TP 02 Langage SQL

Thème 2 : Base de données

02

Résumé cours : langage SQL

Programme Terminale

| Contenus | Capacités attendues | Commentaires |
|-------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|
| Langage SQL : requête d'interrogation et de mise à jour d'une base de données | Identifier les composants d'une requête. Construire des requêtes 'interrogation à l'aide des clauses du langage SQL : SELECT, FROM, WHERE, JOIN Construire des requêtes d'insertion et de mise à jour à l'aide de : UPDATE, INSERT, DELETE | On peut utiliser DISTINCT, ORDER BY OU les fonctions d'agrégation sans utiliser GROUP BY et HAVING |



Voir TP sur Capytal

1. Du modèle relationnel au SGBD

Nous allons maintenant d'aborder la partie logicielle : les SGBD (Systèmes de Gestion de Bases de Données).

Les SGBD jouent le rôle d'interface entre l'être humain et la base de données.

Par l'intermédiaire de **requêtes**, l'utilisateur va consulter ou modifier la base de données. Le SGBD est garant de l'intégrité de cette base, et prévient notamment que les modifications ne soient pas préjudiciables à la base de données.

Le langage utilisé pour communiquer avec le SGBD est le langage **SQL**, pour Structured Query Langage (pour *langage de requêtes structurées*).

Les SGBD les plus utilisés sont basés sur le modèle relationnel. Parmi eux, citons Oracle, MySQL, Microsoft SQL Server, PostgreSQL, Microsoft Access, SQLite, MariaDB...

Mais de plus en plus de SGBD **non-relationnels** sont utilisés, spécialement adaptés à des données plus diverses et moins structurées. On les retrouve sous l'appelation **NoSQL** (pour *Not only SQL*). Citons parmi eux MongoDB,

Cassandra (Facebook), BigTable (Google)...

La quasi-totalité de ces SGBD fonctionnent avec un modèle client-serveur.

Nous allons travailler principalement avec le langage SQLite peut lui s'utiliser directement sans démarrer un serveur : la base de données est entièrement représentée dans le logiciel utilisant SQLite.

Création de tables

```
DROP TABLE IF EXISTS LIVRES;
CREATE TABLE LIVRES
(code INT, titre TEXT, auteur TEXT, ann_publi INT, note INT, PRIMARY KEY (code));
```

1. Création de la tables LIVRES

```
Requête SQL
INSERT INTO LIVRES
(code, titre, auteur, ann_publi, note)
VALUES
(1, '1984', 'Orwell', 1949, 10),
(2, 'Dune', 'Herbert', 1965, 8),
(3, 'Fondation', 'Asimov', 1951, 9),
(4, 'Le meilleur des mondes', 'Huxley', 1931, 7),
(5, 'Fahrenheit 451', 'Bradbury', 1953, 7),
(6, 'Ubik', 'K.Dick', 1969, 9),
(7, 'Chroniques martiennes', 'Bradbury', 1950, 8),
(8, 'La nuit des temps', 'Barjavel', 1968, 7),
(9, 'Blade Runner', 'K.Dick', 1968, 8),
(10, 'Les Robots', 'Asimov', 1950, 9),
(11, 'La Planète des singes', 'Boulle', 1963, 8),
(12, 'Ravage', 'Barjavel', 1943, 8),
(13, 'Le Maître du Haut Château', 'K.Dick', 1962, 8),
(14, 'Le monde des \bar{A}', 'Van Vogt', 1945, 7),
(15, 'La Fin de l'éternité', 'Asimov', 1955, 8),
(16, 'De la Terre à la Lune', 'Verne', 1865, 10);
```

