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 $A_1, A_2, ..., A_n$ from a relation R. Then,

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

RA	SQL
$\Pi_{A_1,A_2,...,A_n}(R)$	select $A_1, A_2, ..., A_n$ 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,

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

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

Table 2: Equivalence RA and SQL

1.3 Composition (Π) and (σ)

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

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

RA	SQL
$\Pi_{A_1,A_2,...,A_n}(\sigma_{condition}(R))$	select $A_1, A_2, ..., A_n$ 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 (x)

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

$$AxB = \{(a, b) : a \in A \ \ \& \ \ b \in B\} = (a_1, b_1), ..., (a_m, b_1), ..., (a_m, b_2) \tag{4}$$

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 ($\bowtie_k$)

Union of two relations A, B by an attribute that has different 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
$\sigma_{A.name=B.name}(AxB)$ $A \bowtie_{A.name=B.name} B$	select $*$ from A INNER JOIN B ON $A.name = B.name$ select $*$ from A JOIN B ON $A.name = B.name$ select $*$ from A, B where $A.name = B.name$

Table 6: Equivalence RA and SQL.

1.7 Natural Join ($\bowtie$)

Union of 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
$\sigma_{A.name=B.name}(AxB)$ $A \bowtie B$	select $*$ from A NATURAL JOIN B

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

1.8 Left Join ($A \ltimes_k B$)

Union of 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 table A then the other values go to be null.

RA	SQL
$A \ltimes_{A.name=B.name} B$	select $*$ from A LEFT JOIN B on $A.name = B.name$ select $*$ from A LEFT OUTER JOIN B on $A.name = B.name$

Table 8: Equivalence RA and SQL.

1.9 Right Join ($A \rtimes_k B$)

Union of 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 table B then the other values go to be null.

RA	SQL
$A \rtimes_{A.name=B.name} B$	select $*$ from A RIGHT JOIN B on $A_{.name} = B_{.name}$ select $*$ from A RIGHT OUTER JOIN B on $A_{.name} = B_{.name}$

Table 9: Equivalence RA and SQL.

1.10 Full Join ($A \ltimes\!\!\rtimes_k B$)

Union of 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 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
$A ⟗_{A.name=B.name} B$	select $*$ from A FULL OUTER JOIN B on $A_{.name} = B_{.name}$ select $*$ from A FULL JOIN B on $A_{.name} = B_{.name}$

Table 10: Equivalence RA and SQL.

1.11 Rename (ρ)

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

RA	SQL
$\rho_{R1}(R)$ $\rho_{R2(AA_1,AA_2,...,AA_n)}(R)$	select $*$ from R AS $R1$ 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, $R_{A_1,...,A_m} := A \cup B = t_A \cup t_B$ where $\exists!\ t_{A_1,...,A_m} = t_{B_1,...,B_m}$.

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, $R_{A_1,...,A_m} := A \cap B = t_A \cap t_B$ where $\exists!\ t_{A_1,...,A_m} = t_{B_1,...,B_m}$.

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, $R_{A_1,...,A_m} := A - B = t_A - t_B$ where $t_{A_1,...,A_m} \neq t_{B_1,...,B_m}$.

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 $A \cap B$ have the attributes $A_{l+1}, A_{l+2}, ..., A_m$. Then, $A \div B$ returns all tuples $t_1, t_2, ..., t_n$ of A that satisfies all tuples of B without taking account attributes $A_{l+1}, A_{l+2}, ..., A_m$.

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 Assignation (:=)

It is possible give a name a relational expression. 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 ($\mathscr{F}$)

$\mathscr{F}_{function\_name(A_1, A_2, ..., An)}(R)$ execute a function over attributes $A_1, A_2, ..., A_n$ into a R relation.

RA	SQL
$\mathscr{F}_{function\_name(A_1, A_2, ..., An)}(R)$	select $Function\_Name$ $(A_1, A_2, ..., A_n)$ from R;

Table 17: Equivalence RA and SQL.

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

Table 18: Equivalence RA and SQL.