



Master Internet des Objets et Systèmes Mobiles
2021-2022

Module: S.I et Bases de données

Rapport des Travaux Pratiques
Série n° 1

Réalisé par :

ZAROUAL Mohammed

Encadré par :

Pr. EL AKKAD Nabil

Les relations utilisées dans les six premiers exercices sont :

- La relation etudiant (id, nom, prénom, age, adresse) :

```
[mysql]> select * from etudiant;
+----+-----+-----+-----+-----+
| id | nom  | prenom | age   | adresse | date_naissance | niveau |
+----+-----+-----+-----+-----+
| 1  | nom1 | prenom1 | 20    | adresse1 | 2001-01-18    | M1    |
| 2  | nom2 | prenom2 | 30    | adresse2 | 1991-04-17    | M2    |
| 3  | nom3 | prenom3 | 25    | adresse3 | 1996-08-19    | M1    |
| 4  | nom4 | Ahmed   | 27    | adresse4 | 1994-03-01    | M2    |
+----+-----+-----+-----+-----+
1 rows in set (0.00 sec)
```

- La relation employé (id, nom, prénom, salaire, adresse, fonction, date_embauche, commission) :

```
[mysql]> select * from employe;
+----+-----+-----+-----+-----+-----+-----+
| id | nom  | prenom | salaire | adresse | fonction | date_embauche | commission |
+----+-----+-----+-----+-----+-----+-----+
| 1  | nom1 | prenom1 | 3000   | adresse1 | professor | 2001-07-18   | 300   |
| 2  | nom2 | prenom2 | 7000   | adresse2 | ingenieur | 2002-09-18   | 330   |
| 3  | nom3 | prenom3 | 6000   | adresse2 | manager   | 2000-09-28   | 240   |
| 4  | nom4 | prenom4 | 5500   | adresse4 | directeur | 2003-03-05   | 500   |
| 5  | nom5 | prenom5 | 8500   | adresse5 | enseignant | 2001-03-11   | 350   |
| 6  | nom6 | prenom6 | 4300   | adresse6 | mechanien | 2004-03-14   | 450   |
+----+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

Exercice 1 (Fonction de manipulation de caractères):

1. Afficher l'identifiant, le nom et le prénom des étudiants dont le prénom est : Ahmed.

```
select id,nom,prenom from etudiant where prenom="Ahmed";
```

```
[mysql]> select id,nom,prenom from etudiant where prenom="Ahmed";
+----+-----+
| id | nom  | prenom |
+----+-----+
| 4  | nom4 | Ahmed  |
+----+-----+
```

2. On supposera qu'aucun enregistrement n'a pas été affiché lors de l'exécution à cause de la casse (lettre en minuscule ou en majuscule). Réécrire la requête pour remédier au problème.

```
select id,nom,prenom from etudiant where upper(prenom)=upper("AHmEd");
```

```
[mysql]> select id,nom,prenom from etudiant where upper(prenom)=upper("AHmEd");
+----+-----+
| id | nom  | prenom |
+----+-----+
| 4  | nom4 | Ahmed  |
+----+-----+
```

3. Afficher le nom et le prénom concaténé de tous les étudiants sous le nom “ Nom de l’étudiant ”.

```
select concat(nom, ' ', prenom) as "nom et prenom" from etudiant;
```

```
[mysql]> select concat(nom, ' ', prenom) as "nom et prenom" from etudiant;
+-----+
| nom et prenom |
+-----+
| nom1 prenom1 |
| nom2 prenom2 |
| nom3 prenom3 |
| nom4 Ahmed   |
+-----+
```

4. Afficher l’identifiant des étudiants ainsi que la taille de leurs noms (nombre de lettres).

```
select id,length(nom) from etudiant;
```

```
[mysql]> select id,length(nom) from etudiant;
+----+-----+
| id | length(nom) |
+----+-----+
| 1  |        4 |
| 2  |        4 |
| 3  |        4 |
| 4  |        4 |
+----+-----+
```

5. Afficher les étudiants dont le prénom contient la lettre ‘a’.

