

# Base de Données Avancées

COURS 1

Introduction à SQL et MySQL

# Plan de la présentation

- ▶ SQL langage ou norme
  - Aspects du langage SQL
  - Avantages du langage SQL
  - Modèle des données utilisé
- ▶ Ce quoi MySQL ?
  - Avantage de MySQL
  - Console de MySQL
- ▶ SQL de Base

# SQL Langage ou norme

- ▶ IBM a implanté le modèle relationnel au travers du langage **SEQUEL** (*Structured English as QUERy Language*), renommé par la suite **SQL** (*Structured Query Language ou langage des requêtes structurées*) au début des années 80.
- ▶ **SQL1** première norme datant de 1987, compromis entre plusieurs constructeurs, mais fortement influencé par le dialecte d'IBM.
- ▶ **SQL2** normalisé en 1992
- ▶ **SQL3** normalisé en 1999....
- ▶ **SQL2011**

Chaque norme a ses propres apports au niveau des caractéristiques et fonctionnalités nouvelles.

# Aspects du langage SQL

- ▶ Définition des données
- ▶ Manipulation des données
- ▶ Interrogation des données
- ▶ Contrôle des données

CREATE - ALTER - DROP - RENAME - TRUNCATE	Définition des données (LDD)
INSERT - UPDATE - DELETE - LOCK TABLE	Manipulation des données (LMD)
SELECT	Interrogation des données (LID)
GRANT - REVOKE - COMMIT - ROLLBACK - SAVEPOINT - SET TRANSACTION	Contrôle des données (LCD)

# Avantages du langage SQL

- ▶ Standard industriel depuis 1987, s'enrichit au fil du temps.
- ▶ Pris en charge par les principaux langages de programmation comme C ou Cobol, mais aussi les langages Orientés Objet comme C++, Java et C#.
- ▶ Possibilité de modifier la structure des données sans pour autant engendrer une refonte importante du programme (Indépendance entre programme et donnée)
- ▶ Adapté aux grandes applications informatiques de gestion
- ▶ Les systèmes basés sur ce langage sont fiables et performants.

# Modèle des données: la table

- ▶ SQL utilise le modèle de données relationnelles (MDR).
- ▶ La table relationnelle (*relational table*) est la structure de données de base qui contient des enregistrements appelés aussi « lignes » (*rows*).
- ▶ Une table est composée de colonnes (*columns*) qui décrivent les enregistrements.
- ▶ Ce modèle est essentiellement basé sur les valeurs. Les associations entre tables sont toujours binaires et assurées par les clés étrangères.
- ▶ Exemple : [Compagnie](#)

comp	nrue	rue	ville	nomComp
AF	10	Gambetta	Paris	Air France
SING	7	Camparols	Singapour	Singapore AL

Pilote

brevet	nom	nbHVol	compa
PL-1	Louise Ente	450	AF
PL-2	Jules Ente	900	AF
PL-3	Paul Soutou	1000	SING

# Modele des données : les clés d'une table

- ▶ La **clé primaire** (*primary key*) d'une table est l'ensemble minimal de colonnes qui permet d'identifier de manière unique chaque enregistrement.
- ▶ Une clé est dite « **candidate** » (*candidate key*) si elle peut se substituer à la clé primaire à tout instant. Une table peut contenir plusieurs clés candidates ou aucune.
- ▶ Une **clé étrangère** (*foreign key*) référence dans la majorité des cas une clé primaire d'une autre table (sinon une clé candidate sur laquelle un index unique aura été défini). Elle est composée d'une ou plusieurs colonnes. Une table peut contenir plusieurs clés étrangères ou aucune.

Les clés primaires et étrangères sont définies dans les tables en SQL à l'aide de **contraintes**.

# Ce quoi MySQL ?

- ▶ Un SGBDR, système de gestion de base de données relationnelle, permettant de gérer un ensemble des données structurées reliées entre-elles.
- ▶ Un système écrit en C et C++, *open source*, fonctionnant sur les systèmes d'exploitation le plus couramment utilisé(Linux, Microsoft, MacOs...etc.).
- ▶ Un serveur développé par une société suédoise dénommé MySQL AB (Uppsala, Suède).
- ▶ Racheté en 2008 par **Sun Microsystems** (USA), elle-même racheté en 2010 par **Oracle Corporation** (USA).

