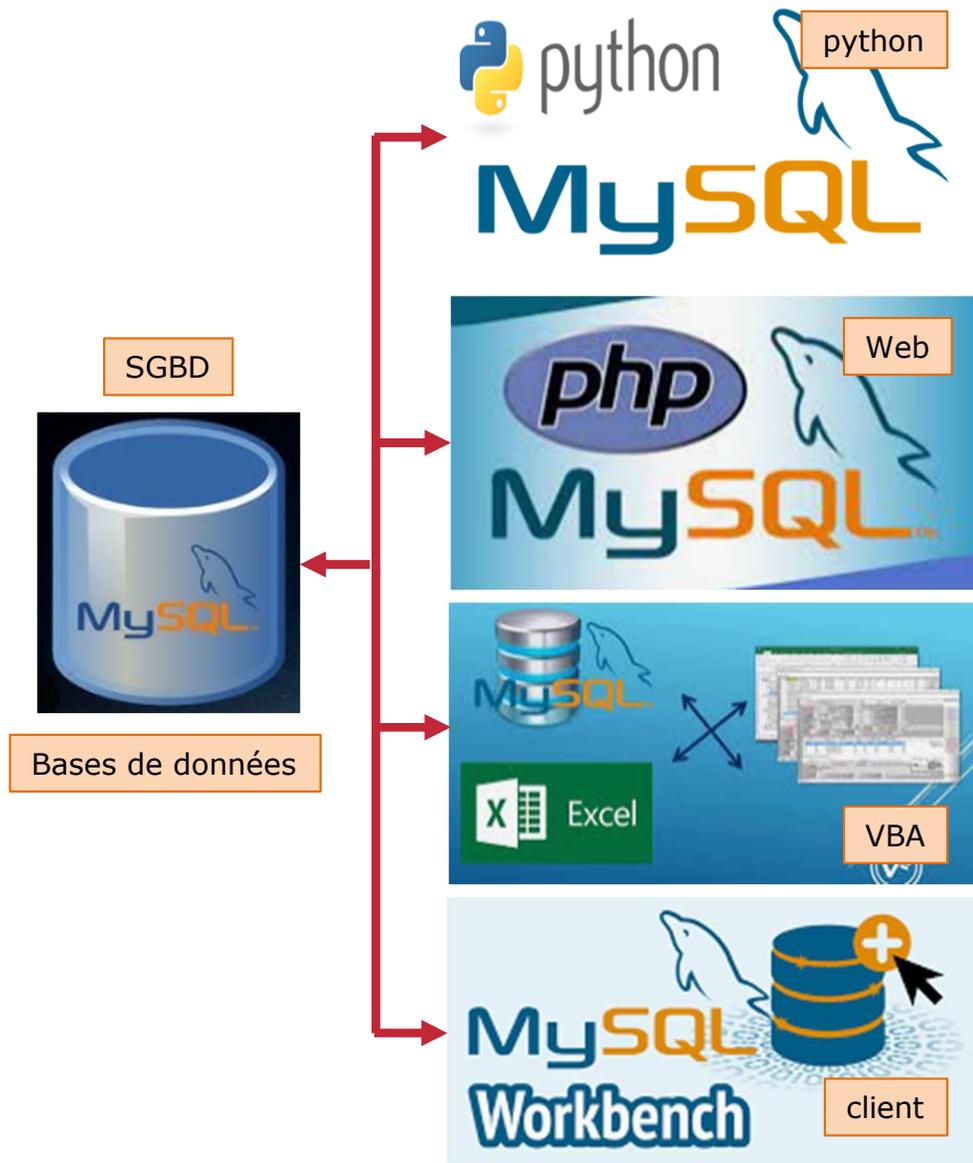


# COURS F1 : SGBD

SGBD = Système de Gestion de Bases de données

BD = Base de données



**SGBD**: Suite de logiciels qui permet la gestion en SQL de plusieurs bases de données.

**BD**: collection de données structurées selon un modèle de données, afin d'en centraliser le dépôt, la sauvegarde, l'accès, l'intégrité et la sécurité.

