

Année scolaire 2025 – 2026

Lycée Victor Hugo - Colomiers

Terminale NSI

BDD

Bases de données

Thomas Ricart

`ricart.lvh@laposte.net`

24 mars 2025

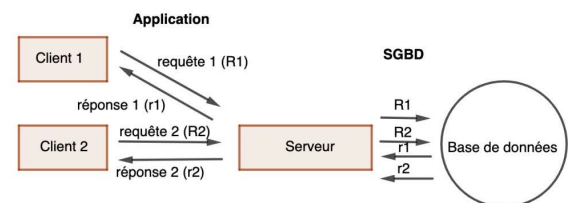
BDD01 – Base de données relationnelles

Lorsque les données deviennent volumineuses, nous sommes obligés d'utiliser des fichiers externes (par exemple des fichiers .csv). On pouvait alors par programmation extraire des informations d'une copie de ce fichier, chacun de son côté ce qui implique que :

1. Les données sont statiques au sens où elles ne sont pas mises à jour régulièrement
2. Plusieurs personnes ne peuvent pas consulter les mêmes données en même temps

Or il est courant que, des données volumineuses et actualisées soient consultées en ligne par plusieurs personnes, par exemple des réservations de locations de vacances ou encore des cartes (OpenstreetMap).

Le principe est le suivant : plusieurs clients peuvent exprimer des requêtes en ligne à un serveur qui est en relation avec une base de données gérée par un logiciel appelé système de gestion de bases de données (SGBD)



Définition 1: Une base de données

Une **base de données** est un ensemble structuré de données stockées sur un support persistant.

Un **système de gestion de base de données (SGBD)** est un logiciel qui gère les données de la base de données

1. Présentation et vocabulaire

Théorisé en 1970 par le Britannique Edgar J. Codd, le modèle relationnel est à ce jour le modèle de base de données le plus utilisé, même si l'ère actuelle du Big Data tend à mettre en avant d'autres modèles non relationnels.

Propriété 1: Les principes de base du modèle relationnel

- Les données sont regroupées dans différentes tables (relations). Chaque relation contient des éléments directement en lien avec le sujet général de la table.
- Autant que possible, des données identiques ne doivent pas se trouver dans des tables différentes : on évite la redondance des données.
- Les données ne doivent pas contenir elles-mêmes d'autres données : on parle d'atomicité des données.

id	titre	auteur	ann_publi	note	en-tête
1	1984	Orwell	1949	10	
2	Dune	Herbert	1965	8	
3	Fondation	Azimov	1951	9	
4	Le meilleur des mondes	Huxley	1931	9	
5	Fahrenheit 451	Bradbury	1953	7	
6	Ubik	K.Dick	1969	6	
7	Chroniques martiennes	Bradbury	1950	7	
8	La nuit des temps	Barjavel	1968	7	
9	Blade Runner	K.Dick	1968	9	
10	Les Robots	Azimov	1950	10	un t-uplet
11	La Planète des singes	Boule	1963	9	
12	Ravage	Barjavel	1943	11	
13	Le Maître du Haut Château	K.Dick	1962	7	
14	La fin de l'éternité	Azimov	1955	9	

attribut : titre

corps

- l'**en – tête** est la première ligne du tableau;
- le **corps** est le reste du tableau;
- Un **enregistrement** ou un t-uplet est une ligne dans le tableau;
- Un **descripteur** ou **attribut** correspond au titre d'une colonne d'un tableau;
- Le **domaine d'un attribut** est l'ensemble des valeurs que l'attribut peut prendre.

2. Clé primaire et clé étrangère

Dans une relation, deux enregistrements identiques sont non autorisés et il faut pouvoir identifier de façon unique chaque enregistrement. Il faut déterminer, parmi les attributs, lequel permet d'identifier de manière unique un enregistrement. Cet attribut sera considéré comme la **clé primaire** de la relation.

