gilles.hunault@univ-angers.fr Angers, novembre 2004

Un petit tuteur MYSQL

Table des matières

1. Présentation de MYSQL

MYSQL est un gestionnaire de bases de données SQL. Il se compose d'un langage de définition de données et de droits ainsi qu'un langage de manipulation des données.

Une base de données regroupes plusieurs tables de données. Par exemple pour une application dont la base est nommée GESTION, les tables pourront s'appeler CLIENTS, FOURNISSEURS, ARTICLES et COMMANDES.

On suppose que le lecteur, la lectrice de ce cours ont quelques connaissances sur les bases de données en général. Sinon, nous recommandons le cours de P. RIGAUX et l'ensemble du site Web correspondant qui s'adresse aux étudiants du CNAM, cycle A à l'adresse

```
http://sikkim.cnam.fr/~rigaux/bdpi.html
```

MYSQL est disponible sous *Windows* et sous *Unix*. Le site officiel est http://www.mysql.com. La version utilisée pour ce tuteur est 3.22.29.

Une fois le logiciel MYSQL installé, configuré et avec des droits utilisateurs corrects, on peut l'utiliser soit en ligne de commande (en interactif ou par programme) soit par l'intermédiaire d'une API (interface de programmation) dans des langages comme *perl*, *php*...

Pour utiliser MYSQL en interactif, on tape en ligne de commande

```
mysql NOM_DE_BASE
```

et on quitte la session MYSQL en tapant

```
quit ;
```

En mode non interactif, si on met les instructions dans un fichier, on peut exécuter le programme correspondant par

```
mysql NOM_DE_BASE < NOM_DU_FICHIER
```

Au lieu d'afficher les résultats à l'écran, il est possible de les rediriger dans un fichie de sortie, suivant la syntaxe

```
mysql NOM_DE_BASE < NOM_DU_FICHIER > NOM_DE_SORTIE
```

Il est à noter que chaque instruction MYSQL peut s'étendre sur plusieurs lignes mais qu'elle doit se terminer par un point-virgule.

2. Création de bases, gestion des droits

C'est en principe le responsable système ("root") qui crée une base, par exemple avec la commande mysqadmin. Les droits peuvent ensuite être transmis par l'instruction GRANT. Par exemple, la création de la base TUTEUR par le responsable système se fait avec

```
mysqladmin create TUTEUR ;
```

ensuite, ce même responsable peut exécuter en ligne

```
mysql TUTEUR -e "GRANT ALL PRIVILEGES ON tuteur.* TO gh@localhost ;"
```

et l'utilisateur *gh* peut alors localement faire tout ce qu'il veut avec la base.

Les droits gérés par GRANT sont

```
ALL PRIVILEGES, ALTER, CREATE, DELETE, DROP, FILE, INDEX, INSERT,
PROCESS, REFERENCES, RELOAD, SELECT, SHUTDOWN, UPDATE, USAGE
```

Signalons que si l'instruction GRANT se termine par

```
WITH GRANT OPTION ;
```

l'utilisateur désigné peut à son tour transmettre des droits.

3. Eléments du langage de MYSQL

Un commentaire est une instruction non exécutable qui commence par un dièse. La première instruction exécutable d'une session ou d'un programme MYSQL doit être

```
USE Nom_Base ;
```

afin de choisir la base (et donc les tables). L'instruction SELECT permet d'afficher des valeurs et d'extraire des données des bases. Par exemple

```
SELECT VERSION() ;
```

affiche le numéro de version courante

```
SELECT COUNT(*) FROM Nom_Table ;
```

indique le nombre d'enregistrements dans la table.

L'instruction SHOW affiche de nombreux renseignements concernant les bases et les tables. Par exemple

```
SHOW DATABASES ;
```

donne la liste de toutes les bases.

```
SHOW VARIABLES ;
```

donne la liste et la valeur de toutes les variables.

```
SHOW STATUS ;
```

décrit l'état de l'ensemble des paramètres de MYSQL.

L'instruction DESCRIBE donne des informations sur une table particulière. Par exemple

```
USE  Test ;
DESCRIBE Tuteur ;
```

décrit toutes les variables de la table Tuteur pour la base Test alors que

```
DESCRIBE Tuteur Qt ;
```

ne décrit que le champ Qt de cette table.