```
select * from etudiant where prenom like "%a%"
```

```
[mysql]> select * from etudiant where prenom like "%a%";
+----+-----+-----+-----+
| id | nom  | prenom | age  | adresse |
+----+-----+-----+-----+
| 4  | nom4 | Ahmed  | 27   | adresse4 |
+----+-----+-----+-----+
```

Exercice 2:

1. Afficher la rémunération de salaire pour tout employé dont la fonction est ‘ingénieur ‘ après avoir divisé leur salaire sur 5000. (Rem=salaire modulo 5000)

```
select id,nom,salaire%5000 from employe where fonction="ingenieur";
```

```
[mysql]> select id,nom,salaire%5000 from employe where fonction="ingenieur";
+----+-----+-----+
| id  | nom  | salaire%5000 |
+----+-----+-----+
| 2   | nom2 |      2000 |
+----+-----+-----+
```

2. Afficher tous les employés embauchés avant le '01/02/03'

```
select * from employe where date_embauche<"2003-02-01";
```

```
Error 1315 (HY000): incorrect date value: '2003-02-01'  
[mysql] > select * from employe where date_embauche<"2003-02-01";  
+----+----+----+----+----+----+----+  
| id | nom | prenom | salaire | adresse | fonction | date_embauche | commission |  
+----+----+----+----+----+----+----+  
| 1 | nom1 | prenom1 | 3000 | adresse1 | professor | 2001-07-18 | 300 |  
| 2 | nom2 | prenom2 | 7000 | adresse2 | ingenieur | 2002-09-18 | 330 |  
| 3 | nom3 | prenom3 | 6000 | adresse2 | manager | 2000-09-28 | 240 |  
| 5 | nom5 | prenom5 | 8500 | adresse5 | enseignant | 2001-03-11 | 350 |  
+----+----+----+----+----+----+----+
```

Exercice 3 (fonctions de date):

1. Afficher le nom et le nombre (entier) de semaines travaillés par tout employé.

```
select id,nom,FLOOR(DATEDIFF(SYSDATE(),date_embauche)/7) as "semaines travaillé" from employe;
```

```
[mysql] > select id,nom,FLOOR(DATEDIFF(SYSDATE(),date_embauche)/7) from employe;  
+----+----+  
| id | nom | FLOOR(DATEDIFF(SYSDATE(),date_embauche)/7) |  
+----+----+  
| 1 | nom1 | 1063 |  
| 2 | nom2 | 1002 |  
| 3 | nom3 | 1105 |  
| 4 | nom4 | 978 |  
| 5 | nom5 | 1081 |  
| 6 | nom6 | 924 |  
+----+----+
```

2. Afficher le nom et le nombre de mois travaillés par tout employé.

```
select nom,DATEDIFF(SYSDATE(),date_embauche)/30 as "semaines travaillé" from employe;
```

```
[mysql] > select id,nom,FLOOR(DATEDIFF(SYSDATE(),date_embauche)/30) from employe;  
+----+----+  
| id | nom | FLOOR(DATEDIFF(SYSDATE(),date_embauche)/30) |  
+----+----+  
| 1 | nom1 | 248 |  
| 2 | nom2 | 233 |  
| 3 | nom3 | 257 |  
| 4 | nom4 | 228 |  
| 5 | nom5 | 252 |  
| 6 | nom6 | 215 |  
+----+----+
```

3. Afficher le nom et la date du jour vendredi après que l'employé soit embauché.

```
select nom, prenom, next_day(date_embauche, "friday") as "premier vendredi" from employe;
```

4. Afficher le nom et la date de titularisation de tout employé (la titularisation se fait après 6 mois).

SQL : select id, nom, prenom, add_months(date_embauche,6) from employe;

MySQL : select id, nom, prenom, TIMESTAMPADD(month,6,date_embauche) from employe;

```
[mysql] 1502 (1200) PERIODIC triggers add_months does not exist
[mysql]> select id, nom, prenom, TIMESTAMPADD(month, 6, date_embauche) from employe;
+----+----+-----+
| id | nom | prenom | TIMESTAMPADD(month, 6, date_embauche) |
+----+----+-----+
| 1 | nom1 | prenom1 | 2002-01-18 |
| 2 | nom2 | prenom2 | 2003-03-18 |
| 3 | nom3 | prenom3 | 2001-03-28 |
| 4 | nom4 | prenom4 | 2003-09-05 |
| 5 | nom5 | prenom5 | 2001-09-11 |
| 6 | nom6 | prenom6 | 2004-09-14 |
+----+----+-----+
```

5. Afficher les employés ayant travaillés moins de 250 mois.

```
select id, nom, DATEDIFF(SYSDATE(), date_embauche)/30 from employe where
(DATEDIFF(SYSDATE(), date_embauche)/30)<250;
```

```
mysql> select id, nom, FLOOR(DATEDIFF(SYSDATE(), date_embauche)/30)
   from employe where (DATEDIFF(SYSDATE(), date_embauche)/30)<250;
+----+----+-----+
| id | nom | FLOOR(DATEDIFF(SYSDATE(), date_embauche)/30) |
+----+----+-----+
| 1 | nom1 | 248 |
| 2 | nom2 | 233 |
| 4 | nom4 | 228 |
| 6 | nom6 | 215 |
+----+----+-----+
```

6. Afficher le dernier jour du mois d'embauche de tous les techniciens de l'entreprise.

