

# Introduction a *MySQL*


# C'est quoi MySql?

- ▶ Système de Gestion de Base de Données : Un Système de Gestion de Base de Données (« SGBD ») est un logiciel (ou un ensemble de logiciels) permettant de manipuler les données d'une base de données.
  - ▶ Manipuler, c'est-à-dire sélectionner et afficher des informations tirées de cette base, modifier des données, en ajouter ou en.
- ▶ MySQL est un système de gestion de bases de données.
- ▶ MySQL implémente la théorie relationnelle ; c'est donc un SGBDR.

# MySQL est un SGBR

Donc les données sont stockées sous forme de tables selon le modèle relationnel.

<i>Numéro</i>	<i>Prénom</i>	<i>Nom</i>	<i>Email</i>
1	Jean	Dupont	jdupont@email.com
2	Marie	Malherbe	mama@email.com
3	Nicolas	Jacques	jacques.nicolas@email.com
4	Hadrien	Piroux	happie@mail.com



<i>Numéro</i>	<i>Client</i>	<i>Produit</i>	<i>Quantité</i>
1	3	Tube de colle	3
2	2	Rame de papier A4	6
3	2	Ciseaux	2

# Le paradigme client - serveur et les SGBDs

- ▶ La plupart des SGBD sont basés sur un modèle Client - Serveur.
- ▶ la base de données se trouve sur un serveur.
- ▶ pour interagir avec cette base de données, il faut utiliser un logiciel « client » qui va interroger le serveur et transmettre la réponse que le serveur lui aura donnée
- ▶ Le serveur peut être installé sur **une machine différente du client** ; c'est souvent le cas. Ce n'est cependant pas obligatoire

- Dans ce cours, nous installerons les logiciels serveur et client sur un seul et même ordinateur.
- Par conséquent, lorsque vous installez un SGBD basé sur ce modèle (c'est le cas de MySQL), vous installez en réalité deux choses (au moins) : le serveur, et le client.
- Chaque requête (insertion/modification/lecture de données) est faite par l'intermédiaire du client. Jamais vous ne discuterez directement avec le serveur (d'ailleurs, il ne comprendrait rien à ce que vous diriez). Vous avez donc besoin d'un langage pour discuter avec le client. Dans le cas de MySQL, ce langage est le SQL.

# Le langage SQL

- ▶ Le SQL (Structured Query Language) est un langage informatique qui permet **d'interagir avec des bases de données relationnelles**. C'est le langage pour base de données le plus répandu, et c'est bien sûr celui utilisé par MySQL.
- ▶ C'est donc le langage que nous allons utiliser pour dire au client MySQL d'effectuer des opérations sur la base de données stockée sur le serveur MySQL. Il a été créé dans les années 1970 et c'est devenu standard en 1986 (pour la norme ANSI - 1987 en ce qui concerne la norme ISO). Il est encore régulièrement amélioré.

# MySQL

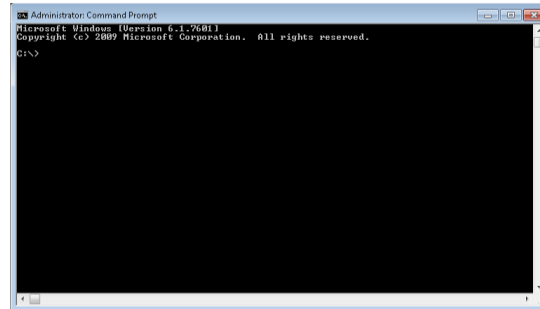
- ▶ MySQL est très utilisé, surtout par les débutants. Vous pourrez faire de nombreuses choses avec ce logiciel, et il convient tout à fait pour découvrir la gestion de bases de données.
- ▶ Sa popularité est due en grande partie au fait qu'il s'agit d'un logiciel Open Source
- ▶ Sachez cependant que MySQL est loin d'être parfait. En effet, il ne suit pas toujours la norme officielle. Certaines syntaxes peuvent donc être propres à MySQL et ne pas fonctionner sous d'autres SGBDR
- ▶ Enfin, il est très permissif, et acceptera donc des requêtes qui généreraient une erreur sous d'autres SGBDR.