Définition 2: Clé primaire

Une clé primaire est un attribut (ou une réunion d'attributs) dont la connaissance suffit à identifier avec certitude un unique enregistrement.

Exemple 1: Dans la relation précédente, on peut dire que:

- id peut jouer le rôle de clé primaire. En effet cette valeur est censée être unique.
- les autres attributs ne peuvent pas être choisis car il peut y avoir plusieurs livres qui ont le même titre, auteur, année de publication ou note.

Ajoutons maintenant une deuxième relation à notre base de données :

id	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	9
5	Fahrenheit 451	5	1953	7
6	Ubik	6	1969	6
7	Chroniques martiennes	5	1950	7
8	La nuit des temps	7	1968	7
9	Blade Runner	6	1968	9
10	Les Robots	3	1950	10
11	La Planète des singes	8	1963	9
12	Ravage	7	1943	11
13	Le Maître du Haut Château	6	1962	7
14	La fin de l'éternité	3	1955	9

Relation : LIVRES

id	nom	prenom	ann_naissance	langue_ecriture
1	Orwell	George	1903	anglais
2	Herbert	Frank	1920	anglais
3	Azimov	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

Relation : AUTEURS

Lors de l'ajout de la nouvelle table, nous avons changé l'attribut `auteur` par l'attribut `id_auteur` permettant de faire le lien entre les deux relations.

`id_auteur` présente dans la relation LIVRES est la clé primaire d'une autre relation (AUTEURS): on dira que c'est une **clé étrangère** de la relation LIVRES.

Définition 3: Clé étrangère

Un attribut d'une relation qui correspond à une clé primaire d'une autre relation est appelé **clé étrangère**. Ce sont ces clés étrangères qui permettent de relier les différentes relations entre elles.

Clé primaire

id	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	9
5	Fahrenheit 451	5	1953	7
6	Ubik	6	1969	6
7	Chroniques martiennes	5	1950	7
8	La nuit des temps	7	1968	7
9	Blade Runner	6	1968	9
10	Les Robots	3	1950	10
11	La Planète des singes	8	1963	9
12	Ravage	7	1943	11
13	Le Maître du Haut Château	6	1962	7
14	La fin de l'éternité	3	1955	9

Relation : LIVRES

Clé étrangère

Clé primaire

id	nom	prenom	ann_naissance	langue_ecriture
1	Orwell	George	1903	anglais
2	Herbert	Frank	1920	anglais
3	Azimov	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

Relation : AUTEURS

3. Les contraintes d'intégrité

Propriété 2: Les 3 contraintes d'intégrité

- **contraintes de domaine:** Tout attribut d'un enregistrement doit respecter le domaine indiqué dans le schéma relationnel. On peut parler aussi de **type**.
- **contraintes de relation:** Tout enregistrement est unique. Cette contrainte est assurée par l'existence d'une clé primaire.
- **contraintes de référence:** Dans une relation, la valeur d'un attribut clé étrangère doit obligatoirement être retrouvée dans la relation dont cet attribut est clé primaire. On en doit pas avoir une clé étrangère qui ne fait pas référence à une clé primaire existante dans une autre relation.

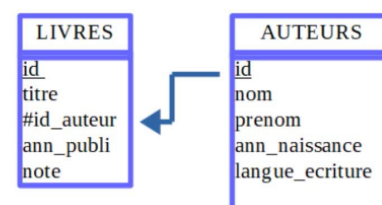
4. Représentation usuelle des bases de données relationnelles

Un **schéma relationnel** est l'ensemble des relations présentes dans une base de données. Il faut:

- les noms des relations
- la liste des attributs et le type de leur domaine
- la clé primaire (on la souligne) et les clés étrangères (précédées d'un #)

Pour notre exemple cela donne :

- LIVRES (id : int, titre: str, #id_auteur: int, ann_publi: int, note: int)
- AUTEURS (id: int, nom: str, prenom: str, ann_naissance: int, langue_ecriture: str)



