TIn311 - Bases de données - IPSA Toulouse - TP 1

Alexandre Condette - [ale](mailto:alexandre.condette@spacebel.fr)[xandre.condette@spacebel.fr](mailto:xandre.condette@spacebel.fr) 2022 - 2023

# SQL : Premiers pas avec une base de données

L’un des objectifs de ce cours est de maîtriser les bases du SQL. Il s’agit donc de taper des requêtes et non pas utiliser les facilités de l’interface graphique (notamment pour créer des tables ou ajouter des données).

## SQL : Strutured Query Language

* C’est le langage universel de manipulation des bases de données relationnelles.
* Créé en 1974, normalisation continue : 86, 92, 99, 2003, 2006, 2008, 2011, 2016, etc.
* On n’écrit pas des programmes, mais des requêtes (queries).
* Pas de compilation / interprétation : on envoie ses requêtes au SGBD-R.
* Une requête agit sur la base, i.e. les données. Son effet est permanent.
* Ne pas confondre la base (les données) et les requêtes !
* Code source des requêtes sous forme de fichier texte à l’extension *.sql*

## SQL, pour faire quoi ?

* Langage de Définition des Données :
  + décrire l’organisation des données
  + créer/supprimer/modifier les tables vues et index de la base
* Langage de Manipulation des Données :
  + insérer de données dans une table
  + modifier tout ou partie des données d’une table
  + supprimer des données, avec ou sans condition
* Administration de la base:
  + Contrôler l’accès, optimiser, sauvegarder, exporter, importer, etc

## Remarques importantes sur la syntaxe SQL

* Une requête se termine toujours par un ";" (même si l’interface graphique le rajoute invisiblement pour vous).
* Le langage n’est pas sensible à la casse.
* Par convention, on essaiera d’écrire les mots-clefs du langage en MAJUSCULE.
* Évitez les noms de tables ou colonnes contenant des accents ou des espaces.
* L’indentation n’est pas nécessaire, on peut écrire une requête sur plusieurs lignes : la lisibilité prime.

## Type de données

Pour chaque colonne, on va devoir choisir un type de données adéquat :

* contrainte d’intégrité. ex : si la donnée est numérique, on ne pourra saisir que des nombres.
* souvent dépendant du SGBD utilisé.

On se contentera avec SQLite (le SGBD que nous allons utiliser) de :

1

INT -- ex : 42 REAL -- ex : 3.14

TEXT -- ex : ' Hello IPSA ' NULL -- Absence de valeur

2

3

4

5

NULL représente l’absence de valeur. C’est-à-dire que pour une ligne d’une table, il est possible que certaines colonnes n’aient pas de valeur.

Pour certaines colonnes, en fonction du modèle, c’est cependant obligatoire : on déclare alors la colonne comme NOT NULL par exemple

1

Email TEXT NOT NULL .

2

Pour comparer une colonne avec NULL, on utilise les opérateurs **IS NULL** et **IS NOT NULL** et non pas ’=’ et ’!=’.

## Opérateur et fonction SQL

|  |  |
| --- | --- |
| **Opérateurs / fonctions** | **Utilisation** |
| +, -, \*, /, % | addition, soustraction, multiplication, division, modulo |
| =, <>, <, <=, >, >= | égale, différent, inférieur, etc. |
| BETWEEN X and Y | compris dans [X, Y] |
| IN (a, b, c), NOT IN (...) | parmi a, b, c ou l’inverse |
| OR, AND, NOT | opérateur booléen : ou, et, non |
| LIKE | texte similaire |
| IS NULL, IS NOT NULL | comparaison avec NULL |
| MIN(), MAX(), AVG(), SUM() | minimum, maximum, moyenne, somme |

**SQLite**

Pour ce cours nous allons utiliser [https://www.sqlite.or](http://www.sqlite.org/)g.

Ce SGBD-R est très populaire,sous licence libre, a la particularité d’être implémenté sous forme d’une biblio- thèque de moins de 1 Mio. Comme il est très compact, beaucoup d’applications embarquent une ou plusieurs bases SQLite. SQLite n’est pas client-serveur. SQLite fournit une interface en ligne de commande. La dis- tribution standard de python fournit le module [sqlite3](https://docs.python.org/3/library/sqlite3.html.) pour manipuler ces bases de données.

Très important: SQLite est très souple par défaut, c’est-à-dire qu’il tolère beaucoup d’approximations.