**Console mysql**: interface texte permettant de gérer les BD d'un SGBD.

**PHPMysqlAdmin**: interface Web en PHP permettant de gérer les BD d'un SGBD.

**SQL**: *Structured Query Language*  
Langage de requêtes structurées

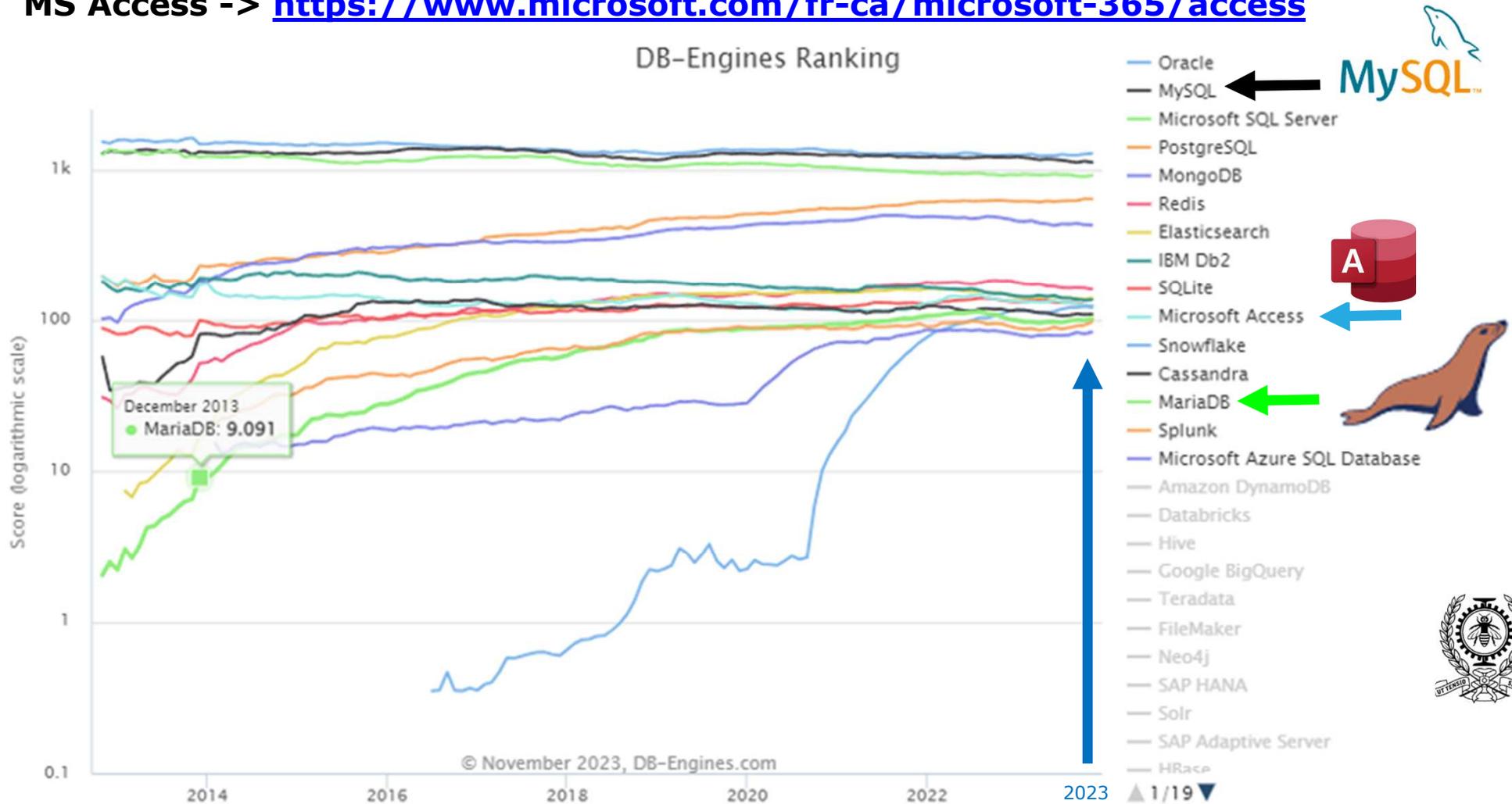
# COURS E1 : SGBD

cogito.meca.polymtl.ca -> MariaBD

MySQL -> <https://www.mysql.com/fr/>

Maria DB -> <https://mariadb.org/>

MS Access -> <https://www.microsoft.com/fr-ca/microsoft-365/access>



# COURS F1 : SGBD

## Modèle de de données relationnelles

**Modèle relationnel:** Ensemble de tables en relation par des clefs primaires et étrangères

**Ligne:** **enregistrement** d'un nouvel événement ou d'un nouvel objet

Ex: un nouveau produit  **enregistrement**

**Colonne:** **attribut** des enregistrements ayant le même **domaine**, et aussi, le même type de données (nombre, texte, booléen, ...)