Exercice 1: Exercice sur le modèle relationnel

codeEquipe	nomEquipe
ALM	AG2R La Mondiale
AST	Astana Pro Team
TBM	Bahrain - McLaren
BOH	BORA - hansgrohe
CCC	CCC Team
COF	Cofidis, Solutions Crédits
DQT	Deceuninck - Quick Step
EF1	EF Pro Cycling
GFC	Groupama - FDJ
LTS	Lotto Soudal
...	...

dossard	nomCoeur	prenomCoeur	codeEquipe
141	LÓPEZ	Miguel Ángel	AST
142	FRAILE	Omar	AST
143	HOULE	Hugo	AST
11	ROGLIČ	Primož	TJV
12	BENNETT	George	TJV
41	ALAPHILIPPE	Julian	DQT
44	CAVAGNA	Rémi	DQT
45	DECLERCQ	Tim	DQT
121	MARTIN	Guillaume	COF
122	CONSONNI	Simone	COF
123	EDET	Nicolas	COF
...	...	...	...

numéroEtape	villeDépart	villeArrivée	km
1	Nice	Nice	156
2	Nice	Nice	185
3	Nice	Sisteron	198
4	Sisteron	Orcières-Merlette	160
5	Gap	Privas	198
...	...	...	...

dossard	numéroEtape	tempsRéalise
41	2	04:55:27
121	4	04:07:47
11	5	04:21:22
122	5	04:21:22
41	4	04:08:24
...	...	...

1. Donner la représentation de ces trois relations

.....

.....

.....

2. Donner le modèle relationnel sous forme de diagramme

3. Quel temps a réalisé Guillaume MARTIN sur l'étape Sisteron / Orcières-Merlette?

4. Quel est le nom de son équipe?

5. A l'arrivée à Privas, qui est arrivé en premier entre Primož ROGLIČ et Simone CONSONNI?

BDD02 – Langage SQL

Nous allons travailler avec le langage **SQLite** qui peut s'utiliser directement sans avoir besoin de serveur.

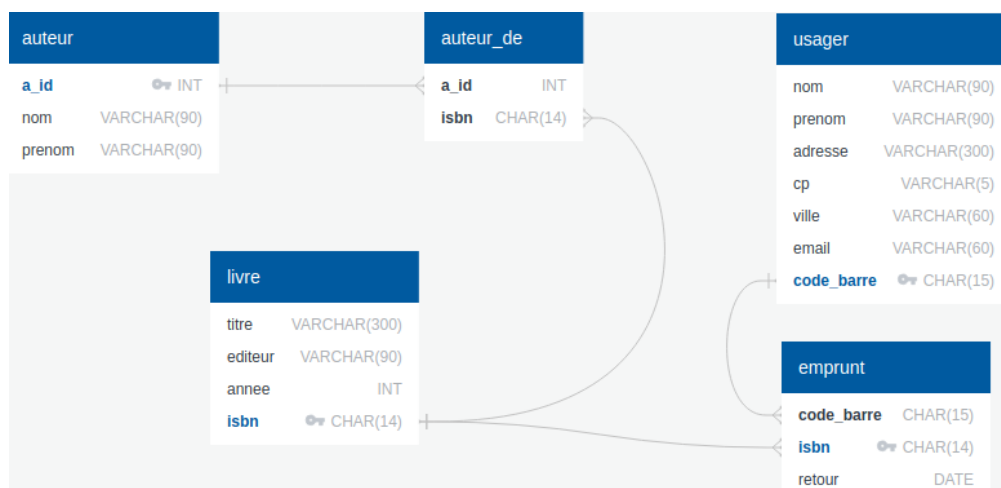
1. Introduction au langage SQL

Dans toute la suite, nous allons travailler avec la base de données livres.db. Créer un répertoire de travail nommé `exo_bdd_sql` et télécharger la base de données: livres.db

Dans la suite vous pourrez interroger cette base de données de deux manières:

