

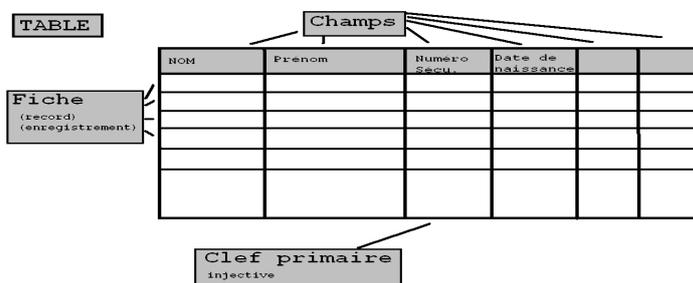
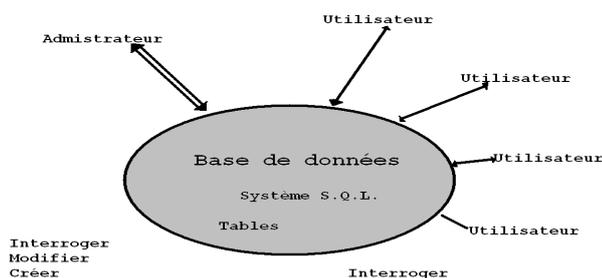
Une base de donnée (*B.D.D.* ou *B.D.D.R.* si relationnelle) est formée de tables qu'on peut interroger séparément mais aussi par jointure  $\bowtie$ .

Pour créer, gérer, interroger une base de données, on dispose d'un Système de Gestion de Bases de Données (*S.G.B.D.*, tel que *MySQL*, *SQLite*, *Oracle*, *Dbase*, *FoxBase*...) qui dispose d'interfaces plus ou moins conviviales et qu'on peut aussi piloter/émuler avec *Python* par exemple.

L'administrateur peut créer, effacer, modifier des tables, insérer des enregistrements.

Suivant son statut (*défini par l'administrateur*) un utilisateur peut interroger la base de données, et éventuellement insérer des enregistrements ou en détruire.

Plusieurs utilisateurs peuvent théoriquement interroger simultanément la base de données sans qu'il n'y ait de problème.



Une table est formée d'enregistrements (*en très grand nombre assez souvent*).

Ces enregistrements sont faits de champs de type numérique/alphabétique/date, de format prédéfini, complétés ou non.

Parmi les différents champs, certains sont des clefs (*il n'y a pas deux enregistrements ayant la même valeur de clef*). L'une d'entre elles sera la clef primaire et servira à relier des tables entre elles.

Ce sera la principale compétence attendue de vous aux concours.

- On peut lire tout le contenu d'une table

```
SELECT * FROM (nom_table) 1.
```

- On peut ne lire que certains champs

```
SELECT (nom_champs) FROM (nom_table).
```

- On peut ne consulter que certaines fiches

```
SELECT (nom_champs) FROM (nom_table) WHERE (condition).
```

- On peut effectuer des calculs et statistiques sur certains champs de certaines fiches

```
SELECT COUNT/MAX/MIN FROM (nom_table) WHERE (condition).
```

- On peut lire des informations issues de plusieurs tables ensemble :

<sup>1</sup> la distinction majuscules/minuscules n'est qu'une convention sans importance mais qui permet de s'y retrouver à la lecture

SELECT (nom_champs.table)	FROM (nom_table) JOIN (autre_table) ON (condition)	WHERE (condition)
---------------------------	---	-------------------

la jointure est une opération qui crée une énorme table “produit cartésien des deux tables en jeu”, n’en garde qu’une partie grâce à une condition introduite par ON.

Les mots et syntaxes à connaître/maîtriser :

SELECT		FROM		WHERE		SORT BY	
	nom des champs séparés par des virgules		nom d’une table		condition booléenne		critère
	COUNT(*) / MAX(champ) / MIN(champ) / AVG(champ) / SUM(champ)		ou jointure de plusieurs tables avec condition de jointure introduite par ON et une for- mule booléenne portant sur les deux tables		tests =, <, > LIKE “%_ %” NOT, AND, OR BETWEEN		
DELETE INSERT UPDATE		INTO				GROUP BY	

Quand deux tables interviennent par jointure (*ou réunion*), il faut préciser de quelle table est issu le champ (*structure table.champ ; exemples : clients.nom, consultations.date, ...*).

Si le nom d’une table devient rébarbatif à écrire, on peut l’alléger par *alias* :

SELECT cl.nom, cons.date FROM clients AS cl, consultations AS cs WHERE ...

MPSI 2/2014	<b>Administrateur</b>	IPT
-------------	-----------------------	-----

Il s’agit ici des prérogatives de l’administrateur :

CREATE DATABASE	créer une base relationnelle
CREATE TABLE	créer une table, en définissant ses champs, leur type, diverses conditions <sup>a</sup> , en spécifiant qui sera clef primaire...  <sup>a</sup> on peut ainsi demander que le contenu d’un champ ne puisse faire partie que d’une liste prédéfinie
ALTER TABLE	supprime ou ajoute des champs dans une table
DESCRIBE	
UPDATE	met à jour le contenu d’une table
DROP	efface une table
GRANT	définit les droits d’accès des différents utilisateurs

On peut mettre à jour une base de donnée en important directement un fichier “texte” contenant déjà toutes les données utiles. Avec Python, on utilisera `load()`, `read()`, `readln()` et `close()`.

*Il est peu probable que les concours vous posent ce type de questions.*

M.P.S.I.2 2014	0 points	2015 Charlemagne
		p IPT q