A télécharger et installer: <http://sqlitebrowser.org/>fournit une interface graphique. C’est ce logiciel que nous utiliserons pour les TP.

L’un des objectifs de ce cours est de maîtriser les bases du SQL. Donc ça se passe dans l’onglet ‘Exécuter le SQL’

* En début de TP : Fichier -> Ouvrir une base de données
* En fin de TP :Fichier -> Enregistrer les modifications
* Tout le SQL que vous allez saisir n’est sauvegardé nulle part : à vous d’enregistrer vos requêtes dans un éditeur de texte si vous souhaitez les conserver.
* Des requêtes ne sont pas un programme. Vous ne pouvez pas saisir toutes vos requêtes et cliquer sur Exécuter.
* Le résultat de l’exécution de vos requêtes est permanent.
* Les autres onglets ne sont qu’une aide graphique.

# Exercice : Internet Of Things (IoT)

L’internet des objets (Internet Of Things) désigne l’interconnexion entre Internet et des objets, des lieux et des environnements physiques.

Cette appellation désigne un nombre croissant d’objets connectés à Internet, permettant ainsi une communica- tion entre nos biens dits physiques et leurs existences numériques. Ces formes de connexions permettent de rassembler de nouvelles masses de données sur le réseau. (source : Wikipédia)

Dans ce contexte, nous souhaitons modéliser une plateforme de gestion de services :

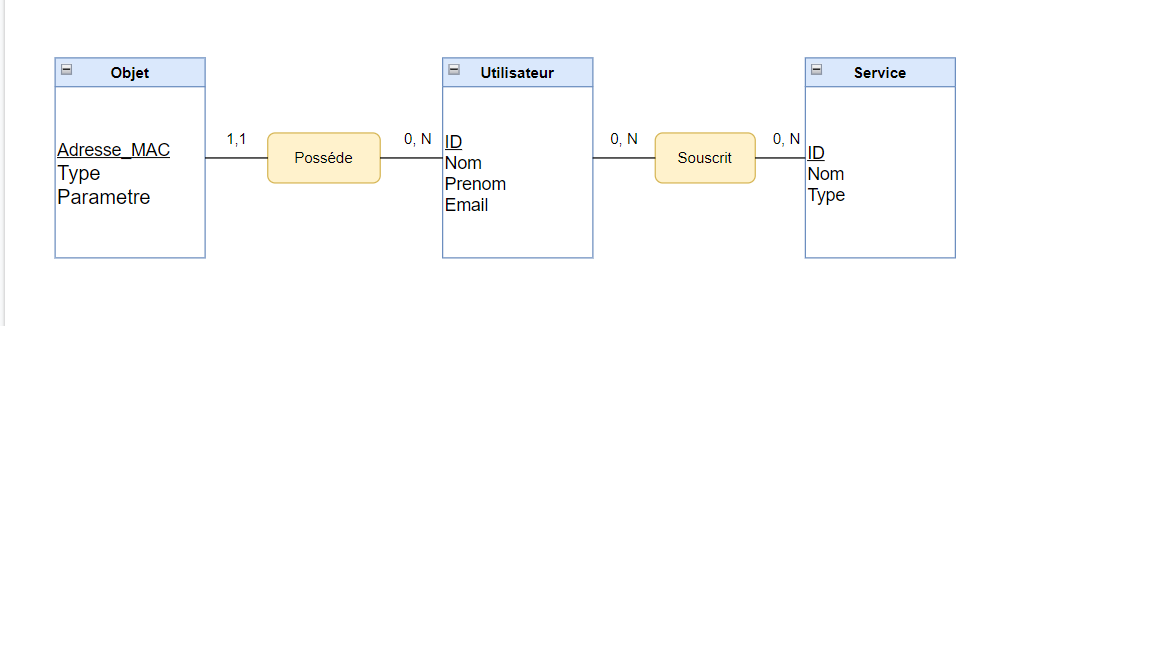
Des utilisateurs qui possèdent un ou plusieurs objets connectés souscrivent à des services. Lorsque les objets envoient des données, elles sont transmises aux services souscrits par leur propriétaire.