2. Création de la table AUTEURS

```
DROP TABLE IF EXISTS AUTEURS;
CREATE TABLE AUTEURS
(id INT, nom TEXT, prenom TEXT, ann_naissance INT, langue_ecriture TEXT, PRIMARY KEY (id)
);
```

```
Requête SQL
```

```
INSERT INTO AUTEURS
(id, nom, prenom, ann_naissance, langue_ecriture)
```

```
VALUES
(1,'Orwell','George',1903,'anglais'),
(2,'Herbert','Frank',1920,'anglais'),
(3,'Asimov','Isaac',1920,'anglais'),
(4,'Huxley','Aldous',1894,'anglais'),
(5,'Bradbury','Ray',1920,'anglais'),
(6,'K.Dick','Philip',1928,'anglais'),
(7,'Barjavel','René',1911,'français'),
(8,'Boulle','Pierre',1912,'français'),
(9,'Van Vogt','Alfred Elton',1912,'anglais'),
(10,'Verne','Jules',1828,'français');
```

3. Sélection de données

1. → Requête basique : SELECT, FROM

```
Requête SQL

SELECT *
FROM LIVRES
```

| code | titre | id_auteur | ann_publi | note |
|------|---------------------------|-----------|-----------|------|
| 1 | 1984 | 1 | 1949 | 10 |
| 2 | Dune | 2 | 1965 | 8 |
| 3 | Fondation | 3 | 1951 | 9 |
| 4 | Le meilleur des mondes | 4 | 1931 | 7 |
| 5 | Fahrenheit 451 | 5 | 1953 | 7 |
| 6 | Ubik | 6 | 1969 | 9 |
| 7 | Chroniques martiennes | 5 | 1950 | 8 |
| 8 | La nuit des temps | 7 | 1968 | 7 |
| 9 | Blade Runner | 6 | 1968 | 8 |
| 10 | Les Robots | 3 | 1950 | 9 |
| 11 | La Planète des singes | 8 | 1963 | 8 |
| 12 | Ravage | 7 | 1943 | 8 |
| 13 | Le Maître du Haut Château | 6 | 1962 | 8 |
| 14 | Le monde des Ā | 9 | 1945 | 7 |
| 15 | La Fin de l'éternité | 3 | 1955 | 8 |
| 16 | De la Terre à la Lune | 10 | 1865 | 10 |

Requête SQL

```
SELECT titre, auteur, note FROM LIVRES
```

| | titre | auteur | note |
|------|-------|--------|------|
| 1984 | | Orwell | 10 |

| titre | auteur | note |
|---------------------------|----------|------|
| Dune | Herbert | 8 |
| Fondation | Asimov | 9 |
| Le meilleur des mondes | Huxley | 7 |
| Fahrenheit 451 | Bradbury | 7 |
| Ubik | K.Dick | 9 |
| Chroniques martiennes | Bradbury | 8 |
| La nuit des temps | Barjavel | 7 |
| Blade Runner | K.Dick | 8 |
| Les Robots | Asimov | 9 |
| La Planète des singes | Boulle | 8 |
| Ravage | Barjavel | 8 |
| Le Maître du Haut Château | K.Dick | 8 |
| Le monde des Ā | Van Vogt | 7 |
| La Fin de l'éternité | Asimov | 8 |
| De la Terre à la Lune | Verne | 10 |

2. ➡ Requête basique : SELECT, FROM, WHERE

🗀 Requête SQL

SELECT titre, ann_publi FROM LIVRES WHERE auteur='Asimov'

| titre | ann_publi |
|----------------------|-----------|
| Fondation | 1951 |
| Les Robots | 1950 |
| La Fin de l'éternité | 1955 |

Requête SQL

SELECT *
FROM LIVRES
WHERE auteur='Asimov'

| code | titre | auteur | ann_publi | note |
|------|----------------------|--------|-----------|------|
| 3 | Fondation | Asimov | 1951 | 9 |
| 10 | Les Robots | Asimov | 1950 | 9 |
| 15 | La Fin de l'éternité | Asimov | 1955 | 8 |

☐ Requête SQL

```
SELECT auteur, titre, ann_publi
FROM LIVRES
WHERE auteur='Asimov' AND note>=9
```

| auteur | titre | ann_publi |
|--------|------------|-----------|
| Asimov | Fondation | 1951 |
| Asimov | Les Robots | 1950 |

4. 🗫 Renommage : AS

Pour rendre l'affichage plus "lisible" on peut renommer les colonnes : AS

• Commande:

```
Requête SQL

SELECT titre, auteur, ann_publi AS publication
FROM LIVRES
WHERE ann_publi >= 1945;
```

• Traduction:

Lors de l'affichage du résulats et dans la suite de la requête (important), la colonne "ann_publi" est renommée "publication".

