R3.07 – SQL dans un langage de programmation

CORRECTION TD3

Question 1 :

```
SELECT d.nomdemandeur, d.prenomdemandeur, COUNT(vm.idvm) AS "nb vm"
FROM demandeur d
  LEFT JOIN vm ON d.iddemandeur=vm.iddemandeur
GROUP BY d.nomdemandeur, d.prenomdemandeur
ORDER BY 1,2;
```

Question 2 :

```
SELECT d.nomdemandeur, d.prenomdemandeur, COUNT(vm.idvm) as "nb vm"
FROM demandeur d
  JOIN vm ON d.iddemandeur=vm.iddemandeur
GROUP BY d.nomdemandeur, d.prenomdemandeur
HAVING COUNT(vm.idvm) >= ALL
      (SELECT COUNT(vm.idvm)
      FROM vm
      GROUP BY iddemandeur)
```

Question 3 :

Si on réutilise la Q1 :

```
SELECT d.nomdemandeur, d.prenomdemandeur
FROM demandeur d
  LEFT JOIN vm ON d.iddemandeur=vm.iddemandeur
GROUP BY d.nomdemandeur, d.prenomdemandeur
HAVING COUNT(vm.idvm)=0
ORDER BY 1,2;
```

OU

```
SELECT d.nomdemandeur, d.prenomdemandeur
FROM demandeur d
  LEFT JOIN vm ON d.iddemandeur=vm.iddemandeur
WHERE vm.idvm IS NULL
ORDER BY 1,2;
```

OU

```
SELECT nomdemandeur, prenomdemandeur
FROM demandeur
WHERE iddemandeur NOT IN (
      SELECT iddemandeur FROM vm
      )
ORDER BY 1,2;
```

OU

```
SELECT d.nomdemandeur, d.prenomdemandeur
FROM demandeur d
WHERE NOT EXISTS (
      SELECT 'X'
      FROM vm
      WHERE vm.iddemandeur=d.iddemandeur)
ORDER BY 1,2;
```

Question 4 :

```
DROP VIEW v_derversionvm;

CREATE VIEW v_derversionvm AS
SELECT vm.idvm, vm.nomvm, vm.description, MAX(v.dateversion) AS "date der version"
FROM vm
  JOIN versionvm v ON vm.idvm=v.idvm
GROUP BY vm.idvm, vm.nomvm, vm.description
ORDER BY 1;
```

```
SELECT * FROM v_derversionvm;
```

Question 5 :

```
DROP VIEW v_derversionvm2;

CREATE VIEW v_derversionvm2 AS
SELECT vm.idvm, vm.nomvm, vm.description, MAX(v.dateversion) AS "date der version"
FROM vm
  JOIN versionvm v ON vm.idvm=v.idvm
GROUP BY vm.idvm, vm.nomvm, vm.description
HAVING to_char(max(v.dateversion), 'YYYY')=to_char(current_date, 'YYYY')
ORDER BY 1;

SELECT * FROM v_derversionvm2;
```

Question 6 :

```
SELECT vm.idvm, v.numversion, COUNT(idlogiciel) as "nb logiciels",
CASE COUNT(idlogiciel)
  WHEN 0 THEN 'OS nu'
  WHEN 1 THEN 'Peu'
  WHEN 2 THEN 'Peu'
  WHEN 3 THEN 'Normal'
  WHEN 4 THEN 'Normal'
  WHEN 5 THEN 'Normal'
  ELSE 'Beaucoup de travail !'
END AS Qualification
FROM vm
  LEFT JOIN versionvm v ON vm.idvm=v.idvm
  LEFT JOIN installation i ON v.idvm=i.idvm AND v.numversion=i.numversion
GROUP BY vm.idvm, v.numversion
ORDER BY 3 desc, 1, 2
```

Comme une VM a toujours une version, la table VM n’est pas nécessaire ici. Mais il est préférable de la conserver si on n’a pas encore créé de version sur une VM existante.