- en ligne avec <https://sqliteonline.com/>
 - par File/OpenDB, ouvrez le fichier livre.db téléchargé
 - écrivez votre requête et cliquer sur Run
- avec le logiciel externe DB Browser for SQLite: <https://sqlitebrowser.org/>
 - ouvrez le fichier livre.db téléchargé
 - choisir Executer le SQL
 - cliquer sur la flèche bleue

Voici le diagramme relationnel de cette base:



✍ Exercice 1: Identifier:

1. par PK les clés primaires des relations (Primary Key)
2. par FK les clés étrangères des relations (Foreign Key)

2. Sélection des données

Propriété 1: Sélection de données basique

```
SELECT titre
FROM livre;
```

On veut tous les titres de la base livre.

- Les mots clés SQL sont traditionnellement écrits en MAJUSCULE.
- Le ; signale la fin de l'instruction.
- l'indentation n'est pas obligatoire.

	titre
1	Les Aventures de Huckleberry Finn
2	Fondation et Empire
3	Akira
4	Les Robots
5	Astérix chez les Pictes
6	Les Monades urbaines

Result: 128 enregistrements ramenés €

✍ Exercice 2: Écrire une requête permettant d'afficher le nom de tous les usagers

Propriété 2: Sélection de données filtrées

```
SELECT titre
FROM livre
WHERE annee >= 1990;
```

On veut les titres de la table `livre` qui sont parus après (ou en) 1990;

Le mot clé `WHERE` doit être suivi d'un booléen. Les opérateurs classiques `=`, `!=`, `>`, `<`, `>=`, `<=` peuvent être utilisés ainsi que le mot clé `IN`.

titre	
1	Les Aventures de Huckleberry Finn
2	Fondation et Empire
3	Akira
4	Les Robots
5	Astérix chez les Pictes
6	Les Monades urbaines
112 enregistrements ramenés en 2	

 **Exercice 3:** Écrire une requête permettant d'afficher le nom des usagers dont le code postal est 75005

Propriété 3: Sélection de données filtrées avec des booléens AND – OR

```
SELECT titre
FROM livre
WHERE annee >= 1990
AND annee <= 1980
AND editeur = 'Dargaud';
```

On veut les titres de la table `livre` qui sont parus entre 1970 et 1980 chez l'éditeur Dargaud;

titre	
1	Astérix chez les Belges
1 enregistrements ramenés en 2ms de ann edi	

 **Exercice 4:** Écrire une requête permettant d'afficher les ISBN des livres de l'éditeur 'Flamarion' ou édités en 2015

Propriété 4: Requete approchée LIKE

```
SELECT titre
FROM livre
WHERE titre LIKE '%Asterix%';
```

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.

titre	
1	Astérix chez les Pictes
2	Astérix et Cléopâtre
3	Le Tour de Gaule d'Astérix
4	Astérix et les Normands
5	Astérix en Corse

 **Exercice 5:** Écrire une requête permettant d'afficher les titres des livres dont l'ISBN contient la séquence 20

Propriété 5: Requete sur plusieurs colonnes

```
SELECT titre, isbn
FROM livre
WHERE annee >= 1990;
```

On veut les titres et les ISBN de la table `livre` qui sont parus après 1990.

	titre	isbn
1	Les Aventures de Huckleberry Finn	978-2081509511
2	Fondation et Empire	978-2207249123
3	Akira	978-2723428262
4	Les Robots	978-2745989857
5	Astérix chez les Pictes	978-2864972662

 **Exercice 6:** Ecrire une requête permettant d'afficher les titres des livres dont l'ISBN contient la séquence 20

Propriété 6: Requete sur toutes les colonnes

```
SELECT *
FROM livre
WHERE annee >= 1990;
```

On veut toutes les colonnes disponibles de la table `livre` pour les livres qui sont parus après 1990.