La création d'une table se fait avec l'instruction CREATE suivie du mot TABLE et du nom de la table. On indique ensuite entre parenthèses le nom et le type des champs en séparant les champs par des virgules. Exemple :

```
CREATE TABLE fournisseurs ( code INT, nom CHAR ) ;
```

Les types de champs possibles sont

```
INT, TINYINT, SMALLINT, MEDIUMINT, BIGINT,
FLOAT, DOUBLE, DECIMAL,
CHAR, VARCHAR, TEXT, TINYTEXT, MEDIUMTEXT, LONGTEXT,
ENUM, SET.
```

Un fois la table créée, il est possible de modifier la structure avec l'instruction ALTER, comme par exemple

```
ALTER TABLE fournisseurs ADD prenom char, ADD ref int ;
```

Les spécifications possibles pour ALTER sont

```
ADD, ALTER, CHANGE, MODIFY, DROP, RENAME.
```

Pour détruire une table, on utilise DROP, comme par exemple

```
DROP TABLE fournisseurs ;
```

Pour ajouter des données, on utilise INSERT, REPLACE, UPDATE ET SET. L'instruction INSERT crée des données pour la première fois. En cas d'index non nul, il n'est pas possible d'insérer deux fois la meme valeur.

Exemples d'INSERT :

```
INSERT INTO fournisseurs VALUES (10,"peper","jean",12) ;
INSERT INTO fournisseurs (nom) VALUES  ("memer") ;
INSERT INTO fournisseurs  (code,ref) VALUES  (100,200) ;
INSERT INTO fournisseurs  (code,ref) VALUES  (100,code+1000) ;
```

Pour détruire des lignes on utilise DELETE, les conditions de suppression étant données dans le WHEN. Par exemple :

```
DELETE FROM fournisseurs WHERE ref<100 ;
```

et pour tout détruire, il suffit de ne donner aucune condition, comme

```
DELETE FROM fournisseurs  ;
```

L'extraction avec SELECT se fait selon la syntaxe générale

```
SELECT expresion INTO
                 FROM
                 WHERE
                 GROUP BY
                 HAVING
                 ORDER BY
                 LIMIT
```

On trouvera dans les programmes qui suivent de nombreux exemples de SELECT.

4. Exemple de programmes de manipulation en MYSQL

Il y a 12 programmes. Le programme 1 montre les instructions USE, SHOW, DESCRIBE. Le programme 2 reprend SHOW. La création d'une table correspond au programme 3 et on vient afficher la structure de la table nouvellement créée dans le programme 4, que l'on modifie par le programme 5. Quant au programme 6, il assure l'insertion de valeurs dans la table, valeurs que l'on affiche et compte par le programme 7. Nous ajoutons une deuxième table avec des valeurs dans le programme 8 de façon à pouvoir, dans le programme 9 introduire la qualification totale, puis nous effectuons diverses jointures et des calculs plus techniques, comme la moyenne, l'écart-type sur des groupes d'enregistrements dans le programme 10. Ensuite, le programme 11 utilise encore une autre base et montre des interrogations sur des champs-texte via des expressions régulières ainsi que divers tris et affichages et enfin le programme 12 montre comment calculer puis remplir des champs à partir d'autres champs.

```
Source du Programme 1  :
-------------------------


 # p01.msql
 # essayons d'utiliser la base tuteur

 use tuteur ;

 # exéction du script par  mysql -f < p01.msql > resultats.p01
 # on peut aussi utiliser le paramètre -v (pour "verbose")

 use  test     ;
 show tables   ;
 describe test.demo ;

 # la commande mysqlshow liste les tables

 # les commandes
 #     use  XXX ;
 #     show tables ;
 # donnent les tables de la base XXX ;
```

```
 # les commandes
 #     use  XXX ;
 #     describe YYY ;
 # décrivent la table YYY de la base XXX

 # fin de p01.msql
```

Résultats du Programme 1 :

```
--------------
show tables
--------------

Tables in test
afone
demo
diremail
eusers
stages
stagesM
stgmass

--------------
describe test.demo
--------------

Field     Type       Null         Key     Default     Extra
nom       char(30)   YES          NULL
prenom    char(30)   YES          NULL
age       int(2)     YES          NULL
naiss     int(4)     YES          NULL
cine      int(1)     YES          NULL
moto      int(1)     YES          NULL
```

