

Exercice 1 :

Par l'intermédiaire de *PHPMyAdmin* ou du client `mysql`, connectez-vous sur votre base de données.

```
mysql -h dbs-perso.luminy.univmed.fr -p -D votre_bd -u votre_login
```

ou

```
http://phpmyadmin.luminy.univmed.fr
```

1. Créez la table suivante **bank** (choisissez soigneusement les types de vos attributs et vos clés primaires) :

ensemblGID	cloneID	bank	plate	position
ENSMUSG038580	H3001A02	FM	001	A02
	H3001H05	FM	003	D01
ENSMUSG023353	H3002B09	FM	001	D01
	634636	MLN	1553	A05
ENSMUSG022811	572816	MTB	LLAM1382	A09
ENSMUSG017335	H3002A01	FM	001	B05

- Entrez les enregistrements successivement
 - En reprenant le TP1, écrivez le fichier `.txt` que vous auriez utilisé si vous pouviez charger les données globalement sous la forme d'un fichier texte (séparer les champs par une tabulation). Vérifiez avec `LOAD DATA` qu'il fonctionne.
2. Affichez les 2 premières lignes de la table **bank** qui concernent les clones de la banque FM.
 3. Sélectionnez les *cloneID* dont l'*ensemblGID* est connu (sous la forme *cloneID*, *ensemblGID*).
 4. Donnez la liste des *cloneID* de la banque FM stockés dans la table **bank**.
 5. Comptez le nombre de clones de la banque FM stockés dans la table dont l'*ensemblGID* est connu.
 6. Créez la table suivante: **bank2**.

cloneID	gene
634636	CLB1
129741	
H301667	TRF4
H3002A01	XRS2

7. Sélectionnez les *cloneID* présents à la fois dans la table **bank** et dans la table **bank2**.
8. Faites une jointure de la table **bank2** sur la table **bank**. La table résultante doit afficher tous les attributs de la table **bank** ainsi que l'attribut *gene* de la table **bank2**.

Exercice 2 :

1. Sous *PHPMysqlAdmin* ou le client `mysql`, créez une table **yeast_table1** contenant les informations suivantes :

SGD_ID	ORF	chromosom	gene
S000028864	YAL069w	1	RBG1
S000002143	YBR009c	2	FUN12
S000031098	YHL188c	8	MTW1
S000028866	YIL274w	9	POP5

2. Créez la table suivante dans votre base de données: **yeast_table2**

SGD_ID	function	length
S000031098	unknown	1358
S000028864	Telomeric maintenance	2894
S000035336	aging	987

3. A l'aide d'une jointure entre les tables **yeast_table1** et **yeast_table2**, créez une table globale **yeast_table3** contenant les attributs *SGD_ID*, *gene*, *function* et *length*.
4. Au lieu d'afficher la table **yeast_table3** à l'écran, imprimez la dans un fichier **yeast_table3.txt**. Ceci peut être fait en exécutant une commande `mysql` à partir du shell (et non plus en mode interactif) :

```
echo 'ma requête SQL' | mysql -h monserveur -u monlogin -D mabasededonnées > yeast_table3.txt
```

L'opérateur `|` permet à une commande d'envoyer son résultat vers une autre commande.
L'opérateur `>` permet à une commande d'écrire dans un fichier au lieu d'afficher à l'écran.
On peut aussi mettre une série de requêtes SQL dans un fichier texte :

```
mysql -h monserveur -u monlogin -D mabasededonnées < fichier.sql > yeast_table3.txt
```

Exercice 3 :

Reprenons notre base de données aérienne. En SQL, exprimez les requêtes suivantes :

- Jointures :
 1. Pour chaque pilote en service, quels sont les numéros des avions pilotés et le nom du pilote ?
 2. Quels sont les noms des pilotes qui pilotent un avion au départ de Paris ?
 3. Quelle est la capacité de chaque vol ?
- Mises à jour :
 4. (INSERT) Ajouter un avion de type 'A 330', de capacité 250, et localisé à Marseille.
 5. (INSERT) Ajouter l'avion immatriculé 'YT-567', type 'B 767', de capacité 300, localisé à Paris.
 6. (DELETE) Supprimer dans la relation VOL le vol de numéro 4 (bien entendu, créez un vol de numéro 4 avant, pour pouvoir tester).
 7. (UPDATE) Changer le type de l'avion de numéro 6 en 'B 777'.
- Double jointure (relier 3 tables) :
 8. Quels sont les noms des pilotes qui pilotent un Airbus ?