L'astérisque `*` est un joker (wildcard en anglais).

	titre	editeur	annee	isbn
1	Les Aventures de Huckleberry Finn	Flammarion	2020	978-2081509511
2	Fondation et Empire	Editions Denoël	1999	978-2207249123
3	Akira	Glénat	2000	978-2723428262
4	Les Robots	Editions Milan	2017	978-2745989857
5	Astérix chez les Pictes	Editions Albert René	2013	978-2864972662

112 enregistrements ramenés en 4ms depuis : `SELECT * FROM livre WHERE annee >= 1990;`

Propriété 7: Renommer les colonnes: AS

```
SELECT titre AS titre_du_livre
FROM livre
WHERE annee >= 1990;
```

Lors de l'affichage du résultats et dans la suite de la requête (important), la colonne `titre` est renommée `titre_du_livre`.

	titre_du_livre
1	Les Aventures de Huckleberry Finn
2	Fondation et Empire
3	Akira
4	Les Robots
5	Astérix chez les Pictes

3. Opération 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.

Propriété 8: Compter: COUNT

```
SELECT COUNT(*) AS total
FROM livre
WHERE titre LIKE '%Asterix%'
```

On veut compter le nombre d'enregistrements de la tables livres comportant le mot 'Astérix'. Le résultat sera le seul élément d'une colonne nommée `total`.

	total
1	10

Propriété 9: Additionner: SUM

```
SELECT SUM(annee) AS somme
FROM livre
WHERE titre LIKE '%Asterix%'
```

On veut additionner les années des livres de la tables livres comportant le mot "Astérix". Le résultat sera le seul élément d'une colonne nommée `somme`. Ce calcul n'a aucun sens...

	somme
1	20046

Propriété 10: Faire une moyenne: AVG

```
SELECT AVG(annee) AS moyenne
FROM livre
WHERE titre LIKE '%Asterix%'
```

On veut calculer la moyenne des années de parution des livres de la table livres comportant le mot "Astérix". Le résultat sera le seul élément d'une colonne nommée `moyenne`.

	moyenne
1	2004.6

Propriété 11: Trouver les extrémum: MIN et MAX

```
SELECT MIN(annee) AS minimum
FROM livre
WHERE titre LIKE '%Asterix%'
```

minimum	
1	1979

On veut trouver la plus petite valeur de la colonne `annee` parmi les livres de la table `livre` comportant le mot "Astérix". Le résultat sera le seul élément d'une colonne nommée `minimum`. Le fonctionnement est identique avec `MAX` pour la recherche du maximum.

Propriété 12: Classer les valeurs: ORDER BY, ASC, DESC

```
SELECT titre, annee
FROM livre
WHERE titre LIKE '%Asterix%'
ORDER BY annee DESC;
```

	titre	annee
1	Astérix et la Transitalique	2017
2	Astérix chez les Pictes	2013
3	Astérix légionnaire	2011
4	L'Odyssée d'Astérix	2008

On veut afficher tous les albums d'Astérix, et leur année de parution, classés par année décroissante. En utilisant le paramètre `ASC` on trie par ordre croissant.

Propriété 13: Limiter le nombre de résultats LIMIT

```
SELECT titre, annee
FROM livre
WHERE titre LIKE '%Asterix%'
ORDER BY annee DESC
LIMIT 1;
```

On veut afficher tous les albums d'Astérix, et leur année de parution, classés par année décroissante. On se limite au premier: on a l'album le plus récent

Propriété 14: Suppression de doublons

```
SELECT DISTINCT editeur
FROM livre;
```

editeur	
1	Flammarion
2	Editions Denoël
3	Glénat
4	Editions Milan

On veut la liste de tous les éditeurs.
Sans le mot-clé `DISTINCT`, beaucoup de doublons apparaîtraient.

4. Des recherches croisées: les jointures

 **Exercice 7:** Écrire la requête permettant d'afficher le contenu complet de la table `emprunt`

Le contenu est peu lisible: qui a emprunté quel livre?

