

Correction LogiSoft™ (séries 3 et 4)

Série 3:

R20 : le nom et prénom des supérieurs hiérarchiques (directs et indirects) de la salariée Sophie Stucké.

```
WITH REC_Salaries (nomSalarié, numSalariéChef, niveau) AS
(
    SELECT Salaries, numSalarié, 1
    FROM Salaries
    WHERE numSalarié = 'Stucké'
)
UNION ALL
SELECT chef, numSalarié, chef, numSalariéChef, niveau + 1
FROM REC_Salaries emp, REC_Salaries chef
WHERE emp.numSalarié = chef.numSalarié
SELECT niveau, nomSalarié, prenomSalarié
FROM REC_Salaries emp, REC_Salaries chef
WHERE emp.numSalarié = chef.numSalarié
ORDER BY niveau;
```

ou encore (sans Oracle):

```
SELECT LEVEL, AS niveau, nomSalarié, prenomSalarié
FROM Salaries
START WITH nomSalarié = 'Stucké'
AND prenomSalarié = 'Sophie'
CONNECT BY PRIOR numSalarié = numSalarié;
```

R21 : le nom et le prénom des subordonnés directs ou indirects du salarié Mélanie Zéotfrats.

```
WITH REC_Salaries (nomSalarié, numSalariéChef, niveau) AS
(
    SELECT Salaries, numSalarié, 1
    FROM Salaries
    WHERE numSalarié = 'Zéotfrats'
)
UNION ALL
SELECT sub, numSalarié, sub, numSalariéChef, niveau + 1
FROM REC_Salaries emp, REC_Salaries chef
WHERE emp.numSalarié = chef.numSalarié
SELECT niveau, nomSalarié, prenomSalarié
FROM REC_Salaries emp, REC_Salaries chef
WHERE emp.numSalarié = chef.numSalarié
ORDER BY niveau;
```

ou encore (sans Oracle)

```
SELECT LEVEL, nomSalarié, prenomSalarié
FROM Salaries
START WITH numSalarié = 'Zéotfrats'
AND prenomSalarié = 'Mélanie'
CONNECT BY PRIOR numSalarié = numSalarié;
```

R22 : le nom et le prénom des salariés qui sont nés à la même date qu'un autre salarié.

```
SELECT nomSalarié, prenomSalarié
FROM Salaries
WHERE dateNaissanceSalarié IN (
    SELECT dateNaissanceSalarié
    FROM Salaries
    WHERE numSalarié != 82, numSalarié)
AND numSalarié != 82, numSalarié
```

R23 : le nom et le prénom des salariés qui ont un salaire supérieur au salaire moyen des salariés de leur catégorie.

```
SELECT nomSalarié, prenomSalarié
FROM Salaries
WHERE salaireSalarié > (
    SELECT AVG(salaireSalarié)
    FROM Salaries
    WHERE categorieSalarié = 82, categorieSalarié)
AND numSalarié != 82, categorieSalarié
```

R24 : pour chaque catégorie de salarié, indiquer le nom et le prénom du salarié le mieux payé.

```
SELECT categorieSalarié, nomSalarié, prenomSalarié
FROM Salaries
WHERE salaireSalarié = (
    SELECT MAX(salaireSalarié)
    FROM Salaries
    WHERE categorieSalarié = 81, categorieSalarié)
AND numSalarié != 81, categorieSalarié
```

R25 : Le nom du salarié qui a le nom le plus long.

```
SELECT nomSalarié
FROM Salaries
WHERE LENGTH(nomSalarié) = (
    SELECT MAX(LENGTH(nomSalarié))
    FROM Salaries);
```

R26 : Le nom des salariés qui ont un nom composé. Le nom des salariés doit être affiché en lettres majuscules.

```
SELECT UPPER(nomSalarié) AS nom
FROM Salaries
WHERE LENGTH(nomSalarié) > 10;
```

R27 : Le nom du projet pour lequel les salaires des salariés ont le plus grand écart-type. On souhaite que le nom du projet soit écrit en lettres minuscules mais que la première lettre soit en majuscules.

```
SELECT INITCAP(nomProjet) AS projet
FROM Projects
WHERE codeProjet IN (
    SELECT codeProjet
    FROM Projects
    WHERE codeProjet IN (
        SELECT MAX(STDDEV(salaireSalarié))
        FROM Salaries
        GROUP BY codeProjet
    )
    ORDER BY STDDEV(salaireSalarié)
);
```

ou encore

```
WITH EcartsTypes AS
(
    SELECT nomProjet, STDDEV(salaireSalarié) AS ecart
    FROM Projects
    WHERE codeProjet IN (
        SELECT codeProjet
        FROM Projects
        WHERE codeProjet IN (
            SELECT MAX(STDDEV(salaireSalarié))
            FROM Salaries
            GROUP BY codeProjet
        )
        ORDER BY STDDEV(salaireSalarié)
    )
)
SELECT INITCAP(nomProjet) AS projet
FROM EcartsTypes
```

R28 : Pour chaque salarié de la table Salaries, le nom, le prénom et le salaire du salarié ainsi que son classement par rapport à son salaire.

```
SELECT nomSalarié, prenomSalarié, salaireSalarié, RANK() OVER (
    ORDER BY salaireSalarié DESC) AS classement
FROM Salaries
```

R29 : Le nom, le prénom et le salaire des trois salariés les mieux payés.

```
SELECT nomSalarié, prenomSalarié, salaireSalarié, RANK() OVER (
    ORDER BY salaireSalarié DESC) AS classement
FROM Salaries
WHERE RANK() <= 3
```

R30 : Afficher la liste des salariés classés par rapport à leur salaire. Pour chacun des salariés on veut les informations suivantes :

```
SELECT nomSalarié, prenomSalarié, salaireSalarié, dateNaissanceSalarié, categorieSalarié AS categorie,
COUNT(*) OVER (
    ORDER BY categorieSalarié) AS nbCat,
RANK() OVER (
    PARTITION BY categorieSalarié ORDER BY salaireSalarié DESC) AS classement,
LAG(salaireSalarié) OVER (
    ORDER BY salaireSalarié DESC) AS salairePrecedent
FROM Salaries
ORDER BY salaireSalarié DESC
```