# Avantages de MySQL :

- ▶ Système *open source*
- ▶ Fonctionnalités de plus en plus riche version après version.
- ▶ Système performant
- ▶ Système ouvert à tous les principaux langages de programmation
- ▶ Système fonctionnant sur le SE le plus couramment utilisé
- ▶ Sa facilité d'utilisation pour des applications Web de taille moyenne.

# MySQL: Notion de *database*

- ▶ ***Database*** (*base de donnée*) un regroupement logique d'objets (*tables, index, vues, déclencheurs, procédures cataloguées, etc.*) pouvant être stockés à différents endroits de l'espace disque.
- ▶ Pour MySQL, un objet appartient à son schéma (*database*), il n'y a pas de notion d'appartenance d'un objet (table, index, etc.) à un utilisateur.
- ▶ Pour Oracle ou d'autres SGBD, chaque objet appartient à son *user*.
- ▶ En MySQL, dès que le serveur est installé dans votre système, il y'a l'existence de trois base de données (*mysql, test et information\_schema*).  
Ces bases sont dites **base système**, le serveur ne peut fonctionner sans elles.

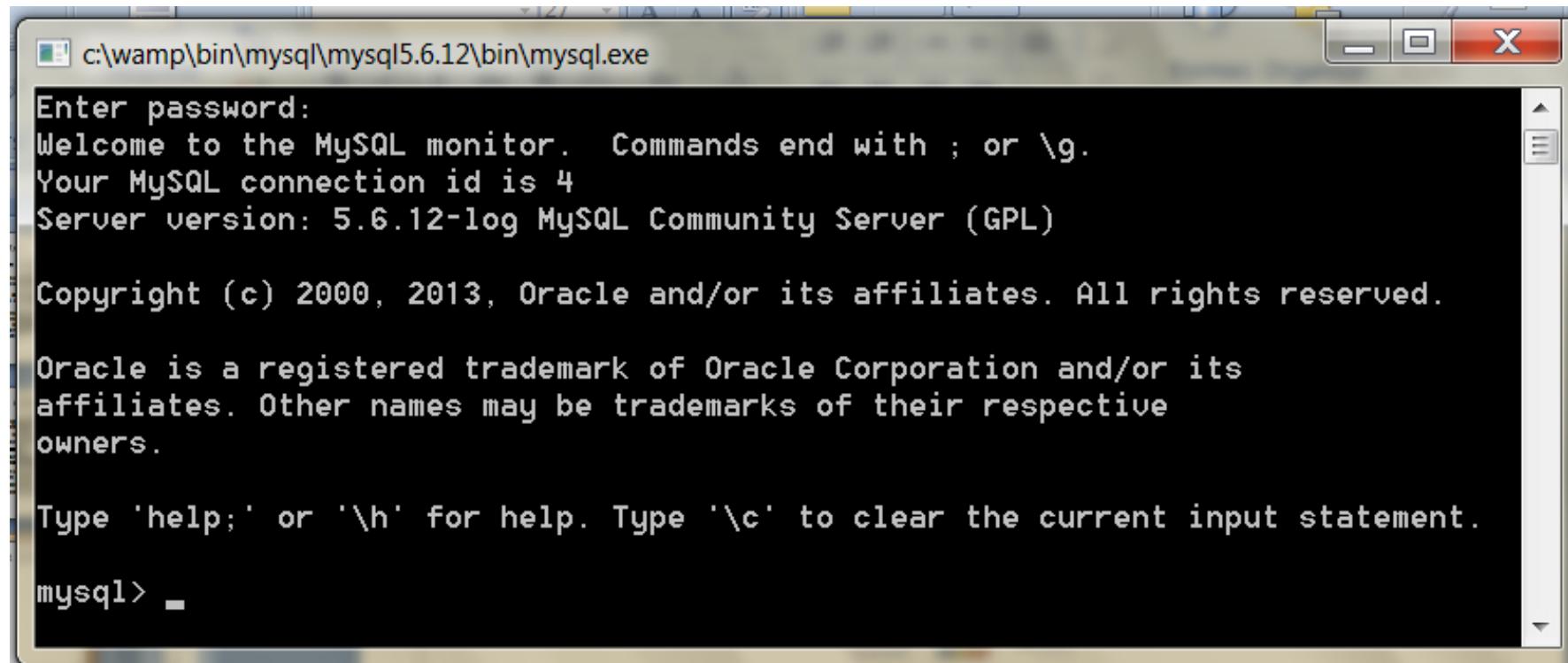
# MySQL: Notion d'hôte

- ▶ Avec MySQL, *host* c'est le nom de la machine hébergeant le SGBD.
- ▶ Il est possible de distinguer des accès d'un même utilisateur se connectant depuis plusieurs machines différentes.
- ▶ La notion d'identité est basée sur le couple **nom d'utilisateur MySQL (*user*) côté serveur, machine cliente**.
- ▶ Exemple: si l'user *Said* se connecte à partir de la machine *clt1* vers le serveur une fois, et une autre fois, il se connecte à partir de la machine *clt5*, il faudra prévoir deux identités différentes au niveau du serveur (*said@clt1* et *said@clt5*).

# MySQL: Interface de commande

- ▶ L'interface de commande en ligne (*Console MySQL*) ressemble à une fenêtre DOS.
- ▶ Elle permet de **dialoguer** de la plus simple façon avec la **base de données**. L'utilisation peut être souvent *interactive* ou « *batch* ».
- ▶ En mode *interactive* , le résultat des extractions sont représentés sous forme **tabulaire**.
- ▶ Une fois la connexion établie vers le serveur, des **instructions** (ordres SQL) sont envoyées à la base qui retourne des **Résultats** affichés dans la même fenêtre de commande.

# Une fenêtre Console de MySQL



The screenshot shows a Windows command-line window titled "c:\wamp\bin\mysql\mysql5.6.12\bin\mysql.exe". The window displays the MySQL monitor welcome message, connection information, server version, copyright notice, trademark information, help instructions, and a prompt.