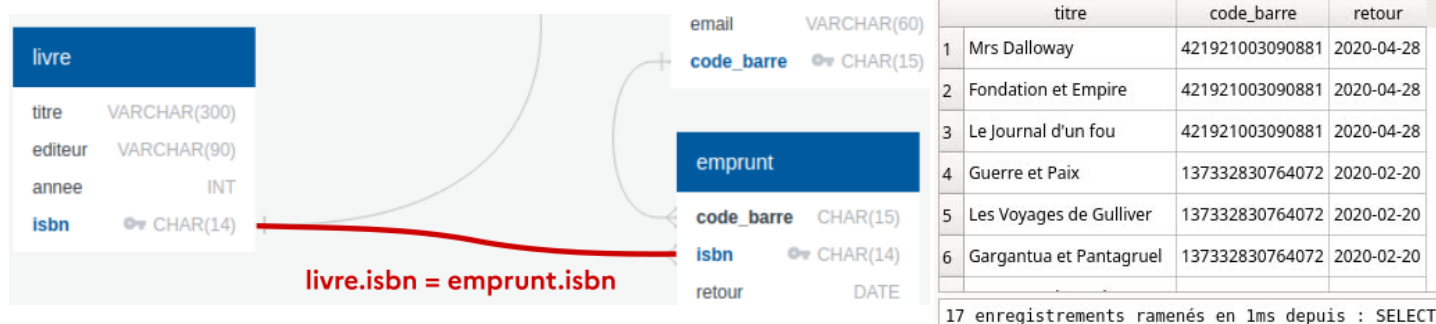
Pour que la table `emprunt` soit lisible, il faudrait (dans un premier temps) que l'on affiche à la place de l'ISBN le titre de l'ouvrage. Or ce titre est disponible dans la table `livres`. On va donc procéder à une jointure de ces deux tables.

Propriété 15: Jointure de 2 tables: JOIN

```
SELECT livre.titre, emprunt.code_barre, emprunt.retour
FROM emprunt
JOIN livre ON emprunt.isbn = livre.isbn;
```

Comme plusieurs tables sont appelées, nous préfixons chaque colonne avec le nom de la table. Nous demandons ici l'affichage de la table `emprunt`, mais où on aura remplacé l'ISBN (peu lisible) par le titre du livre. Cette commande doit se comprendre comme:

- on invite la table `livre` (dont on va afficher la colonne `titre`)
- la correspondance entre la table `livre` et la table `emprunt` doit se faire sur l'attribut `ISBN`, qui est la clé primaire de `livres` et une clé étrangère d'`emprunts`.



Le résultat précédent a permis d'améliorer la visibilité de la table **emprunt**, mais il reste la colonne **code_barre** qui est peu lisible. Nous pouvons la remplacer par le titre du livre, en faisant une nouvelle jointure, en invitant maintenant les deux tables **livre** et **usager**.

Propriété 16: Jointure de 3 tables: JOIN

```
SELECT usager.nom, usager.prenom, livre.titre, emprunt.retour
FROM emprunt
JOIN livre ON emprunt.isbn = livre.isbn
JOIN usager ON emprunt.code_barre = usager.code_barre;
```

Il faut bien comprendre que la table principale qui nous intéresse ici est **emprunts**, mais qu'on modifie les valeurs affichées en allant chercher des correspondances dans deux autres tables.

	nom	prenom	titre	retour
1	MOREAU	ALAIN	Mrs Dalloway	2020-04-28
2	MOREAU	ALAIN	Fondation et Empire	2020-04-28
3	MOREAU	ALAIN	Le Journal d'un fou	2020-04-28
4	DUBOIS	PHILIPPE	Guerre et Paix	2020-02-20
5	DUBOIS	PHILIPPE	Les Voyages de Gulliver	2020-02-20

