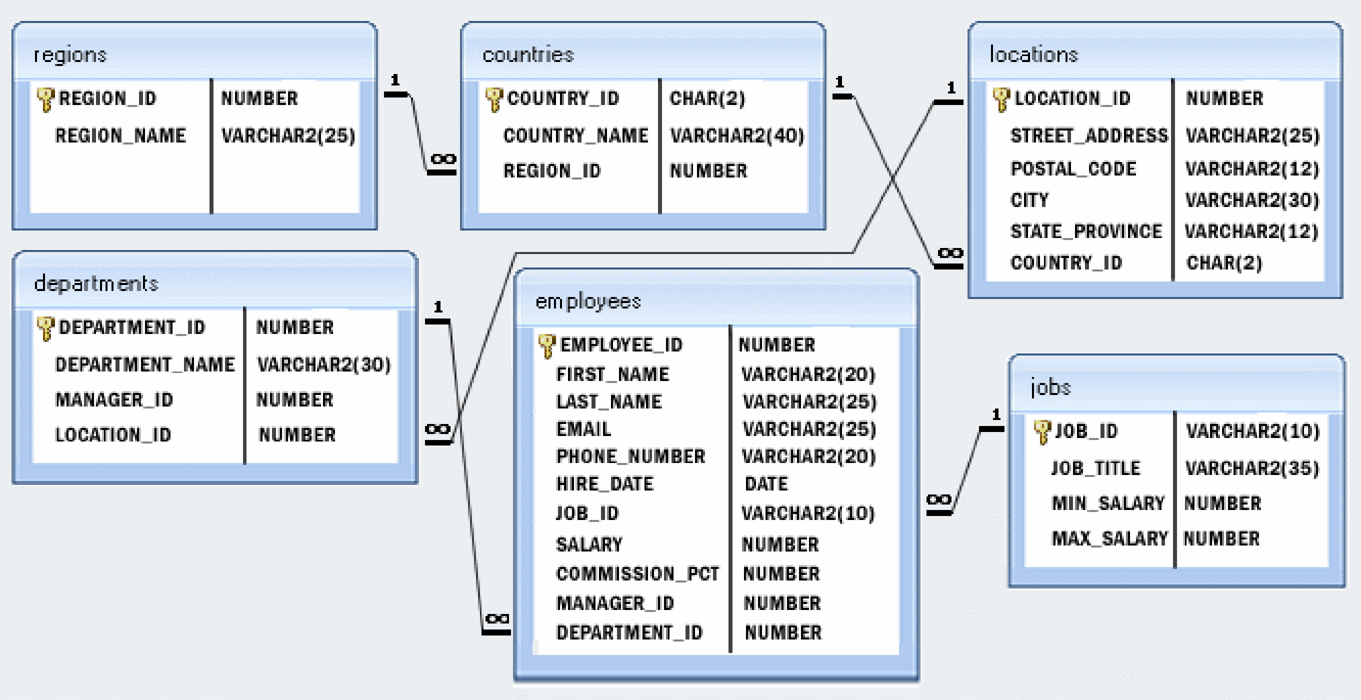
TD - Séance 1

La base de données associée au fichier `Lab12_TD1_SQL_humanresources.sql` contient 7 tables :



Partie 1 : SELECT

Question 1 : Write a query to display the content of the table regions

```
SELECT * FROM regions;
```

Question 2 : Write a query to display the content of the table employees

```
SELECT * FROM employees;
```

Question 3 : Write a query to display the titles of the jobs

```
SELECT job_title FROM jobs;
```

Question 4 : Write a query to display the employees working in the department #20

```
SELECT * FROM employees
WHERE department_id = 20;
```

Question 5 : Write a query to display the first and last name of the employee working in department #20

```sql
SELECT first_name, last_name
FROM employees
WHERE department_id = 20;
```

Question 6 : Write a query to display the first and last name of employees from departments #20, #30

```sql
SELECT first_name, last_name
FROM employees
WHERE department_id IN (20, 30);
```

Question 7 : Write a query to get the names of the departments, using the dname alias, ordered from A to

```sql
SELECT department_name AS dname
FROM departments
ORDER BY department_name ASC;
```

Question 8 : Write a query to get the lastname, firstname and salary of the three employees having the largest salaries

```sql
SELECT last_name, first_name, salary
FROM employees
ORDER BY salary DESC
LIMIT 3;
```

Question 9 : Write a query to return the number of employees as employeeCount

```sql
SELECT COUNT(*) AS employeeCount
FROM employees;
```

Question 10 : Write a query to display the manager ids, without duplicates

```sql
SELECT DISTINCT manager_id
FROM employees
WHERE manager_id IS NOT NULL;
```