Source du Programme 2 :

```
 # p02.msql

 use test ;
```

```
show tables ;

# si on est root, on peut ouvrir la table
# user de la base mysql et gérer les
# droits...

#   mysql> use mysql ;
#   Database changed
#   mysql> show tables ;
#   +-----------------+
#   | Tables in mysql |
#   +-----------------+
#   | db              |
#   | func            |
#   | host            |
#   | user            |
#   +-----------------+
#   4 rows in set (0.00 sec)
#
#   mysql> describe user ;
#   +---------------+----------+------+-----+---------+-------+
#   | Field         | Type     | Null | Key | Default | Extra |
#   +---------------+----------+------+-----+---------+-------+
#   | Host          | char(60) |      | PRI |         |       |
#   | User          | char(16) |      | PRI |         |       |
#   | Password      | char(16) |      |     |         |       |
#   | Select_priv   | char(1)  |      |     | N       |       |
#   | Insert_priv   | char(1)  |      |     | N       |       |
#   | Update_priv   | char(1)  |      |     | N       |       |
#   | Delete_priv   | char(1)  |      |     | N       |       |
#   | Create_priv   | char(1)  |      |     | N       |       |
#   | Drop_priv     | char(1)  |      |     | N       |       |
#   | Reload_priv   | char(1)  |      |     | N       |       |
#   | Shutdown_priv | char(1)  |      |     | N       |       |
#   | Process_priv  | char(1)  |      |     | N       |       |
#   | File_priv     | char(1)  |      |     | N       |       |
#   +---------------+----------+------+-----+---------+-------+
#
#   13 rows in set (0.00 sec)
#
#
```

```
 #   mysql> describe host ;
 #   +-------------+----------+------+-----+---------+-------+
 #   | Field       | Type     | Null | Key | Default | Extra |
 #   +-------------+----------+------+-----+---------+-------+
 #   | Host        | char(60) |      | PRI |         |       |
 #   | Db          | char(32) |      | PRI |         |       |
 #   | Select_priv | char(1)  |      |     | N       |       |
 #   | Insert_priv | char(1)  |      |     | N       |       |
 #   | Update_priv | char(1)  |      |     | N       |       |
 #   | Delete_priv | char(1)  |      |     | N       |       |
 #   | Create_priv | char(1)  |      |     | N       |       |
 #   | Drop_priv   | char(1)  |      |     | N       |       |
 #   +-------------+----------+------+-----+---------+-------+
 #   8 rows in set (0.00 sec)

 # fin de p02.msql
```

Résultats du Programme 2 :

```
--------------
show tables
--------------

Tables in test
afone
demo
diremail
eusers
stages
stagesM
stgmass
```

Source du Programme 3 :

```
 # -- p03.msql

 # on peut utiliser ce programme par mysql < p03.msql
 # ou, pour plus de detail par mysql -v < p03.msql
```

```
 # créons une nouvelle table

 use test ;
 create table tuteur ( usr char(2) not null, qt int(4) )  ;

 # si on relance le programme une seconde fois, on a le message
 #    ERROR 1050 at line 6: Table 'tuteur' already exists

 # pour détruire la table
 #    drop   table tuteur ;

 # on peut écrire en ligne de commande ou en script
 #    mysql test -e "drop table tuteur " ;

 # -- fin de p03.msql
```

```
Résultats  du Programme 3 :

--------------
create table tuteur ( usr char(2) not null, qt int(4) )
```

```
Source du Programme 4  :
------------------------

# -- p04.sql

# demandons la structure de la table tuteur dans la base test

use test ;
describe tuteur ;

#   +-------+---------+------+-----+---------+-------+
#   | Field | Type    | Null | Key | Default | Extra |
#   +-------+---------+------+-----+---------+-------+
#   | qt    | int(4)  | YES  |     | NULL    |       |
#   | usr   | char(2) |      |     |         |       |
#   +-------+---------+------+-----+---------+-------+
#   2 rows in set (0.00 sec)# fin de p04.sql
# -- fin de p04.msql
```

Résultats du Programme 4 :

```
--------------
describe tuteur
--------------

Field     Type      Null          Key     Default     Extra
usr       char(2)
qt        int(4)     YES          NULL
```

Source du Programme 5 :

```
# -- p05.msql

# changeons la structure de la table tuteur

use test ;
alter table tuteur DROP usr, ADD user char(2) not null ;

# -- fin de p05.msql
```

Résultats du Programme 5 :

```
--------------
alter table tuteur DROP usr, ADD user char(2) not null
--------------
```

Source du Programme 6 :

```
# -- p06.msql

# ajoutons des valeurs à la main dans la table tuteur
# de la base test (champs qt et user)

use test ;
insert into tuteur values (1,'GH') ;
insert into tuteur values (2,'GG') ;
insert into tuteur values (3,'HU') ;
insert into tuteur values (4,'GH') ;
```

```
insert into tuteur values (5,'GH') ;
insert into tuteur values (3,'AA') ;
insert into tuteur values (5,'GH') ;
insert into tuteur values (8,'GG') ;
insert into tuteur values (7,'HU') ;
insert into tuteur values (2,'GH') ;
insert into tuteur values (5,'GH') ;
insert into tuteur values (9,'AA') ;


# -- fin de p06.msql
```

Résultats du Programme 6 :

```
--------------
insert into tuteur values (1,'GH')
--------------
insert into tuteur values (2,'GG')
--------------
insert into tuteur values (3,'HU')
--------------
insert into tuteur values (4,'GH')
--------------
insert into tuteur values (5,'GH')
--------------
...
```

Source du Programme 7 :

```
# -- p07.msql

use test ;

# demandons à voir tous les enregistrements

select * from tuteur ;
# en cas d'erreur, on vide la table par
# mysql test -e " delete from tuteur ; "

# demandons le nombre d'enregistrements...
```

```
select count(*) from  tuteur ;

# les différents utilisateurs

select distinct(user) from tuteur ;

# divers affichages

select distinct(qt) from tuteur order by qt ;

select user,qt from tuteur order by user ;
select qt,user from tuteur order by user ;
select user,qt,' soit ',qt*3.25 from tuteur order by user ;

# -- fin de p07.msql
```