17 enregistrements ramenés en 1ms depuis : SELECT

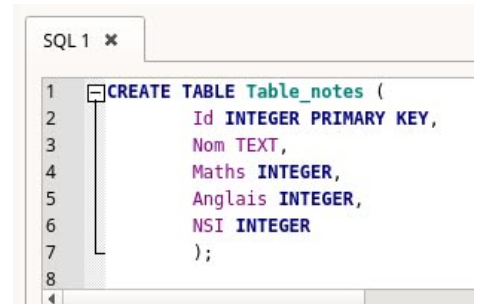
Exercice 8: Utilisation de la base Mondial Database

1. Trouver les noms des pays dont les codes sont les suivants: COCO, XMAS, GROX, SMAR, SBAR
2. La liste de tous les pays et de leur population trié par ordre croissant de la population
3. Le nom des dix pays ayant la plus petite superficie
4. La liste de tous les aéroports dans le monde ? Combien y-a-t-il d'aéroports?
5. La liste des aéroports (noms) situés en France
6. La liste des aéroports (noms) situés à plus de 2000 m d'altitude
7. Quels sont tous les fleuves qui se jettent dans la mer Méditerranée triés dans l'ordre décroissant de leur longueur)
8. Quel est le fleuve le plus court? le plus long?
9. Quel est le désert le plus vaste et où se trouve-t-il? le moins vaste?
10. La liste de toutes les capitales dont la latitude est supérieure à 48 degrés
11. La liste de toutes les capitales dont la latitude est supérieure à celle de Paris
12. La liste de toutes les capitales européennes dont la latitude est supérieure à celle de Paris
13. Quelles sont toutes les montagnes situées en Islande (nom et altitude)?
14. Quelles sont toutes les montagnes situées sur une île ayant un sommet à plus de 3000 mètres?
15. Rédiger une requête pour avoir les 10 premiers pays européens les plus densément peuplés
16. La liste de tous les pays (leurs noms) limitrophes à la France ainsi que la longueur de la frontière
17. Quels sont les fleuves dans le Monde de plus de 1000 km de longueur ? Combien sont ils?
18. Afficher la liste des pays d'Europe avec leur PIB, la part de l'industrie, et le taux de chômage
19. Calculer la moyenne du taux de chômage en Europe continentale. Afficher les pays européens dont le taux de chômage est inférieure à cette moyenne
20. Calculer la moyenne de la part de l'industrie pour les pays européens. Afficher les pays dont la part de l'industrie est supérieure à la moyenne en Europe
21. Afficher les noms des villes et des pays que traversent le Nil (Nile)
22. Afficher tous les lacs situés en Grande-Bretagne

5. Création et modification d'une base de données

Propriété 17: Création d'une table: CREATE TABLE

```
CREATE TABLE Table_notes (  
    Id INTEGER PRIMARY KEY,  
    Nom TEXT,  
    Maths INTEGER,  
    Anglais INTEGER,  
    NSI INTEGER  
);
```



C'est l'utilisateur qui spécifie, éventuellement, quel attribut sera une clé primaire.

Propriété 18: Insertion de valeurs: INSERT INTO, VALUES

```
INSERT INTO Table_notes  
VALUES (1, 'Alice', 16, 11, 17),  
       (2, 'Bob', 12, 15, 10),  
       (3, 'Charles', 9, 11, 18);
```

```
INSERT INTO Table_notes (Id,Nom,Maths,Anglais,NSI)  
VALUES (1, 'Alice', 16, 11, 17),  
       (2, 'Bob', 12, 15, 10),  
       (3, 'Charles', 9, 11, 18);
```

Propriété 19: Violation de contrainte de relation

```
INSERT INTO Table_notes  
VALUES (3, 'Denis', 18, 10, 12);
```

On obtient un message d'erreur du type
UNIQUE constraint failed.

Le SGBD protège la base de données en n'acceptant pas la proposition d'insertion. La base de données n'est pas modifiée.