```
select id, nom, last_day(date_embauche) from employe where fonction="ingenieur";
```

```
[mysql]> select id, nom, last_day(date_embauche) from employe where fonction="ingenieur";
+----+----+-----+
| id | nom | last_day(date_embauche) |
+----+----+-----+
| 2 | nom2 | 2002-09-30 |
+----+----+-----+
```

Exercice 4 (fonctions de conversion):

1. Afficher la date d'embauche de tout employé sous la forme 'Monday 17 junary 2011'.

```
select DATE_FORMAT(date_embauche,"%W %d %M %Y") as "date d'embauche" from employe;  
en sql : select to_char(date_embauche,'day dd month yyyy');
```

```
[mysql]> select DATE_FORMAT(date_embauche,"%W %d %M %Y") as "date d'embauche" from employe;  
+-----+  
| date d'embauche |  
+-----+  
| Wednesday 18 July 2001 |  
| Wednesday 18 September 2002 |  
| Thursday 28 September 2000 |  
| Wednesday 05 March 2003 |  
| Sunday 11 March 2001 |  
| Sunday 14 March 2004 |  
+-----+
```

2. Afficher le salaire de toute secrétaire sous la forme : 6 chiffres avant la virgule, deux chiffres après la virgule sans oublier le symbole du dollar.

```
select concat(cast(round(salaire,2) as decimal(6,2))," $") from employe;
```

```
[mysql]> select concat(cast(round(salaire,2) as decimal(6,2))," $") from employe;  
+-----+  
| concat(cast(round(salaire,2) as decimal(6,2))," $") |  
+-----+  
| 3000.00 $ |  
| 7000.00 $ |  
| 6000.00 $ |  
| 5500.00 $ |  
| 8500.00 $ |  
| 4300.00 $ |  
+-----+
```

3. Afficher le nom et la date d'embauche de tout employé commençant le 24 mai 2000

```
select * from employe where DATE_FORMAT(date_embauche,"%Y %b  
%d")=DATE_FORMAT("2002-09-18", "%Y %b %d");
```

```
[mysql]> select * from employe where DATE_FORMAT(date_embauche,"%Y %b %d")=DATE_FORMAT("2002-09-18", "%Y %b %d");  
+-----+-----+-----+-----+-----+-----+-----+  
| id | nom | prenom | salaire | adresse | fonction | date_embauche | commission |  
+-----+-----+-----+-----+-----+-----+-----+  
| 2 | nom2 | prenom2 | 7000 | adresse2 | ingenieur | 2002-09-18 | 330 |  
+-----+-----+-----+-----+-----+-----+-----+
```

Exercice 5 (imbrication de fonctions):

1. Afficher les 3 premières lettres des noms des employés concaténés à '_MAR' en majuscule.

```
select concat(substr(prenom,1,3),'_MAR') from employe;
```

```
[mysql> select concat(substr(prenom,1,3),'_MAR') from employe;
+-----+
| concat(substr(prenom,1,3),'_MAR') |
+-----+
| pre_MAR
| pre_MAR
| pre_MAR
| pre_MAR
| pre_MAR
| pre_MAR
+-----+
6 rows in set (0.00 sec)
```

2. Afficher tous les employés en remplaçant toute valeur nulle de : fonction par 'ouvrier', adresse par 'SDF' et date d'embauche par '01-JAN-2011'.

Le tableau "employe" avant la modification (avec les valeurs NULL) :

```
[mysql> select * from employe;
+-----+-----+-----+-----+-----+-----+-----+-----+
| id | nom | prenom | salaire | adresse | fonction | date_embauche | commission |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | nom1 | prenom1 | 3000 | adresse1 | professor | NULL | 300 |
| 2 | nom2 | prenom2 | 7000 | NULL | ingenieur | NULL | 330 |
| 3 | nom3 | prenom3 | 6000 | NULL | NULL | 2000-09-28 | 240 |
| 4 | nom4 | prenom4 | 5500 | adresse4 | directeur | 2003-03-05 | 500 |
| 5 | nom5 | prenom5 | 8500 | adresse5 | NULL | 2001-03-11 | 350 |
| 6 | nom6 | prenom6 | 4300 | NULL | mechanien | 2004-03-14 | 450 |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

Apres la modification :