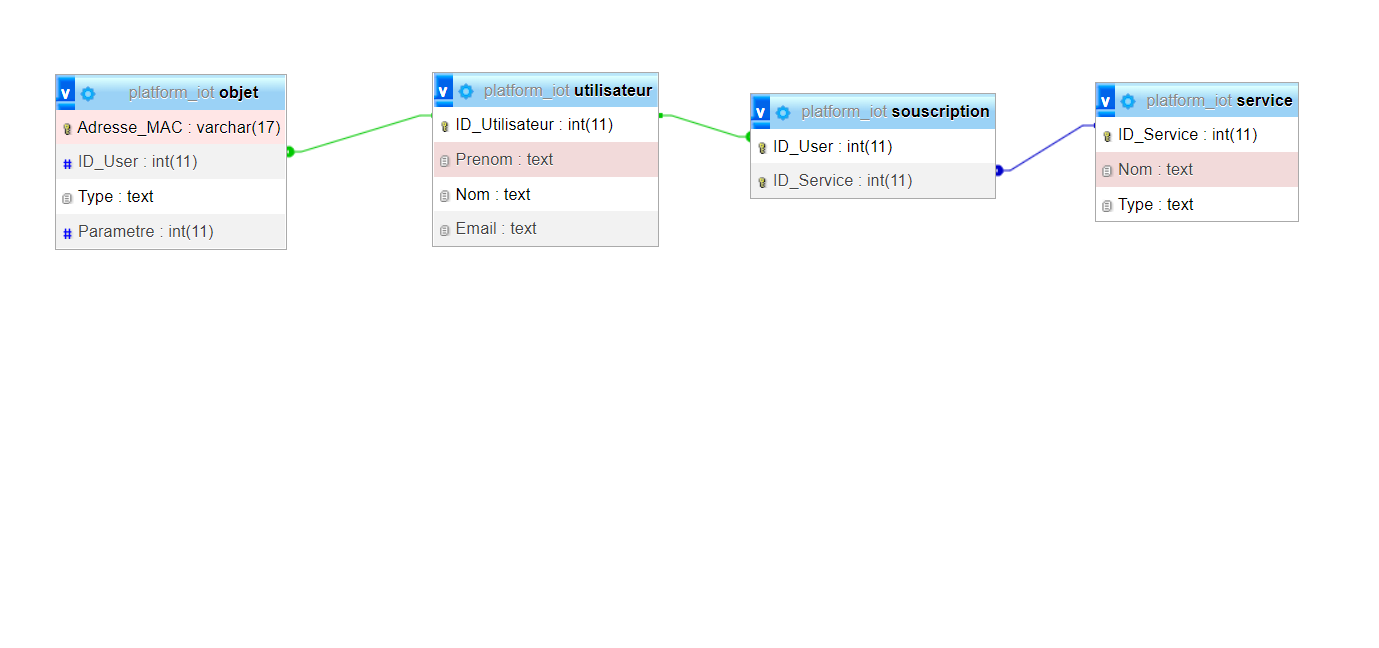
Afin de savoir à qui les objets connectés appartiennent et quels sont les services souscrits par leur propriétaire, nous utilisons une base de données.

* + Un utilisateur (Nom + Prénom) possède un identifiant unique ainsi qu’une adresse mail.
  + Chaque objet connecté est identifié par son adresse réseau physique unique (adresse MAC) sur 17 carac- tères, peut avoir un type particuliers et fournir des paramètres optionnels.
  + Un service est donné par son identifiant, éventuellement son nom, et peut être de type particulier.

## MCD / MLD

**Question 1 : Donner le MCD du contexte présenté**

**Question 2 : Donner le MLD à partir du MCD précédent**

****

## Créer et supprimer une table

**Langage SQL**

1

-- Requete SQL Pour creer une table CREATE TABLE Ma Table (

Cle\_primaire UN\_TYPE , Cle\_etrangere UN\_TYPE ,

PRIMARY KEY ( Cle\_primaire )

FOREIGN KEY ( Cle\_etrangere ) REFERENCES Autre Table ( Cle\_Primaire )

);

2

3

4

5

6

7

8

9

En cas d’erreur vous pouvez toujours recréer la table, pour celà, il vous faudra d’abord la supprimer avec la requête suivante :

1

-- Requete SQL Pour supprimer une table et toutes les donnees qu ' elle contient DROP TABLE Ma Table ;

2

3

**Question 1 : Créer une nouvelle Table**

Dans DB Browser créer une nouvelle base de données ”*Plat f orm*\_*IoT* "

Dans la base de données nouvellement créée, ajouter une nouvelle table "Utilisateur" avec les attributs définis dans le MCD / MLD de la question 1