- Auto-jointure (jointure d'une table avec elle-même) + surnommage (SELECT `PILOTE_ID` AS "Numéro"). Le mot-clé 'AS' est optionnel mais contribue à la lisibilité :
 9. Quels sont les numéros des pilotes qui pilotent à la fois les avions de numéros 3 et 4 ?
 10. Quels sont les noms des pilotes qui pilotent un avion piloté aussi par le pilote numéro 3 ?
- Sous-requête + fonction MAX () :
 11. Quelles sont les villes desservies par les pilotes dont le numéro est plus grand que celui de Ringo et Paul ?
- Complémentaire (NOT IN ou éventuellement NOT EXISTS) + sous-requête :
 12. Quelles sont toutes les caractéristiques des avions qui ne participent à aucun vol ?
 13. Quels sont les noms des pilotes qui n'effectuent aucun vol au départ de Paris ?
- Double sous-requête avec complémentaire et projection distincte (DISTINCT) :
 14. Quels sont les numéros des pilotes qui pilotent tous les avions de la compagnie ?

Pour pouvoir répondre à cette question, répondez, dans l'ordre aux questions suivantes :

 15. Donnez les couples (AVION_ID, PILOTE_ID) de tous les vols, sans doublon,
 16. Donnez tous les couples (AVION_ID, PILOTE_ID) possibles,
 17. Donnez parmi tous les couples (AVION_ID, PILOTE_ID) possibles, ceux qui n'apparaissent pas dans VOL,
 18. Donnez les numéros des pilotes qui ne pilotent pas tous les avions (= les PILOTE_ID de tous les couples possibles (produit cartésien de AVION_ID * PILOTE_ID) qui n'apparaissent pas dans VOL).
 19. Quels sont les noms des pilotes qui pilotent au moins tous les Airbus de la compagnie? Pour répondre à cette question, énoncez et répondez d'abord aux 4 questions équivalentes.
- Agrégats (GROUP BY et HAVING) :
 20. (fonction COUNT (*)) Comptez le nombre d'Airbus de la compagnie.
 21. Donnez les couples (AVION_ID, PILOTE_ID) de VOL sans doublons pour lesquels l'avion est un Airbus.
 22. Donnez pour chaque pilote son numéro et le nombre d'Airbus (différents) pilotés.
 23. Répondez aux questions 14 et 19 grâce à des agrégats.

Exercice 4 :

1. Réalisez un *dump* d'une partie de la base de données d'ENSEMBL :

```
mysqldump -h ensembl.db.ensembl.org -P 4306 -u anonymous --skip-opt --no-data --skip-comments --add-drop-table saccharomyces_cerevisiae_core_47_1g > saccharomyces.sql
```

Le fichier `saccharomyces.sql` contient le schéma relationnel de toutes les tables de la base de données `saccharomyces_cerevisiae_core_47_1g` mais ne contient aucune donnée (`--no-data`). Plus d'informations sur <http://dev.mysql.com/doc/refman/5.0/fr/mysqldump.html>.
2. Ouvrez ce fichier, repérez comment on définit une clé primaire, une clé unique, une clé simple (appelé aussi *index*). Quelles sont les propriétés de ces trois objets ? Leurs différences ? Notez au passage qu'une clé peut être un n-uplet.
3. Réalisez un *dump* des données de la table `external_db` dans le fichier `external_db.sql`. Ce *dump* doit contenir uniquement les données, pas la création de la table.

En conjonction avec la commande `source`, vous pouvez de cette façon transférer une base de données d'une machine à une autre.