→ **SQL** : select id, nom, prenom, nvl(adresse,"SDF"), nvl(fonction,"ouvrier"), nvl (date_embauche, "2011-01-01") from employe;

MySQL : select id, nom, prenom, coalesce(adresse,"SDF"), coalesce(fonction,"ouvrier"), coalesce(date_embauche,"2011-01-01") from employe;

```
[mysql> select id,nom,prenom,coalesce(adresse,"SDF"),coalesce(fonction,"ouvrier"),coalesce(date_embauche,"2011-01-01") from employe;
+-----+-----+-----+-----+-----+
| id | nom | prenom | coalesce(adresse,"SDF") | coalesce(fonction,"ouvrier") | coalesce(date_embauche,"2011-01-01") |
+-----+-----+-----+-----+-----+
| 1 | nom1 | prenom1 | adresse1 | professeur | 2011-01-01 |
| 2 | nom2 | prenom2 | SDF | ingenieur | 2011-01-01 |
| 3 | nom3 | prenom3 | SDF | ouvrier | 2000-09-28 |
| 4 | nom4 | prenom4 | adresse4 | directeur | 2003-03-05 |
| 5 | nom5 | prenom5 | adresse5 | ouvrier | 2001-03-11 |
| 6 | nom6 | prenom6 | SDF | mechanien | 2004-03-14 |
+-----+-----+-----+-----+-----+
```

Exercice 6 (fonctions de restitution des valeurs nulles) :

1. Afficher le nom et le salaire annuel de tout employé (la colonne commission contient des valeurs ‘null’) ;

```
select nom,salaire,commission from employe where isnull(commission);
```

```
[mysql> select nom,salaire,commission from employe where isnull(commission);
+-----+-----+-----+
| nom   | salaire | commission |
+-----+-----+-----+
| nom1  |    3000 |      NULL |
| nom3  |    6000 |      NULL |
| nom6  |    4300 |      NULL |
+-----+-----+-----+
```

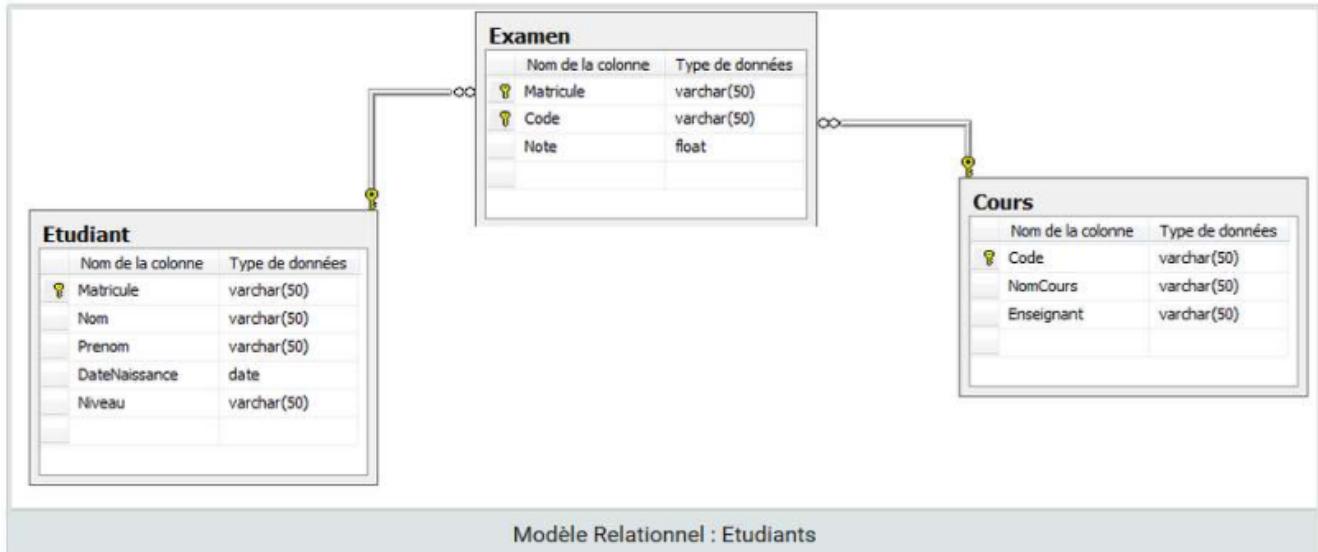
2. Afficher le revenu mensuel de tout employé, qui égale au salaire si la commission est nulle et le salaire + commission dans le cas contraire.

```
select id,nom,salaire+coalesce(commission,0) as "revenu mensuel" from employe;
```

```
[mysql> select id,nom,salaire+coalesce(commission,0) as "revenu mensuel" from employe;
+-----+-----+-----+
| id   | nom   | revenu mensuel |
+-----+-----+-----+
| 1    | nom1  |      3000 |
| 2    | nom2  |      7330 |
| 3    | nom3  |      6000 |
| 4    | nom4  |      6000 |
| 5    | nom5  |      8850 |
| 6    | nom6  |      4300 |
+-----+-----+-----+
```