Table Produits

Produits			
ip	Modele	Watt	Prix
101	Eolienne horizontale 3 pales	5000	8 000.00 \$
102	Eolienne horizontale 5 pales	7000	9 000.00 \$
103	Eolienne verticale 2 pales	6000	4 000.00 \$
104	Eolienne verticale 6 pales	7500	5 500.00 \$
105	Controleur de charge	8000	3 400.00 \$
106	Module de batterie SOK	5200	2 750.00 \$

Diagram annotations: An orange bracket above the table spans the width of the data rows. Two red arrows point down to the 'Watt' and 'Prix' columns. A blue box highlights the row for 'ip' 103. A green box highlights the 'Watt' column, with a green arrow pointing to the text 'Ex: le domaine de l'attribut Watt est un entier positif' below the table.

Ex: le **domaine** de l'attribut **Watt** est un **entier positif**



# COURS F1 : SGBD

## Modèle de données relationnelles

Produits			
ip	Modele	Watt	Prix
101	Eolienne horizontale 3 pales	5000	8 000.00 \$
102	Eolienne horizontale 5 pales	7000	9 000.00 \$
103	Eolienne verticale 2 pales	6000	4 000.00 \$
104	Eolienne verticale 6 pales	7500	5 500.00 \$
105	Controleur de charge	8000	3 400.00 \$
106	Module de batterie SOK	5200	2 750.00 \$

↑  
Clef primaire  
unique

Relation

Ventes			
iv	RefProduit	Date	Qte
1	102	"2023-11-01"	3
2	105	"2023-11-01"	3
3	102	"2023-11-05"	2
4	106	"2023-11-06"	4

↑  
Clef primaire  
unique

↑  
Clef étrangère  
non unique

**Clef primaire:** identifiant unique et obligatoire des enregistrements d'une table

Ex: l'attribut **ip** est la clef primaire de la table **Produits**

**Clef étrangère:** référence non unique vers la clef primaire d'une autre table

Ex: l'attribut **RefProduit** de la table **Ventes** pointe vers l'attribut **ip** de la table **Produits**



