

TP MySQL N° 2

Licence 2 Informatique

Université du Littoral Côte d'Opale

Document à rendre :

un rapport de TP qui contient

les commandes utilisées,

avec description et références à l'énoncé,

et commentaires éventuels.

**Attention**

| Lisez ATTENTIVEMENT les explications et les consignes de travail.

Préparer l'environnement de travail et accéder au SGBDR MySQL

(se reporter au support TP1)

1.1 A réaliser pour l'OS Linux

- ouvrir une session en utilisant vos identifiants
- lancer un premier Terminal (dans Applications > Accessoires)
- afficher le nom du répertoire courant (commande **pwd** : 'print working directory')
- lister le contenu du répertoire courant (commande **ls** : 'list')
- créer un répertoire de travail Tp2SQL (commande **mkdir** : 'make directory')
- se placer dans ce répertoire (commande **cd** : 'change directory')
- lancer un éditeur pour créer le fichier de script du Tp2 : **gedit** tp2.sql ou **geany** tp2.sql : vous remettrez ce fichier à la fin de chaque séance
- compléter le fichier avec les informations suivantes (les 3 premières lignes comme ci-dessous, le reste sera complété au fur et à mesure du TP):

```
-- Date : 18/11/2020
-- Classe : L2Info
-- Etudiant : Jacques Dupont / Magali Dubois

-- exercice 1
    requete SQL    exercice 1
-- question 2
    -- reponse a la question 2
```

- enregistrer ce fichier (mais le laisser ouvert !)

Attention

Remarque : vous conserverez une trace numérique de toutes les actions réalisées dans le fichier de scripts. Cela vous permettra de relancer des ensembles de commandes en une seule fois. Vous remettrez le fichier à la fin de chaque séance.

Attention

Chaque requête SQL devrait d'abord être écrite dans le fichier de script , puis copiée derrière l'invite mysql pour être exécutée (Ctrl-C pour copier, clic-droit+Coller dans la fenêtre en mode commande). En cas d'erreur de syntaxe dans une requête, la modification devrait être réalisée dans le fichier de script avant d'être soumise à nouveau. (Pensez à enregistrer le fichier régulièrement)

1.2 Rappels : structure d'accueil, contraintes et contenu

1. DDL : `create table` : créer une structure d'accueil des données
2. DDL : `alter table` + `add constraint` : ajouter des contraintes afin que le SGBD assure l'intégrité d'entité (pas de doublon dans la clef primaire) et l'intégrité référentielles (la valeur d'une clef étrangère doit se retrouver comme clef primaire dans une autre table)
3. DML : `insert into`, `update`, `delete from` : gérer le contenu des tables

1.3 Créer les tables de la BD 'cinema' : DDL

1.3.1 A réaliser

1. récupérer le fichier 'bdcine_DDL.sql'
2. modifier le fichier pour remplacer le nom de la base de données : il s'agit de votre login au lieu de 'cinema9'
3. après l'invite de commande MySQL, taper la commande suivante en adaptant le chemin d'accès au fichier :

```
mysql > source c:\temp\bdcine_DDL.sql
```

4. vérifier quelques tables en utilisant la commande MySQL 'describe' :

```
mysql > describe acteur;
```

1.4 Ajouter des données (DML) LOAD DATA

Vous avez créé la structure d'accueil des données. Celle-ci est vide. Vous insérer les données de la base de données 'cinema' (créer pour le TP 1) pour les exercices de ce TP.

Vous allez maintenant charger 'sérieusement' les tables de la base de données.

L'ordre INSERT INTO est très efficace et sûr. Cependant, pour des volumes de données importants, les SGBD proposent des outils pour le chargement de données à partir de fichiers 'plats', fichiers textes bruts, dans lesquels :

— chaque ligne est séparée de la suivante par un retour à la ligne,

— chaque colonne est séparée de la suivante par un caractère donné : ',' , ' par exemple ;

C'est le format Comma Separated Values (CSV) des exportations à partir des feuilles de calcul, par exemple. Exemple de fichier 'salle.csv' :

```
"num_salle","nombre_places" 1,50
2,70
3,250
```

...

