

Base de données

MySQL

Introduction

- Définition des Bases de données
- SGBD: Qu'est-ce c'est ?
- SGBD: Caractéristiques
- SGBD: Objectifs
- Historique et types de BD et de SGBD

Définition des Bases de données

- **Bases de données:**
- Collection de fichiers reliés par des pointeurs multiples, bien organisés, répondre efficacement à des demandes variées
- Collection d'informations via la modélisation du monde réel

SGBD : Qu'est-ce c'est ?

- SGBD (DBMS): **Databases Management System**
Système de Gestion de Bases de Données
 - C'est un logiciel qui permet d'interagir avec une base de données
 - Ensemble de logiciels systèmes permettant de stocker et d'interroger un ensemble de fichiers indépendants. Il est aussi un outil permettant de modéliser et de gérer des données

SGBD : Caractéristiques

- Management des données persistantes
- Accéder aux larges données avec efficacité
- Supporter les modèles de données
- Transaction
- Permettre de définir des données(structure, accès, manipuler)
- Recouverte des données
- Contrôle d'accès

SGBD : Objectifs

- Garantir l'intégrité des données
- Indépendance:
 - Physique des programmes aux données
 - Logique des programmes aux données
 - Administration facile
 - Permettre de manipuler par un langage de requête structurée (SQL)
- Offre de bonnes performances des données alors que plusieurs peuvent accéder simultanément à la BD

Historique et types de BD et de SGBD

- Il existe différents types de BDs en fonction du modèle d'organisation des données qu'elles utilisent.

A ce modèle est associé un langage de manipulation des données.

- Années 60 : les anciens modèles. Modèle hiérarchique, modèle réseau, langages d'accès navigationnels permettant de circuler dans des structures de types graphes.

Historique et types de BD et de SGBD

- 1970 : Création du modèle relationnel. Article de CODD.
- Années 80 : Premiers SGBD-Relationnels sur le marché
(ORACLE, Ingres, Informix, sybase, Dbase IV, etc.)
- Années 90 : Premiers SGBD-Objets.
- Aujourd'hui : L'Objet n'a pas remplacé le Relationnel. Le marché est dominé par les SGBD-R (ORACLE, SQL-Server (microsoft), DB2 (IBM), MySQL, PostgreSQL, ACCESS...). Mais les technologies sont souvent mixtes (Relationnel et objet).

MySQL

- Définition
- Avantages
- Architecture
- Objets

Définition

MySQL est un système de gestion de bases de données relationnelles. Le SQL dans “MySQL” signifie “Structured Query Language” : le langage standard pour les traitements de bases de données.

Avantages

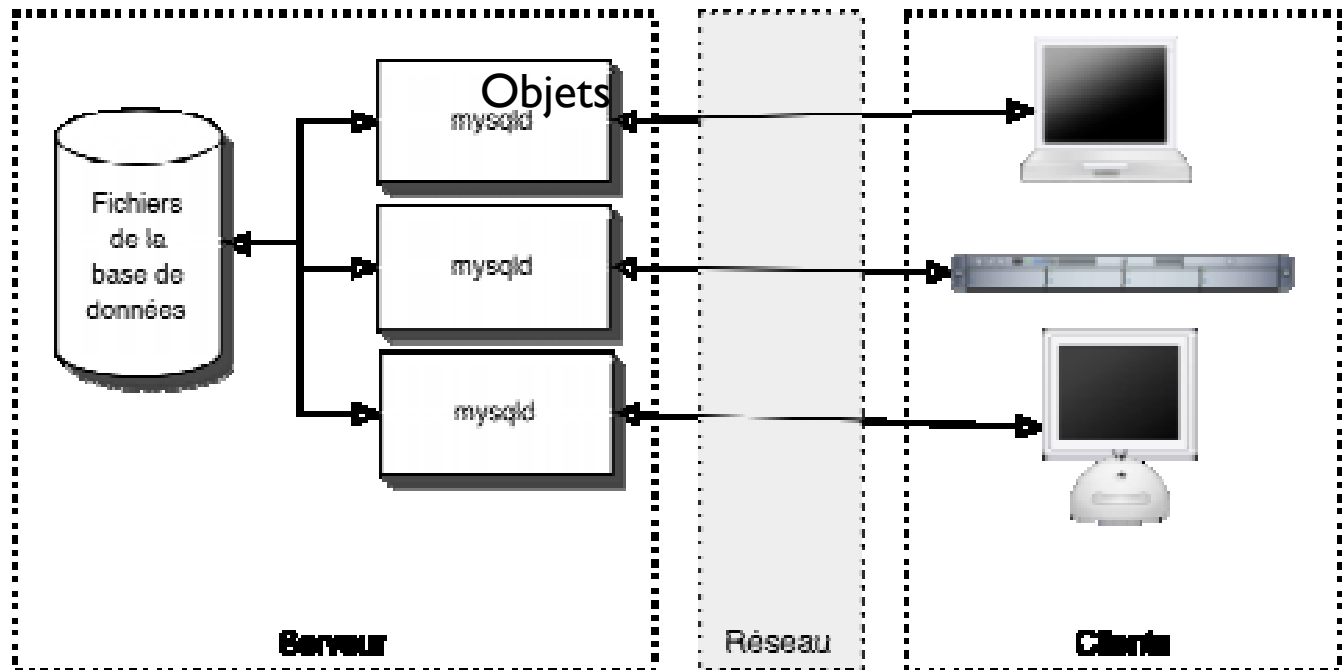
- très rapide ;
- multi-utilisateurs ;
- disponible sur Internet
(<http://www.mysql.com>) ;
- gratuit (sauf si vous commercialisez un service ou un logiciel qui utilise MySQL) ;
- facile à utiliser, un manuel de référence est fourni ;