Résultats du Programme 7 :

```
--------------
select * from tuteur
--------------

qt    user
1     GH
2     GG
3     HU
4     GH
5     GH
3     AA
5     GH
8     GG
7     HU
2     GH
5     GH
9     AA

--------------
select count(*) from  tuteur
--------------
```

```
count(*)
12

--------------
select distinct(user) from tuteur
--------------

user
AA
GG
GH
HU

--------------
select distinct(qt) from tuteur order by qt
--------------

qt
1
2
3
4
5
7
8
9

--------------
select user,qt from tuteur order by user
--------------

user   qt
AA     3
AA     9
GG     2
GG     8
...

--------------
select qt,user from tuteur order by user
--------------
```

```
qt    user
3     AA
9     AA
2     GG
8     GG
1     GH
4     GH
5     GH
5     GH
2     GH
5     GH
3     HU
7     HU

--------------
select user,qt,' soit ',qt*3.25 from tuteur order by user
--------------

user   qt     soit      qt*3.25
AA     3      soit      9.75
AA     9      soit      29.25
GG     2      soit      6.50
GG     8      soit      26.00
GH     1      soit      3.25
GH     4      soit      13.00
GH     5      soit      16.25
GH     5      soit      16.25
GH     2      soit      6.50
GH     5      soit      16.25
HU     3      soit      9.75
HU     7      soit      22.75




Source du Programme 8  :
------------------------
```

```
# -- p08.msql

# rajoutons une table decode des noms correspondant
# mis dans le champ nom ; les initiales dans le champ usr
# correspondent aux initiales mises dans le champ user
# de la table tuteur

use test ;

create table decode ( usr char(2) not null, nom char(40) )  ;

insert into decode values ('AA','Andrée Tarkowsky') ;
insert into decode values ('GG','Gérard Manvussa') ;
insert into decode values ('GH','Gilles Hunault') ;
insert into decode values ('HU','Hubert Hubert') ;

select * from decode ;

# -- fin de p08.msql
```

```
Résultats  du Programme 8 :

--------------
create table decode ( usr char(2) not null, nom char(40) )
--------------
insert into decode values ('AA','Andrée Tarkowsky')
--------------
insert into decode values ('GG','Gérard Manvussa')
--------------
insert into decode values ('GH','Gilles Hunault')
--------------
insert into decode values ('HU','Hubert Hubert')
--------------



select * from decode
--------------

usr     nom
```

```
AA      Andrée Tarkowsky
GG      Gérard Manvussa
GH      Gilles Hunault
HU      Hubert Hubert

Source du Programme 9  :
------------------------

# -- p09.msql

use test ;

# utilisons la qualification totale

select distinct(test.tuteur.user) from tuteur ;

# ceci est une erreur :
# select usr,distinct(test.decode.nom) from decode ;

# mais pas :

select distinct(usr),test.decode.nom from decode ;

# plus lisible

select distinct(usr),' est le code de : ',test.decode.nom from decode ;

# autre possibilité

select ' code : ',usr,' utilisateur : ',test.decode.nom from decode ;

# -- fin de p09.msql




Résultats  du Programme 9 :

--------------
select distinct(test.tuteur.user) from tuteur
```

```
--------------

user
AA
GG
GH
HU

--------------
select distinct(usr),test.decode.nom from decode
--------------

usr    nom
AA     Andrée Tarkowsky
GG     Gérard Manvussa
GH     Gilles Hunault
HU     Hubert Hubert

--------------
select distinct(usr),' est le code de : ',test.decode.nom from decode
--------------

usr     est le code de :      nom
AA      est le code de :      Andrée Tarkowsky
GG      est le code de :      Gérard Manvussa
GH      est le code de :      Gilles Hunault
HU      est le code de :      Hubert Hubert

--------------
select ' code : ',usr,' utilisateur : ',test.decode.nom from decode
--------------

 code :      usr     utilisateur :      nom
 code :      AA      utilisateur :      Andrée Tarkowsky
 code :      GG      utilisateur :      Gérard Manvussa
 code :      GH      utilisateur :      Gilles Hunault
 code :      HU      utilisateur :      Hubert Hubert
```