Exercice7 :

Soit le modèle relationnel suivant :



Les valeurs des Tableaux :

```

mysql> select * from etudiant;
+----+-----+-----+-----+-----+-----+
| id | nom  | prenom | age  | adresse | date_naissance | niveau |
+----+-----+-----+-----+-----+-----+
| 1  | nom1 | prenom1 | 20   | adresse1 | 2001-01-18   | M1    |
| 2  | nom2 | prenom2 | 30   | adresse2 | 1991-04-17   | M3    |
| 3  | nom3 | prenom3 | 25   | adresse3 | 1996-08-19   | M1    |
| 4  | nom4 | Ahmed    | 27   | adresse4 | 1994-03-01   | M3    |
| 5  | nom5 | prenom5 | 23   | adresse5 | 1997-05-18   | M2    |
| 6  | nom6 | prenom6 | 25   | adresse5 | 1996-01-18   | M2    |
+----+-----+-----+-----+-----+-----+

```

```

mysql> select * from examen
-> ;
+----+-----+
| id | code | note |
+----+-----+
| 1  | c3  | 13  |
| 1  | c2  | 16  |
| 1  | c4  | 10  |
| 1  | c5  | 16  |
| 1  | c1  | 9   |
| 2  | c1  | 11  |
| 2  | c2  | 16  |
| 2  | c3  | 18  |
| 2  | c4  | 8   |
| 2  | c5  | 13  |
| 3  | c1  | 9   |
| 3  | c2  | 11  |
| 3  | c3  | 12.5 |
| 3  | c4  | 15.5 |
| 3  | c5  | 7.5  |
| 4  | c1  | 14  |
| 4  | c2  | 18  |
| 4  | c3  | 16  |
| 4  | c4  | 19  |
| 4  | c5  | 20  |
| 5  | c1  | 5.5 |
| 5  | c2  | 9.5 |
| 5  | c3  | 11.5 |
| 5  | c4  | 11  |
| 5  | c5  | 10  |
+----+-----+

```

```

mysql> select * from cours;
+----+-----+-----+
| code | nomCours | enseignant |
+----+-----+-----+
| c1  | nomCours1 | E1      |
| c2  | nomCours2 | E1      |
| c3  | nomCours3 | E5      |
| c4  | nomCours4 | E6      |
| c5  | nomCours5 | E5      |
+----+-----+-----+

```

1. Afficher la liste des étudiants triés par ordre croissant de date de naissance.

```
select * from etudiant order by date_naissance;
```

```
[mysql]> select * from etudiant order by date_naissance;
+----+-----+-----+-----+-----+-----+
| id | nom  | prenom | age   | adresse | date_naissance | niveau |
+----+-----+-----+-----+-----+-----+
|  2 | nom2 | prenom2 | 30    | adresse2 | 1991-04-17    | M2     |
|  4 | nom4 | Ahmed   | 27    | adresse4 | 1994-03-01    | M2     |
|  6 | nom6 | prenom6 | 25    | adresse5 | 1996-01-18    | M2     |
|  3 | nom3 | prenom3 | 25    | adresse3 | 1996-08-19    | M1     |
|  5 | nom5 | prenom5 | 23    | adresse5 | 1997-05-18    | M2     |
|  1 | nom1 | prenom1 | 20    | adresse1 | 2001-01-18    | M1     |
+----+-----+-----+-----+-----+-----+
```

2. Afficher tous les étudiants inscrits à M1 et tous les étudiants inscrits à M2.

```
select * from etudiant where niveau="M1" or niveau="M2";
```

```
[mysql]> select * from etudiant where niveau="M1" or niveau="M2";
+----+-----+-----+-----+-----+-----+
| id | nom  | prenom | age   | adresse | date_naissance | niveau |
+----+-----+-----+-----+-----+-----+
|  1 | nom1 | prenom1 | 20    | adresse1 | 2001-01-18    | M1     |
|  3 | nom3 | prenom3 | 25    | adresse3 | 1996-08-19    | M1     |
|  5 | nom5 | prenom5 | 23    | adresse5 | 1997-05-18    | M2     |
|  6 | nom6 | prenom6 | 25    | adresse5 | 1996-01-18    | M2     |
+----+-----+-----+-----+-----+-----+
```

