Difference Par rapport a oracle

Certaines particularités par rapport à Oracle
Les fichiers de journalisations Contrairement à Oracle, MySQL ne possède pas toujours des fichiers de journalisations. Cela dépend de son engin (voir plus bas) de tables. Ce qui veut dire qu'il est possible que les données soient perdues si la base de données plante. Ouch !

Les fichiers de données
Oracle groupe les fichiers de données selon leur tablespace. Par exemple USER01.DBF et USER02.DBF. Dans le cas de tablespaces permanents, ces fichiers peuvent contenir des index ou des tables.

MySQL regroupe les fichiers de données selon un schéma/base de données. Chaque fichier correspond à une table.

Exemple : nomBaseDeDonnées/table1.frm nomBaseDeDonnées/table2.frm

L'installation et l'administration
Le niveau de connaissance afin d'administrer et de gérer une base de données MySQL est moindre que pour Oracle. En effet, son installation est rapide et la courbe d'apprentissage du fonctionnement de MySQL est inférieure à Oracle.

La base de données MySQL est également moins lourde sur un ordinateur et son démarrage est presque instantané. Il prend seulement la RAM nécessaire à son bon fonctionnement, contrairement à Oracle.

Les clusters
Oracle (RAC – Real Application Cluster) - Chaque machine possède une instance, qui accède à la même base de données

MySQL - Il existe une version de MySQL pour les clusters : MySQL Cluster

La gestion des usagers
Les droits d'un usager sur une base de données varient dépendamment de l'endroit (machine) où il se connecte.

Par exemple :
CREATE USER 'usr_facturation'@'localhost' IDENTIFIED BY 'xyz_pwd'; GRANT ALL ON

ma_bd_facturation.* TO 'usr_facturation'@'localhost' IDENTIFIED BY 'xyz_pwd';

Différences entre le serveur « community » et le « enterprise » Community (sous la licence GPL) - Ce que la compagnie dit sur l'utilisation du serveur MySQL Community : o Tu peux l'utiliser pour contribuer à son développement o Tu peux faire un projet dérivé et qui utilise MySQL, mais celui-ci doit être open source o Tu peux tester

- Ce que la licence GPL dit : o Si le projet est distribué avec MySQL, alors ce projet doit être open source sous la même licence. Il est également possible d'acheter une licence commerciale.

o Si le projet n'a pas à être distribué avec MySQL, alors il n'a pas besoin d'être open source.

Par exemple, un site Web qui se connecte à la base de données MySQL sur un serveur d'hébergement partagé. Celui-ci n'a pas à être open source. Enterprise - Pour les besoins commerciaux o Serveur plus stable o Meilleur support et mises à jour régulières

Les engines de tables de MySQL
innoDB

Engin par défaut de MySQL
Supporte les transactions (donc les clés étrangères également)
MyISAM

Jusqu'à tout récemment, c'était l'engin par défaut de MySQL
Ne supporte pas les transactions
Supporte les index sur plusieurs mots (FULL TEXT INDEXING)
MEMORY

Table dont le contenu est gardé en mémoire uniquement.
Comme toutes les données sont en mémoire (incluant les index), c'est extrêmement rapide. Il faut cependant faire attention. Si le serveur plante ou se ferme, les données seront perdues.
Quelques-unes des différences sur la syntaxe
LIMIT Afin de limiter les résultats d'une requête, la clause LIMIT est utilisée. Par exemple : <source lang='sql'>

SELECT
id, name FROM users ORDER BY name LIMIT 10, 20; </source> Ceci retourne les résultats à partir de la ligne #11 jusqu'à la ligne #35 Pour retourner les 10 premières lignes = LIMIT 0,10;

Pour faire identique avec Oracle, il faut plutôt faire : <source lang='sql'> SELECT * FROM (SELECT USERS_2.*, ROWNUM RNUM FROM (SELECT ID, NAME FROM USERS ORDER BY NAME) USERS_2 WHERE ROWNUM < 36) WHERE RNUM >= 10; AUTO_INCREMENT </source> Il n'est pas nécessaire de faire des séquences comme avec Oracle. AUTO_INCREMENT suffit ! <source lang='sql'> CREATE TABLE users (id INT NOT NULL AUTO_INCREMENT, … PRIMARY KEY pk_users (id)) ENGINE = innoDB; </source>

Syntaxe
La syntaxe des commandes suivantes est en simplifiée. En effet, certaines clauses peuvent être ajoutées (optionnel).