Source du Programme 10 :

```
# -- p10.msql
```

```
# lions les deux bases

use test ;

select decode.nom,tuteur.qt from decode,tuteur
       where tuteur.user=decode.usr ;

# comptage simple

select decode.nom,count(tuteur.qt) from decode,tuteur
       where tuteur.user=decode.usr group by decode.nom ;

# recherche du minimum

select decode.nom,min(tuteur.qt) from decode,tuteur
       where tuteur.user=decode.usr group by decode.nom  ;

# recherche du minimum en tant qu'alias et tri sur ce minimum

select min(tuteur.qt) as minQt, decode.nom from decode,tuteur
       where tuteur.user=decode.usr group by decode.nom order by minQt ;

# valeur moyenne de qt

select nom,avg(qt) as moyQt from decode,tuteur
       where user=usr group by nom order by moyQt ;

# valeur moyenne de qt*15 avec cadrage

select left(concat(nom,"................"),24),
avg(qt*15.0) as moyQt from decode,tuteur
       where user=usr group by nom order by moyQt ;



# moyenne, écart-type de qt avec cadrage et divers  tris

select left(concat(upper(nom),"           ..........."),35),avg(qt) as moyQt,
       std(qt) from decode,tuteur
       where user=usr group by user order by moyQt desc ;
```

```
select left(concat(upper(nom),"           ..........."),35),
       avg(qt),std(qt) as stdQt
       from decode,tuteur
       where user=usr group by user order by stdQt desc ;

select left(concat(upper(nom),"           ..........."),35),avg(qt),
       std(qt),std(qt)*100/avg(qt) as valCv from decode,tuteur
       where user=usr group by user order by valCv desc ;

# -- fin de p10.msql

Résultats  du Programme 10 :

--------------
select decode.nom,tuteur.qt from decode,tuteur
       where tuteur.user=decode.usr
--------------

nom                 qt
Gilles Hunault      1
Gérard Manvussa     2
Hubert Hubert       3
Gilles Hunault      4
Gilles Hunault      5
Andrée Tarkowsky    3
Gilles Hunault      5
Gérard Manvussa     8
Hubert Hubert       7
Gilles Hunault      2
Gilles Hunault      5
Andrée Tarkowsky    9

--------------
select decode.nom,count(tuteur.qt) from decode,tuteur
       where tuteur.user=decode.usr group by decode.nom
--------------

nom                  count(tuteur.qt)
Andrée Tarkowsky     2
Gérard Manvussa      2
```

```
Gilles Hunault       6
Hubert Hubert        2

--------------
select decode.nom,min(tuteur.qt) from decode,tuteur
       where tuteur.user=decode.usr group by decode.nom
--------------

nom                  min(tuteur.qt)
Andrée Tarkowsky     3
Gérard Manvussa      2
Gilles Hunault       1
Hubert Hubert        3

--------------
select min(tuteur.qt) as minQt, decode.nom from decode,tuteur
where tuteur.user=decode.usr group by decode.nom order by minQt
--------------

minQt nom
1     Gilles Hunault
2     Gérard Manvussa
3     Hubert Hubert
3     Andrée Tarkowsky

--------------
select nom,avg(qt) as moyQt from decode,tuteur
       where user=usr group by user order by moyQt
--------------

nom                  moyQt
Gilles Hunault       3.6667
Gérard Manvussa      5.0000
Hubert Hubert        5.0000
Andrée Tarkowsky     6.0000

--------------
select left(concat(nom,"................"),24),
       avg(qt*15.0) as moyQt from decode,tuteur
       where user=usr group by user order by moyQt
--------------
```

```
left(concat(nom,"................"),24)     moyQt
Gilles Hunault..........     55.00000
Gérard Manvussa.........     75.00000
Hubert Hubert...........     75.00000
Andrée Tarkowsky........     90.00000

--------------
select left(concat(upper(nom),"           ..........."),35),
       avg(qt) as moyQt,std(qt) from decode,tuteur
       where user=usr group by user order by moyQt desc
--------------
```