**Correction :**

-- Question 1 : Créer une nouvelle Table

**CREATE** **TABLE** **IF** **NOT** EXIST Utilisateur**(**

ID\_Utilisateur **INTEGER** **PRIMARY** **KEY** AUTOINCREMENT**,**

Prenom TEXT **NOT** **NULL,**

Nom TEXT **NOT** **NULL,**

Email TEXT

**);**

**Question 2 : Faire de même avec les autres tables du MLD**

**Correction :**

-- Question 2 : Faire de même avec les autres tables du MLD

**CREATE** **TABLE** **IF** **NOT** EXIST Service**(**

ID\_Service **INTEGER** **PRIMARY** **KEY** AUTOINCREMENT**,**

Nom TEXT **NOT** **NULL,**

**Type** TEXT

**);**

**CREATE** **TABLE** **IF** **NOT** EXIST Objet**(**

Adresse\_MAC TEXT **PRIMARY** **KEY,**

ID\_User **INTEGER** **NOT** **NULL,**

**Type** TEXT**,**

Parametre **INT,**

**FOREIGN** **KEY(**ID\_User**)** **REFERENCES** Utilisateur**(**ID\_Utilisateur**)**

**);**

**CREATE** **TABLE** **IF** **NOT** EXIST Souscription**(**

ID\_User **INTEGER** **NOT** **NULL,**

ID\_Service **INTEGER** **NOT** **NULL,**

**PRIMARY** **KEY(**ID\_User**,** ID\_Service**),**

**FOREIGN** **KEY(**ID\_User**)** **REFERENCES** Utilisateur**(**ID\_Utilisateur**),**

**FOREIGN** **KEY(**ID\_Service**)** **REFERENCES** Service**(**ID\_Service**)**

**);**

## Insertion de données

**Langage SQL**

|  |  |  |
| --- | --- | --- |
| 1  2  3  4 | -- Creer un utilisateur en donnant les valeurs de chaque colonnes  INSERT INTO Utilisateur VALUES (1 , ' MARTIN ', ' Solveig ', ' martin . solveig@ ipsa .  fr ')  /\* Si on essaie de re - creer un utilisateur avec le meme ID , on obtient une erreur | |
| 5 | Il ne peut pas y avoir 2 lignes avec la meme valeur de cle pri | maire \*/ |
| 6 | INSERT INTO Utilisateur VALUES (1 , ' Dupont ', ' Rose ', ' rose . dup | ont@ ipsa . fr ') |
| 7 |  |  |
| 8 | -- Inserer une valeur en ne specifiant que les colonnes NOT NU | LL |
| 9 | INSERT INTO Utilisateur ( ID , nom , prenom ) VALUES (2 , ' Mickey ', | ' Mouse ') |
| 10 |  |  |
| 11 | -- Erreur si on ne donne pas la valeur d' une colonne NOT NULL |  |
| 12 | INSERT INTO Utilisateur ( ID , nom ) VALUES (3 , ' Dingo ') |  |
| 13 |  |  |

**Question 1 : Insérer un premier utilisateur**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Nom** | **Prenom** | **Mail** |
| 1 | Frederic | Sacre | [frederic.sacre@ipsa.fr](mailto:frederic.sacre@ipsa.fr) |

**Correction :**

-- Question 1 : Insérer un premier utilisateur

**INSERT** **INTO** Utilisateur**(**ID\_Utilisateur**,** Nom**,** Prenom**,** Email**)** **VALUES**

**(**1**,** 'Frederic'**,** 'Sacre'**,** 'frederic.sacre@ipsa.fr'**);**

**Question 2 : Insérer encore d’autres utilisateurs - N’utiliser qu’une seule requete SQL**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Nom** | **Prenom** | **Mail** |
| 2 | Heylen | Alexander | [alexandre.he](mailto:alexandre.heylen@gmail.com)[ylen@gmail.com](mailto:ylen@gmail.com) |
| 3 | Meunier | Anne | [anne.meunier@ipsa.fr](mailto:anne.meunier@ipsa.fr) |
| 4 | Garcia | Margaret | [margaret.g](mailto:margaret.garcia@spacebel.fr)[arcia@spacebel.fr](mailto:arcia@spacebel.fr) |
| 5 | Andre | Zacharie | [zacharie.andre@ipsa.fr](mailto:zacharie.andre@ipsa.fr) |
| 6 | Joly | Christine | [cjy@cnes.fr](mailto:cjy@cnes.fr) |
| 7 | Leleu | Maurice | NULL |
| 8 | Begue | Georges | [georges.be](mailto:georges.begue@ipsa.fr)[gue@ipsa.fr](mailto:gue@ipsa.fr) |
| 9 | Pereira | Agathe | [ag](mailto:agathe.pereira@ipsa.fr)[athe.pereira@ipsa.fr](mailto:athe.pereira@ipsa.fr) |