Exercice 5 :

En SQL standard, il est possible de rajouter des contraintes sur les valeurs des attributs (exemple, s'assurer qu'un pourcentage est bien inférieur à 100, ou que les dates d'arrivées des vols sont supérieures aux dates de départ, ...). En MySQL, ce genre de contrainte du style

```
create table VOL (  
    ...  
    CONSTRAINT chkTime CHECK (`HEURE_DEP` < `HEURE_ARR`)  
)
```

n'est pas implémenté.

1. Heureusement, les triggers (*déclencheurs*) existent : Ouf ! Un trigger est une fonction qui sera appelée dès qu'une condition sera remplie. Récupérez le fichier *Trigger.sql* et activez le trigger (via la commande *source*). Insérez un nouveau vol avec une mauvaise date d'arrivée et vérifiez que MySQL rejette l'insertion.
 2. **Question subsidiaire** : créez un second *trigger* vérifiant qu'un avion n'effectue pas 2 vols différents en même temps (vérifier que deux vols différents avec le même avion).
- Pour voir les triggers : `SHOW TRIGGERS;` . Pour supprimer un trigger spécifique: `DROP TRIGGER chkTime;`

Exercice 6 :

Au moment de l'installation d'un serveur MySQL, un compte administrateur de bases de données (DBA) *root* est créé et un mot de passe est demandé à l'administrateur.

1 - Création de comptes

- Le DBA se connecte avec la commande : `mysql -u root -p`
- Pour créer une base de données :
`CREATE DATABASE IF NOT EXISTS `tichitdb`;`
- Pour créer un utilisateur 'tichit' ayant comme mot de passe 'passwd de tichit' et ayant tous les droits sur cette base de données uniquement :
`GRANT ALL PRIVILEGES ON `tichitdb`.* TO 'tichit'@'localhost' IDENTIFIED BY 'le passwd de tichit';`
- 1. Comment faire pour donner des droits plus globaux (pas spécifiques à une base de données particulière) ?
- 2. Comment faire pour donner par la suite des droits particuliers à une table précise (droits locaux?).

2 - Supprimer les droits

- Comment enlever des droits en écriture sur une base de données particulière ?

Exercice 7 :

Les vues ont trois rôles :

- Créer une table virtuelle regroupant le résultat d'une requête `SELECT` pour lui donner des privilèges spécifiques. Exemple :

```
create view PAYE_PILOTE as  
    select nom, prenom, num_secu, salaire  
    from PILOTE;
```

Le service financier aura des privilèges sur cette vue que n'aura pas le service de contrôle aérien par exemple. Bien entendu il est fréquent qu'une vue soit le résultat d'une *jointure*.
- Stocker de façon efficace le résultat d'une requête que l'on utilisera fréquemment (comme sous-requête par exemple).
- Jouer le rôle de filtre entre l'utilisateur et les tables en cas de mise à jour des données.

1. Créez une vue regroupant vol, avion et pilote et affichant tous les attributs sauf les ID. Visualisez-là.
2. Créez une table (`create table nomDeMaTable as ...`) regroupant les mêmes éléments. Visualisez-là. Il ne devrait pas y avoir de différence avec la vue.
3. Faites une requête de type *update* dans la vue pour changer l'immatriculation de l'un des avions les plus fréquemment utilisés.
4. Le résultat s'est-il répercuté pour toutes les immatriculations identiques de la vue ?
5. Le résultat s'est-il répercuté dans la (nouvelle) table ?
6. Le résultat s'est-il répercuté sur la table AVION ?
7. Faites une seconde vue dont le résultat est un `select * from VOL` mais pour lequel on assure que la date d'arrivée est supérieure ou égale à la date de départ + 1 heure. Votre création de vue doit finir par `WITH CHECK OPTION`.
8. Supprimer les éventuels triggers en *insertion* sur la table VOL pour éviter tout conflit.
9. Insérez dans cette vue un mauvais vol. Résultat ?
 - Pour voir les vues : `SHOW TABLES;` (pas terrible : ils auraient pu nommer ça « show views »)
 - Pour supprimer une vue : `DROP VIEW maVue;`