

Secteur Tertiaire Informatique
Filière « Etude et développement »

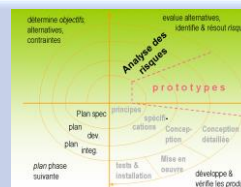
TP – Initiation requêtage SQL

BDD – requêtage SQL

Apprentissage

Mise en situation

Evaluation



Objectif

A l'issue de cette activité, vous serez capable de requêter une base de données en utilisant le langage SQL (« Structured Query Language »), en particulier les opérations de manipulation des données (« DML » pour « Data Manipulation Language »).

Objectifs pédagogiques :

- Comprendre la structure simple d'une base de données
- Effectuer des requête « SELECT »
- Manipuler les paramètres des clauses « SELECT »

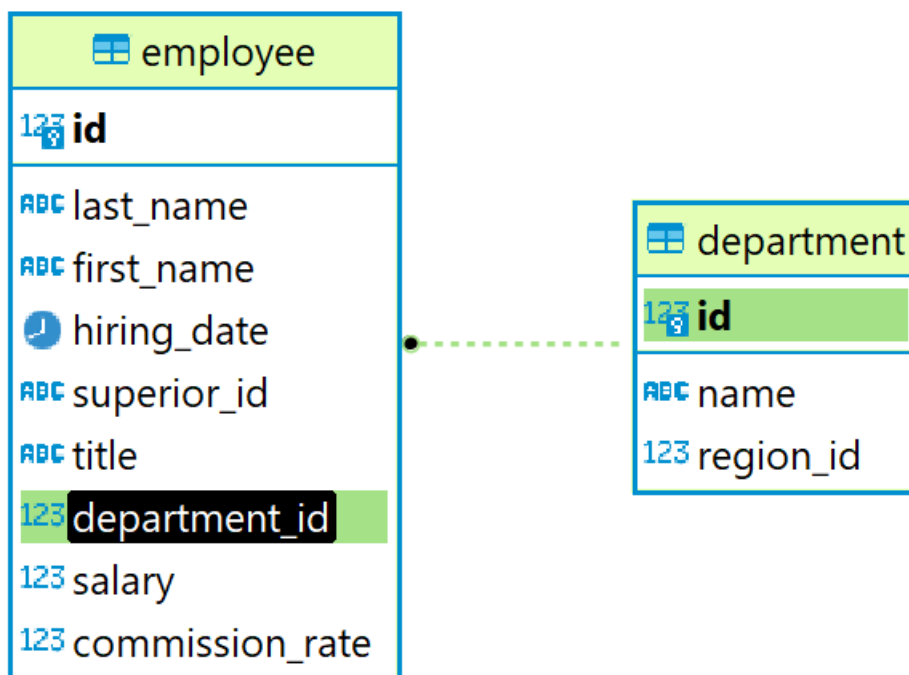
1. STRUCTURE DE LA BASE DE DONNEES

Cette base représente d'une manière simpliste (et remarquablement incomplète) le personnel d'une entreprise. Elle n'est pas destinée à être réaliste, mais seulement à être simple à comprendre pour une première approche de SQL.

Elle est composée de deux tables :

- La table « employee » contient les informations sur les employés
- La table « department » recense les différents départements de l'entreprise.

Ci-dessous un schéma « entité-relation » présentant la structure de la base de données :



1.1 DICTIONNAIRE DE DONNEES

Ci-dessous les détails concernant les différents champs des tables.

1.1.1 Table « employee »

id :

L'identifiant de la personne. Chaque employé a un « id », tous les « id » sont différents et uniques. Il s'agit d'une **clef primaire** (« primary key »).

last_name :

Le nom de la personne.

first_name :

Le prénom de la personne.

hiring_date :

Sa date d'embauche.

superior_id :

L'identifiant de son supérieur hiérarchique.

Par exemple, Sophie Fonfec (dont l'identifiant est 17) a 6 pour « superior_id », ce qui signifie que l'identifiant de son supérieur hiérarchique est 6.

title :

La fonction de l'employé (directeur, secrétaire, représentant, etc.).

department_id :

Le numéro du département (dans l'entreprise) dont dépend l'employé. Ce numéro fait référence à la clef primaire de la table « department ». C'est une **clef étrangère** (foreign key).

salary :

Salaire mensuel net de l'employé.

commission_rate :

Taux de commission, exprimé en pourcentage.

1.1.2 Table « department »

id :

L'identifiant du département. Chaque département a un identifiant, tous les identifiants sont différents. C'est une clef primaire (primary key).

name :

Le nom du département. On peut remarquer que des départements différents peuvent avoir le même nom.

region_id :

Référence à une région. Ce numéro fait référence à la clef primaire d'une table qui ne sera pas décrite ici. C'est une **clef étrangère** (foreign key).

2. REQUETE « SELECT »

La commande de recherche est le verbe SELECT.

Dans un premier temps, l'étude de la commande SELECT et des différentes clauses et fonctions est faite en ne considérant qu'une seule table.