**Correction :**

-- Question 2 : Insérer encore d’autres utilisateurs - N’utiliser qu’une seule requete SQL

**INSERT** **INTO** Utilisateur**(**ID\_Utilisateur**,** Nom**,** Prenom**,** Email**)** **VALUES**

**(**2**,** 'Heylen'**,** 'Alexandre'**,** 'alexandre.heylen@gmail.com'**),**

**(**3**,** 'Meunier'**,** 'Anne'**,** 'anne.meunier@ipsa.fr'**),**

**(**4**,** 'Garcia'**,** 'Margaret'**,** 'margaret.garcia@spacebel.fr'**),**

**(**5**,** 'Andre'**,** 'Zacharie'**,** 'zacharie.andre@ipsa.fr'**),**

**(**6**,** 'Joly'**,** 'Christine'**,** 'cjy@cnes.fr'**),**

**(**7**,** 'Leleu'**,** 'Maurice'**,** **NULL),**

**(**8**,** 'Begue'**,** 'Georges'**,** 'georges.begue@ipsa.fr'**),**

**(**9**,** 'Pereira'**,** 'Agathe'**,** 'agathe.pereira@ipsa.fr'**);**

**Question 3 : Insérer les services suivants dans la table Service**

|  |  |
| --- | --- |
| **Nom** | **Type** |
| myKWHome | smarthome |
| FridgAlert | smarthome |
| RUNstats | quantifiedself |
| traCARE | quantifiedself |
| dogWATCH | NULL |
| CarUse | NULL |

**Correction :**

-- Question 3 : Insérer les services suivants dans la table Service

**INSERT** **INTO** Service**(**ID\_Service**,** Nom**,** **Type)** **VALUES**

**(**1**,** 'myKWHome'**,** 'smarthome'**),**

**(**2**,** 'FridgAlert'**,** 'smarthome'**),**

**(**3**,** 'RUNstats'**,** 'quantifiedself'**),**

**(**4**,** 'traCARE'**,** 'quantifiedself'**),**

**(**5**,** 'dogWATCH'**,** **NULL),**

**(**6**,** 'carUSE'**,** **NULL);**

**Question 4 : Insérer les objets connectés suivants dans la table Objet**

|  |  |  |  |
| --- | --- | --- | --- |
| **Adresse\_MAC** | **ID\_Utilisateur** | **type** | **param** |
| f0:de:f1:39:7f:17 | 1 | NULL | NULL |
| f0:de:f1:39:7f:18 | 2 | NULL | NULL |
| f0:de:f1:39:7f:19 | 2 | thingtempo | 60 |

**Correction :**

-- Question 4 : Insérer les objets connectés suivants dans la table Objet

**INSERT** **INTO** Objet**(**Adresse\_MAC**,** ID\_User**,** **Type,** Parametre**)** **VALUES**

**(**'f0:de:f1:39:7f:17'**,** 1**,** **NULL,** **NULL),**

**(**'f0:de:f1:39:7f:18'**,** 2**,** **NULL,** **NULL),**

**(**'f0:de:f1:39:7f:19'**,** 2**,** thingtempo**,** 60**);**

**Question 5 : Insérer les souscriptions adéquates dans la table Souscription**

|  |  |
| --- | --- |
| **ID\_User** | **ID\_Service** |
| 2 | 1 |
| 2 | 2 |
| 1 | 3 |

**Correction :**

-- Question 5 : Insérer les souscriptions adéquates dans la table Souscription

**INSERT** **INTO** Souscription**(**ID\_User**,** ID\_Service**)** **VALUES**

**(**2**,**1**),**

**(**2**,**2**),**

**(**1**,**3**);**

## Sélectionner et filtrer des données

**Langage SQL**

1

-- Lister tous les utilisateurs SELECT \* FROM Utilisateurs ;

-- En triant par nom et prenom

SELECT \* FROM Utilisateur ORDER BY nom , prenom ;