```
c:\wamp\bin\mysql\mysql5.6.12\bin\mysql.exe

Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.6.12-log MySQL Community Server (GPL)

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> _
```

# Options de base de la Console MySQL

- ▶ Les principales options au lancement de la Console MySQL sont résumées dans le tableau suivant :

Option	Commentaire
<code>--help</code> ou <code>-?</code>	Affiche les options disponibles, l'état des variables d'environnement et rend la main.
<code>--batch</code> ou <code>-B</code>	Toute commande SQL peut être lancée dans la fenêtre de commande système sans pour autant voir l'invite ; les résultats (colonnes) sont séparés par des tabulations.
<code>--database=nomBD</code> ou <code>-D nomBD</code>	Sélection de la base de données à utiliser après la connexion.
<code>--host=nomServeur</code> ou <code>-h nomServeur</code>	Désignation du serveur.
<code>--html</code> ou <code>-H</code>	Formatte le résultat des extractions en HTML.
<code>--one-database</code> ou <code>-O</code>	Restreint les instructions à la base de données spécifiée initialement.
<code>-p</code>	Demande le mot de passe sans l'employer en tant que paramètre.
<code>--password=motdePasse</code>	Transmission du mot de passe de l'utilisateur à connecter. Évitez cette option et préférez la précédente...
<code>--prompt=parametre</code>	Personnalise l'invite de commande (par défaut <code>mysql&gt;</code> ).
<code>--silent</code> ou <code>-s</code>	Configure le mode silence pour réduire les messages de MySQL.

# Options de base de la Console MySQL

--skip-column-names ou -N	N'écrit aucun en-tête de colonne pour les résultats d'extraction.
--table ou -t	Format le résultat des extractions en tables à en-tête de colonne (par défaut dans le mode interactif).
--tee=cheminNomFichier	Copie la trace de toute la session dans le fichier que vous indiquez.
--user=utilisateur ou -u utilisateur	Désigne l'utilisateur devant se connecter.
--verbose ou -v	Mode verbeux pour avoir davantage de messages du serveur.
--version ou -V	Affiche la version du serveur et rend la main.
--vertical ou -E	Affiche les résultats des extractions verticalement (non plus en lignes horizontales).
--xml ou -X	Format le résultat des extractions en XML. Les noms de balises générées sont <resultset> pour la table résultat, <row> pour chaque ligne et <field> pour les colonnes.

