**Extended relational algebra** (relations are multisets)

| | |
|---|---|
| Union: $R \cup S$ | (sum of appearances) |
| Intersection: $R \cap S$ | (minimum of appearances) |
| Difference: R - S | (difference of appearances) |
| Selection: $\sigma_C(R)$ | (where C is a condition) |
| Projection: $\pi_L(R)$ | (in L: arithmetic expressions, renaming) |
| Product: $R \times S$ | |
| Join: $R \bowtie S$ ($R \bowtie_\theta S$) | (where $\theta$ is a condition) |
| Duplicate elimination: $\delta(R)$ | |
| Grouping and aggregation: $\gamma_L(R)$ | |
| (in L: grouping expressions and aggregated expressions, plus renaming) | |
| Sorting: $\tau_L(R)$ | |

**Examples:**

| | |
|---|---|
| $\pi_{A,B+C->X}(R)$ | `SELECT A, B+C AS X FROM R;` |
| $\delta(R)$ | `SELECT DISTINCT * FROM R;` |
| $R \cup S$ | `SELECT * FROM R UNION ALL SELECT * FROM S;` (**multiset**) |
| $R \cap S$ | `SELECT * FROM R INTERSECT ALL SELECT * FROM S;` **(!)** |
| R - S | `SELECT * FROM R MINUS ALL SELECT * FROM S;` **(!)** |
| $\delta(R \cup S)$ | `SELECT * FROM R UNION SELECT * FROM S;` (**set**) |
| $\delta(R \cap S)$ | `SELECT * FROM R INTERSECT SELECT * FROM S;` (**set**) |
| $\delta(R) - \delta(S)$ | `SELECT * FROM R MINUS SELECT * FROM S;` (**set**) |
| $R \bowtie S$ | `SELECT * FROM R NATURAL JOIN S;` |
| $R \bowtie_\theta S$ | `SELECT * FROM R JOIN S ON (θ);` |
| $R \times S$ | `SELECT * FROM R CROSS JOIN S;` or `SELECT * FROM R, S;` |
| $\gamma_{A,SUM(B)}(R)$ | `SELECT A, SUM(B) FROM R GROUP BY A;` |
| $\gamma_{A,COUNT(B)}(\delta\ \pi_{A,B}\ R)$ | `SELECT A, COUNT(DISTINCT B) FROM R GROUP BY A;` |
| $\tau_{A,\ B+C}(R)$ | `SELECT * FROM R ORDER BY A, B+C;` |

**(!) Attention**

Oracle does not support INTERSECT ALL and MINUS ALL, so $R \cap S$ and $R - S$ cannot be expressed in Oracle in multiset meaning.

**A complex sample query in SQL and Ralax extended relational algebra:**

```
SELECT DISTINCT dname, AVG(sal) + 100 AS sal_plus
FROM emp e, dept d
WHERE e.deptno = d.deptno
GROUP BY dname
HAVING COUNT(empno) > 3
ORDER BY dname;
```

In Relax we do not have δ, because the result of every operation is a set.

In Relax syntax:

τ dname (π dname, av+100→sal_plus
(σ cnt>3 (γ dname;avg(sal)→av,count(empno)→cnt (σ e.deptno=d.deptno (ρ e Emp × ρ d Dept)))))

Execution steps of a SELECT statement expressed in relational algebra:

1. replace all usages of the temporary-tables defined in the WITH-clause
2. ⋈ joins or product operations after FROM-clause
3. σ selection based on the WHERE-clause
4. γ creating groups and computing aggregations, based on GROUP BY-clause
5. σ selection for the groups or tuples created from the groups, based on HAVING-clause
6. π projection based on SELECT-clause
7. ρ rename result attributes based on AS keyword
8. ∪ ∩ – UNION, INTERSECT, MINUS set operations
9. δ duplicate elimination if we have DISTINCT