MySQL propose l'ordre LOAD DATA INFILE :

LOAD DATA LOCAL INFILE 'c:/temp/salle.csv' INTO TABLE salle FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"' IGNORE 1 LINES;

Soit :

- chargement du contenu du fichier 'c:/temp/salle.csv' (exemple sous Windows) — dans la table 'salle',
- les colonnes sont séparées par ',' ,
- les chaînes de caractères peuvent être encadrées par le caractère '"' (guillemets), — et la première ligne du fichier est ignorée (c'est une ligne d'entête de colonne)

ATTENTION : l'ordre des colonnes du fichier texte doit être le même que celui des colonnes de la table.

1.4.1 A réaliser

1. récupérer le fichier archive 'cinema.zip' (le site de téléchargement vous sera fourni)
2. en extraire les fichiers textes CSV
3. placer ces fichiers dans votre répertoire TpSQL
4. quitter la session MySQL et rouvrir une session MySQL en utilisant la commande mysql avec le paramètre --local-infile qui autorisera le fait d'envoyer un fichier d'un poste client vers le serveur

sur linux

```
> mysql -h 192.168.22.48 --local-infile -u john -p
```

Sur windows

set local-infile=1;

OU SET GLOBAL local_infile = true;

Pour le verifier: SHOW GLOBAL VARIABLES LIKE 'local_infile';

Sur MySQLWorkbench : éditer la connexion puis dans 'Advanced>OTHERS' ajouter la ligne : OPT_LOCAL_INFILE=1

Pour retrouver le chemin sécurisé :

SHOW GLOBAL VARIABLES LIKE '%SECUREFILE%';

OU SHOW VARIABLES LIKE "secure_file_priv";

OU SELECT @@global.secure_file_priv;

Vous devrez placer les fichiers à importer sur ce chemin sécurisé.

Faites attention aux slashes (barres obliques "/") ou anti-slashes dans le chemin. Ils doivent être remplacés par "/" ou par double antislash "\\".

```
LOAD DATA LOCAL INFILE "C:/ProgramData/MySQL/MySQL Server
8.0/Uploads/film.csv "
INTO TABLE film
FIELDS TERMINATED BY ','
OPTIONALLY ENCLOSED BY '"'
IGNORE 1 LINES;
```

5. utiliser l'ordre LOAD DATA pour charger chacun des fichiers dans les tables créées précédemment.

— modifier l'adresse du fichier à charger en spécifiant une adresse relative à votre répertoire de connexion (sous Linux :)

— attention : les tables ne peuvent être chargées dans n'importe quel ordre : les contraintes d'intégrités référentielles nécessitent que certaines tables soient chargées avant d'autres...

6. vérifier le bon chargement en listant le contenu de chacune des tables

```
mysql > SELECT * FROM acteur;
```

7. la colonne photo_act de la table acteur va nous gêner : la supprimer
8. la colonne resume_film de la table film va nous gêner : la supprimer
9. ajouter le realisateur de numéro 14, Georges Lucas, année de naissance à null
10. ajouter les films de Georges Lucas suivants : 13, red Tails, sortie en 2012, et 14, Star wars et 15, Indiana jones4 sortis en 2008 ; on ne connaît ni le budget, ni des résumés de ces 3 films (valeurs null)

1.5 Interroger la base de données (DQL) : projection,sélection, classement

1.5.1 A réaliser

En utilisant le support SQL au chapitre traitant des ordres DQL, effectuer les opérations suivantes :

1. Lister les film dont le numéro de réalisateur est 1, 4, 7, ou 14 ou 2540

r	num_film	titre_film	anSortie_film	budget_film	genre_film	num_real	resume_film
	1	The Last Samurai	2003	8000000	épopée	1	
	4	Stella	2008	4000000	comédie dramatique	7	