3. Afficher les matricules des étudiants qui ont passé l'examen du cours 002.

```
select id from examen where code="c2" group by id;
```

```
[mysql]> select id from examen where code="c2" group by id;
+----+
| id |
+----+
|  1 |
|  2 |
|  3 |
|  4 |
|  5 |
+----+
```

4. Afficher les matricules de tous les étudiants qui ont passé l'examen du cours 001 et de tous les étudiants qui ont passé l'examen du cours 002.

```
select id from examen where code="c1" or code="c2" group by id;
```

```
mysql> select id from examen where code="c1" or code="c2" group by id;
+----+
| id |
+----+
| 1 |
| 2 |
| 3 |
| 4 |
| 5 |
+----+
```

5. Afficher la matricule, code, note /20 et note /40 de tous les examens classés par ordre croissant de matricule et de code.

```
select id,note as "note/20",note*2 as "note/40" from examen order by id,code;
```

```
mysql> select id,note as "note/20",note*2 as "note/40" from examen order by id,code;
+----+-----+-----+
| id | note/20 | note/40 |
+----+-----+-----+
| 1 |      9 |     18 |
| 1 |     16 |     32 |
| 1 |     13 |     26 |
| 1 |     10 |     20 |
| 1 |     16 |     32 |
| 2 |     11 |     22 |
| 2 |     16 |     32 |
| 2 |     18 |     36 |
| 2 |      8 |     16 |
| 2 |     13 |     26 |
| 3 |      9 |     18 |
| 3 |     11 |     22 |
| 3 |    12.5 |     25 |
| 3 |    15.5 |     31 |
| 3 |     7.5 |     15 |
| 4 |     14 |     28 |
| 4 |     18 |     36 |
| 4 |     16 |     32 |
| 4 |     19 |     38 |
| 4 |     20 |     40 |
| 5 |     5.5 |     11 |
| 5 |     9.5 |     19 |
| 5 |    11.5 |     23 |
| 5 |     11 |     22 |
| 5 |     10 |     20 |
+----+-----+-----+
```

6. Trouver la moyenne de notes de cours 002.

```
select avg(note) as "Moyenne de notes" from examen where code="c2";
```

```
mysql> select avg(note) as "Moyenne de notes" from examen where code="c2";
+-----+
| Moyenne de notes |
+-----+
|        14.1 |
+-----+
```

7. Compter les examens passés par un étudiant (exemple avec matricule 'e1')

```
select id,count(code) as "nbre d'examen passés" from examen where id=1;
```

```
[mysql]> select id,count(code) as "nbre d'examen passés" from examen where id=1;
+-----+
| id | nbre d'examen passés |
+-----+
| 1  | 5 |
+-----+
```

8. Compter le nombre d'étudiants qui ont passé l'examen du cours 002.

```
select count(id) as "nombre d'étudiants" from examen where code="c2";
```

```
[mysql]> select count(id) as "nombre d'étudiants" from examen where code="c2";
+-----+
| nombre d'étudiants |
+-----+
| 5 |
+-----+
```

9. Calculer la moyenne des notes d'un étudiant (exemple avec matricule 'e1').

```
select id,avg(note) as "moyenne des notes" from examen where id=1;
```

```
[mysql]> select id,avg(note) as "moyenne des notes" from examen where id=1;
+-----+
| id | moyenne des notes |
+-----+
| 1  | 12.8 |
+-----+
```

10. Compter les examens passés par chaque étudiant.

```
select id,count(code) as "nbre d'examen passés" from examen group by id;
```

```
[mysql]> select id,count(code) as "nombres d'examen passés" from examen group by id;
+-----+
| id | nombres d'examen passés |
+-----+
| 1  | 5 |
| 2  | 5 |
| 3  | 5 |
| 4  | 5 |
| 5  | 5 |
+-----+
```

11. Calculer la moyenne des notes pour chaque étudiant.

```
select id,avg(note) as "moyenne des notes" from examen group by id;
```

```
[mysql]> select id,avg(note) as "moyenne des notes" from examen group by id;
+-----+
| id | moyenne des notes |
+-----+
| 1  |      12.8 |
| 2  |      13.2 |
| 3  |      11.1 |
| 4  |      17.4 |
| 5  |
+-----+
```

12. La même au-dessus, mais afficher seulement les étudiants (et leurs moyennes) dont la moyenne est ≥ 15 .

```
select id,avg(note) as "moyenne des notes" from examen group by id having avg(note)>=15;
```

```
[mysql]> select id,avg(note) as "moyenne des notes" from examen group by id having avg(note)>=15;
+-----+
| id | moyenne des notes |
+-----+
| 4  |      17.4 |
+-----+
```