Il est possible de déléguer la gestion des valeurs de la clé primaire avec l'instruction AUTOINCREMENT:

```
Id INTEGER PRIMARY KEY AUTOINCREMENT,
```

Propriété 20: Modification d'une valeur UPDATE, SET

```
UPDATE Table_notes  
SET Maths = 18  
WHERE Nom = 'Alice';
```

On modifie la note de Maths d'Alice

Propriété 21: Suppression d'un enregistrement: DELETE

```
DELETE FROM Table_notes  
WHERE Nom = 'Charles';
```

On supprime totalement la ligne concernant Charles

Si une autre table contient par exemple l'attribut id comme clé étrangère, et si l'id de Charles fait partie de cette table, le SGBD refusera de supprimer cette ligne, afin de ne pas violer la contrainte de référence.

Propriété 22: Suppression totale d'une table: DROP TABLE

```
DROP TABLE Table_notes;
```

On supprime totalement la table. Si une autre table est reliée à Table_notes par une clé étrangère, la suppression sera bloquée par le SGBD.

Memento SQL

Vocabulaires

table	relation
ligne	tuple ou enregistrement
colonne	attribut avec domaine de valeurs (entier, date, chaîne de caractères)
clé	primaire ou externe/étrangère

Type

TEXT	mais aussi STR, STRING, VARCHAR
INT	mais aussi INTEGER
REAL	mais aussi FLOAT
DATE	au format YYYY-MM-DD
ENUM	liste des valeurs possibles

Sélection :

SELECT * FROM clients ;
SELECT nom, prenom FROM Clients ;
SELECT nom, prenom FROM Clients WHERE ville= 'Lorient' ;
SELECT nom, prenom FROM Clients WHERE ville= 'Lorient' AND naissance < '1967-03-04' ;
SELECT nom, prenom FROM Clients WHERE ville IN ('Lorient','Brest') ;
SELECT nom, prenom FROM Clients WHERE nom LIKE "D%" ;
SELECT nom, prenom FROM Clients WHERE ville IS NULL ;
SELECT nom, prenom FROM Clients ORDER BY naissance DESC / ASC ;
SELECT nom, prenom FROM Clients ORDER BY nom, naissance ;
SELECT nom, prenom FROM Clients LIMIT 20 ; (MySQL, PostgreSQL, sqlLite)
SELECT DISTINCT ville FROM Clients ;
SELECT DISTINCT ville, pays FROM Clients ; (Paris, France et Paris,Texas)

Comptage

SELECT COUNT(*) FROM Clients ;
SELECT COUNT(DISTINCT ville) FROM Clients ;

Jointure

SELECT * FROM Table1 JOIN Table2 ON Table1.foreignkey = Table2.primarykey ;
SELECT newName.nom FROM Table1 JOIN Table2 AS newName ON Table1.fk = newName.pk ;

Modification

INSERT INTO Table VALUES ('valeur 1', 'valeur 2', ...) ;
INSERT INTO Table (nom, prenom, age) VALUES ('Durand', 'Paul',45) ;
UPDATE Table SET age = age +1 ;
UPDATE Table SET nom = 'Durant' WHERE nom='Durand' AND prenom='Paul' ;
DELETE FROM Table WHERE nom='Durant' ;

Fonctions

textuelles	UPPER/LOWER	SELECT nom FROM Clients WHERE UPPER(ville) = 'LORIENT'
	SUBSTR	SELECT SUBSTR(first,nb,prenom) pour l'initiale : first = 1 pas 0 !
numériques	+ - * /	SELECT nombre * prix FROM Commande ;
	ROUND	SELECT ROUND(prix,2) FROM Commande ;
d'agrégation	COUNT	SELECT COUNT(*) FROM Table ;
	MAX	SELECT MAX(id) FROM Table ;
	MIN	SELECT MIN(note) FROM Devoir ;
	AVG	SELECT AVG(note) FROM Devoir ;
	SUM	SELECT SUM(prix*nombre) FROM Commande ;