	13	red tails	2012	null	null	14	null
	14	star wars	2008	null	null	14	null
	15	Indiana jones4	2008	null	null	14	null

nombre de lignes : 6

2. Lister les acteurs dont le nom commence par 'isa'

r	num_act	nom_act	anNais_act
	9	Isabelle Huppert	1953
	18	Isabelle Ferrari	1964
	27	Isabelle Adjani	1955

nombre de lignes : 3

3. Lister les acteurs dont le nom se termine par 'ani'

r	num_act	nom_act	anNais_act
	27	Isabelle Adjani	1955

nombre de lignes : 1

4. Lister les acteurs dont le nom contient 'ol' et ne contient pas 'go'

r	num_act	nom_act	anNais_act
	4	Angelina Jolie	1975
	12	Nicolas Giraud	1978
	20	Karole Rocher	1974
	22	Benjamin Biolay	1973
	25	André Dussolier	1946

nombre de lignes : 5

5. Lister les acteurs dont le nom comporte la lettre 'n' en 3ème position

r	num_act	nom_act	anNais_act
	6	Ben Stiller	1965
	15	Nanni Moretti	1953
	22	Benjamin Biolay	1973
	26	Annie Cordy	1928

nombre de lignes : 4

6. Lister le numéro de film, son titre et le numéro de réalisateur du film dont le titre est 'home'

r	num_film	titre_film	num_real
	7	Home	4

nombre de lignes : 1

7. Lister les acteurs nés avant 1950 ou après 1960 et dont le nom contient l'une des syllabes suivantes : 'cru', 'roc' et 'cor'

r	num_act	nom_act	anNais_act
	1	Tom Cruise	1962
	20	Karole Rocher	1974
	26	Annie Cordy	1928

nombre de lignes : 3

8. Lister les réalisateurs dont l'année de naissance n'est pas renseignée

r	num_real	nom_real	anNais_real
	14	Georges Lucas	null

nombre de lignes : 1

9. Lister les films sortis en 2005 ou après, ayant un numéro pair et dont le genre ne contient pas 'drame' (il existe un opérateur modulo comme en C/C++)

r	num_film	titre_film	anSortie_film	budget_film	genre_film	num_real	resume_film
	2	Woman on the beach	2007	9000000	Comédie dramatique	9	
	4	Stella	2008	4000000	comédie dramatique	7	
	8	Tonnere sous les tropiques	2008	6000000	Comédie, Aventure, action	3	

nombre de lignes : 3

10. utiliser les fonctions SQL month et now : select month(now) from dual; (remarquer le format de la date : aaaa-mm-jj)

r	now()	month(now())	year(now())
	2018-11-20 14 :52 :54.0	11	2018

nombre de lignes : 1

11. Lister les projections de films en mars 2017 classées par numéro de film

r	num_film	num_salle	date_projection	heure_projection	nbSpectateurs_proj
---	----------	-----------	-----------------	------------------	--------------------

nombre de lignes : 0

12. Lister les différents numéros de films projetés dans une séance entre 18 :00 et 20 :00

r	num_film
	1
	4
	...
	9
	10
	11

nombre de lignes : 7

13. Lister le nom et la longueur du nom des acteurs : nommer cette dernière colonne 'longueur' (cf. alias de colonne, opération de renommage)(il existe une fonction length qui retourne la longueur d'une chaîne)

r	nom_act	longueur
	Tom Cruise	10
	Jean-Pierre Darrousin	21

	Isabelle Adjani	15
	Clint Eastwood	14
	Sophie Marceau	14

nombre de lignes : 29

14. Lister les acteurs dont la longueur du nom est supérieure à 15 caractères

r	num_act	nom_act	anNais_act
	2	Jean-Pierre Darrousin	1953
	8	Robert Downer jr	1965