# COURS F1 : SGBD

## Modèle de données relationnelles

Conception d'un système d'information

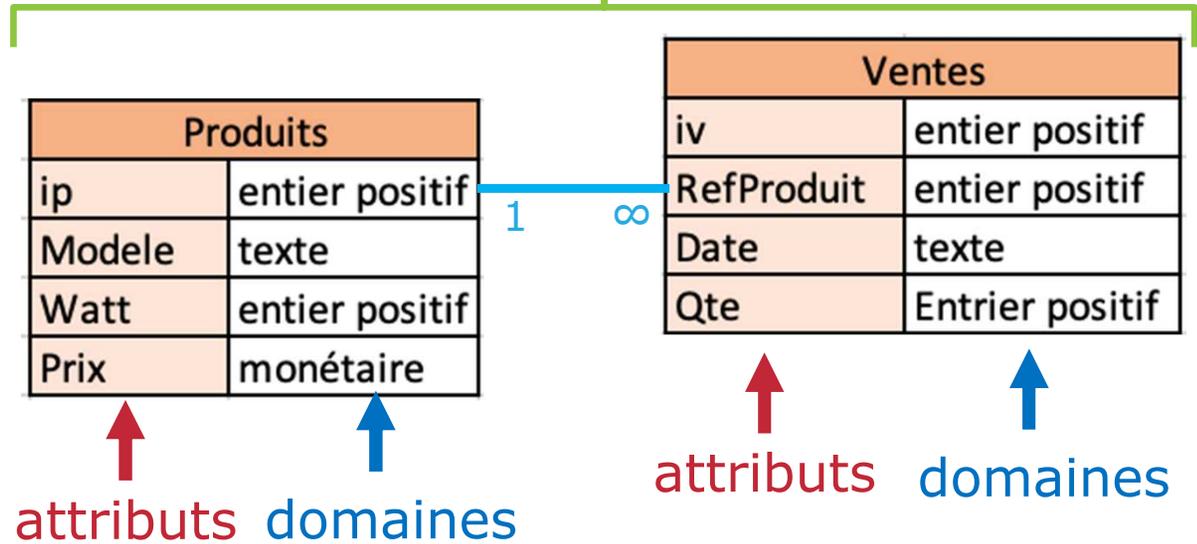


Diagramme relationnel



Code SQL

### Diagramme relationnel



```
CREATE TABLE Produits (  
  ip INT UNSIGNED NOT NULL PRIMARY KEY,  
  Modele VARCHAR(30) NOT NULL,  
  Watt INT UNSIGNED NOT NULL,  
  Prix DECIMAL(8,2) DEFAULT 1000.00);
```

```
CREATE TABLE Ventes (  
  iv INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  RefProduit INT UNSIGNED NOT NULL REFERENCES Produits,  
  Date CHAR(10) DEFAULT "2023-11-01",  
  Qte INT UNSIGNED);
```



```
> mysql -u ti660 -p ← Démarre la console mysql sur cogito avec
Enter password: ***** votre code usager MySQL et mot de passe
Welcome to MariaDB monitor
...
> SHOW DATABASES; ← Affiche les BD dont vous avez accès
                          incluant votre BD personnelle ti660

+-----+
| Database |
+-----+
| H2006    |
| auto2014 |
| ti660    |
+-----+

xx rows in set (0.001 sec)

> USE ti660; ← Utilisez votre BD personnelle ti660
Database changed

> Source CoursF1.sql; ← Exécute les commandes SQL du fichier CoursF1.sql sur cogito
Query OK, 0 rows affected (0.021 sec)

> SHOW TABLES; ← Affiche les tables de la BD ti660

+-----+
| Tables_in_ti660 |
+-----+
| Produits        |
| Ventes          |
| Zeffacer        |
+-----+

3 rows in set (0.000 sec)

> DROP TABLE Zeffacer; ← Efface la table Zeffacer
Query OK, 0 rows affected (0.015 sec)

> EXIT; ← Quitte la console d'accès à MariaDB
Bye
>
```

**Attention**, le ; demande l'exécution de la commande inscrite sur une ou plusieurs lignes

Nous venons de créer les tables Produits et Ventes, ainsi que la **Zeffacer**.



```
> DESCRIBE Produits;
```

Field	Type	Null	Key	Default	Extra
ip	int(10) unsigned	NO	PRI	NULL	
Modele	varchar(30)	NO		NULL	
Watt	int(10) unsigned	NO		NULL	
Prix	decimal(8,2)	YES		1000.00	

```
4 rows in set (0.001 sec)
```

```
> SELECT * FROM Produits;
```

ip	Modele	Watt	Prix
101	Eolienne horizontale 3 pales	5000	8000.00
102	Eolienne horizontale 5 pales	7000	9000.00
103	Eolienne verticale 2 pales	6000	4000.00
104	Eolienne verticale 6 pales	7500	5500.00
105	Controleur de charge	8000	3400.00
106	Module de batterie SOK	5200	2750.00

```
6 rows in set (0.000 sec)
```

```
> SELECT Watt, Modele, Prix FROM Produits ORDER BY Watt ASC;
```

Watt	Modele	Prix
5000	Eolienne horizontale 3 pales	8000.00
5200	Module de batterie SOK	2750.00
6000	Eolienne verticale 2 pales	4000.00
7000	Eolienne horizontale 5 pales	9000.00
7500	Eolienne verticale 6 pales	5500.00
8000	Controleur de charge	3400.00

```
6 rows in set (0.001 sec)
```

Clef primaire (entier positif)

Modèle du produit (max 30 carac.)