Avantages

- interfaces de programmation (API) : C, Perl, PHP, Python et Java ;
- langage de requête : SQL (langage de requête le plus répandu) ;
- portabilité, MySQL fonctionne sous :
 - unix et linux ;
 - solaris (OS/2) ;
 - Mac Os X server ;
 - windows.
- open software, i.e. le code source est disponible (les bugs sont corrigés par la communauté et des nouvelles versions sont disponibles sur Internet dès que les corrections sont effectuées).

Architecture

- MySQL est basé sur une architecture client/serveur.



Objets

Un serveur MySQL gère une ou plusieurs base de données.

Chaque base de données contient différents types d'objets (tables, index, fonctions).

L'objet le plus représenté d'une base de données est la table.

Chaque table (appelées encore « relation ») est caractérisée par une ou plusieurs colonnes (ou « attributs »).

Objets

Le langage qui permet de gérer ces objets est appelé « Langage de Description des Données » (LDD).

Les données sont stockées dans les tables sous forme de lignes (ou « tuples »).

Le langage qui permet de manipuler les données est appelé « Langage de Manipulation des Données » (LMD)

Clés primaires et étrangères

Une table contient une clé primaire.

Une clé primaire est constituée d'une ou plusieurs colonnes.

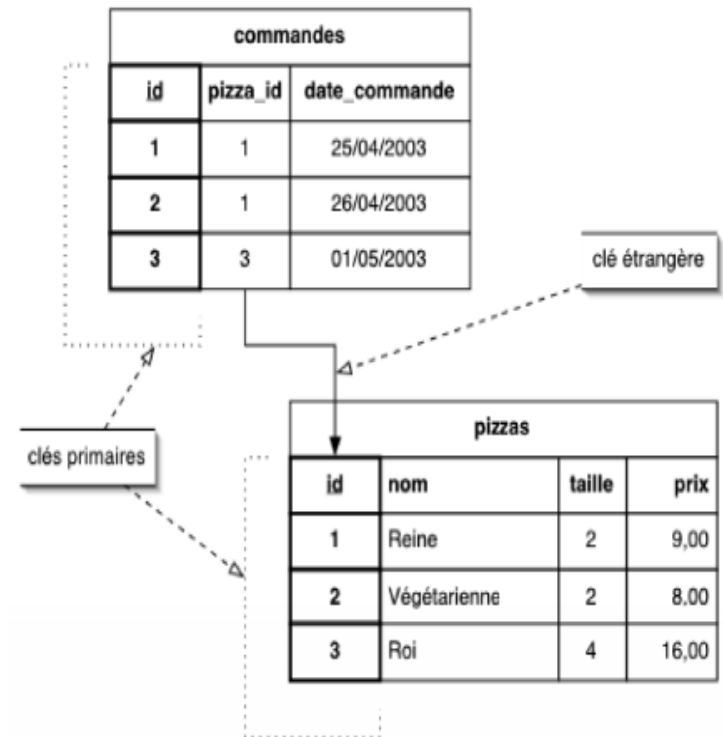
Les valeurs des colonnes qui constituent la clé primaire d'une table sont **uniques** pour toutes les lignes de la table. La clé primaire d'une table permet donc de faire **référence** de manière univoque à chaque ligne de la table.