☐ Requête SQL

```
SELECT titre, auteur, ann_publi AS publication
FROM livres
WHERE ann_publi >= 1945;
```

| titre | auteur | publication |
|---------------------------|----------|-------------|
| 1984 | Orwell | 1949 |
| Dune | Herbert | 1965 |
| Fondation | Asimov | 1951 |
| Fahrenheit 451 | Bradbury | 1953 |
| Ubik | K.Dick | 1969 |
| Chroniques martiennes | Bradbury | 1950 |
| La nuit des temps | Barjavel | 1968 |
| Blade Runner | K.Dick | 1968 |
| Les Robots | Asimov | 1950 |
| La Planète des singes | Boulle | 1963 |
| Le Maître du Haut Château | K.Dick | 1962 |
| Le monde des Ā | Van Vogt | 1945 |
| La Fin de l'éternité | Asimov | 1955 |

1. • Mettre dans l'ordre les réponses la clause ORDER BY

Il est aussi possible de rajouter la clause SQL ORDER BY afin d'obtenir les résultats classés dans un ordre précis.

Requête SQL SELECT titre FROM LIVRES WHERE auteur='K.Dick' ORDER BY ann_publi

titre Le Maître du Haut Château Blade Runner Ubik

⚠ Remarques :"

• Comportement par défaut : Si le paramètre ASC ou DESC est omis, le classement se fait par ordre croissant (donc ASC est le paramètre par défaut).

5. 🕶 La clause DISTINCT

Il est possible d'éviter les doublons grâce à la clause DISTINCT

☐ Requête SQL

SELECT auteur FROM LIVRES

auteur

Orwell

Herbert

Asimov

Huxley

Bradbury

K.Dick

Bradbury

Barjavel

K.Dick

Asimov

Boulle

Barjavel

K.Dick

auteur Van Vogt Asimov

Verne

Requête SQL

SELECT DISTINCT auteur FROM LIVRES

auteur

Orwell

Herbert

Asimov

Huxley

Bradbury

K.Dick

Barjavel

Boulle

Van Vogt

Verne

1. ◆ La clause LIKE

On veut les titres de la table «livre» dont le titre contient la chaîne de caractères "Astérix". Le symbole % est un joker qui peut symboliser n'importe quelle chaîne de caractères.

Requête SQL

```
SELECT titre
FROM livres
WHERE titre LIKE 'F%';
```

permet d'obtenir les titres de livres commençant par F

🗀 Requête SQL

```
SELECT titre
FROM livres
WHERE titre LIKE '%s';
```

permet d'obtenir les titres de livres finissant par s

☐ Requête SQL

```
SELECT titre
FROM livres
```

```
WHERE titre LIKE 'F%';
```

titre

Fondation

Fahrenheit 451

Requête SQL

```
SELECT titre
FROM livres
WHERE titre LIKE '%s';
```

titre

Le meilleur des mondes

Chroniques martiennes

La nuit des temps

Les Robots

La Planète des singes

6. Opérations sur les données : sélection avec agrégation

Les requêtes effectuées jusqu'ici ont juste sélectionné des données grâce à différents filtres : aucune action à partir de ces données n'a été effectuée.

Nous allons maintenant effectuer des opérations à partir des données sélectionnées.

On appelle ces opérations des opérations d'agrégation.

7. 🗫 La clause COUNT

On veut compter le nombre d'enregistrements de la tables livres publiés en 1968.

