

Correction de l'exercice complémentaire

- 1- Les entités
- 2- Les clés
- 3- Les requêtes

1 Les entités

AVION = {av, type, capacite }
PILOTE = {pl, nom, adresse, sal}
VOL = {vol, vpl, vav, ville_dep, ville_arr }

Relations entre ces entités

2 Les clés

Les clés principales (**PRIMARY KEY**) de chaque table : tri implicite => le SGBDR créera automatiquement l'index :

AVION = {av, type, capacite}
PILOTE = {pl, nom, adresse, sal}
VOL = {vol, vpl, vav, ville_dep, ville_arr}

Les clés étrangères : (**REFERENCE**)
VOL = {vol, *vpl*, *vav*, ville_dep, ville_arr}

On créera des index sur des champs qui seront utilisés pour des jointures (rapprochement entre deux tables) : pourront être créées ultérieurement.

3 Les requêtes de l'exercice

```
D:\EasyPHP1-7\mysql\bin>mysql -u root
```

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
```

```
Your MySQL connection id is 20 to server version: 4.0.15-max-debug
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
```

```
mysql> use mabase
```

```
Database changed
```

```
mysql> create table pilote (  
-> pl int(3) not null auto_increment,  
-> nom varchar(25),  
-> adresse varchar(10),
```

```
-> sal int(5),
-> primary key (pl)
-> );
```

Query OK, 0 rows affected (0.05 sec)

```
mysql> create table avion (
-> av int(3) not null auto_increment,
-> type varchar(10),
-> cap int(3),
-> primary key (av)
-> );
```

```
mysql> create table vol (
-> vol int(3) not null auto_increment,
-> vpl int(3) not null,
-> vav int(3) not null,
-> villedep varchar(10),
-> villearr varchar(10),
-> primary key (vol)
-> );
```

```
mysql> insert into pilote
-> values (1,"Miranda","Sophia",1800);
```

```
mysql> show tables;
```

```
+-----+
| Tables_in_mabase |
+-----+
| avion             |
| pilote            |
| vol               |
+-----+
```

3 rows in set (0.01 sec)

R1 : select pl,nom,adresse,sal from pilote

R2 :

```
mysql> select pl "Numéro", nom "Nom pilote", sal "Salaire Mensuel"
-> from pilote;
```

```
+-----+-----+-----+
| Numéro | Nom pilote | Salaire Mensuel |
+-----+-----+-----+
| 1 | Miranda | 1800 |
| 2 | Armstrong | 1200 |
| 3 | Tintin | 2400 |
| 4 | Gagarine | 2100 |
| 5 | Baudry | 2500 |
| 6 | Berunard | 2500 |
| 7 | Ruskoi | 2200 |
| 8 | Math | 2200 |
| 9 | Yen | 2200 |
```

```
| 10 | Mopolo | 2600 |
| 11 | Chretien | 1600 |
| 12 | SaintEx | 2600 |
| 13 | Vernes | 24500 |
| 14 | Tournesol | 2450 |
```

```
+-----+-----+-----+
```

14 rows in set (0.01 sec)

R3 :réécrire la requete avec un alias de la table pilote : p

**select p.pl “No pilote”, p.nom “Nom pilote”, p.sal “Salaire”, p.adresse “Adresse
From pilote p;**

```
mysql> select nom "Nom", sal "Salaire Mensuel", pl "Numéro"
-> from pilote;
```

```
+-----+-----+-----+
| Nom      | Salaire Mensuel | Numéro |
```

```
+-----+-----+-----+
```

```
| Miranda | 1800 | 1 |
| Armstrong | 1200 | 2 |
| Tintin | 2400 | 3 |
| Gagarine | 2100 | 4 |
| Baudry | 2500 | 5 |
| Berunard | 2500 | 6 |
| Ruskoi | 2200 | 7 |
| Math | 2200 | 8 |
| Yen | 2200 | 9 |
| Mopolo | 2600 | 10 |
| Chretien | 1600 | 11 |
| SaintEx | 2600 | 12 |
| Vernes | 24500 | 13 |
| Tournesol | 2450 | 14 |
```

```
+-----+-----+-----+
```

14 rows in set (0.00 sec)

