**Chapitre 1**

CRÉATION DE LA TABLE

CREATE TABLE `liste\_livres` (

`id\_livre` int(3) NOT NULL,

`titre` varchar(128) NOT NULL,

`isbn\_10` bigint(11) NOT NULL,

`auteur` varchar(128) NOT NULL,

`prix` float NOT NULL

)

INSERT INTO `liste\_livres` (`id\_livre`, `titre`, `isbn\_10`, `auteur`, `prix`) VALUES

(1, 'Forteresse digitale', 2709626306, 'Dan Brown', 20.5),

(2, 'La jeune fille et la nuit ', 2253237620, 'Guillaume Musso', 21.9),

(3, 'T\'choupi se brosse les dents', 2092589547, 'Thierry Courtin', 5.7),

(4, 'La Dernière Chasse', 2226439412, 'Jean-Christophe Grangé', 22.9),

(5, 'Le Signal', 2226319484, 'Maxime Chattam', 23.9);

Exercices

### # 1

Quelle requête utiliser pour afficher l'ensemble des enregistrements de la table liste\_livres ?

### #2

Quelle requête utiliser pour sélectionner uniquement les livres qui ont un **prix strictement supérieur à 20** de la table liste\_livres ?

### #3

Quelle requête utiliser pour trier les enregistrements de la table liste\_livres du prix le plus élevé aux prix le plus bas ?

### #4

Quelle requête utiliser pour récupérer le prix du livre le plus élevé de la table liste\_livres ?

### #5

Quelle requête utiliser pour récupérer les livres de la table liste\_livres qui ont un prix compris entre 20 et 22 ?

### #6

Quelle requête utiliser pour récupérer tous les livres de la table liste\_livres à l'exception de celui portant la valeur pour la colonne **isbn\_10** : **2092589547** ?

### #7

Quelle requête utiliser pour récupérer le prix du livre le moins élevé de la table liste\_livres en renommant la colonne dans les résultats par **minus** ?

### #8

Quelle requête utiliser pour sélectionner uniquement les 3 premiers résultats sans le tout premier de la table liste\_livres ?

**Chapitre 2**

CRÉATION DE LA TABLE

CREATE TABLE `liste\_etudiants` (

`id\_etudiant` int(3) NOT NULL,

`prenom` varchar(128) NOT NULL,

`nom` varchar(128) NOT NULL

)

INSERT INTO `liste\_etudiants` (`id\_etudiant`, `prenom`, `nom`) VALUES

(30, 'Joseph', 'Biblo'),

(31, 'Paul', 'Bismuth'),

(32, 'Jean', 'Michel'),

(33, 'Ted', 'Bundy'),

(34, 'Caroline', 'Martinez'),

(35, 'Joséphine', 'Henry');

--------------------------------------------------------

CREATE TABLE `liste\_examens` (

`id` int(3) NOT NULL,

`id\_examen` int(3) NOT NULL,

`id\_etudiant` int(3) NOT NULL,

`matiere` varchar(128) NOT NULL,

`note` float NOT NULL

)

INSERT INTO `liste\_examens` (`id`, `id\_examen`, `id\_etudiant`, `matiere`, `note`) VALUES

(788, 45, 30, 'Histoire-Geographie', 10.5),

(789, 87, 33, 'Mathématiques', 14),

(790, 87, 34, 'Mathématiques', 4),

(791, 45, 31, 'Histoire-Geographie', 15.5),

(792, 45, 32, 'Histoire-Geographie', 8),

(793, 87, 31, 'Mathématiques', 14),

(794, 45, 33, 'Histoire-Geographie', 9.5),

(795, 45, 36, 'Histoire-Geographie', 13),

(796, 45, 34, 'Histoire-Geographie', 17),

(797, 87, 30, 'Mathématiques', 7.5);

QuelQUES NOTES SQL

L’utilisation de la commande **SELECT** en SQL permet de lire toutes les données d’une ou plusieurs colonnes. Cette commande peut potentiellement afficher des lignes en doubles. Pour éviter des redondances dans les résultats il faut simplement ajouter **DISTINCT** après le mot SELECT.

En SQL, la fonction d’agrégation **COUNT()** permet de compter le nombre d’enregistrement dans une table.

La fonction d’agrégation **AVG()** dans le langage SQL permet de calculer une valeur moyenne sur un ensemble d’enregistrement de type numérique et non nul.

Dans le langage SQL, la fonction d’agrégation **MAX()** permet de retourner la valeur maximale d’une colonne dans un set d’enregistrement.

Une image contenant texte, clipart

Description générée automatiquement

Dans le langage SQL la commande **INNER JOIN** est un type de jointures très communes pour lier plusieurs tables entre-elles. Cette commande retourne les enregistrements lorsqu’il y a au moins une ligne dans chaque colonne qui correspond à la condition.

Une image contenant texte, clipart

Description générée automatiquement

Dans le langage SQL, la commande **LEFT JOIN** est un type de jointure entre 2 tables. Cela permet de lister tous les résultats de la table de gauche (left = gauche) même s’il n’y a pas de correspondance dans la deuxième table.

Une image contenant texte, clipart, graphiques vectoriels

Description générée automatiquement

En SQL, la commande **RIGHT JOIN** est un type de jointure entre 2 tables qui permet de retourner tous les enregistrements de la table de droite (right = droite) même s’il n’y a pas de correspondance avec la table de gauche.

Une image contenant texte, clipart

Description générée automatiquement

Dans le langage SQL, la commande **FULL JOIN** permet de faire une jointure entre 2 tables. L’utilisation de cette commande permet de combiner les résultats des 2 tables, les associer entre eux grâce à une condition et remplir avec des valeurs NULL si la condition n’est pas respectée.

EXERCICES

#1

Quelle requête utiliser pour afficher l'id des étudiants qui ont participés à au moins un examen ?

#2

Quelle requête utiliser pour compter le nombre d'étudiants qui ont participés à au moins un examen ?

#3

Quelle requête utiliser pour calculer la moyenne de l'examen portant l'id : **45** ?

#4

Quelle requête utiliser pour récupérer la meilleure note de l'examen portant l'id : **87** ?

#5

Quelle requête utiliser pour afficher l'id des étudiants qui ont eu **plus de 11 à l'examen 45** ou **plus de 12 à l'examen 87** ?

#6

Quelle requête utiliser pour afficher tous les enregistrements de la table liste\_examens avec en plus, si c'est possible, le prénom et le nom de l'étudiant ?

*Tester la requête avec INNER JOIN et LEFT JOIN. Quelle différence ?*

Note : Pour simplifier l’écriture des critères de jointure, il existe une technique appelée « surnommage », c'est-à-dire que l'on attribue un surnom à chacune des tables présentes dans la partie **FROM** du **SELECT** ou dans le **SELECT** directement**.**

#7

Quelle requête utiliser pour afficher les enregistrements de la table liste\_examens avec le prénom et le nom de l'étudiant, uniquement quand les étudiants sont présents dans la table liste\_etudiants ?

#8

Quelle requête utiliser pour afficher les 3 meilleures examens, du meilleur au moins bon, avec le prénom et le nom de l'étudiant associé ?