Il est possible de combiner ces options en les séparant simplement par un espace (exemple :

```
mysql --tee=D:\\dev\\sortiemysql.txt --database=bdtest)
```

# Mode Batch de la Console MySQL

- ▶ Pour lancer plusieurs commandes regroupées dans un fichier à extension « `.sql` », il faut préciser le chemin du fichier (`.sql`) et celui qui contiendra les éventuels résultats., c'est le mode batch de la console.
- ▶ Exemple :

```
mysql --user=root --password=tor dbtest  
<D:\\testdv\\Testbatch.sql >D:\\testdv\\sortie.txt
```

- ▶ l'instruction suivante exécute dans la base `dbtest`, sous l'autorité de l'utilisateur `root`, les commandes contenues dans le fichier `Testbatch.sql` situé dans le répertoire `D:\\dev`. Le résultat sera consigné dans le fichier `sortie.txt` du même répertoire.

# Commandes de base de la Console

- Après connexion au serveur, il est possible d'utiliser des commandes ou faire des copier-coller d'un éditeur de texte dans la Console MySQL.
- Le tableau suivant résume les principales instructions pour manipuler la console d'entrée de l'interface.
- Exemple : prompt (\u0@\h) [\d]>

Commande	Commentaire
?	Affichage des commandes disponibles.
delimiter chaîne	Modifie le délimiteur (par défaut « ; »).
use nomBase	Rend une base de données courante.
prompt chaîne	Modifie l'invite de commande avec les paramètres vus précédemment.
quit ou exit	Quitte l'interface.
source cheminNomFichier.sql	Charge et exécute dans le buffer le contenu du <i>cheminNomFichier.sql</i> (ex : source D:\\\\dev\\\\Testbatch.sql exécutera le script Testbatch.sql situé dans D:\\dev).
tee nomFichiersortie	Création nomFichiersortie dans le répertoire C:\\Program Files\\MySQL\\MySQL Server n.n\\bin qui contiendra la trace de la session.

# Autres caractéristiques de MySQL

- ▶ **Délimiteurs** : la directive `delimiter` permet de choisir le symbole qui terminera chaque instruction , le symbole par défaut de MySQL c'est « ; ». exemple:

```
delimiter #
```

```
CREATE TABLE Test (t CHAR(8)) #
```

- ▶ **Sensibilité à la casse** : MySQL est sensible par défaut à la casse dans la plupart des distributions Unix, il ne l'est pas pour Windows.
- ▶ La variable `lower_case_table_names` permet de forcer la sensibilité à la casse pour les noms des tables et des bases de données.

# Les commentaires et conventions

- ▶ **Conventions de MySQL :**
  - Tous les mots-clés de SQL sont notés en majuscules.
  - Les noms de tables sont notés en Minuscules (excepté la première lettre, mais stockés dans le système en minuscules).
  - Les noms de colonnes et de contraintes en minuscules.
- ▶ **Un commentaire** se déclare : sur une même ligne précédée de deux tirets « -- », en fin de ligne à l'aide du dièse « # », au sein d'une ligne ou sur plusieurs lignes entre « /\* » et « \*/ »).

Exemple:

```
CREATE TABLE
    -- nom de la table
    Test( #début de la description
        COLONNE DECIMAL(38,8)
    ) -- fin, ne pas oublier le point-virgule.
;
CREATE TABLE Test (
    /* une plus grande description
    des colonnes */
    COLONNE /* type : */ DECIMAL(38,8));
```

# Quelques restrictions de MySQL

- ▶ Nom des colonnes **unique** pour une même table;
- ▶ Les colonnes SET sont évaluées par des chaînes de caractères séparés par des « , » ('Airbus, Boeing'). Ainsi aucune valeur d'un SET ne doit contenir le symbole « , ».
- ▶ Les noms des objets (base, tables, colonnes, contraintes, vues, etc.) ne doivent pas emprunter des mots-clés de MySQL : TABLE, SELECT, INSERT, IF...etc.

# SQL de Base de MySQL

# SQL de base: CREATE TABLE

- ▶ La syntaxe simplifié est la suivante:

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] [nomBase.]nomTable
( colonne1 type1
[NOT NULL | NULL] [DEFAULT valeur1] [COMMENT 'chaine1']
[, colonne2 type2
[NOT NULL | NULL] [DEFAULT valeur2] [COMMENT 'chaine2' ] ]
[CONSTRAINT nomContrainte1 typeContrainte1] ...)
[ENGINE= InnoDB | MyISAM | ...];
```

# Exemple avec CREATE TABLE