Source du Programme 11 :

```
# -- p11.msql

# on utilise la table diremail de la base test

use test ;

# la structure de diremail est

#   mysql> describe diremail ;
#   +--------+----------+------+-----+---------+-------+
#   | Field  | Type     | Null | Key | Default | Extra |
#   +--------+----------+------+-----+---------+-------+
#   | jourR  | int(2)   | YES  |     | NULL    |       |
#   | moisR  | int(2)   | YES  |     | NULL    |       |
#   | anneR  | int(4)   | YES  |     | NULL    |       |
#   | nomvu  | char(80) | YES  |     | NULL    |       |
#   | pnmvu  | char(30) | YES  |     | NULL    |       |
#   | email  | char(80) |      | PRI |         |       |
#   | source | char(20) | YES  |     | NULL    |       |
#   +--------+----------+------+-----+---------+-------+
#   7 rows in set (0.00 sec)
#

# nombre d'enregistrements
```

```
select count(email) from diremail ;

# les années

select distinct(anneR) from diremail order by anneR ;

# les mois par année

# INVALIDE :
# select distinct(moisR),distinct(anneR) from diremail order by anneR ;
select distinct(moisR),anneR from diremail order by anneR desc,moisR desc;

# le détail des jours

select distinct(jourR),moisR,anneR from diremail
       order by anneR desc,moisR desc, jourR desc ;

# un peu de comptage

select anneR,moisR from diremail group by moisR,anneR
       order by anneR desc,moisR desc ;

select anneR,moisR,count(email) from diremail group by moisR,anneR
       order by anneR desc,moisR desc ;

select anneR,moisR,jourR,count(email) from diremail group by jourR,moisR,anneR
       order by anneR desc,moisR desc,jourR desc ;

# comptons les emails par abonnement

select count(*) from diremail ;
select count(*) from diremail where email like "%wanadoo.fr" ;
select count(*) from diremail where email like "%yahoo.fr" ;
select count(*) from diremail
       where not(email like "%yahoo.fr"  or email like "%wanadoo.fr") ;

# puis cherchons des noms avec like et regexp

select * from diremail where nomvu like   "aar%" ;
select * from diremail where nomvu like   "%caa" ;
select * from diremail where nomvu regexp "AA.*R" ;
```

```
# et enfin calculons des moyennes

create table TmpMoy ( an int(4), mois int(2), jour int(2), nb int ) ;

insert into TmpMoy
       select anneR,moisR,jourR,count(email)
       from diremail group by jourR,moisR,anneR
       order by anneR desc,moisR desc,jourR desc ;

select an,mois,count(nb) from TmpMoy
       group by mois,an order by an desc,mois desc  ;
select an,mois,avg(nb)   from TmpMoy
       group by mois,an order by an desc,mois desc  ;

drop table TmpMoy ;

# -- fin de p11.msql
```

Résultats du Programme 11 :

```
--------------
select count(email) from diremail
--------------

count(email)
5476

--------------
select distinct(anneR) from diremail order by anneR
--------------

anneR
1999
2000

--------------
select distinct(moisR),anneR from diremail order by anneR desc,moisR desc
--------------

moisR  anneR
```

```
10     2000
 9     2000
10     1999
 9     1999
 8     1999

--------------
select distinct(jourR),moisR,anneR from diremail
       order by anneR desc,moisR desc, jourR desc
--------------

jourR  moisR  anneR
21     10     2000
21      9     2000
21     10     1999
20     10     1999
19     10     1999
18     10     1999
15     10     1999
28      9     1999
19      9     1999
31      8     1999
20      8     1999

--------------
select anneR,moisR from diremail group by moisR,anneR
       order by anneR desc,moisR desc
--------------

anneR   moisR
2000     10
2000      9
1999     10
1999      9
1999      8

--------------
select anneR,moisR,count(email) from diremail
       group by moisR,anneR order by anneR desc,moisR desc
--------------
```

```
anneR    moisR  count(email)
2000     10     1
2000      9     1
1999     10     4122
1999      9     25
1999      8     1327

--------------
select anneR,moisR,jourR,count(email) from diremail
       group by jourR,moisR,anneR order by anneR desc,moisR desc,jourR desc
--------------

anneR    moisR  jourR  count(email)
2000     10     21     1
2000      9     21     1
1999     10     21     1400
1999     10     20     1554
1999     10     19     592
1999     10     18     25
1999     10     15     551
1999      9     28     13
1999      9     19     12
1999     8     31     1323
1999     8     20     4

--------------
select count(*) from diremail
--------------

count(*)
5476

--------------
select count(*) from diremail where email like "%wanadoo.fr"
--------------

count(*)
3080

--------------
select count(*) from diremail where email like "%yahoo.fr"
```

```
--------------

count(*)
1730



--------------
select count(*) from diremail
       where not(email like "%yahoo.fr"  or email like "%wanadoo.fr")
--------------

count(*)
666

--------------
select * from diremail where nomvu like   "aar%"
--------------

jourR  moisR  anneR    nomvu     pnmvu     email                  source
20     10     1999     AARID     Najim      n_aarid@yahoo.fr      YahooPS
20     10     1999     AARIF     Bnimellal  bnimellal@yahoo.fr    YahooPS

--------------
select * from diremail where nomvu like   "%caa"
--------------

jourR  moisR  anneR    nomvu           pnmvu       email                source
15     10     1999     MINCHE AGNCAA   Christian   agncaa@wanadoo.fr   SwitchBoardP

--------------
select * from diremail where nomvu regexp "AA.*R"
--------------

jourR  moisR  anneR    nomvu      pnmvu       email                        source
20     10     1999     AZZAANKARA Hicham      copycat1980@yahoo.fr         YahooPS
20     10     1999     AARID      Najim       n_aarid@yahoo.fr             YahooPS
20     10     1999     AARIF      Bnimellal   bnimellal@yahoo.fr           EbayAdr
21     10     1999     AJAAFAR    Mokhtar     image.service@wanadoo.fr     YahooPS

--------------
```

```
create table TmpMoy ( an int(4), mois int(2), jour int(2), nb int )
--------------
insert into TmpMoy
       select anneR,moisR,jourR,count(email)
       from diremail group by jourR,moisR,anneR
       order by anneR desc,moisR desc,jourR desc
--------------
select an,mois,count(nb) from TmpMoy
       group by mois,an order by an desc,mois desc
--------------

an      mois    count(nb)
2000     10     1
2000      9     1
1999     10     5
1999      9     2
1999      8     2

--------------
select an,mois,avg(nb) from TmpMoy
       group by mois,an order by an desc,mois desc
--------------

an     mois     avg(nb)
2000     10       1.0000
2000      9       1.0000
1999     10     824.4000
1999      9      12.5000
1999      8     663.5000

--------------
drop table TmpMoy
--------------
```