Requête SQL

```
SELECT COUNT(*) AS total
FROM livres
WHERE ann_publi=1968;
```

total

8. 🔷 La clause : SUM - Additionner

• Commande:

```
Requête SQL

SELECT SUM(ann_publi) AS somme

FROM livres

WHERE auteur LIKE "F%";
```

· Traduction:

On veut additionner les années des livres de la tables livres commençant par F.

Le résultat sera le seul élément d'une colonne nommée «somme». Attention : dans notre cas précis, ce calcul n'a aucun sens...



9. 🔷 La clause : AVG - Moyenne''

On veut calculer la moyenne des notes des livres de la table livres de l'auteur "Bradbury". Le résultat sera le seul élément d'une colonne nommée «moyenne».

```
Requête SQL

SELECT AVG(note) AS note moyenne
FROM livres
WHERE auteur="Bradbury";
```

```
moyenne
7.5
```

1. ◆ La clause : MIN, MAX - Trouver les extremums:

• Commande:

```
Requête SQL

SELECT MIN(note) AS minimum

FROM livres;
```

| auteur | titre | minimum |
|--------|------------------------|---------|
| Huxley | Le meilleur des mondes | 7 |

10. Des recherches croisées sur les tables : les jointures

Nous avons 2 tables, grâce aux jointures nous allons pouvoir associer ces 2 tables dans une même requête.

Repartons sur la bases LIVRES légèrement modifiées.

Requête SQL

```
DROP TABLE IF EXISTS LIVRES;
CREATE TABLE LIVRES
(code INT, titre TEXT, id_auteur TEXT, ann_publi INT, note INT, PRIMARY KEY (code,id_auteur));
```

Requête SQL

```
INSERT INTO LIVRES
(code, titre, id_auteur, ann_publi, note)
VALUES
(1, '1984', 1, 1949, 10),
(2, 'Dune', 2, 1965, 8),
(3, 'Fondation', 3, 1951, 9),
(4, 'Le meilleur des mondes', 4, 1931, 7),
(5, 'Fahrenheit 451', 5, 1953, 7),
(6, 'Ubik', 6, 1969, 9),
(7, 'Chroniques martiennes', 5, 1950, 8),
(8, 'La nuit des temps', 7, 1968, 7),
(9, 'Blade Runner', 6, 1968, 8),
(10, 'Les Robots', 3, 1950, 9),
(11, 'La Planète des singes', 8, 1963, 8),
(12, 'Ravage', 7, 1943, 8),
(13, 'Le Maître du Haut Château', 6, 1962, 8),
(14, 'Le monde des \bar{A}', 9, 1945, 7),
(15, 'La Fin de l'éternité', 3, 1955, 8),
(16, 'De la Terre à la Lune', 10, 1865, 10);
```

1. → Jointures simples

En général, les jointures consistent à associer des lignes de 2 tables. Elles permettent d'établir un lien entre 2 tables.

• Commande:

```
Requête SQL

SELECT *
FROM LIVRES
INNER JOIN AUTEURS ON LIVRES.id_auteur = AUTEURS.id
```

• Traduction: Comme plusieurs tables sont appelées, nous préfixons chaque colonne avec le nom de la table.

| code | titre | id_auteur | ann_publi | note | id | nom | prenom | ann_naissance | langue_ecriture |
|------|------------------------|-----------|-----------|------|----|----------|--------|---------------|-----------------|
| 1 | 1984 | 1 | 1949 | 10 | 1 | Orwell | George | 1903 | anglais |
| 2 | Dune | 2 | 1965 | 8 | 2 | Herbert | Frank | 1920 | anglais |
| 3 | Fondation | 3 | 1951 | 9 | 3 | Asimov | Isaac | 1920 | anglais |
| 4 | Le meilleur des mondes | 4 | 1931 | 7 | 4 | Huxley | Aldous | 1894 | anglais |
| 5 | Fahrenheit 451 | 5 | 1953 | 7 | 5 | Bradbury | Ray | 1920 | anglais |
| 6 | Ubik | 6 | 1969 | 9 | 6 | K.Dick | Philip | 1928 | anglais |
| 7 | Chroniques martiennes | 5 | 1950 | 8 | 5 | Bradbury | Ray | 1920 | anglais |
| 8 | La nuit des temps | 7 | 1968 | 7 | 7 | Barjavel | René | 1911 | français |