# Interface de MySQL

Il existe plusieurs manières d'utiliser MySQL :

- Ligne de commande : il s'agit d'une fenêtre toute simple, dans laquelle toutes les instructions sont tapées à la main.

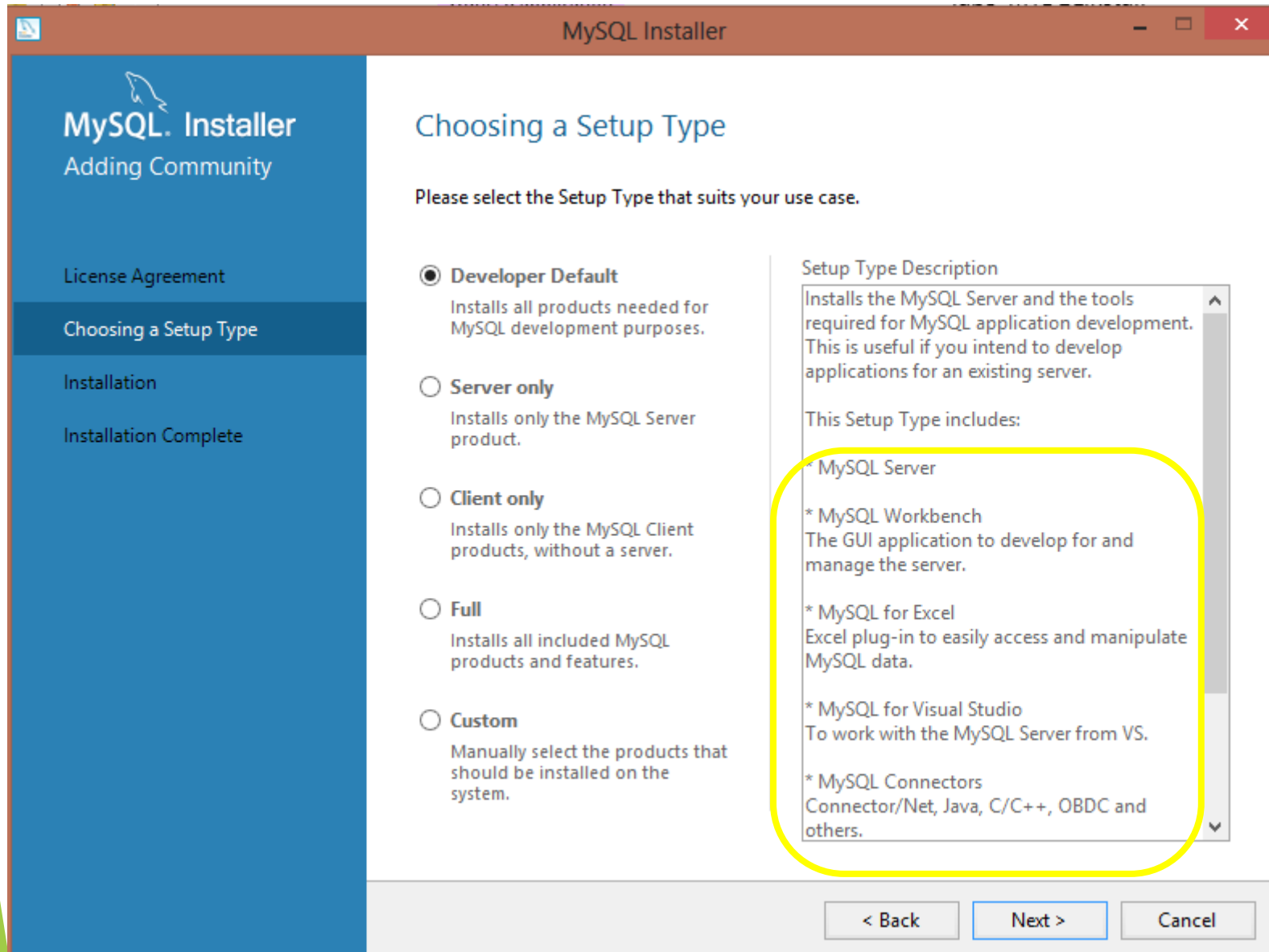
Les utilisateurs de Linux connaissent très certainement. Pour Mac, il faut utiliser l'application « Terminal » que vous trouverez dans Applications > Utilitaires. Quant aux utilisateurs de Windows, c'est le « Command Prompt » que vous devez utiliser ; il se trouve dans Démarrer > Tous les programmes > Accessoires



- Interface graphique : Comme interface graphique pour MySQL, on peut citer MySQL Workbench, Php-MyAdmin (souvent utilisé pour créer un site web en combinant MySQL et PHP) ou MySQL Front par exemple.