Puissance du produit (entier positif)

Prix à 8 chiffres et 2 décimales afin d'éviter les arrondis en FLOAT

Affiche tous les attributs de la table **Produits**

Affiche les attributs **Watt**, **Modele** et **Prix** de la table **Produits** avec les enregistrements en ordre croissant de **Prix**

**ASC** peut être omit puisque c'est l'ordre par défaut

**DESC** ordre descendant



# COURS F1 : SGBD

## Console d'accès à MariaDB sur cogito (3)

### Table Produits

```
> SELECT Modele, Watt AS Puissance FROM Produits Order By Puissance DESC Limit 4;
```

Modele	Puissance
Contrôleur de charge	8000
Eolienne verticale 6 pales	7500
Eolienne horizontale 5 pales	7000
Eolienne verticale 2 pales	6000



Affiche les attributs **Modele** et **Watt** renommé **Puissance**, puis ordonne par ordre décroissant de puissance et limite l'affichage à 4 lignes

```
4 rows in set (0.000 sec)
```

```
> SELECT Modele, Watt FROM Produits WHERE Prix<5000 AND Watt>5000;
```

Modele	Watt
Eolienne verticale 2 pales	6000
Contrôleur de charge	8000
Module de batterie SOK	5200

Une seule éolienne répond aux 2 critères simultanément, soient un prix inférieur à 5000\$ et une puissance supérieure à 5000 Watt.

```
3 rows in set (0.000 sec)
```

```
> SELECT Modele, Prix FROM Produits WHERE Modele LIKE "%verticale%";
```

Modele	Prix
Eolienne verticale 2 pales	4000.00
Eolienne verticale 6 pales	5500.00

Affiche les attributs **Modele** et **Prix** de la table **Produits** avec seulement les enregistrements ayant le mot verticale dans le nom de modèle.

```
2 rows in set (0.000 sec)
```

```
> SELECT COUNT(*),AVG(Prix) FROM Produits WHERE Modele Like "Eolienne%";
```

COUNT(*)	AVG(Prix)
4	6625.000000

% signifie n'importe quelle chaîne de caractères

Calcul le nombre d'enregistrement débutant par "Eolienne", ainsi que la moyenne de l'attribut **Prix** de ces 4 lignes.

Les fonctions sont: **MAX()** = maximum; **MIN()** = minimum;

**AVG()** = moyenne; **SUM()** = somme; Le génie en première classe 8/14



# COURS F1 : SGBD

## Console d'accès à MariaDB sur cogito (4)

### Table Ventes

```
> DESCRIBE Ventes;
```

Field	Type	Null	Key	Default	Extra
iv	int(10) unsigned	NO	PRI	NULL	auto_increment
RefProduit	int(10) unsigned	NO		NULL	
Date	char(10)	YES		2023-11-01	
Qte	int(10) unsigned	YES		NULL	

Clef primaire (entier positif)  
à incrément automatique

Clef étrangère vers Produits

Date (10 caractères max.)

Qte (entier positif)

```
4 rows in set (0.001 sec)
```

```
> SELECT * FROM Ventes;
```

iv	RefProduit	Date	Qte
1	102	2023-11-01	3
2	105	2023-11-01	3
3	102	2023-11-05	2
4	106	2023-11-06	4

Affiche tous les attributs de la table **Ventes**

```
4 rows in set (0.000 sec)
```

```
> SELECT Date, Modele, Watt, Qte, Prix From Ventes, Produits Where RefProduit=ip;
```

Date	Modele	Watt	Qte	Prix
2023-11-01	Eolienne horizontale 5 pales	7000	3	9000.00
2023-11-01	Controleur de charge	8000	3	3400.00
2023-11-05	Eolienne horizontale 5 pales	7000	2	9000.00
2023-11-06	Module de batterie SOK	5200	4	2750.00

Affiche des attributs en provenance  
des 2 tables

Il faut assurer la relation :  
**RefProduit = ip**  
ou en général

**Ventes.RefProduit = Produits.ip**



Dans cet exemple, les attributs ont tous des noms uniques.

Il n'est donc pas nécessaire d'indiquer le nom de la table de leur provenance.

