

1 Activité: Recherchez des données à l'aide de requêtes SQL

1.1 Objectifs

Le but de cette activité est de rédiger des requêtes sur une base de données afin de répondre à des questions. La base de données modélise une entreprise, dans laquelle des personnes sont employées dans différents départements.

1.2 Contexte

Elle contient 6 tables : - *employees* : les employés - *titles* : le titre sous lequel un employé est embauché - *salaries* : les salaires des employés - *departments* : les départements de l'entreprise - *dept_manager* : les managers (chefs) de départements - *dept_emp* : les employés associés à un département donné

Ces tables sont résumées dans ce schéma :

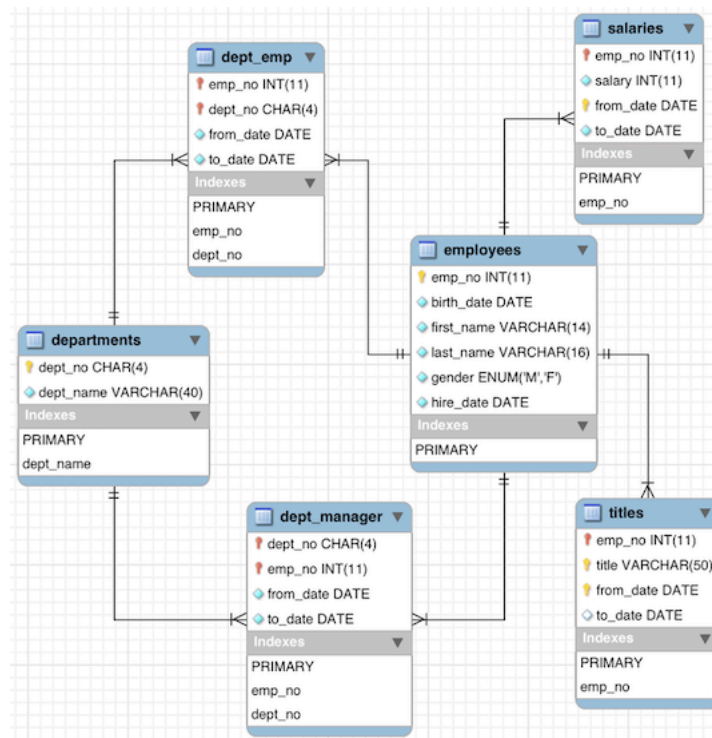


Figure 1: Schéma de la base de données

1.3 Consigne

Vous accéderez à la base de données via cette console interactive.

Il est conseillé de garder une copie des requêtes que vous rédigez sur votre ordinateur, car la mémoire de la console interactive se réinitialise si la page web est rafraîchie.

Pour chaque question, vous fournirez la requête SQL vous ayant permis d'obtenir le résultat, ainsi qu'une capture d'écran de la table renvoyée par votre requête. Si le résultat d'une requête contient beaucoup de ligne, seule une capture d'écran des premières lignes suffit.

1.4 Questions

1. Affichez toutes les lignes de la table *employees*, en n'affichant que les noms et prénoms des employés.
2. Afficher tous les employés ayant été embauché après le 1er août 1999 (non compris).
3. Dans cette base de données, deux tables ont le même schéma. Réaliser leur union.

4. Afficher les différents salaires qu'a eu l'employé dont l'identifiant est **499593** grâce à une jointure.
5. Grâce à une jointure, produisez une table indiquant le ou les départements dans lesquels a travaillé l'employé dont l'identifiant est **499902** (la table doit contenir également les dates de début et de fin d'embauche dans chacun des départements).
6. En utilisant une agrégation, trouver combien de personnes ont pour nom de famille "Gewali" (le nom de famille est donné dans la colonne *last_name* de la table "*employees*").