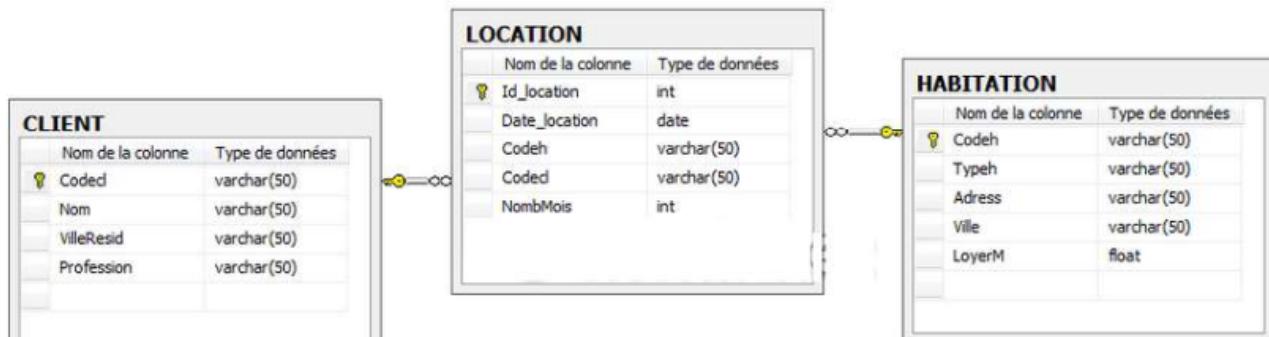
13. Trouver la moyenne de notes de chaque cours.

```
select code,avg(note) as "moyenne des notes" from examen group by code;
```

```
[mysql]> select code,avg(note) as "moyenne des notes" from examen group by code;
+-----+
| code | moyenne des notes |
+-----+
| c1  |      9.7 |
| c2  |     14.1 |
| c3  |     14.2 |
| c4  |     12.7 |
| c5  |     13.3 |
+-----+
```

Exercice 8 :

Soit le modèle relationnel suivant :



Modèle relationnel : Locations

Les valeurs des Tableaux :

```

mysql> select * from habitation;
+-----+-----+-----+-----+-----+
| codeh | typeh | adress | ville   | loyerM |
+-----+-----+-----+-----+-----+
| ch1   | t1   | adresse1 | fes    | 1500   |
| ch2   | t2   | adresse2 | marrakech | 2500   |
| ch3   | t2   | adresse3 | rabat  | 2100   |
| ch4   | t1   | adresse4 | tanger | 1200   |
| ch5   | t3   | adresse5 | marrakech | 3000   |
| ch6   | t3   | adresse6 | tanger | 3100   |
+-----+-----+-----+-----+-----+
    
```

```

mysql> select * from client;
+-----+-----+-----+-----+
| coded | nom  | villeResid | profession |
+-----+-----+-----+-----+
| c1    | nom1 | tanger    | ingenieur  |
| c2    | nom2 | marrakech | enseignant |
| c3    | nom3 | fes       | manager    |
| c4    | nom4 | rabat    | secretaire |
| c5    | nom5 | marrakech | manager    |
| c6    | nom6 | fes       | directeur  |
| c7    | nom7 | tanger    | professeur |
+-----+-----+-----+-----+
    
```

```

mysql> select * from location;
+-----+-----+-----+-----+-----+
| id_location | date_location | codeh | coded | nombMois |
+-----+-----+-----+-----+-----+
| 1            | 2020-01-17    | ch4   | c1   | 9      |
| 2            | 2019-08-27    | ch5   | c2   | 15     |
| 3            | 2019-06-19    | ch1   | c3   | 7      |
| 4            | 2018-11-19    | ch3   | c4   | 20     |
| 5            | 2020-11-19    | ch2   | c5   | 19     |
| 6            | 2017-02-03    | ch1   | c6   | 30     |
| 7            | 2021-02-03    | ch6   | c7   | 3      |
+-----+-----+-----+-----+-----+
    
```

1. Trouver les habitations de type T1 à Tanger et de type T2 à Marrakech.

```
select * from habitation where (typeh="t1" and ville="tanger") or (typeh="t2" and ville="marrakech");
```

```
mysql> select * from habitation where (typeh="t1" and ville="tanger")  
or (typeh="t2" and ville="marrakech");  
+-----+-----+-----+-----+  
| codeh | typeh | adress | ville | loyerM |  
+-----+-----+-----+-----+  
| ch2 | t2 | adresse2 | marrakech | 2500 |  
| ch4 | t1 | adresse4 | tanger | 1200 |  
+-----+-----+-----+-----+
```