# Installation de MySQL et workbench



# Pourquoi utiliser la ligne de commande ?

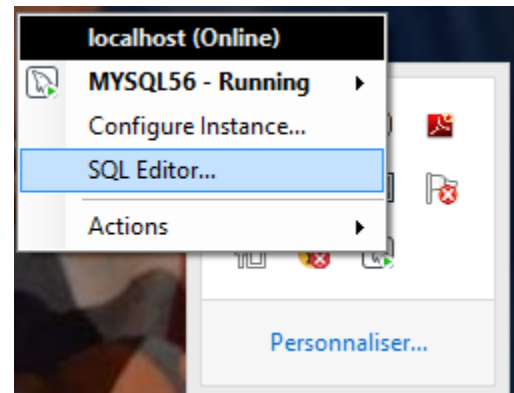
Deux raisons :

- d'abord, parce que je veux que vous maîtrisiez vraiment les commandes. En effet, les interfaces graphiques permettent de faire pas mal de choses, mais une fois que vous serez bien lancés, vous vous mettrez à faire des choses subtiles et compliquées, et il ne serait pas étonnant qu'il vous soit obligatoire d'écrire vous-mêmes vos requêtes;
- ensuite, parce qu'il est fort probable que vous désiriez utiliser MySQL en combinaison avec un autre langage de programmation (si ce n'est pas votre but immédiat, ça viendra probablement un jour). Or, dans du code PHP (ou Java, ou Python, ou. . .), on ne va pas écrire « Ouvre PhpMyAdmin et clique sur le bon bouton pour que je puisse insérer une donnée dans la base ». On va devoir écrire en dur les requêtes. Il faut donc que vous sachiez comment faire.