Source du Programme 12 :

```
# -- p12.msql
```

```
# on rajoute le prix a payer dans le champ prix
# avec un meme prix unitaire de 325000.17 F

use    test ;
alter  table  tuteur add prix float ;
update tuteur set prix=325000.17*qt ;
select * from tuteur ;

# on reprend en prixAmi avec un prix unitaire de 100 F
# et une remise pour GH

alter  table  tuteur add prixAmi float ;
update tuteur set prixAmi=if(user="GH",90*qt,100*qt) ;
select * from tuteur ;

# -- fin de p12.msql
```

Résultats du Programme 12 :

```
--------------
alter  table  tuteur add prix float
--------------
update tuteur set prix=325000.17*qt
--------------



select * from tuteur
--------------

qt    user   prix
1     GH      325000.16
2     GG      650000.31
3     HU      975000.50
4     GH     1300000.62
5     GH     1625000.88
3     AA      975000.50
5     GH     1625000.88
8     GG     2600001.25
7     HU     2275001.25
2     GH      650000.31
```

```
5     GH     1625000.88
9     AA     2925001.50

--------------
alter  table  tuteur add prixAmi float
--------------


update tuteur set prixAmi=if(user="GH",90*qt,100*qt)
--------------


select * from tuteur
--------------

qt    user    prix          prixAmi

1     GH      325000.16      90.00
2     GG      650000.31     200.00
3     HU      975000.50     300.00
4     GH     1300000.62     360.00
5     GH     1625000.88     450.00
3     AA      975000.50     300.00
5     GH     1625000.88     450.00
8     GG     2600001.25     800.00
7     HU     2275001.25     700.00
2     GH      650000.31     180.00
5     GH     1625000.88     450.00
9     AA     2925001.50     900.00
```

5. Interaction Web : PHP et MYSQL

Nous terminons ce petit tuteur par quelques programmes en PHP qui interfacent MYSQL pour le WeB. Notre tuteur PHP est disponible sur le web à l'adresse