Clés primaires et étrangères

Exemple : Les tables pizzas et commandes possèdent toutes les deux une **clé primaire** qui est constituée de leur colonne id respective.

La colonne pizza_id de la table commandes fait référence à la colonne id (donc à la clé primaire) de la table pizzas.
Ce type de référence est appelée « **clé étrangère** ».

MySQL n'implémente pas les clés étrangères sous forme de contrainte.



Clés uniques

En complément à la clé primaire, une ou plusieurs clés uniques (appelées également « clés secondaires » ou « clés alternatives ») peuvent être associées à une table.

Les clés uniques sont identiques aux clés primaires à la différence près que les colonnes qui les constituent peuvent contenir une valeur NULL.

La plupart du temps, toute table devrait posséder au moins une clé primaire.

Clés uniques

Exemple: Si la clé primaire de la table pizzas est constituée de la colonne id et s'il existe une clé unique constituée de la seule colonne nom, le schéma ci-contre propose trois nouvelles lignes à l'insertion dont une seule pourra être ajoutée à la table.

pizzas			
<u>id</u>	<i>nom</i>	taille	prix
1	Reine	2	9,00
2	Végétarienne	2	8,00
3	Roi	4	16,00
3	Calzone	4	13,00
4	Roi	2	11,00
4	Calzone	4	14,00

insertion interdite

insertion interdite

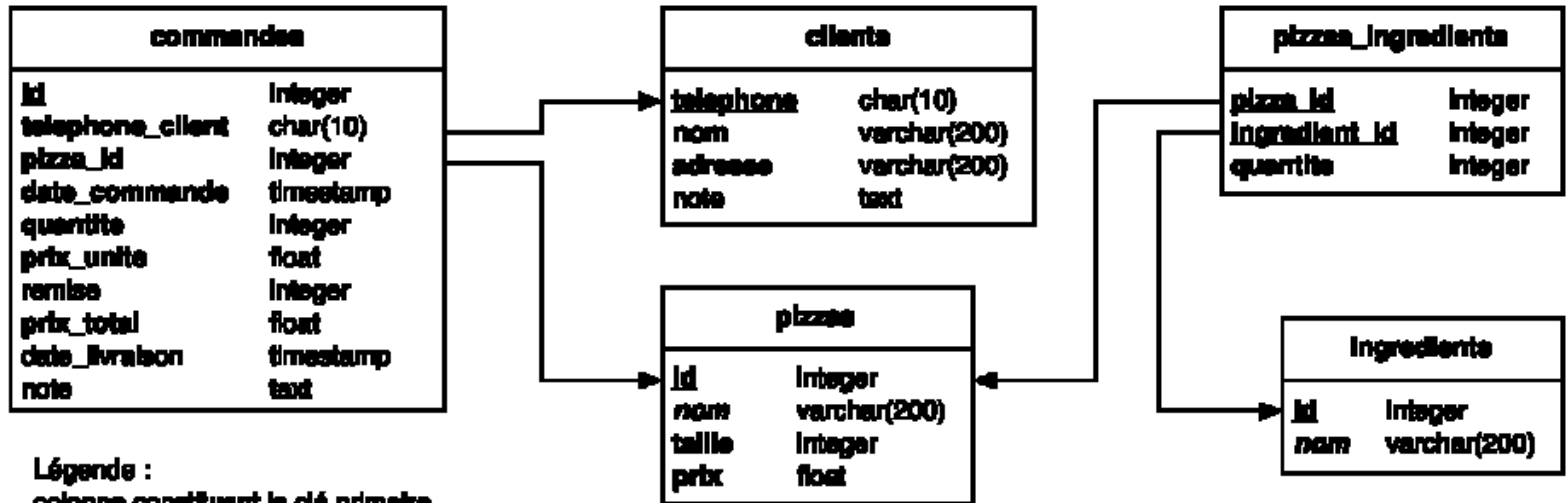
insertion ok

Base de données

Tout au long de ce support de cours les exemples se référeront à une base de données fictive. Cette base modélise (très simplement) une activité de fabrication et de livraison de pizzas.

Le Modèle Physique de Données (MPD) est un diagramme qui permet de présenter les tables avec leurs colonnes ainsi que les références entre elles.