1.5 Réponses

1.5.1 Question 1

Pour afficher les lignes de la table *employees*, on utilise la commande SELECT de la façon suivante:

```
SELECT last_name, first_name FROM employees;
```

La requête donne le résultat suivant:

The screenshot shows a SQL IDE interface. At the top, there are tabs for 'README' and 'query.sql'. The 'query.sql' tab is active, displaying the following SQL query:

```
1 SELECT last_name, first_name FROM employees;
2
```

Below the query editor, there is a tab labeled 'SQL View'. This tab displays the results of the query in a table format. The table has two columns: 'last_name' and 'first_name'. The results are as follows:

| last_name | first_name |
|-----------|------------|
| Lowrie | Mihalis |
| Kawashimo | Nalini |
| Kawashima | Serenella |
| Wixon | Moti |
| Gewali | Morrie |
| Mawatari | Kellie |
| Pews | Jaideep |
| Pews | Weidon |
| Gewali | Kazuhisa |
| Kawashima | Maia |
| Sewelson | Sugwoo |
| Mawatari | Miomir |

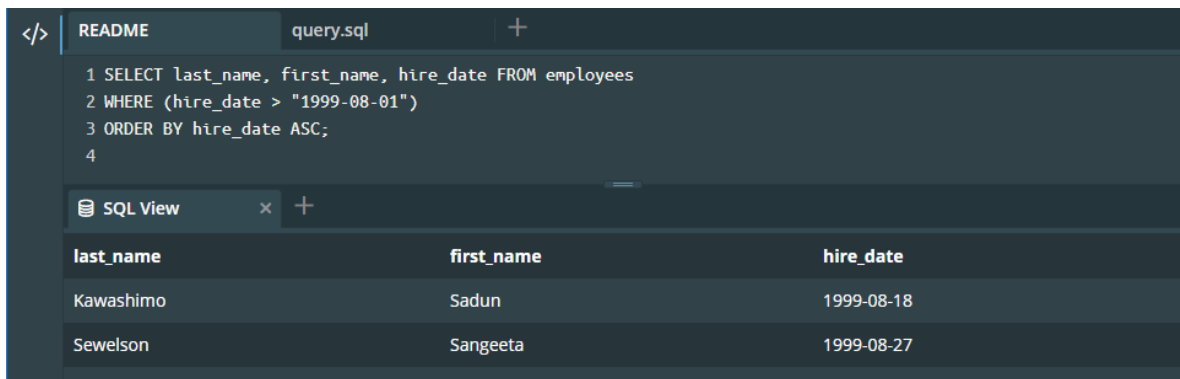
Figure 2: Résultat de la question 1.

1.5.2 Question 2

Pour afficher tous les employés ayant été embauché après le 1er aout 1999:

```
SELECT last_name, first_name, hire_date FROM employees
WHERE (hire_date > "1999-08-01")
ORDER BY hire_date ASC;
```

La requête donne le résultat suivant:



The screenshot shows a SQL IDE interface. At the top, there are tabs for 'README' and 'query.sql'. The 'query.sql' tab is active, displaying the following SQL query:

```
1 SELECT last_name, first_name, hire_date FROM employees
2 WHERE (hire_date > "1999-08-01")
3 ORDER BY hire_date ASC;
4
```

Below the query editor, there is a tab labeled 'SQL View'. The results of the query are displayed in a table with three columns: last_name, first_name, and hire_date. The results are ordered by hire_date in ascending order.

| last_name | first_name | hire_date |
|-----------|------------|------------|
| Kawashimo | Sadun | 1999-08-18 |
| Sewelson | Sangeeta | 1999-08-27 |

Figure 3: Résultat de la question 2.

Note: la commande `ORDER BY hire_date ASC` permet d'ordonner les résultats de la requête.

1.5.3 Question 3

Les deux tables ayant le schéma sont les tables: dept_emp et dept_manager

```
SELECT emp_no, dept_no, from_date, to_date FROM dept_emp
UNION
SELECT emp_no, dept_no, from_date, to_date FROM dept_manager;
```