-- En triant par nom dans l' ordre croissant ( defaut ) SELECT \* FROM Utilisateur ORDER BY nom ASC ;

-- En triant par nom dans l' ordre decroissant SELECT \* FROM Utilisateur ORDER BY nom DESC ;

-- Recuperer uniquement le nom SELECT nom FROM Utilisateur ;

-- Recuperer tous les prenoms sans doublons SELECT DISTINCT prenom FROM Utilisateur ;

-- Compter le nombre d' utilisateurs SELECT COUNT (\*) FROM Utilisateur ;

-- Ceux qui s' appellent " Martin " ( peut - etre aucun )

SELECT nom , prenom FROM Utilisateur WHERE nom = ' Martin ';

-- Ceux dont le nom commence par la lettre D

SELECT nom , prenom FROM Utilisateur WHERE nom LIKE 'D%';

-- Ceux dont le nom fini par la lettre E

SELECT nom , prenom FROM Utilisateur WHERE nom LIKE '% E';

-- Ceux qui ont un ID < 3

SELECT \* FROM UTILISATEUR WHERE ID < 3;

-- Ceux qui possedent une adresse mail

SELECT \* FROM Utilisateur WHERE Email IS NOT NULL ;

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

**Question 1 : Lister les noms des services**

**Correction :**

-- Question 1 : Lister les noms des services

**SELECT** Nom **FROM** Service**;**

**Question 2 : Lister les noms et prénoms des utilisateurs par ordre croissant de nom**

**Correction :**

-- Question 2 : Lister les noms et prénoms des utilisateurs par ordre croissant de nom

**SELECT** Nom**,** Prenom **FROM** Utilisateur **ORDER** **BY** Nom **ASC** **;**

**Question 3 : Lister les utilisateurs qui n’ont pas d’adresse mail ’ipsa’**

**Correction :**

-- Question 3 : Lister les utilisateurs qui n’ont pas d’adresse mail ’ipsa’

**SELECT** **\*** **FROM** Utilisateur **WHERE** Email **NOT** **LIKE** '%ipsa%' **;**

**Question 4 : Afficher l’utilisateur avec l’ID 5**

**Correction :**

-- Question 4 : Afficher l’utilisateur avec l’ID 5

**SELECT** **\*** **FROM** Utilisateur **WHERE** ID\_Utilisateur**=**5**;**

**Question 5 : Lister les adresses MAC des objets connectés que possède l’utilisateur d’ID 2**

**Correction :**

-- Question 5 : Lister les adresses MAC des objets connectés que possède l’utilisateur d’ID 2

**SELECT** Adresse\_MAC **FROM** Objet **WHERE** ID\_User**=**2**;**

**Question 6 : Lister les adresses MAC des objets de type "thingtempo"**

**Correction :**

-- Question 6 : Lister les adresses MAC des objets de type "thingtempo"

**SELECT** Adresse\_MAC **FROM** Objet **WHERE** **type=**'thingtempo'

**Question 7 : Lister les ID et les noms de utilisateurs qui ne possèdent pas d’objet**

**Correction :**

-- Question 7 : Lister les ID et les noms de utilisateurs qui ne possèdent pas d’objet

**SELECT** **DISTINCT** ID\_Utilisateur**,** Nom **FROM** Utilisateur **WHERE** ID\_Utilisateur **NOT** **IN** **(SELECT** **DISTINCT** ID\_User **FROM** Objet**);**

**Question 8 : Lister les ID et les noms de utilisateurs qui possèdent des d’objets**

**Correction :**

-- Question 8 : Lister les ID et les noms de utilisateurs qui possèdent des d’objets

**SELECT** **DISTINCT** ID\_Utilisateur**,** Nom **FROM** Utilisateur **WHERE** ID\_Utilisateur **IN** **(SELECT** **DISTINCT** ID\_User **FROM** Objet**);**

## Mettre à jour les données

**Langage SQL**

|  |  |  |
| --- | --- | --- |
| 1 | -- Mettre a jour l' adresse mail d' un utilisateur choisi |  |
| 2 | UPDATE Utilisateur SET Email = " XXX@ ipsa . fr" WHERE ID = votre\_ID ; |
| 3 |  |
| 4 | -- Ajouter la colonne " Age " a la table Utilisateur |
| 5 | ALTER TABLE Utilisateur ADD Age INT NOT NULL DEFAULT ' 18 '; |
| 6 |  |
| 7 | -- Modifier le type de donnee du hamp Email de la table Utilisateur | pour |
| 8  9  10  11  12 | pouvoir contenir au maximum 200 caracteres  ALTER TABLE Utilisateur MODIFY Email VARCHAR (200)  -- Changer le nom du champ ID de la table Utilisateur pour devenir ID\_Utilisateur  ALTER TABLE Utilisateur CHANGE ID ID\_Utilisateur INT NOT NULL ; | |