Par la suite, l'utilisation de plusieurs tables dans le même SELECT, sera prise en compte.

La syntaxe de base est :

```
SELECT colonnes d'une ou plusieurs tables séparées par « , »  
FROM tables séparées par « , »  
WHERE conditions logiques séparées par « AND » ou « OR »  
ORDER BY colonnes séparées par « , »
```

Le caractère « * » permet de demander l'affichage de toutes les colonnes d'une table.

Par exemple, pour obtenir toutes les lignes issues de la table « employee » :

```
SELECT * FROM employee;
```

3. REQUETES A ECRIRE

Dans la suite de ce TP vous allez découvrir une partie des fonctionnalités offertes par le langage SQL. Chaque requête à écrire sera indiqué par un numéro. Répertoriez toutes vos requêtes dans un fichier « .sql ».

3.1 SELECT

1. Afficher toutes les informations concernant les départements

A la place de « * » on peut donner la liste des colonnes souhaitées, dans l'ordre souhaité, en écrivant la requête sur une ou plusieurs lignes.

L'écriture sur plusieurs lignes est conseillée en vue de rendre plus simple la relecture et la modification de la requête.

2. Afficher le nom, la date d'embauche, le numéro du supérieur, le numéro de département et le salaire de tous les employés.

Le titre des colonnes obtenues par un SELECT sont les noms des champs.

On peut cependant changer ces titres en utilisant « AS ».

La requête

```
SELECT last_name AS "Nom" FROM employee;
```

produira le même effet que

```
SELECT last_name FROM employee;
```

la différence que la colonne ne sera pas intitulé « last_name » mais « Nom ».

3.2 RESULTAT SANS DOUBLON

Il peut être utile de supprimer les doublons, d'où utilisation de la clause « **DISTINCT** ». Elle a pour effet de n'afficher qu'une seule fois les lignes qui seraient semblables à l'affichage.

On l'utilise sous la forme :

```
SELECT DISTINCT <suite-requete>;
```

3. Afficher le titre de tous les employé.e.s
4. Afficher les différentes valeurs des titres des employés

3.3 REQUETES AVEC RESTRICTIONS

Les **clauses de restriction** s'écrivent derrière le mot clef « **WHERE** ». Elles permettent de sélectionner les lignes à afficher.

Voici la structure d'une requête faisant figurer une clause de restriction :

```
SELECT colonnes d'une ou plusieurs tables séparées par « , »  
FROM tables séparées par « , »  
WHERE conditions logiques séparées par « AND » ou « OR »
```

Les conditions logiques de la clause de restriction peuvent utiliser les tous les opérateurs suivants :

Opérateur	Description	Exemple
=	Egalité : Vérifie si la valeur des deux opérandes est égale. Retourne « VRAI » si c'est le cas.	5 = 5 retourne VRAI
!=	Différence : Vérifie si la valeur des deux opérandes est différente. Retourne « VRAI » si c'est le cas.	5 != 6 retourne VRAI
<>	Différence (syntaxe alternative) : Même chose que « != » avec une autre syntaxe.	5 <> 4 retourne VRAI
>	Plus grand que : Vérifie si l'opérande de gauche est plus grand que l'opérande de droite.	4 > 5 retourne VRAI
<	Plus petit que : Vérifie si l'opérande de gauche est plus petit que l'opérande de droite.	4 < 5 retourne VRAI
>=	Plus grand ou égal : Vérifie si l'opérande de gauche est plus grand ou égal à	4 >= 5 retourne

	l'opérande de droite.	VRAI
<=	Plus petit ou égal : Vérifie si l'opérande de gauche est plus petit ou égal à l'opérande de droite.	4 <= 5 retourne VRAI
LIKE	Egalité pour les chaînes des caractères :	'Bonjour' LIKE 'Aurevoir' retourne FAUX

Pour plus d'informations sur la clause « **WHERE** » : <https://librecours.net/module/bdd0/sql-select/restriction.xhtml>

5. Afficher toutes les informations des salariés ayant un salaire supérieur à 25000.
6. Afficher le nom, le numéro d'employé et le numéro du département des employés dont le titre est « Secrétaire ».
7. Afficher le nom et le numéro de département dont le numéro de département est supérieur à 40.

3.4 RESTRICTION EN COMPARANT LES COLONNES ENTRE ELLES

La restriction peut mettre en jeu la comparaison de deux ou plusieurs colonnes entre elles. Il suffit de les appeler par leurs noms.

8. Afficher le nom et le prénom des employés dont le nom est alphabétiquement antérieur au prénom.
9. Afficher le nom, le salaire et le numéro du département des employés dont le titre est « Représentant », le numéro de département est 35 et le salaire est supérieur à 20000
10. Afficher le nom, le titre et le salaire des employés dont le titre est « Représentant » ou dont le titre est « Président ».
11. Afficher le nom, le titre, le numéro de département, le salaire des employés du département 34, dont le titre est « Représentant » ou « Secrétaire ».
12. Afficher le nom, le titre, le numéro de département, le salaire des employés dont le titre est Représentant, ou dont le titre est « Secrétaire » dans le département numéro 34.
13. Afficher le nom, et le salaire des employés dont le salaire est compris entre 20000 et 30000.