Question 11 : Write a query that displays the number of unique managers as managerCount

```
SELECT COUNT(DISTINCT manager_id) AS managerCount
FROM employees
WHERE manager_id IS NOT NULL;
```

Question 12 : Write a query to get the number of employees having a salary strictly greater than 10000, using the riches alias

```
SELECT COUNT(*) AS riches
FROM employees
WHERE salary > 10000;
```

Question 13 : Write a query to get the number of departments where some employees have a commission percentage

```
SELECT COUNT(DISTINCT department_id) AS departmentCount
FROM employees
WHERE commission_pct IS NOT NULL;
```

Question 14 : Write a query to get the average salary within the company

```
SELECT AVG(salary) AS averageSalary
FROM employees;
```

Question 15 : Write a query to get the average and standard deviation of salary from employees of the department 80

```
SELECT AVG(salary) AS averageSalary, STDDEV(salary) AS salaryStandardDeviation
FROM employees
WHERE department_id = 80;
```

Question 16 : Write a query to get the total bonuses (salary times commission) earnt by the employee of department 80, as well as the minimum and maximum. Return also the number of employees in this department.

```
SELECT
    SUM(salary * commission_pct) AS totalBonuses,
    MIN(salary * commission_pct) AS minBonus,
```

```sql
    MAX(salary * commission_pct) AS maxBonus,
    COUNT(*) AS employeeCount
FROM
    employees
WHERE
    department_id = 80;
```

Partie 2 : Manipulation de strings

Question 1 : Write a query that displays the five first characters of the five first employees' last names by alphabetical order

```sql
SELECT
    LEFT(last_name, 5) AS firstFiveChars
FROM
    employees
ORDER BY
    last_name
LIMIT 5;
```

Question 2 : Your new french employee speaks a very poor english. He is looking for an employee whom he calls « Brousse », but nobody matches this first name in the database. Using the soundex function, find this employee and display the information about him.

```sql
SELECT *
FROM employees
WHERE SOUNDEX(last_name) = SOUNDEX('Brousse');
```

Question 3 : Write a query to display the information about the employee who has the longest last name

```sql
SELECT *
FROM employees
ORDER BY LENGTH(last_name) DESC
LIMIT 1;
```

Question 4 : Write a query to display in a single column called fullname the first and last names of the three first employees. The last name must be in upper case

```sql
SELECT CONCAT(first_name, ' ', UPPER(last_name)) AS fullname
FROM employees
ORDER BY employee_id
LIMIT 3;
```

Partie 3 : Dates

Question 1 : Write a query to display how long the employees have been working for the company, in days

```
SELECT employee_id,
       DATEDIFF(CURDATE(), hire_date) AS days_worked
FROM employees;
```

Question 2 : Write a query to extract the month from the current date

```
SELECT MONTH(CURDATE()) AS current_month;
```

Question 3 : Write a query to get the first name and hire date from employees table where hire date between '1987-06-01' and '1987-07-30'

```
SELECT first_name, hire_date
FROM employees
WHERE hire_date BETWEEN '1987-06-01' AND '1987-07-30';
```

Question 4 : Write a query to get the firstname, lastname of employees who joined in 1987

```
SELECT first_name, last_name
FROM employees
WHERE YEAR(hire_date) = 1987;
```

Question 5 : Write a query to get the firstname, lastname who joined in a Monday.

```
SELECT first_name, last_name
FROM employees
WHERE DAYOFWEEK(hire_date) = 2;
```

Question 6 : Write a query to display the first day of the month (in datetime format) three months before the current month.

```
SELECT DATE_FORMAT(DATE_SUB(CURRENT_DATE, INTERVAL 3 MONTH), '%Y-%m-01') AS
first_day_three_months_ago;
```

Questions supplémentaires

Bonus 02 : Trouver tous les employés embauchés au mois de juin

```sql
SELECT firstname, lastname
FROM employees
WHERE MONTH(hire_date) = 6;
```

Bonus 03 : Trouver les employés dont le prénom commence par la lettre D

```sql
SELECT firstname, lastname
FROM employees
WHERE firstname LIKE 'D%';
```

Bonus 04 : Trouver les employés dont le nom contient 4 caractères

```sql
SELECT firstname, lastname
FROM employees
WHERE LENGTH(lastname) = 4;
```

Bonus 05 : Trouver les adresses emails correctement formées avec un arobase

```sql
SELECT email
FROM employees
WHERE email LIKE '%_@_%._%';
```

Bonus 06 : Afficher le prénom et le nom des employés dont leur prénom et nom commencent par la même lettre

```sql
SELECT firstname, lastname
FROM employees
WHERE LEFT(firstname, 1) = LEFT(lastname, 1);
```