Base de données



Légende :
colonne constituant la clé primaire
colonne constituant une clé unique
colonne not null

LMD, verbes

Le langage de manipulation des données (LMD) est constitué de quatre verbes :

SELECT INSERT UPDATE DELETE

Chacun de ces verbes possède une ou plusieurs clauses.

Lorsqu'elles sont entrées dans l'utilitaire mysql, les commandes SQL doivent se terminer par un point-virgule.

Dans d'autres interfaces le point-virgule final est généralement accepté mais pas souvent nécessaire.

Par exemple :

```
select * from pizzas where nom='Roi';
```

LMD, verbes

La clause **from** permet de spécifier le nom de la table qui est interrogée, la clause **where** (condition) permet de préciser un critère de sélection.

« * » est utilisé pour sélectionner toutes les colonnes d'une table.

LMD,select

Le verbe **SELECT** permet de faire des requêtes sur une ou plusieurs tables.

Il ne modifie jamais les données.

Une forme simple de select est :

select colonnes **from** tables **where**
condition **order by** colonnes [**asc/desc**]

LMD,select

Les opérateurs de comparaison sont les suivants :

- arithmétiques : = < > <= >= !=
- plage de valeur : between v1 and v2
- appartenance : in (v1, v2, etc.)
- nullité : is null, is not null
- comparaison de chaîne : like 'modèle' (caractères joker : « % » et « _ »)

Exemples :

select * from commandes where quantite >= 2;

**select nom, prix from pizzas where prix
between 5 and 10;**

LMD,select

select * from ingredients where id in (1, 2, 4);

select nom, note from clients where note is not null;

select telephone, nom from clients where telephone like '01%' order by nom;

Opérateur booléens : and et or.

Exemple :

select * from pizzas where taille = 4 and prix < 10;

LMD, insert

Le verbe **insert** permet d'ajouter des lignes à une table.
Il s'utilise de deux manières :

insert into table (colonnes) values (valeurs)

insert into table (colonnes) select colonnes from tables

La liste des colonnes entre parenthèses est optionnelle. Si elle est omise la liste des valeurs doit fournir une valeur pour toute les colonnes de la table dans l'ordre dans lequel elles ont été spécifiées lors de la création de la table.

La seconde forme permet d'insérer dans la table de destination le résultat d'une requête.

LMD,update

Le verbe **update** permet de modifier des lignes déjà présentes dans une table.

Sa syntaxe est la suivante :

update table set colonne=valeur, colonne=valeur where condition

La clause **where** est optionnelle. Si elle est omise, les modifications sont appliquées à la totalité des lignes de la table.

Par exemple, nous livrons la commande de d'Alex Feinberg à 20h28 :

update commandes set date_livraison = '2003-05-14 20:28' where id = 7;

Il est possible d'introduire une formule dans la requête de mis-à-jour.

LMD,delete

Le verbe **delete** est utilisé pour supprimer des lignes d'une table.

Sa syntaxe est la suivante :

delete from table where condition

La clause where est optionnelle. Toutes les lignes sont supprimées si elle est omise.

LDD, create table

Le verbe **create table** est utilisé pour créer une table avec l'ensemble de ses colonnes.

Sa syntaxe est :

create table table

(colonne1 type [not null]

, colonne2 type [not null]

, ...

, primary key (colonnes)

[, unique (colonnes)]

, ...)

Les colonnes qui constituent la clé primaire doivent toutes être not null.

LDD, alter table

La commande **alter table** permet de modifier certaines caractéristiques d'une table ou de l'une de ses colonnes.

Elle est fréquemment utilisée pour ajouter une colonne à un table.

Exemple : Il nous est maintenant nécessaire d'ajouter la colonne `livreur_id` à la table `commandes`. Cette colonne contiendra l'identifiant du livreur ou la valeur `NULL` si le client est venu chercher la pizza sur place.

```
alter table commandes add column  
livreur_id integer;
```

LDD, alter table

La commande **alter table** permet également :

- d'ajouter une clé unique

alter table table add unique (colonnes)

- supprimer une colonne

alter table table drop column colonne

- changer le nom et le type d'une colonne

**alter table table change column colonne
nouveau_nom nouveau_type**

- renommer une table

alter table table rename as nouveau_nom

La commande **drop table** supprime une table :

drop table table

Installation et configuration

<http://www.youtube.com/watch?v=iP1wOSsKjW8&list=PLSIQu1Wo1RlahlYDqHWZb8IqsKgEvPiHn&index=1>