

Corrigés des Exercices SQL Part II

Données provenant de plusieurs tables

1. Ecrivez une interrogation produisant l'adresse de tous les départements. Utilisez les tables LOCATIONS et COUNTRIES. Affichez dans les résultats l'ID de lieu, la rue, la ville, le département et le pays. Utilisez une jointure naturelle (NATURAL JOIN) pour produire les résultats.

```
SELECT location_id, street_address, city, state_province, country_name  
FROM locations NATURAL JOIN countries
```

2. Ecrivez une interrogation permettant d'afficher le nom, ainsi que le numéro et le nom de département, pour tous les employés.

```
SELECT last_name, department_id, department_name  
FROM employees JOIN departments USING (department_id);
```

3. Affichez le nom, le poste, ainsi que le numéro et le nom de département, pour tous les employés qui travaillent à Toronto.

```
SELECT e.last_name, e.job_id, e.department_id, d.department_name FROM employees e  
JOIN departments d ON (e.department_id = d.department_id) JOIN locations l ON  
(d.location_id = l.location_id) WHERE LOWER(l.city) = 'toronto';
```

4. Créez un état permettant d'afficher le nom et l'ID des employés, ainsi que le nom et l'ID de leur manager. Intitulez respectivement les colonnes Employee, Emp#, Manager et Mgr#.

```
SELECT w.last_name "Employee", w.employee_id "EMP#", m.last_name "Manager",  
m.employee_id "Mgr#" FROM employees w join employees m ON (w.manager_id =  
m.employee_id);
```

Sous-interrogations

1. Ecrire une interrogation qui affiche le nom et la date d'embauche de tous les employés travaillant dans le même département que l'employé indiqué (en excluant ce dernier). Par exemple, si l'utilisateur saisit Zlotkey, cherchez tous les employés qui travaillent avec Zlotkey (en excluant ce dernier).

```
SELECT last_name, hire_date FROM employees WHERE department_id = (SELECT  
department_id FROM employees WHERE last_name = '&&Nom') AND last_name <> '&Nom';
```

2. Créez un état qui affiche le numéro d'employé, le nom et le salaire de tous les employés qui gagnent plus que le salaire moyen. Triez les résultats par ordre croissant sur la base du salaire.

```
SELECT employee_id, last_name, salary FROM employees  
WHERE salary > (SELECT AVG(salary) FROM employees) ORDER BY salary
```

3. Ecrivez une interrogation qui affiche le numéro d'employé et le nom de tous les employés qui travaillent dans un département comprenant un employé dont le nom contient la lettre "u"

```
SELECT employee_id, last_name FROM employees  
WHERE department_id IN (SELECT department_id FROM employees WHERE last_name like  
'%u%');
```

4. Créez un état affichant la liste de tous les employés dont le salaire est plus que le salaire de tout employé du département 60

```
SELECT last_name FROM employees WHERE salary > ANY (SELECT salary  
FROM employees WHERE department_id=60);
```

Opérateurs ensemblistes

1. Affichez la liste des ID des départements qui ne contiennent pas l'ID de poste ST_CLERK. Utilisez les opérateurs ensemblistes pour créer cet état.

```
SELECT department_id FROM departments  
MINUS  
SELECT department_id FROM employees WHERE job_id = 'ST_CLERK'
```

2. Produisez la liste des postes des départements 10, 50 et 20, dans cet ordre. Affichez l'ID de poste et l'ID de département à l'aide des opérateurs ensemblistes

```
SELECT DISTINCT job_id, department_id FROM employees WHERE department_id = 10  
UNION ALL  
SELECT DISTINCT job_id, department_id FROM employees WHERE department_id = 50  
UNION ALL  
SELECT DISTINCT job_id, department_id FROM employees WHERE department_id = 20
```

3) Le département des ressources humaines a besoin de la liste des pays dans lesquels il n'existe aucun département. Affichez l'ID et le nom des pays. Utilisez les opérateurs ensemblistes pour créer cet état.

```
SELECT country_id, country_name  
FROM countries  
MINUS  
SELECT l.country_id, c.country_name  
FROM locations l JOIN countries c  
ON (l.country_id = c.country_id)  
JOIN departments d  
ON d.location_id=l.location_id;
```