	18	Isabelle Ferrari	1964
	21	Melissa Rodrigues	1988
	23	Guillaume Depardieu	1971

nombre de lignes : 7

15. Lister les noms des réalisateurs en lettres capitales

r	upper(nom_real)
	EDWARD ZWICK
	CLINT EASTWOOD
	...
	JOANN SFAR
	CLAUDE PINOTEAU
	GEORGES LUCAS

nombre de lignes : 14

16. Lister les prénom et nom des acteurs (le prénom est situé avant le 1er espace de nom_act, le nom après...) (Index : fonctions locate, substring et length)

r	prenom	nom
	Tom	Cruise
	Jean-Pierre	Darrousin

	Isabelle	Adjani
	Clint	Eastwood
	Sophie	Marceau

nombre de lignes : 29

17. Lister les acteurs classés par année de naissance (croissant) puis par nom (croissant)

r	num_act	nom_act	anNais_act
	26	Annie Cordy	1928
	14	Horst Westphal	1929

	11	Salomé Stéverin	1985
	21	Melissa Rodrigues	1988
	19	Leora Barbara	1996

nombre de lignes : 29

18. Lister les projections classées par tarif, puis heure de projection décroissante et date de projection décroissante

r	date_projection	heure_projection	tarif_projection
	2017-02-14	16 :20 :00	3.00
	2017-02-14	14 :00 :00	3.00

	2017-02-19	20 :15 :00	6.00
	2017-03-01	20 :10 :00	6.00
	2017-02-13	20 :20 :00	6.50

nombre de lignes : 18

2 Deux parenthèses avec le produit cartésien

2.1 Parenthèse 1 : inventer un nom d'entreprise

Vous êtes chargé de découvrir un nom pour une entreprise : pour vous aider, vous allez créer une requête qui va lister toutes les possibilités d'assembler plusieurs mots pour former un nom.

2.1.1 à réaliser

1. créer la table 'mots' comportant une colonne 'mot' de type chaine variable de longueur maximale 4
2. ajouter les mots suivants dans la table : 'info','tech','net','comm'
3. créer une requête donnant toutes les combinaison possibles de 2 mots
4. modifier la requête précédente en définissant une seule colonne résultat (indice : fonction concat)
5. améliorer la requête pour qu'elle donne tous les agencements possibles de 2 mots différents (indice : sélection des mots différents)
6. créer une requête donnant tous les agencements possibles de 3 mots différents
7. créer une requête donnant tous les agencements possibles de 2 ou 3 mots classés alphabétiquement (indice : union)

r	entreprise
	comminfo
	comminfonet
	...
	technet
	technetcomm
	technetinfo

nombre de lignes : 36

2.2 Parenthèse 2 : Aider Mr Jourdain à composer son billet doux...

Vous allez aider Mr Jourdain à composer sa célèbre tirade (dans le Bourgeois Gentilhomme) en lui proposant toutes les possibilités d'agencement de 5 termes : "On les peut mettre premièrement comme vous avez dit : "Belle Marquise, vos beaux yeux me font mourir d'amour". Ou bien : "D'amour mourir me font, belle Marquise, vos beaux yeux". Ou bien : "Vos yeux beaux d'amour me font, belle Marquise, mourir". Ou bien : "Mourir vos beaux yeux, belle Marquise, d'amour me font". Ou bien : "Me font vos yeux beaux mourir, belle Marquise, d'amour." Le Bourgeois gentilhomme, II, 4.

2.2.1 à réaliser

1. créer la table 'termes' comportant une colonne 'terme' de type chaîne variable de longueur maximale 16
2. ajouter les mots suivants dans la table : 'belle marquise ', 'vos beaux yeux ', 'me font ', 'mourir', 'd'amour'
3. créer la requête qui va permettre de lister toutes les permutations possibles des 5 termes différents

r	tirade
	belle marquise d'amour me font mourir vos beaux yeux
	belle marquise d'amour me font vos beaux yeux mourir
	...
	vos beaux yeux mourir d'amour me font belle marquise
	vos beaux yeux mourir me font belle marquise d'amour
	vos beaux yeux mourir me font d'amour belle marquise

nombre de lignes : 120