| code | titre | id_auteur | ann_publi | note | id | nom | prenom | ann_naissance | langue_ecriture |
|------|---------------------------|-----------|-----------|------|----|----------|--------------|---------------|-----------------|
| 9 | Blade Runner | 6 | 1968 | 8 | 6 | K.Dick | Philip | 1928 | anglais |
| 10 | Les Robots | 3 | 1950 | 9 | 3 | Asimov | Isaac | 1920 | anglais |
| 11 | La Planète des singes | 8 | 1963 | 8 | 8 | Boulle | Pierre | 1912 | français |
| 12 | Ravage | 7 | 1943 | 8 | 7 | Barjavel | René | 1911 | français |
| 13 | Le Maître du Haut Château | 6 | 1962 | 8 | 6 | K.Dick | Philip | 1928 | anglais |
| 14 | Le monde des Ā | 9 | 1945 | 7 | 9 | Van Vogt | Alfred Elton | 1912 | anglais |
| 15 | La Fin de l'éternité | 3 | 1955 | 8 | 3 | Asimov | Isaac | 1920 | anglais |
| 16 | De la Terre à la Lune | 10 | 1865 | 10 | 10 | Verne | Jules | 1828 | français |

Des informations (id et id_auteur) sont en double.

On peut être plus précis.

Requête SQL

SELECT AUTEURS.nom, LIVRES.titre, LIVRES.note, AUTEURS.ann_naissance AS Naissance, LIVRES.ann_publi AS Publication, AUTEURS.langue_ecriture AS Langue FROM LIVRES

INNER JOIN AUTEURS ON LIVRES.id_auteur = AUTEURS.id

ORDER BY AUTEURS.nom

| nom | titre | note | Naissance | Publication | Langue |
|----------|---------------------------|------|-----------|-------------|----------|
| Asimov | Fondation | 9 | 1920 | 1951 | anglais |
| Asimov | Les Robots | 9 | 1920 | 1950 | anglais |
| Asimov | La Fin de l'éternité | 8 | 1920 | 1955 | anglais |
| Barjavel | La nuit des temps | 7 | 1911 | 1968 | français |
| Barjavel | Ravage | 8 | 1911 | 1943 | français |
| Boulle | La Planète des singes | 8 | 1912 | 1963 | français |
| Bradbury | Fahrenheit 451 | 7 | 1920 | 1953 | anglais |
| Bradbury | Chroniques martiennes | 8 | 1920 | 1950 | anglais |
| Herbert | Dune | 8 | 1920 | 1965 | anglais |
| Huxley | Le meilleur des mondes | 7 | 1894 | 1931 | anglais |
| K.Dick | Ubik | 9 | 1928 | 1969 | anglais |
| K.Dick | Blade Runner | 8 | 1928 | 1968 | anglais |
| K.Dick | Le Maître du Haut Château | 8 | 1928 | 1962 | anglais |
| Orwell | 1984 | 10 | 1903 | 1949 | anglais |
| Van Vogt | Le monde des Ā | 7 | 1912 | 1945 | anglais |
| Verne | De la Terre à la Lune | 10 | 1828 | 1865 | français |

11. Modifications d'une base

1. → INSERT

Insérer les données suivantes dans la base Auteurs:

INSERT INTO LIVRES (code, titre, id_auteur, ann_publi, note) VALUES (17, 'Hypérion', 'Simmons', 1989, 8)

2. ◆ UPDATE

```
Requête SQL

UPDATE LIVRES
SET note=7
WHERE titre = 'Hypérion'
```

3. **◆** DELETE

Requête SQL DELETE FROM LIVRES WHERE titre='Hypérion'