# COURS F1 : SGBD

## Console d'accès à MariaDB sur cogito (5)

Fichier: CoursF1.sql

```
/* MEC1315: Fichiers CoursF1.sql
   Copyright Luc Baron Nov. 2023 */
/* Table des Produits */
DROP TABLE IF EXISTS Produits;
CREATE TABLE Produits (
  ip INT UNSIGNED NOT NULL PRIMARY KEY,
  Modele VARCHAR(30) NOT NULL,
  Watt INT UNSIGNED NOT NULL,
  Prix DECIMAL(8,2) DEFAULT 1000.00);
```

```
INSERT INTO Produits VALUES(101, "Eolienne horizontale 3 pales", 5000, 8000.00);
INSERT INTO Produits VALUES(102, "Eolienne horizontale 5 pales", 7000, 9000.00);
INSERT INTO Produits VALUES(103, "Eolienne verticale 2 pales", 6000, 4000.00);
INSERT INTO Produits VALUES(104, "Eolienne verticale 6 pales", 7500, 5500.00);
INSERT INTO Produits VALUES(105, "Controleur de charge", 8000, 3400.00);
INSERT INTO Produits VALUES(106, "Module de batterie SOK", 5200, 2750.00);
```

```
/* Table des Ventes */
DROP TABLE IF EXISTS Ventes;
CREATE TABLE Ventes (
  iv INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
  RefProduit INT UNSIGNED NOT NULL REFERENCES Produits,
  Date CHAR(10) DEFAULT "2023-11-01",
  Qte INT UNSIGNED);
```

```
INSERT INTO Ventes VALUES(NULL, 102, "2023-11-01", 3);
INSERT INTO Ventes VALUES(NULL, 105, "2023-11-01", 3);
INSERT INTO Ventes VALUES(NULL, 102, "2023-11-05", 2);
INSERT INTO Ventes VALUES(NULL, 106, "2023-11-06", 4);
```

```
CREATE TABLE Zeffacer ( /* Table pour effacer */
  id INT PRIMARY KEY,
  a VARCHAR(10) NOT NULL)
```

Pour exécuter un fichier texte **.sql**, il faut:

- 1- télécharger le fichier **.sql** sur cogito;
- 2- déposer le fichier dans le répertoire de l'utilisateur **ti660**, c'est-à-dire **/home/ti660**;
- 3- démarrer la console mysql à partir du répertoire **/home/ti660**;
- 4- indiquer l'usage de votre BD (**USE ti660**);
- 5- utiliser la commande **source fichier.sql**

Insertion des données dans la table **Produits**:

4 attributs = 4 données par enregistrement

Insertion des données dans la table **Ventes**:

4 attributs = 4 données par enregistrement

sauf que l'attribut **iv** est en mode AUTO INCREMENT, et donc, ne doit pas être spécifié.

On utilise alors NULL.



# COURS F1 : SGBD

## Exploration d'une BD - Horaire de Poly H2006 (1)

La base de données **H2006** contient  tables.  
`SHOW TABLES;`

La table **Cours** contient  attributs.  
`DESCRIBE Cours;`

L'attribut **intitule** peut contenir un maximum de  caractères.  
`DESCRIBE Cours;`

La table **Cours** contient  enregistrements.  
`SELECT * FROM Cours;`      `SELECT COUNT(*) FROM Cours;`

Combien y-a-t'il de cours de 2 crédits ?   
`SELECT * FROM Cours WHERE credits=2;`

Quel est le nom du cours de 0 crédit ?   
`SELECT * FROM Cours WHERE credits=0;`

Quel est le sigle du cours de 12 crédits ?   
`SELECT * FROM Cours WHERE credits=12;`

Quel est le sigle du cours de 6 crédits ?   
`SELECT * FROM Cours WHERE credits=6;`



# COURS F1 : SGBD

## Exploration d'une BD - Horaire de Poly H2006 (2)

Préparer un histogramme du nombre de cours par nombre de crédit ?

```
SELECT Credits, COUNT(*) FROM Cours GROUP BY Credits;
```

Credits	COUNT(*)
0	1
1	27
2	69
3	220
4	14
6	15
9	11
12	9
15	8

Préparer une requête SQL permettant d'afficher le sigle, le titre et le nombre de crédits des cours de mathématique de 2 crédits ? **Noter l'ordre et le titre des colonnes.**

Sigle	Cours de mathematique	Credits
MTH1006	ALGEBRE LINEAIRE	2
MTH1101	CALCUL I	2
MTH1102	CALCUL II	2
MTH1110	EQUATIONS DIFFERENTIELLES ORD.	2
MTH2111	METH. MATH. DE LA PHYSIQUE II	2

```
SELECT sigle AS Sigle, intitule AS "Cours de mathematique", credits AS Credits FROM Cours WHERE credits=2 AND sigle LIKE "MTH%";
```

Préparer une requête SQL permettant d'afficher les cours MECx5xx (où x est caractère inconnu) en ordre croissant de crédit et décroissant de sigle (tel que ci-dessous) ?

