are a same tuple in $A \cap B$ with same tuple in A_{sol} .

If we have two tables A, B where the name of the attributes of A are 1,2 and the name for attribute B is 2. Then, $A \div B$ in the relational algebra could be found using

select $A.1$ from A, B where $A.2 = B.2$ group by $A.1$ having $count(*) = (select\ count(*)\ from\ B);$

If we have two tables A, B where the name of the attributes of A are 1,2,3,4, and the name for attributes B are 3,4. Then, $A \div B$ in the relational algebra could be found using

select $A.1, A.2$ from A, B where $A.3 = B.3$ and $A.4 = B.4$ group by $A.1, A.2$ having $count(*) = (select\ count(*)\ from\ B);$

Using this deduction, we could use the function “divide.” that returns a “div” temporal table. Then, we could obtain a division of two tables with

| RA | SQL |
|------------|--|
| $A \div B$ | select DIVIDE('A','B'); select * from DIV; |

Table 15: Equivalence RA and SQL.

1.16 Assigination ($:=$)

It gives a relational expression name. For instance: $R' := \Pi_{A_1}(R)$.

| RA | SQL |
|---------------------|---|
| $R := R1$ | alter table R to $R1$ |
| $R_{A1} := R_{AA1}$ | alter table R rename column $A1$ to $AA1$; |

Table 16: Equivalence RA and SQL.

1.17 Aggregation Function (\mathcal{F})

$\mathcal{F}_{function_name(A_1, A_2, \dots, A_n)}(R)$ execute a function over attributes A_1, A_2, \dots, A_n into a R relation.

| RA | SQL |
|---|--|
| $\mathcal{F}_{function_name(A_1, A_2, \dots, A_n)}(R)$ | select $Function_Name(A_1, A_2, \dots, A_n)$ from R ; |

Table 17: Equivalence RA and SQL.

| RA FUNCTIONS | SQL FUNCTIONS | Description |
|--|---|-----------------------------|
| $\mathcal{F}_{max(A_1)}(R)$ | select $max(A_1)$ from R ; | maximum value of tuples |
| $\mathcal{F}_{min(A_1)}(R)$ | select $min(A_1)$ from R ; | minimum value of tuples |
| $\mathcal{F}_{count(A_1)}(R)$ | select $count(A_1)$ from R ; | tuples sum |
| $\mathcal{F}_{avg(A_1)}(R)$ | select $avg(A_1)$ from R ; | tuples average |
| $\mathcal{F}_{concat(A_1, ' ', A_2)}(R)$ | select $concat(A_1, ' ', A_2)$ from R ; | tuples concatenated |
| $\mathcal{F}_{(A_1 ' ' A_2)}(R)$ | select $A_1 ' ' A_2$ from R ; | tuples concatenated |
| $\mathcal{F}_{generate_series(1,5)}(R)$ | select $generate_series(1, 5)$; | tuples with int numbers 1:5 |
| $\mathcal{F}_{generate_series(1,5)}(R)$ | select * from $generate_series(1, 5)$; | tuples with int numbers 1:5 |

Table 18: Equivalence RA and SQL.