```
CREATE TABLE bdbeile.Compagnie  
  (comp CHAR(4),  
   nrule INTEGER(3),  
   rue CHAR(20),  
   ville CHAR(15) DEFAULT 'Paris'  
   COMMENT 'Par defaut : Paris',  
   nomComp CHAR(15) NOT NULL);
```

# Les contraintes dans les tables

- ▶ But: programmer des règles de gestion au niveau des colonnes des tables. Elles peuvent alléger un développement côté client.
- ▶ Elles se déclarent de deux manières:
  - En même temps que la colonne: contraintes « en ligne » et/ou **mono-colonne**,
  - Après déclaration d'une colonne, celles-ci ne sont pas limitées à une seule colonne et personnalisables par un nom;
- ▶ La contrainte NOT NULL est déclarée « en ligne » (recommandé)

# Les contraintes les plus utilisées

## **CONSTRAINT *nomContrainte***

UNIQUE (*colonne1 [,colonne2]...*)

PRIMARY KEY (*colonne1 [,colonne2]...*)

FOREIGN KEY (*colonne1 [,colonne2]...*)

REFERENCES *nomTablePere* [*(colonne1 [,colonne2]...)*]

[ON DELETE {RESTRICT|CASCADE|SET NULL|NO ACTION}]

[ON UPDATE {RESTRICT|CASCADE|SET NULL|NO ACTION}]

CHECK (*condition*)

**(exemple:** CHECK (*note BETWEEN 0 AND 20*) , CHECK  
(*grade='Copilote'* OR *grade='Commandant'*)).

# Les contraintes : exemple

```
CREATE TABLE Pilote
(brevet CHAR(6), nom CHAR(15) NOT NULL,
nbHVol DECIMAL(7,2), compa CHAR(4),
CONSTRAINT pk_Pilote PRIMARY KEY(brevet),
CONSTRAINT ck_nbHVol
CHECK(nbHVol BETWEEN 0 AND 20000),
CONSTRAINT un_nom UNIQUE (nom),
CONSTRAINT fk_Pil_compa_Comp
FOREIGN KEY (compa)
REFERENCES Compagnie(comp));
```

# Les types des colonnes:caractères

Pour décrire les colonnes d'une table, MySQL fournit les types prédéfinis suivants (*built-in datatypes*) : parmi les types caractères disponibles, il y'en a:

- ▶ CHAR, chaîne de caractères à taille fixe (max:255 cars);
- ▶ VARCHAR, chaîne de caractères à taille variable (max:65535 cars);
- ▶ BINARY et VARBINARY sont similaires à CHAR et VARCHAR, mais contiennent des chaînes d'octets , pas de jeu de caractères en particulier.
- ▶ Les quatre types permettant aussi de stocker du texte sont TINYTEXT, TEXT, MEDIUMTEXT, et LONGTEXT. Ils sont associés à un jeu de caractères, pas de DEFAULT, ni de suppression d'espaces de fin.

# Les types de colonnes: les nombres

Valeurs exactes, positifs ou négatifs

- ▶ INTEGER (entier sur 4 octets. signé ou non)
- ▶ SMALLINT (entier sur 2 octets. signé ou non)
- ▶ BOOL ou BOOLEAN (valeur booléenne 0 ou 1)

Valeurs à virgule fixe ou flottante

- ▶ FLOAT, nombre à virgule flottante (sur 4 à 8 octets  
Avec 7decimales max)
- ▶ DOUBLE, nombre à virgule flottante (sur 8 octets  
Avec 15decimales max)
- ▶ DECIMAL , décimal à virgule fixe (max 30 chiffres  
après virgule)

Les entiers restreints (TINYINT, MEDIUMINT et BIGINT).

# Les types de colonnes : Dates/heures

- ▶ DATE (sur 3 octets, du 01/01/1000 au 31/12/9999, format ‘YYYY-MM-DD’);
- ▶ DATETIME (date et heures, sur 8 octets. ‘YYYY-MM-DD HH:MM:SS’)
- ▶ YEAR (année, sur 2 (70 à 69 [1970–2069] ou 4 position (1901–2155), 1 octet, format ‘YYYY’)
- ▶ TIME (heure, sur 3 octets, de -838h59m59s à 838h59m59s, format ‘HHH:MM:SS’)
- ▶ TIMESTAMP (instants du 01/01/1970 0h0m0s à l'an 2037, mis à jour à chaque modification de la table)

Les fonctions NOW() et SYSDATE() retournent la date et l'heure courante du système.

# Les autres types de colonnes

## Données binaires :

- ▶ Les quatre types BLOB (*Binary Large Object*) [TINYBLOB, BLOB, MEDIUMBLOB et LONGBLOB] permettent de stocker des données non organisées comme le multimédia (images, sons, vidéo, etc.), considérés comme des flux d'octets sans jeu de caractère.

## Deux types collections proposés par MySQL:

- ▶ ENUM définit une liste de valeurs permises (chaînes de caractères).
- ▶ SET permet de comparer une liste à une combinaison de valeurs permises à partir d'un ensemble de référence (chaînes de caractères).

# DESCRIBE ou structure d'une table

- ▶ Elle permet d'afficher la structure d'une table ou d'une vue. Sa syntaxe est la suivante :

DESCRIBE [nomBase.] nomTableouVue [colonne];

- ▶ Exemple :