Bonus 09 : Afficher le/les noms des employés qui ont le nom le plus court (en terme de nombre de caractères)

```sql
SELECT lastname
FROM employees
WHERE LENGTH(lastname) = (SELECT MIN(LENGTH(lastname)) FROM employees);
```

Bonus 11 : Afficher le nom du premier jour de l'année d'emploi de chaque employé

```
SELECT firstname, lastname, DATE_FORMAT(hire_date, '%Y-01-01') AS
first_day_of_year
FROM employees;
```

Bonus 15 : Prénom de l'employé ayant été embauché le plus récemment

```
SELECT firstname
FROM employees
ORDER BY hire_date DESC
LIMIT 1;
```

Bonus 17 : Donner la date du 14e mercredi de l'année de dix ans avant votre naissance

```
SELECT DATE_ADD(DATE(CONCAT(YEAR(CURDATE()) - 10, '-01-01')), INTERVAL 14 * 7 DAY)
AS fourteenth_wednesday;
```

Bonus 18 : Donner les 20 premiers nombres de la séquence de Fibonacci. Vous aurez besoin du mot clef RECURSIVE

```
WITH RECURSIVE fibonacci AS (
    SELECT 0 AS num, 1 AS next_num, 1 AS index
    UNION ALL
    SELECT next_num, num + next_num, index + 1
    FROM fibonacci
    WHERE index < 20
)
SELECT num
FROM fibonacci;
```

Questions supplémentaires (avec jointures)

Bonus 01 : Donner le nombre d'employés pour chaque département, ainsi que le nom du département

```
SELECT d.department_name, COUNT(e.employee_id) AS employee_count
FROM departments d
LEFT JOIN employees e ON d.department_id = e.department_id
GROUP BY d.department_name;
```

Bonus 07 : Afficher les noms des employés et leur pays de résidence

```sql
SELECT e.firstname, e.lastname, c.country_name
FROM employees e
JOIN countries c ON e.country_id = c.country_id;
```

Bonus 08 : Quels sont les différents postes occupés par le plus vieil employé ?

```sql
SELECT DISTINCT job_title
FROM employees
WHERE employee_id = (SELECT employee_id FROM employees ORDER BY hire_date ASC
LIMIT 1);
```

Bonus 10 : Afficher le nom et prénom de chaque employé ainsi que le nom du pays, de la région et du lieux

```sql
SELECT e.firstname, e.lastname, c.country_name, r.region_name, l.location_name
FROM employees e
JOIN locations l ON e.location_id = l.location_id
JOIN countries c ON l.country_id = c.country_id
JOIN regions r ON c.region_id = r.region_id;
```

Bonus 12 : Afficher le nom et le prénom de l'employé ayant occupé le même poste durant le moins de temps

```sql
SELECT e.firstname, e.lastname
FROM employees e
JOIN (
    SELECT employee_id, job_id, MIN(end_date - start_date) AS min_duration
    FROM job_history
    GROUP BY employee_id, job_id
) j ON e.employee_id = j.employee_id
WHERE j.min_duration = (SELECT MIN(end_date - start_date) FROM job_history);
```

Bonus 13 : Afficher le nombre d'emplois qu'à eu chaque employé

```sql
SELECT employee_id, COUNT(job_id) AS job_count
FROM job_history
GROUP BY employee_id;
```

Bonus 14 : Afficher la liste des employés qui travaille aux Etats-Unis

```
SELECT firstname, lastname
FROM employees
WHERE country_id = (SELECT country_id FROM countries WHERE country_name = 'United
States');
```

Bonus 16 : Trouver le nom du département où se trouve le plus d'employés

```
SELECT d.department_name
FROM departments d
JOIN employees e ON d.department_id = e.department_id
GROUP BY d.department_name
ORDER BY COUNT(e.employee_id) DESC
LIMIT 1;
```

Bonus 19 : Pour un poste donné, un employé a pu être remplacé. Trouver les noms de ces employés remplacés et remplaçants. Trouver les dates de ces remplacements, et identifier les postes vacants

```
SELECT old_employee.firstname AS replaced_firstname, old_employee.lastname AS
replaced_lastname,
       new_employee.firstname AS replacer_firstname, new_employee.lastname AS
replacer_lastname,
       r.replacement_date, j.job_title
FROM replacements r
JOIN employees old_employee ON r.old_employee_id = old_employee.employee_id
JOIN employees new_employee ON r.new_employee_id = new_employee.employee_id
JOIN jobs j ON old_employee.job_id = j.job_id;
```