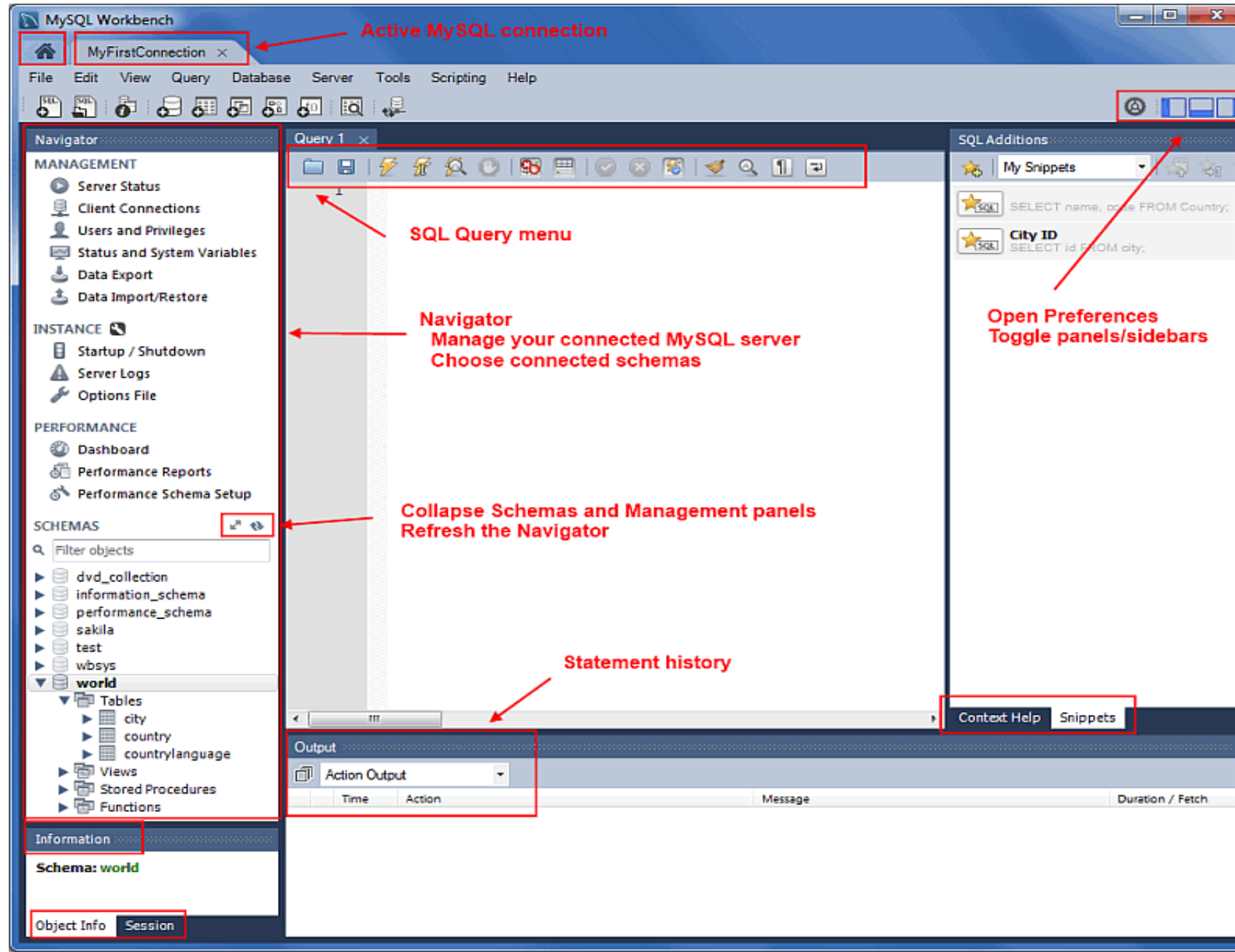
# Utilisation de MYSQL

- ▶ Après avoir installé : mysql+workbench dans le prochain lab
- ▶ Allez dans la barre de tâche, faites un clic droit sur le sigle de MySQL -le dauphin-  
Puis choisissez dans le menu contextuel **SQL Editor**.

L'éditeur SQL vous permet d'écrire du code SQL pour communiquer avec le serveur de MySQL



## L'éditeur SQL sous workbench



# Utilisation de *MySQL*

## Quelques commandes SQL usuelles

- Démarrer client *MySQL*, par exemple l'éditeur SQL sous workbench:

`use sakila;` --> utilise une BD dont le nom est sakila. Sakila est une base de données de test qui existe par défaut.