```
mysql> select * from pilote;
```

```
+---+-----+-----+-----+
```

```
| pl | nom      | adresse | sal |
```

```
+---+-----+-----+-----+
```

```
| 1 | Miranda | Sophia | 1800 |
| 2 | Armstrong | Wapakoneta | 1200 |
| 3 | Tintin | Bordeaux | 2400 |
| 4 | Gagarine | Klouchino | 2100 |
| 5 | Baudry | Toulouse | 2500 |
| 6 | Berunard | Milton | 2500 |
| 7 | Ruskoi | Moscou | 2200 |
| 8 | Math | Paris | 2200 |
| 9 | Yen | Munich | 2200 |
| 10 | Mopolo | Nice | 2600 |
| 11 | Chretien | | 1600 |
```

```
| 12 | SaintEx | Lyon | 2600 |
| 13 | Vernes | Paris | 24500 |
| 14 | Tournesol | Bruxelles | 2450 |
+-----+-----+-----+
```

14 rows in set (0.00 sec)

R4 calculer le salaire annuel des pilotes, le lister pour chaque pilote
mysql> **select pl, nom, sal*12 "Salaire annuel"**
 -> from pilote;

```
+-----+-----+-----+
| pl | nom | Salaire annuel |
+-----+-----+-----+
| 1 | Miranda | 21600 |
| 2 | Armstrong | 14400 |
| 3 | Tintin | 28800 |
| 4 | Gagarine | 25200 |
| 5 | Baudry | 30000 |
| 6 | Berunard | 30000 |
| 7 | Ruskoi | 26400 |
| 8 | Math | 26400 |
| 9 | Yen | 26400 |
| 10 | Mopolo | 31200 |
| 11 | Chretien | 19200 |
| 12 | SaintEx | 31200 |
| 13 | Vernes | 294000 |
| 14 | Tournesol | 29400 |
+-----+-----+-----+
```

14 rows in set (0.01 sec)

R5 calculer la somme des salaires des pilotes
mysql> **select sum(sal) from pilote;**

```
+-----+
| sum(sal) |
+-----+
| 52850 |
+-----+
```

1 row in set (0.01 sec)

R6 : donner tous les types d'avion de la compagnie
mysql> **select distinct type from avion;**

```
+-----+
| type |
+-----+
| A300 |
| A320 |
| B707 |
| B727 |
| CARAVELLE |
| CONCORDE |
+-----+
```

6 rows in set (0.03 sec)

R7 : donner les numeros d'avions et type d'avion de capacité sup à 300

```
mysql> select av, type
      -> from avion
      -> where cap >300;
```

```
+-----+
| av  | type |
+-----+
| 4   | B727 |
+-----+
1 row in set (0.00 sec)
```

R8 : donner les noms de pilotes habitants Paris ou Nice

```
Select nom,adresse
```

```
From pilote
```

```
Where adresse in ("Paris","Nice") ;
```

R9 : quels pilotes ont un 't' dans leur nom en 3eme position?

```
mysql> select nom
      -> from pilote
      -> where nom LIKE '___t%';
```

```
+-----+
| nom  |
+-----+
| Tintin |
+-----+
1 row in set (0.00 sec)
```

R10 :Quels sont les vols au depart de Nice, Paris ou Bordeaux ?

```
mysql> select vol
      -> from vol
      -> where villedep IN ("Nice","Paris","Bordeaux");
```

```
+-----+
| vol |
+-----+
| 1 |
| 2 |
| 3 |
| 4 |
+-----+
4 rows in set (0.00 sec)
```

R11 : donner les noms des pilotes ayant un a et un e dans leur nom

```
mysql> select nom
      -> from pilote
      -> where nom LIKE "%a%"
      -> and nom LIKE "%e%";
```

```
+-----+
| nom  |
+-----+
| Gagarine |
| Berunard |
| SaintEx  |
+-----+
3 rows in set (0.00 sec)
```

```
mysql> select nom, adresse
```

-> **from pilote**
 -> **where adresse is not null;**

```
+-----+-----+
| nom    | adresse |
+-----+-----+
| Miranda | Sophia  |
| Armstrong | Wapakoneta |
| Tintin  | Bordeaux |
| Gagarine | Klouchino |
| Baudry  | Toulouse |
| Berunard | Milton  |
| Ruskoi  | Moscou  |
| Math    | Paris   |
| Yen     | Munich  |
| Mopolo  | Nice    |
| Chretien |         |
| SaintEx | Lyon    |
| Vernes  | Paris   |
| Tournesol | Bruxelles |
+-----+-----+
14 rows in set (0.01 sec)
```

R13 : donner le salaire le + eleve et l'afficher sous la forme : <valeur du salaire max> « salaire le plus fort»

mysql> **select max(sal) "salaire le plus fort"**
 -> **from pilote;**