La requête donne le résultat suivant:

| <div> <div></></div> <div>README</div> <div>query.sql</div> <div>+</div> </div> | | | |
|---|---------|------------|------------|
| <pre> 1 SELECT emp_no, dept_no, from_date, to_date FROM dept_emp 2 UNION 3 SELECT emp_no, dept_no, from_date, to_date FROM dept_manager; 4 </pre> | | | |
| <div> <div>SQL View</div> <div>×</div> <div>+</div> </div> | | | |
| emp_no | dept_no | from_date | to_date |
| 10184 | d009 | 1997-12-26 | 9999-01-01 |
| 10264 | d005 | 1997-07-16 | 9999-01-01 |
| 10328 | d005 | 1998-07-17 | 9999-01-01 |
| 10605 | d006 | 1993-07-01 | 9999-01-01 |
| 10735 | d009 | 1985-10-29 | 9999-01-01 |
| 11217 | d004 | 1990-03-11 | 9999-01-01 |
| 11245 | d004 | 1994-11-25 | 9999-01-01 |
| 11257 | d005 | 1992-02-16 | 1998-12-11 |
| 11257 | d008 | 1990-07-13 | 1992-02-16 |
| 11336 | d004 | 1998-09-01 | 2001-04-14 |
| 11542 | d003 | 1985-06-20 | 9999-01-01 |
| 11780 | d004 | 1999-11-10 | 9999-01-01 |
| 11780 | d005 | 1987-09-11 | 1999-11-10 |

Figure 4: Résultat de la question 3.

1.5.4 Question 4

```

SELECT e.first_name, e.last_name, e.emp_no, s.salary FROM employees e
JOIN salaries s
ON (e.emp_no = s.emp_no) AND (e.emp_no = 499593);

```

La requête donne le résultat suivant:

| <div> <div></></div> <div>README</div> <div>query.sql</div> <div>+</div> </div> | | | |
|--|-----------|--------|--------|
| <pre> 1 SELECT e.first_name, e.last_name, e.emp_no, s.salary FROM employees e 2 JOIN salaries s 3 ON (e.emp_no = s.emp_no) AND (e.emp_no = 499593); 4 </pre> | | | |
| <div> <div>SQL View</div> <div>×</div> <div>+</div> </div> | | | |
| first_name | last_name | emp_no | salary |
| Mantis | Kawashima | 499593 | 58087 |
| Mantis | Kawashima | 499593 | 57888 |
| Mantis | Kawashima | 499593 | 58710 |

Figure 5: Résultat de la question 4.

1.5.5 Question 5

SELECT

```

e.first_name,
e.last_name,
d.dept_name,
de.from_date,
de.to_date

```

FROM

```

employees e,
dept_emp de,
departments d

```

WHERE

```

e.emp_no = de.emp_no
AND de.dept_no = d.dept_no
AND e.emp_no = 499902 ;

```

La requête donne le résultat suivant:

| <div> <div></></div> <div>README</div> <div>query.sql</div> <div>+</div> </div> | | | | |
|--|-----------|------------------|------------|------------|
| <pre> 1 SELECT 2 e.first_name, 3 e.last_name, 4 d.dept_name, 5 de.from_date, 6 de.to_date 7 FROM 8 employees e, 9 dept_emp de, 10 departments d 11 WHERE 12 e.emp_no = de.emp_no 13 AND de.dept_no = d.dept_no 14 AND e.emp_no = 499902 15 </pre> | | | | |
| <div> <div>SQL View</div> <div>×</div> <div>+</div> </div> | | | | |
| first_name | last_name | dept_name | from_date | to_date |
| Aloke | Wuwongse | Sales | 1999-08-02 | 2000-11-28 |
| Aloke | Wuwongse | Customer Service | 2000-11-28 | 9999-01-01 |

Figure 6: Résultat de la question 5.

1.5.6 Question 6

```

SELECT * FROM
  (SELECT last_name, COUNT(*)
   FROM employees
   GROUP BY last_name) t
WHERE t.last_name="Gewali";

```

La requête donne le résultat suivant:

| <div> <div></></div> <div>README</div> <div>query.sql</div> <div>+</div> </div> | |
|---|----------|
| <pre> 1 SELECT * FROM 2 (SELECT last_name, COUNT(*) 3 FROM employees 4 GROUP BY last_name) t 5 WHERE t.last_name="Gewali"; 6 </pre> | |
| <div> <div>SQL View</div> <div>×</div> <div>+</div> </div> | |
| last_name | COUNT(*) |
| Gewali | 209 |

Figure 7: Résultat de la question 6.

Notons qu'on peut obtenir un résultat identique plus simplement ... mais sans agrégation avec la requête suivante:

```
SELECT last_name, COUNT(*) FROM employees
WHERE last_name = "Gewali";
```