Extended relational algebra

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, renaming)
Product: $R \times S$
Join: $R \bowtie S$ ($R \bowtie_\theta S$) (where θ 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;`
$R \bowtie S$	`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;`

Complex example query in SQL and extended relational algebra:

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

$\tau_{dname}(\pi_{dname,av+100->sal\_plus}(\sigma_{cnt>3}(\gamma_{dname,AVG(sal)->av,COUNT(empno)->cnt}(\sigma_{E.deptno=D.deptno}(Emp \times Dept)))))$

Exercises (SQL and relational algebra)

```
NIKOVITS.EMP (empno, ename, job, mgr, hiredate, sal, comm, deptno)
NIKOVITS.DEPT(deptno, dname, loc)
NIKOVITS.SAL_CAT(category, lowest_sal, highest_sal)
NIKOVITS.LIKES(name, fruits)
```

1.
List the department number, department name and location for the departments having an employee with salary category 1. (**deptno, dname, loc**)

E := Emp; D := Dept; S := Sal_cat;

T := $\pi_{deptno}\ \sigma_{cat=1\ AND\ sal >= lowest\_sal\ AND\ sal <= highest\_sal}$ (E x S)
$\pi_{deptno,\ dname,\ loc}\ \sigma_{T.deptno=D.deptno}$ (T x D)

```
SELECT deptno, dname, loc from dept WHERE deptno IN
  (SELECT deptno FROM emp, sal_cat
   WHERE category = 1 AND sal BETWEEN lowest_sal AND highest_sal);
```

2.
List the department number, department name and location for the departments **having no employee** with salary category 1. (**deptno, dname, loc**)

E := Emp; D := Dept; S := Sal_cat;
T1 := $\pi_{deptno}\ \sigma_{cat=1\ AND\ sal >= lowest\_sal\ AND\ sal <= highest\_sal}$ (E x S)
T2 := $\pi_{deptno,\ dname,\ loc}\ \sigma_{T1.deptno=D.deptno}$ (T1 x D)
$\pi_{deptno,\ dname,\ loc}$ D – T2

```
SELECT deptno, dname, loc from dept
 MINUS
SELECT deptno, dname, loc from dept WHERE deptno IN
  (SELECT deptno FROM emp, sal_cat
   WHERE category = 1 AND sal BETWEEN lowest_sal AND highest_sal);
```

Other solution:

```
SELECT deptno, dname, loc from dept WHERE deptno NOT IN
  (SELECT deptno FROM emp, sal_cat
   WHERE category = 1 AND sal BETWEEN lowest_sal AND highest_sal);
```

3.
List the department number, department name and location for the departments **having at least two employees** with salary category 1. (**deptno, dname, loc**)

E := Emp; D := Dept; S := Sal_cat;
T := $\gamma_{deptno,COUNT(ename)->cnt}\ \sigma_{cat=1\ AND\ sal >= lowest\_sal\ AND\ sal <= highest\_sal}$ (E x S)
$\pi_{deptno,\ dname,\ loc}\ \sigma_{T.deptno=D.deptno\ AND\ cnt >= 2}$ (T x D)

```
SELECT deptno, dname, loc from dept WHERE deptno IN
  (SELECT deptno FROM emp, sal_cat
   WHERE category = 1 AND sal BETWEEN lowest_sal AND highest_sal
   GROUP BY deptno HAVING COUNT(ename) >= 2);
```

4.
List the employees who have maximal salary within their own department. Give the department number, employee name and salary for them. (**deptno, ename, sal**)

T := $\gamma_{deptno,Max(sal)->ms}$(Emp); E:= Emp;
$\sigma_{T.deptno=E.deptno\ AND\ E.sal = ms}$(T x E)

```
SELECT e.deptno, ename, sal
FROM emp e, (SELECT deptno, MAX(sal) ms FROM emp GROUP BY deptno) t
WHERE e.deptno = t.deptno and e.sal = ms;
```

5.
List the jobs where this job occurs only on one department, and give the name of this department too. **(job, dname)**

$T := \sigma_{dn=1}\, \gamma_{job,Count(deptno)->dn}(\delta\, \pi_{job,\, deptno}\, Emp)$; E := Emp; D := Dept;
$\pi_{job,\, dname}\, T \bowtie E \bowtie D$

```
SELECT DISTINCT job, dname FROM emp NATURAL JOIN dept
WHERE job IN
  (SELECT job FROM emp
   GROUP BY job HAVING COUNT(DISTINCT deptno) = 1);
```

6.
Give the names who like every fruit. (relation Likes)

$T := \gamma_{Count(fruits)->fr}\, \delta\, (\pi_{\, fruits}\, (Likes))$;
$\pi_{name}\, (T \bowtie \gamma_{name,\, Count(fruits)->fr}\, Likes)$

```
SELECT name FROM likes
GROUP BY name
HAVING COUNT(fruits) = (SELECT COUNT(DISTINCT fruits) FROM likes);
```

Other solution without grouping:

L1 := Likes; L2 := Likes; NotLikes := $\delta(\pi_{L1.name,\, L2.fruits}\, (L1 \times L2)) - Likes$
$\pi_{name}\, Likes - \pi_{name}\, NotLikes$

```
SELECT name FROM likes
 MINUS
SELECT DISTINCT name FROM
(SELECT DISTINCT l1.name, l2.fruits FROM likes l1, likes l2
  MINUS
 SELECT name, fruits FROM likes) NotLikes;
```

7.
Give the salary and salary category of the employee who has the lowest salary among the employees having a subordinate. (**sal, category**)

Manager := $\pi_{mgr->empno}\, Emp$; $T := \gamma_{Min(sal)\, ->ms}\, Manager \bowtie Emp$; S := Sal_cat;
$\pi_{ms,\, category}\, \sigma_{\, ms >= lowest\_sal\ AND\ ms <= highest\_sal}\, (S \times T)$

```
SELECT minsal, category FROM sal_cat,
(SELECT MIN(sal) minsal
 FROM emp NATURAL JOIN (SELECT mgr empno FROM emp)) t
WHERE t.minsal BETWEEN lowest_sal AND highest_sal;
```

Other solution:

```
SELECT minsal, category FROM sal_cat,
(SELECT MIN(sal) minsal FROM emp WHERE empno IN (SELECT mgr FROM emp)) t
WHERE t.minsal BETWEEN lowest_sal AND highest_sal;
```