OU

```
SELECT vm.idvm, v.numversion, COUNT(idlogiciel) as "nb logiciels",
CASE
  WHEN COUNT(idlogiciel)=0 THEN 'OS nu'
  WHEN COUNT(idlogiciel) BETWEEN 1 AND 2 THEN 'Peu'
  WHEN COUNT(idlogiciel) BETWEEN 3 AND 5 THEN 'Normal'
  ELSE 'Beaucoup de travail !'
END AS Qualification
FROM vm
  LEFT JOIN versionvm v ON vm.idvm=v.idvm
  LEFT JOIN installation i ON v.idvm=i.idvm AND v.numversion=i.numversion
GROUP BY vm.idvm, v.numversion
ORDER BY 3 desc, 1, 2
```

Question 7 :

```
DROP TABLE INFOSVM;

CREATE TABLE INFOSVM
AS
SELECT idvm, nomvm, description, tailledd, memoireram, typestockage, nomdepartement,
nomdemandeur, prenomdemandeur
FROM vm
  JOIN departement dep ON vm.iddepartement=dep.iddepartement
  JOIN demandeur dem ON vm.iddemandeur=dem.iddemandeur;

SELECT * FROM INFOSVM;
```

Pas de contraintes (FK, PK, CHECK, etc.) hormis not null sur la table car les contraintes ne sont pas créées lors d’un `CREATE TABLE AS…`

Question 8 :

```
SELECT idlogiciel, nomlogiciel
FROM logiciel l
WHERE NOT EXISTS (
  SELECT 'X'
  FROM OS
  WHERE NOT EXISTS (
    SELECT 'X'
    FROM compatibilite c
    WHERE c.idlogiciel=l.idlogiciel AND c.idos=os.idos
                    )
  );
```

OU

```
SELECT l.idlogiciel, l.nomlogiciel
FROM compatibilite c
  JOIN logiciel l ON c.idlogiciel=l.idlogiciel
GROUP BY l.idlogiciel, l.nomlogiciel
HAVING COUNT(DISTINCT c.idos) = (
                                   SELECT COUNT(*) FROM os
                                )
ORDER BY 1;
```

Question 9 :

```
DROP VIEW v_vmpardept;

CREATE VIEW v_vmpardept
AS
      SELECT nomufr, nomdepartement, COUNT(*) AS "Nb VM"
      FROM departement d
        JOIN vm ON d.iddepartement=vm.iddepartement
      GROUP BY nomufr, nomdepartement;

SELECT * FROM v_vmpardept;
```

Question 10 :

```
SELECT vm.idvm, vm.nomvm
FROM vm
  JOIN versionvm v ON vm.idvm=v.idvm
WHERE CURRENT_DATE - INTERVAL '36 months' <= v.dateversion
GROUP BY vm.idvm, vm.nomvm
HAVING count(*)>=2;
```

Question 11 :

```
SELECT idlogiciel, nomlogiciel
FROM logiciel
WHERE idlogiciel NOT IN (
      SELECT idlogiciel
      FROM installation i
        JOIN versionvm v ON i.idvm=v.idvm AND i.numversion=v.numversion
      WHERE CURRENT_DATE - INTERVAL '8 years' <= v.dateversion
);
```

Question 12 :

```
SELECT l.idlogiciel, l.nomlogiciel
FROM compatibilite c
  JOIN logiciel l ON c.idlogiciel=l.idlogiciel
  JOIN os ON c.idos=os.idos
  JOIN typeos t ON os.idtypeos=t.idtypeos
WHERE t.typeos='Mac OS/X'
GROUP BY l.idlogiciel, l.nomlogiciel
HAVING COUNT(DISTINCT c.idos) = (
    SELECT COUNT(*)
    FROM os
      JOIN typeos t ON os.idtypeos=t.idtypeos
    WHERE t.typeos='Mac OS/X'
    )
ORDER BY 1;
```

OU

```
SELECT idlogiciel, nomlogiciel
FROM logiciel l
WHERE NOT EXISTS
  (
  SELECT 'X'
  FROM OS
  WHERE NOT EXISTS (
    SELECT 'X'
    FROM compatibilite c
    WHERE c.idlogiciel=l.idlogiciel AND c.idos=os.idos
                    )
  AND idos IN (
   SELECT idos
   FROM os
    JOIN typeos t ON os.idtypeos=t.idtypeos
   WHERE t.typeos='Mac OS/X'
              )
  );
```

OU

```
SELECT idlogiciel, nomlogiciel
FROM logiciel l
WHERE NOT EXISTS
  (
  SELECT 'X'
  FROM OS
      JOIN typeos t ON os.idtypeos=t.idtypeos
   WHERE t.typeos='Mac OS/X'
      AND NOT EXISTS (
                      SELECT 'X'
                      FROM compatibilite c
                      WHERE c.idlogiciel=l.idlogiciel AND c.idos=os.idos
                      )
  );
```