**Question 1 : Mettre la valeur NULL pour toute la colonne type de la table service**

**Correction :**

-- Question 1 : Mettre la valeur NULL pour toute la colonne type de la table service

**UPDATE** Service **SET** **type=NULL;**

**Question 2 : Mettre le nom ’WatchDOG2’ pour la ligne qui a l’ID 5**

**Correction :**

-- Question 2 : Mettre le nom ’WatchDOG2’ pour la ligne qui a l’ID 5

**UPDATE** Service **SET** Nom**=**'WatchDOG2' **WHERE** ID\_Service **=** 5**;**

**Question 3 : Multiplier par 1.5 les valeurs de la colonne param des objets connectés**

**Correction :**

-- Question 3 : Multiplier par 1.5 les valeurs de la colonne param des objets connectés

**UPDATE** Objet **SET** Parametre **=** Parametre **\*** 1.5**;**

**Question 4 : Modifier le nom du champ Parametre de la table Objet pour devenir Param**

**Correction :**

-- Question 4 : Modifier le nom du champ Parametre de la table Objet pour devenir Param

**ALTER** **TABLE** Objet **RENAME** Parametre **TO** Param **;**

**Question 5 : Ajouter un champ "Anniversaire" à la table Utilisateur**

**Correction :**

-- Question 5 : Ajouter un champ "Anniversaire" à la table Utilisateur

**ALTER** **TABLE** Utilisateur **ADD** Anniversaire TEXT **;**

## Supprimer des données

**Langage SQL**

1

-- Supprimer l' utilisateur avec un nom donne DELETE FROM Utilisateur WHERE Nom = " Martin " ;

-- Supprimer toutes les lignes de la table DELETE FROM Ma Table ;

-- Supprimer une colonne de la table ALTER TABLE Ma Table DROP Colonne ;

2

3

4

5

6

7

8

9

**Question 1 : Supprimer l’utilisateur avec l’ID 1**

**Correction :**

-- Question 1 : Supprimer l’utilisateur avec l’ID 1

**DELETE** **FROM** Utilisateur **WHERE** ID\_Utilisateur **=** 1 **;**

**Question 2 : Supprimer la souscription de l’utilisateur d’ID 2 au service d’ID 1**

**Correction :**

-- Question 2 : Supprimer la souscription de l’utilisateur d’ID 2 au service d’ID 1

**DELETE** **FROM** Souscription **WHERE** ID\_User**=**2 **AND** ID\_Service**=**1**;**

**Question 3 : Supprimer toutes les lignes de la table Objet**

**Correction :**

-- Question 3 : Supprimer toutes les lignes de la table Objet

**DELETE** **FROM** Objet **;**

**Question 4 : Supprimer la colonne "Anniversaire"**

**Correction :**

-- Question 4 : Supprimer la colonne "Anniversaire"

**ALTER** **TABLE** Utilisateur **DROP** Anniversaire**;**

**Question 5 : Supprimer la table "Objet"**

**Correction :**

-- Question 5 : Supprimer la table "Objet"

**DROP** **TABLE** Objet**;**

## A votre tour !

Pour cette partie, récupérer la base de données "tp\_rugby.db" dans l’archive

**Question 1 : Donner la requête pour obtenir le nombre d’élèves qui ont entre 10 et 12 ans Question 2 : Donner la requête pour compter le nombre d’élèves qui se prénomment Pierre Question 3 : Donner la requête pour lister les élèves dont le nom ou le prénom est Leroy**

**Question 4 : Donner la requête pour obtenir la moyenne d’âge des élèves dont le nom commence par une voyelle Question 5 : Donner la requête pour obtenir le prénom des élèves triés par âge**

**Question 6 : Donner la requête pour obtenir la liste des élèves qui ont un nuéro de licence dont les derniers chiffres sont 42, triés par âge**

**Question 7 : Donner la requête pour lister les éducateurs qui n’ont pas de date de brevet**

**Question 8 : Donner la requête pour lister les élèves dont le nom est également le prénom d’un autre élève**