```
+-----+
| salaire le plus fort |
+-----+
|          24500      |
+-----+
1 row in set (0.00 sec)
```

R14 : quels sont les noms, adresse et salaire des pilotes triés par ordre croissant sur l'adresse et pour une meme adresse ordre decroissant de salaire

Select nom, adresse, sal
From pilote
Order by adresse ASC, sal DESC ;

R15 : donner les paires de pilotes habitant la meme ville (auto-jointure ! + alias) (ne pas avoir le meme nom dans le couple)

Select p1.nom,p2.nom,p1.adresse
From pilote p1, pilote p2
Where p1.adresse=p2.adresse
And p1.nom != p2.nom
And p1.nom > p2.nom ;

R16 : donner tous les noms de pilote qui ont un nom d'avion

mysql> **select distinct nom**
 -> **from pilote, avion**
 -> **where UPPER(nom)=UPPER(type);**

```
+-----+
| nom    |
+-----+
```

```
+-----+
| Concorde |
+-----+
```

1 row in set (0.00 sec)

R17 : Donner les noms des pilotes qui conduisent un A300 ou un B727

```
mysql> select nom,type from pilote p, avion a, vol v
-> where v.vpl=p.pl and v.vav=a.av and type="B727"
-> UNION
-> select nom,type from pilote p, avion a, vol v
-> where v.vpl=p.pl and v.vav=a.av and type="A300";
```

```
+-----+-----+
| nom    | type |
+-----+-----+
| Gagarine | B727 |
| Armstrong | A300 |
+-----+-----+
```

2 rows in set (0.03 sec)

OU

```
mysql> select distinct nom, type
-> from pilote p, avion a, vol v
-> where a.av=v.vav
-> and p.pl=v.vpl
-> and type IN ("B727","A300");
```

```
+-----+-----+
| nom    | type |
+-----+-----+
| Armstrong | A300 |
| Gagarine | B727 |
+-----+-----+
```

2 rows in set (0.00 sec)

R18 : jointures

Select nom, vol from pilote, vol
Where pilote.pl = vol.vpl(+);

NB : on retient tous les enregistrements de la 1ere table, select * from table1 LEFT JOIN table2 on (colonne) where table2.colonne=5;

Équivaut à écrire :

Select * from table1, table2

Where table2.colonne=5

And table2.colonne=table1.colonne;

Select nom, vol
From pilote left join vol
On pilote.pl = vol.vpl;

R19 :Quels pilotes parmi les no 1 à 4 et 13 assurent au moins 2 vols

```
mysql> select vpl,count(*) "nombre de vols"
```

```
-> from vol
```

```
-> where vpl in (1,2,3,4,13)
```

```
-> group by vpl
```

```
-> having count(*) >= 2;
```

```
+-----+-----+
| vpl | nombre de vols |
+-----+-----+
| 1 | 2 |
+-----+-----+
1 row in set (0.02 sec)
```

```
mysql> select * from vol;
```

```
+-----+-----+-----+-----+-----+
| vol | vpl | vav | villedep | villearr |
+-----+-----+-----+-----+-----+
| 1 | 1 | 2 | Paris | Bordeaux |
| 2 | 1 | 2 | Paris | Moscou |
| 3 | 2 | 1 | Nice | Moscou |
| 4 | 5 | 3 | Nice | Moscou |
| 5 | 4 | 4 | Lyon | Paris |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

R20 : Mettre à jour le salaire du pilote no 3

```
mysql> update pilote set sal=3000 where pl=3;
```

Query OK, 1 row affected (0.00 sec)

Enregistrements correspondants: 1 Modifiés: 1 Warnings: 0

R21 supprimer le pilote no 3

```
Delete from pilote where pl=3 ;
```

R22 : Ajouter colonne age a la table pilote

```
mysql> alter table pilote
```

```
-> ADD age int(2);
```

Query OK, 15 rows affected (0.10 sec)

Enregistrements: 15 Doublons: 0 Avertissements: 0