Hicham ROLDAN

Création de la BDD via Workbench

Mise en place des requêtes SQL nécessaires

```sql
CREATE DATABASE IF NOT EXISTS compagnie_creole;

USE compagnie_creole;

CREATE TABLE services (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    office_number VARCHAR(20) DEFAULT NULL
);

CREATE TABLE employees (
    id INT AUTO_INCREMENT PRIMARY KEY,
    first_name VARCHAR(50) NOT NULL,
    last_name VARCHAR(50) NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL,
    salary DECIMAL(10, 2) NOT NULL,
    service_id INT,
    FOREIGN KEY (service_id) REFERENCES services(id) ON DELETE SET NULL
);

CREATE TABLE manage (
    service_id INT NOT NULL,
    employee_id INT NOT NULL,
    start_date DATE NOT NULL,
    PRIMARY KEY (service_id, employee_id),
    FOREIGN KEY (service_id) REFERENCES services(id) ON DELETE CASCADE,
    FOREIGN KEY (employee_id) REFERENCES employees(id) ON DELETE CASCADE
);
```

Remplissage de données pour les colonnes de la BDD

```sql
INSERT INTO services (name, office_number) VALUES
('Cuisine Centrale', 'CC101'),
('Département Fromage', 'DF202'),
('Boulangerie Pâtisserie', 'BP303'),
('Pôle Charcut', 'PC404'),
('Équipe Sauce Secrète', 'ESS505');

INSERT INTO employees (first_name, last_name, email, salary, service_id) VALUES
('Brie', 'DeMeaux', 'brie.demeaux@compagniegourmande.com', 45000.00, 2),
('Camille', 'Embert', 'camille.embert@compagniegourmande.com', 50000.00, 1),
('Jean', 'Bon', 'jean.bon@compagniegourmande.com', 55000.00, 3),
('Tom', 'Ato', 'tom.ato@compagniegourmande.com', 60000.00, 5),
('Pat', 'ÉChou', 'pat.echou@compagniegourmande.com', 48000.00, 4),
('Alain', 'Proviste', 'alain.proviste@compagniegourmande.com', 53000.00, 1),
('Charlotte', 'Auxfraises', 'charlotte.auxs@compagniegourmande.com', 57000.00, 3),
('Léa', 'Roux', 'lea.roux@compagniegourmande.com', 62000.00, 4),
('Michou', 'Croute', 'michou.croute@compagniegourmande.com', 52000.00, 5),
('Max', 'Lamelasse', 'max.melasse@compagniegourmande.com', 67000.00, 2);

INSERT INTO manage (service_id, employee_id, start_date) VALUES
(1, 2, '2021-01-15'),  -- Camille Embert gère Cuisine Centrale
(2, 1, '2020-03-20'),  -- Brie DeMeaux gère Département Fromage
(3, 3, '2021-07-10'),  -- Jean Bon gère Boulangerie R&D
(4, 8, '2022-06-01'),  -- Léa Roux gère Pôle Charcuterie
(5, 4, '2022-12-25');  -- Tom Ato gère Équipe Sauce Secrète
```

J'ai tout à fait j'ai à cœur de créer les requêtes permettant de trouver les informations suivantes :

```sql
-- Nombre total d'employés
SELECT COUNT(*) AS total_employees
FROM employees;
```

Result Grid | Filter Rows: | Export:

total_employees
10

```sql
-- Moyenne des salaires dans l'entreprise
SELECT AVG(salary) AS average_salary
FROM employees;
```

Result Grid | Filter Rows: | Export:

average_salary
54900.000000

```sql
-- Moyenne des salaires par service
SELECT
    s.name AS service_name,
    ROUND(AVG(e.salary), 2) AS average_salary
FROM employees e
LEFT JOIN services s ON e.service_id = s.id
GROUP BY s.name;
```

Result Grid | Filter Rows: | Export:

service_name	average_salary
Département Fromage	56000.00
Cuisine Centrale	51500.00
Boulangerie Pâtisserie	56000.00
Équipe Sauce Secrète	56000.00
Pôle Charcut	55000.00

```sql
-- Classement du nb d'employés par service
SELECT
    s.name AS service_name,
    COUNT(e.id) AS employee_count
FROM services s
LEFT JOIN employees e ON s.id = e.service_id
GROUP BY s.name
ORDER BY employee_count DESC;
```

Result Grid | Filter Rows: | Export:

service_name	employee_count
Cuisine Centrale	2
Département Fromage	2
Boulangerie Pâtisserie	2
Pôle Charcut	2
Équipe Sauce Secrète	2

```sql
-- Top 5 des services par masse salariale
SELECT
    s.name AS service_name,
    SUM(e.salary) AS total_salary
FROM services s
LEFT JOIN employees e ON s.id = e.service_id
GROUP BY s.name
ORDER BY total_salary DESC
LIMIT 5;
```

Result Grid | Filter Rows: | Export:

service_name	total_salary
Département Fromage	112000.00
Boulangerie Pâtisserie	112000.00
Équipe Sauce Secrète	112000.00
Pôle Charcut	110000.00
Cuisine Centrale	103000.00

```sql
-- Liste des managers et leurs services
SELECT
    CONCAT(e.first_name, ' ', e.last_name) AS manager_name,
    s.name AS service_name,
    m.start_date AS management_start_date
FROM manage m
JOIN employees e ON m.employee_id = e.id
JOIN services s ON m.service_id = s.id
ORDER BY m.start_date ASC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

manager_name	service_name	management_start_date
Brie DeMeaux	Département Fromage	2020-03-20
Camille Embert	Cuisine Centrale	2021-01-15
Jean Bon	Boulangerie Pâtisserie	2021-07-10
Léa Roux	Pôle Charcut	2022-06-01
Tom Ato	Équipe Sauce Secrète	2022-12-25

```sql
-- Fonction permettant de calculer l'écart entre + gros et + petit salaire
DELIMITER $$

CREATE FUNCTION SalaryGap()
RETURNS DECIMAL(10, 2)
BEGIN
    DECLARE max_salary DECIMAL(10, 2);
    DECLARE min_salary DECIMAL(10, 2);
    DECLARE difference DECIMAL(10, 2);

    SELECT MAX(salary) INTO max_salary FROM employees;
    SELECT MIN(salary) INTO min_salary FROM employees;

    SET difference = max_salary - min_salary;

    RETURN difference;
END $$

DELIMITER ;

SELECT SalaryGap() AS salary_difference;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

salary_difference
22000.00