3.5 NEGATION ET RECHERCHE APPROCHEE

Les opérateurs arithmétiques sont niés avec « ! » : par exemple « non égal » s'écrit « != » (ou « <> »).

Les autres opérateurs sont niés par « NOT » : par exemple « pas dans » s'écrit « NOT IN ».

Les caractères jokers sont « % » pour une chaîne et « _ » pour un caractère. On les utilise avec l'opérateur LIKE.

- 14. Afficher le nom des employés commençant par la lettre « H ».**
- 15. Afficher le nom des employés se terminant par la lettre « n ».**
- 16. Afficher le nom des employés contenant la lettre « u » en 3ème position.**

3.6 TRI DES RESULTATS

La clause « ORDER BY » permet de trier les résultats en fonction d'une colonne. Vous pouvez vous référer à la ressource suivante pour une compréhension de son fonctionnement : <https://sql.sh/cours/order-by>

- 17. Afficher le salaire et le nom des employés du service 41 classés par salaire croissant.**
- 18. Afficher le salaire et le nom des employés du service 41 classés par salaire décroissant.**
- 19. Afficher le titre, le salaire et le nom des employés classés par titre croissant et par salaire décroissant.**

3.7 VALEURS NON RENSEIGNEES

En SQL il existe une valeur correspondant à la valeur vide. C'est l'équivalent de la valeur « Null » de certains langages de programmation (Java, C++, Python). C'est la valeur que prennent les champs non renseignés.

Elle s'appelle également « NULL ». Cette valeur n'est pas prise en compte dans les classements.

- 20. Afficher le taux de commission, le salaire et le nom des employés classés par taux de commission croissant.**

Si on veut utiliser les valeurs non renseignées dans une restriction ce sera avec les clauses « IS NULL » ou « IS NOT NULL ».

- 21. Afficher le nom, le salaire, le taux de commission et le titre des employés dont le taux de commission n'est pas renseigné.**
- 22. Afficher le nom, le salaire, le taux de commission et le titre des employés dont le taux de commission est renseigné.**
- 23. Afficher le nom, le salaire, le taux de commission, le titre des employés dont le taux de commission est inférieur à 15.**
- 24. Afficher le nom, le salaire, le taux de commission, le titre des employés dont le taux de commission est supérieur à 15.**

3.8 EXPRESSIONS ARITHMETIQUES

Des expressions arithmétiques peuvent être utilisées après : `SELECT`, `WHERE`, et `ORDER BY`.

- 25. Afficher le nom, le salaire, le taux de commission et la commission des employés dont le taux de commission n'est pas nul. (la commission est calculée en multipliant le salaire par le taux de commission).**
- 26. Afficher le nom, le salaire, le taux de commission, la commission des employés dont le taux de commission n'est pas nul, classé par taux de commission croissant.**

3.9 CONCATENATION

La fonction qui permet la concaténation de chaînes de caractères est [`CONCAT`](#)

- 27. Afficher le nom et le prénom (concaténés) des employés. Renommer les colonnes.**

3.10 CHAINES DE CARACTERE

La fonction extraction de chaîne de caractères « [`SUBSTRING\(expr, p, n\)`](#) » permet d'extraire un sous-chaîne commençant à la position « p » et de taille « n » de « `expr` ».

La fonction « `POSITION(search_string IN main_string)` » permet de retrouver l'index du début de la chaîne « `search_string` » dans « `main_string` ».

Les fonctions « [`UPPER\(str\)`](#) » et « [`LOWER\(str\)`](#) » permettent respectivement de forcer à la majuscule ou à la minuscule.

La fonction « [`LENGTH\(arg1\)`](#) » permet d'obtenir le nombre de caractères d'une chaîne de caractères.

- 28. Afficher les 5 premières lettres du nom des employés**
- 29. Afficher le nom et le rang de la lettre « r » dans le nom des employés.**
- 30. Afficher le nom, le nom en majuscule et le nom en minuscule de l'employé dont le nom est « Vrante ».**
- 31. Afficher le nom et le nombre de caractères du nom des employés.**

CREDITS

ŒUVRE COLLECTIVE DE l'AFPA

Sous le pilotage de la DIIP et du centre d'ingénierie sectoriel Tertiaire-Services

Equipe de conception (IF, formateur, mediatiseur)

Michel Coulard – Formateur Evry

Chantal Perrachon – IF Neuilly sur Marne

Date de mise à jour : 10/09/2023

Reproduction interdite

Article L 122-4 du code de la propriété intellectuelle.

« Toute représentation ou reproduction intégrale ou partielle faite sans le consentement de l'auteur ou de ses ayants droits ou ayants cause est illicite. Il en est de même pour la traduction, l'adaptation ou la reproduction par un art ou un procédé quelconque. »