`show tables;` --> liste les tables dans la BD courante

`describe actor;` --> liste les colonnes de la table actor

`create database cours;` --> création de la BD cours

`use cours;` --> utilise la BD cours

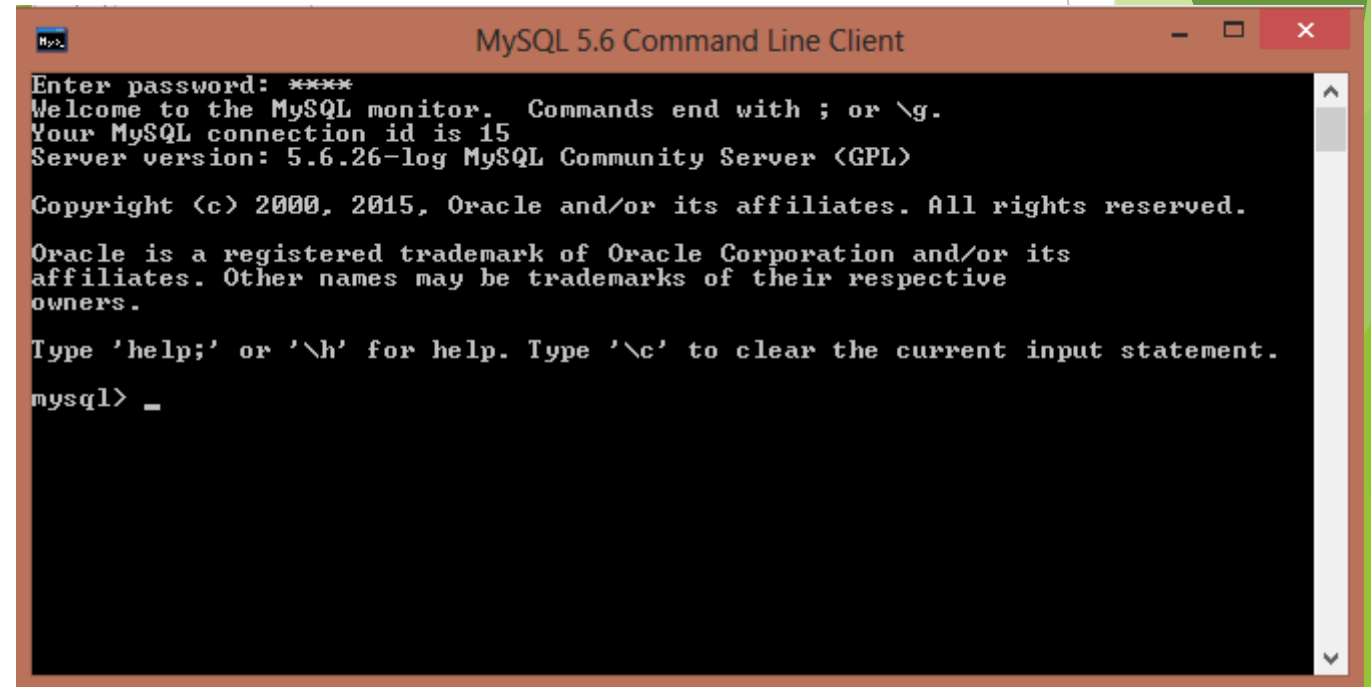
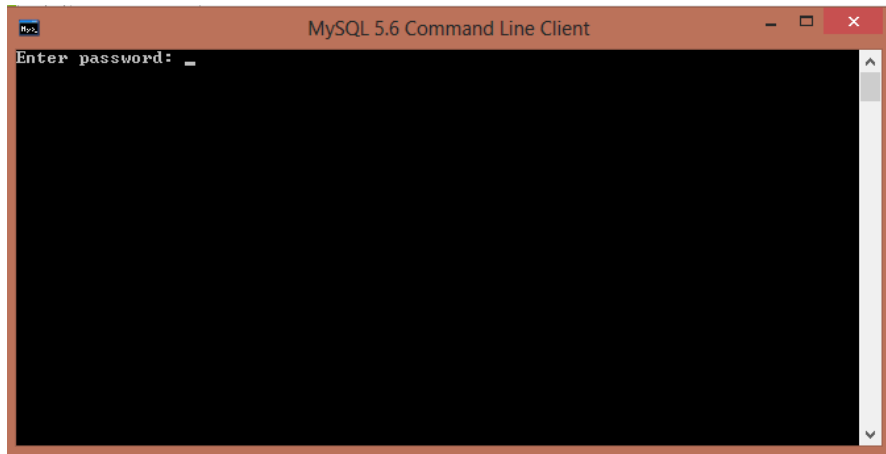
`create database tp2;` --> création de la BD tp2

`use tp2;` --> utilise la BD tp2

Note: les commandes dans *MySQL* se terminent toujours par ;

# Utilisation de MySQL via la console

- ▶ Cherchez MySQL command Line Editor dans les programmes installés.
- ▶ A son ouverture, il faut taper le mot de passe saisi pendant la configuration



# Utilisation de MySQL via la console

```
MySQL 5.6 Command Line Client

mysql> use sakila;
Database changed
mysql> show tables
-> ;
+-----+
| Tables_in_sakila |
+-----+
| actor             |
| actor_info        |
| address           |
| category          |
| city              |
| country           |
| customer          |
| customer_list     |
| film              |
| film_actor        |
| film_category     |
| film_list         |
| film_text         |
| inventory         |
| language          |
| nicer_but_slower_film_list |
| payment           |
| rental            |
| sales_by_film_category |
| sales_by_store    |
| staff             |
| staff_list        |
| store             |
+-----+
23 rows in set (0.00 sec)

mysql> _
```

```
mysql> describe actor
-> ;
+-----+-----+-----+-----+-----+-----+
| Field      | Type                | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| actor_id   | smallint(5) unsigned | NO   | PRI | NULL    | auto_increment |
| first_name | varchar(45)          | NO   |     | NULL    |                 |
| last_name  | varchar(45)          | NO   | MUL | NULL    |                 |
| last_update | timestamp            | NO   |     | CURRENT_TIMESTAMP | on update CURRENT_TIMESTAMP |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```