`http://www.univ-angers.fr/info/pub/gh/`

```
Exemple 1 de programme PHP3
----------------------------

    <html>
      <head>
         <title> (gH) : un premier exemple en PHP</title>
      </head>
      <body>
         Bonjour,
         <? echo " tout le monde " ;?>
      </body>
    </html>


Exemple 2 de programme PHP3
----------------------------

<? # comptage de fichiers dans des eépertoires

function dirstat($nomdir) {

$d  = dir($nomdir) ;
$nbf = 0 ;
 while ($entry=$d->read()) {
    $nbf = $nbf +1 ;
 } ; # fin tant que
 $d->close() ;
 return $nbf-2 ;

} # fin fonction dirstat

######### programme principal

$dirbase = "/home/gh/Rch/Directe/" ;

echo "<font size=+5>Statistiques fichiers,
répertoire $dirbase</h1> " ;


$cdir = $dirbase ;
$nbf = dirstat($cdir) ;
```

```
echo   "<font size=+5>racine : $nbf<p>" ;

$ldir = "Adresses/" ;
$cdir = $dirbase.$ldir ;
$nbf  = dirstat($cdir) ;
echo   "<font size=+5>$ldir : $nbf<p>" ;

$ldir = "Archivadr/" ;
$cdir = $dirbase.$ldir ;
$nbf  = dirstat($cdir) ;
echo   "<font size=+5>$ldir : $nbf" ;

?>

Exemple 3 de programme simple PHP3 avec formulaire
---------------------------------------------------

 FORMULAIRE

 <html>
      <head>
         <title> (gH) : programme BONJOUR en PHP, partie 1 :
                        formulaire de saisie du prénom </title>
      </head>
      <body>
      <form action="bonjour.php3">
      <font color="red">Bonjour</font>.  
      Quel est votre nom ? <input type="text" name="prenom">
      <input type="submit" value=" envoi ! ">
      </form>
      </body>
 </html>

 SCRIPT PHP DE REPONSE

 <html>

      <head>
         <title> (gH) : programme BONJOUR en PHP,
                        partie 2 : réponse </title>
      </head>
```

```
      <body>
      <?
         if ($prenom=="") { $nom = " bel inconnu " ; }
                     else { $nom = $prenom  ; } ;
         $ladate  = date("d/m/Y") ;
         $lheure  = date("h:i") ;
         $nomMaj  = strtoupper($nom) ;
         echo " Le $ladate à $lheure, au revoir, $nomMaj." ;
      ?>
      </body>

 </html>
```

```
Exemple 4 de programme simple PHP3 qui utilise MYSQL
----------------------------------------------------

<html><? # connection et comptage d'enregistrements

mysql_connect("localhost","root","");
mysql_select_db("test");

$res  = mysql_query("select count(email) from afone") ;
$ligr = mysql_fetch_array($res) ;
$nbst = $ligr["count(email)"] ;

echo " il y a donc $nbst enregistrement(s). " ;
?></html>
```

```
Exemple 5 de programme sophistiqué qui utilise MYSQL
---------------------------------------------

<? # nommons les mois en français

function nomMois($im) {

if       ($im==1) { $nomlong  = "Janvier  " ;
} elseif ($im==2) { $nomlong  = "Février  " ;
} elseif ($im==3) { $nomlong  = "Mars     " ;
} elseif ($im==4) { $nomlong  = "Avril    " ;
```

```
} elseif ($im==5) { $nomlong  = "Mai      " ;
} elseif ($im==6) { $nomlong  = "Juin     " ;
} elseif ($im==7) { $nomlong  = "Juillet  " ;
} elseif ($im==8) { $nomlong  = "Aout     " ;
} elseif ($im==9) { $nomlong  = "Septembre" ;
} elseif ($im==10){ $nomlong  = "Octobre  " ;
} elseif ($im==11){ $nomlong  = "Novembre " ;
} elseif ($im==12){ $nomlong  = "Décembre " ; }

return $nomlong ;

} ; # fin fonction nomMois


# connection SQL et ouverture de la base

mysql_connect("localhost","root","");

mysql_select_db("test");

# comptons les emails

$res  = mysql_query("select count(email) from afone") ;
$ligr = mysql_fetch_array($res) ;
$nbst = $ligr["count(email)"] ;

$dthr = date(" d/m/Y à H:i ") ;
echo " <font size=6>    " ;
echo " le $dthr " ;
echo " <br><font size=6>    " ;
echo " on a : " ;
echo " <font size=+4 color=000088><b>$nbst </b></font>adresses " ;

# détaillons mois par mois et proposons
# d'ailler voir jour par jour pour un mois donné

# comptons les années

$res  = mysql_query("select distinct anneR from afone order by anneR ") ;
$nbans  = 0 ;
while ($ligr=mysql_fetch_array($res)) {
```

```
  $nbans++ ;   $ans[$nbans] = $ligr["anneR"] ;
} ; # fin de tant que

echo "<h1>choisissez le mois : </h1>         " ;

# comptons les mois

$res  = mysql_query("select distinct moisR from afone order by moisR ") ;
$nbmois = 0 ;
while ($ligr=mysql_fetch_array($res)) {
  $nbmois++ ;   $mois[$nbmois] = $ligr["moisR"] ;
} ; # fin de tant que



# passage en revue des années et de mois
# on crée une ancre <a href=...> si pour le mois
# et l'année courante il y a un nombre non nul
# d'adresses


echo "<table> " ;
$ian = $nbans ;
while ($ian>=1) {
   $van = $ans[$ian] ;

   $imois = $nbmois ;
   $jmois = 0 ;
   while ($imois>=1) {
     $vmois = $mois[$imois] ;
     $res   = mysql_query("select count(email) from afone
                     where (moisR=".$vmois." and anneR=".$van.")") ;
     $ligr  = mysql_fetch_array($res) ;
     $nbpa  = $ligr["count(email)"] ;
     if ($nbpa>0) {
        $jmois++ ;
        $nbpa  = sprintf("%3d",$nbpa) ;
        $pct   = sprintf("%3d",round(100.0*$nbpa/$nbst)) ;
        $nmois = nomMois($vmois) ;
        echo   "<tr><td><font size=+3>  " ;
        if ($jmois==1) { echo $van ; } ;
```

```
        echo   "  </td><td><font size=+3>" ;
        $choix = $van."*".$vmois."*" ;
        echo   "<b><a href=statmois.php3?choix=$choix>$nmois</a></td>" ;
        echo   "<td align=right><font size=+4 color=880000>
                              $nbpa</td>" ;
        echo   "<td align=right><font size=+4 color=008800>
                              $pct %</td></tr>" ;
     } ; # finsi sur $nbpa>0
     $imois-- ;
   } ; # fin de tant que sur $imois

  $ian-- ;
} ; # fin de tant que sur $ian
echo "</table> " ;

?>
```