```
mysql> DESCRIBE Pilote;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| brevet | char(6) | NO   | PRI |          |        |
| nom    | char(15) | YES  | UNI | NULL    |        |
| nbHVol | double(7,2)| YES |      | NULL    |        |
| compa  | char(4)  | YES  | MUL | NULL    |        |
+-----+-----+-----+-----+-----+
mysql> DESCRIBE Compagnie;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| comp  | char(4) | NO   | PRI |          |        |
| nrue  | int(3)  | YES  |      | NULL    |        |
| rue   | char(20) | YES  |      | NULL    |        |
| ville | char(15) | YES  |      | Paris   |        |
| nomComp | char(15) | NO   |      |          |        |
+-----+-----+-----+-----+-----+
```

# Index d'une colonne avec MySQL

- ▶ Il permet d'accélérer l'accès aux données d'une table, utile pour éviter de parcourir une table séquentiellement .
- ▶ Un index est associé à une table, plus ou moins défini sur une ou plusieurs colonnes (dites « indexées »), peut être déclaré unique dans le besoin.
- ▶ Une table peut « accueillir » plusieurs index ou aucun, les colonnes indexées sont mis à jour obligatoirement après rafraîchissement de la table.

# Création d'index sur une colonne

- ▶ Avoir le privilège INDEX sur la base concernée pour créer ou supprimer des index.
- ▶ CREATE INDEX permet de créer un index sur une colonne;
- ▶ DROP INDEX permet de le supprimer;
- ▶ La syntaxe est la suivante :

**CREATE** [UNIQUE | FULLTEXT | SPATIAL] **INDEX**  
***nomIndex***

[USING BTREE | HASH]

ON *nomTable* (*colonne1* [(*taille1*) ] [ASC | DESC], . . .) ;

# Exemple de déclaration d'index

```
CREATE INDEX idx_Pilote_compa  
USING BTREE  
ON Pilote (compa);
```

```
CREATE UNIQUE INDEX idx_Pilote_nom3  
USING BTREE  
ON Pilote (nom(3) DESC);
```

# Remarques sur les index

- ▶ Un index ralentit l'actualisation de la base, mais accélère l'accès.
- ▶ Il est recommandé de créer des index sur des colonnes utilisées dans les clauses de jointures.
- ▶ Possibilité de créer des index sur les toutes les colonnes d'une table (max 16).
- ▶ Pénalisant sur une table très souvent modifiée, ou contenant peu des enregistrements.

# Destruction d'une table : DROP TABLE

- ▶ Posséder le privilège DROP sur cette base.
- ▶ L'ordre de destruction des tables est table « fils » ensuite « père ».
- ▶ DROP TABLE entraîne la suppression des données, de la structure, de la description dans le dictionnaire des données, des index, des déclencheurs associés (*triggers*) et la récupération de la place dans l'espace de stockage. Syntaxe :

```
DROP [TEMPORARY] TABLE [IF EXISTS]  
[nomBase.] nomTable1  
[, [nomBase2.] nomTable2, ...]  
[RESTRICT | CASCADE]
```

# Exemple de destruction d'une table

## ► Avec CASCADE

```
DROP TABLE Compagnie CASCADE;
```

```
DROP TABLE Pilote;
```

## ► « fils » puis « père »

```
DROP TABLE Pilote;
```

```
DROP TABLE Compagnie;
```

# **FIN DU CHAPITRE**