Création d’une base de données
CREATE {DATABASE | SCHEMA } xyz_db CHARACTER SET = utf8 (ou ascii, greek, …)

Database et schema sont synonymes pour MySQL.

Création d’un usager
CREATE USER 'foo'@'192.168.0.1' IDENTIFIED BY 'pwd';

foo : Le nom de l’usager 192.168.0.1 : Nom de l’hôte d’où il se connecte. Pour n’importe quel hôte, on utilise « % ». pwd : Mot de passe de l’usager

Assignation de droits d’un usager à une base de données GRANT v ON w.x TO 'y'@'z';

V: Privilège donné - Exemple : o ALL - Tous les privilèges o INSERT - Privilège d’insertion o UPDATE - Privilège de mise à jour o SELECT - Privilège de SELECT o EXECUTE - Permet d’exécuter des procédures stockées o ALTER - Permet de modifier la structure d’une table avec ALTER TABLE W: Nom de la base de données (ex : xyz_db) X: Nom de la table, ou * pour toutes les tables Y: nom de l’usager (ex: foo) Z: hôte (ex: %)

Exemple complet : GRANT SELECT, INSERT ON xyz_db.* TO 'foo'@'192.168.0.1';

MySQL supporte également les rôles, mais pas les profiles.

Suppression de droits d’un usager sur une base de données REVOKE INSERT ONT xyz_db.* TO 'foo'@'192.168.0.1';

Exemple de création d’une table simple La colonne “status” permet une énumération de valeurs permises. <source lang='sql'> CREATE TABLE users (

id INT NOT NULL AUTO_INCREMENT,
status ENUM("pending", "inactive", "active") DEFAULT "pending",
password VARCHAR(40) NOT NULL,
email VARCHAR(70) NOT NULL,
PRIMARY KEY pk_users(id),
INDEX idx_users_email (email)
) ENGINE = innoDB; </source> Exemple de création d’une table avec une clef étrangère <source lang='sql'> CREATE TABLE forgot_passwords (

id INT NOT NULL AUTO_INCREMENT,
id_user INT NOT NULL,
access_key VARCHAR(70),
PRIMARY KEY pk_temporary_passwords (id),
CONSTRAINT fk_temporary_passwords_id_user FOREIGN KEY (id_user) REFERENCES users (id)

) ENGINE = innoDB; </source>

Le dictionnaire de données
Depuis la version 5.0 de MySQL, les informations sur les bases de données (tables, colonnes, accès, etc) se trouve dans le schéma/database « INFORMATION_SCHEMA ». C'est la version MySQL, du dictionnaire de données Oracle.

Par exemple, pour avoir le nombre de connexions, on utiliserait la vue processlist du schéma INFORMATION_SCHEMA. Par exemple : SELECT * FROM INFORMATION_SCHEMA.PROCESSLIST;

Pour avoir la liste des colonnes des tables : SELECT * FROM INFORMATION_SCHEMA.COLUMNS;

MySQL Workbench SQL Development Est similaire à SQL Developer pour Oracle.

Pour afficher tous les schémas de MySQL (incluant le schéma du dictonnaire de données), il y a l'option Préférences SQL Editor Show Metadata schemata.

Data Modeling Permet de faire des diagrammes relationnels. Ces diagrammes peuvent par la suite être exportés en script SQL.

Il est également possible d'importer des schémas en script MySQL. MySQL Administration Permet de gérer la base de données MySQL.

Création de copies de sauvegarde
Les connexions actives à MySQL
Le CPU + RAM usage
- Etc.

Revision #1
Created 6 September 2022 14:04:15 by ggpilou2
Updated 6 September 2022 14:04:48 by ggpilou2