Credits	Sigle	Intitule
2	MEC1515	DAO EN INGENIERIE
3	MEC4530	FABRICATION MECANIQUE AVANCEE
3	MEC4510	FAO ET MACHINES-OUTILS
3	MEC2510	TECHN. INFORMATIONN. EN MEC.
3	MEC2500	DEF. TECHNOLOG. DE PRODUITS MEC.
3	MEC1510	MODEL. SYSTEMES MECANQUES
4	MEC3500	FABRICATION ET METAUX

**Noter l'ordre et les titres**

**Le symbole \_ indique un caractère quelconque**

**Les premières lettres des attributs peuvent être mises en majuscules sans la commande d'alias AS**

```
SELECT Credits, Sigle, Intitule FROM Cours WHERE sigle LIKE "MEC_5__" ORDER BY Credits ASC, sigle DESC;
```



# COURS F1 : SGBD

## Exploration d'une BD - Horaire de Poly H2006 (3)

Composition de la table **Sections** ?

Field	Type	Null	Key	Default	Extra
sigle	varchar(8)	NO	PRI		
groupe	int(11)	NO	PRI	0	
type	char(2)	NO	PRI		
frequence	char(3)	YES		NULL	
professeur	varchar(25)	YES		NULL	

**DESCRIBE Sections;**

Quel sont les sections de cours et/ou laboratoire enseignés par le prof. « P.Tanguy » ?

sigle	groupe	type	frequence	professeur
GCH4290	1	L	H	P.Tanguy
GCH4390	1	L	H	P.Tanguy
GCH4590	1	L	H	P.Tanguy
GCH4690	1	L	H	P.Tanguy

**SELECT \* FROM Sections WHERE professeur="P.Tanguy";**

Quel est le nombre total de groupe cours et laboratoire enseigné en GCH... ?

Total GCH	COUNT(*)
Total GCH	39

Il est possible de remplacer le nom d'un attribut par un titre fixe

**SELECT "Total GCH", COUNT(\*) FROM Sections WHERE sigle LIKE "GCH%";**

La clef primaire de la table **Sections** est formée des attributs :  
**sigle + groupe + type**

À l'hiver 2006, le cours **MEC1310** avait 3 groupes de cours (type C) et 3 groupes de Laboratoire (type L), c'est-à-dire:  
**MEC1310 + 1 + C & MEC1310 + 1 + L**  
**MEC1310 + 2 + C & MEC1310 + 2 + L**  
**MEC1310 + 3 + C & MEC1310 + 3 + L**

**Attention**, pas d'espace dans le nom des enseignants

Toutes des sections de laboratoires



# COURS F1 : SGBD

## Exploration d'une BD - Horaire de Poly H2006 (4)

Préparer une requête permettant d'afficher le sigle, le type et le nombre de section ?

sigle	type	Nb section
MTH0102	C	2
MTH0102	L	2
MTH0103	C	2
MTH0103	L	2
. . .	. . .	. . .
MTH2305	C	2
MTH2305	L	2
MTH2401	C	1
MTH2401	L	1

Attention au titre de la dernière colonne

```
SELECT sigle, type, COUNT(groupe) AS "Nb section" FROM Sections WHERE sigle LIKE "MTH%" GROUP By sigle, type;
```

Préparer une requête permettant d'afficher le nombre de section de laboratoire des cours MEC avec les titres tels que montré ci-dessous ?

Labo	Section
MEC	61

Noter les 2 titres de colonne, ainsi que le remplacement de l'attribut à la 1<sup>ère</sup> colonne

```
SELECT "MEC" AS "Labo", COUNT(*) AS Section FROM Sections WHERE type = "L" AND sigle LIKE "MEC%";
```