2. Déterminer le prix minimum, maximum et moyen des habitations de chaque type.

```
select typeh,min(loyerM),max(loyerM),avg(loyerM) from habitation group by typeh;
```

```
mysql> select typeh,min(loyerM),max(loyerM),avg(loyerM) from habitation group by typeh;  
+-----+-----+-----+-----+  
| typeh | min(loyerM) | max(loyerM) | avg(loyerM) |  
+-----+-----+-----+-----+  
| t1 | 1200 | 1500 | 1350 |  
| t2 | 2100 | 2500 | 2300 |  
| t3 | 3000 | 3100 | 3050 |  
+-----+-----+-----+-----+
```

3. La même que la requête précédente, mais donner les prix pour type et ville et seulement pour les villes de Tanger et Marrakech.

```
select typeh,ville,min(loyerM),max(loyerM),avg(loyerM) from habitation where  
ville="marrakech" or ville="tanger" group by typeh,ville;
```

```
mysql> select typeh,ville,min(loyerM),max(loyerM),avg(loyerM) from habitation  
where ville="marrakech" or ville="tanger" group by typeh,ville;  
+-----+-----+-----+-----+  
| typeh | ville | min(loyerM) | max(loyerM) | avg(loyerM) |  
+-----+-----+-----+-----+  
| t2 | marrakech | 2500 | 2500 | 2500 |  
| t1 | tanger | 1200 | 1200 | 1200 |  
| t3 | marrakech | 3000 | 3000 | 3000 |  
| t3 | tanger | 3100 | 3100 | 3100 |  
+-----+-----+-----+-----+
```

4. Pour chaque habitation, afficher son code, son type, la ville où elle se trouve, les noms des locataires et leur profession.

```
select h.codeh,h.typeh,h.ville,c.nom,c.profession from habitation h, location l,client c where
h.codeh=l.codeh and l.coded=c.coded;
```

```
mysql> select h.codeh,h.typeh,h.ville,c.nom,c.profession from habitation h,
location l,client c where h.codeh=l.codeh and l.coded=c.coded;
+-----+-----+-----+-----+
| codeh | typeh | ville      | nom    | profession |
+-----+-----+-----+-----+
| ch1   | t1    | fes        | nom3   | manager    |
| ch1   | t1    | fes        | nom6   | directeur  |
| ch2   | t2    | marrakech | nom5   | manager    |
| ch3   | t2    | rabat      | nom4   | secretaire |
| ch4   | t1    | tanger     | nom1   | ingenieur  |
| ch5   | t3    | marrakech | nom2   | enseignant |
| ch6   | t3    | tanger     | nom7   | professeur |
+-----+-----+-----+-----+
```

5. La même que la requête précédente mais afficher aussi les habitations qui n'ont jamais été loué.

```
select h.codeh,h.typeh,h.ville,c.nom,c.profession from habitation h, location l,client c where
h.codeh=l.codeh and l.coded=c.coded or l.date_location=NULL;
```

```
mysql> select h.codeh,h.typeh,h.ville,c.nom,c.profession from habitation h, location l,client c
where h.codeh=l.codeh and l.coded=c.coded or l.date_location=NULL;
+-----+-----+-----+-----+
| codeh | typeh | ville      | nom    | profession |
+-----+-----+-----+-----+
| ch4   | t1    | tanger     | nom1   | ingenieur  |
| ch5   | t3    | marrakech | nom2   | enseignant |
| ch1   | t1    | fes        | nom3   | manager    |
| ch3   | t2    | rabat      | nom4   | secretaire |
| ch2   | t2    | marrakech | nom5   | manager    |
| ch1   | t1    | fes        | nom6   | directeur  |
| ch6   | t3    | tanger     | nom7   | professeur |
+-----+-----+-----+-----+
```

6. Trouver les clients qui ont loué une habitation dans la même ville où ils ont leur résidence

```
select c.nom,c.villeResid,c.profession from client c,habitation h, location l where
c.coded=l.coded and l.codeh=h.codeh and h.ville=c.villeResid;
```

```
mysql> select c.nom,c.villeResid,c.profession from client c,habitation h, location l
where c.coded=l.coded and l.codeh=h.codeh and h.ville=c.villeResid;
+-----+-----+
| nom  | villeResid | profession |
+-----+-----+
| nom3 | fes       | manager    |
| nom6 | fes       | directeur  |
| nom5 | marrakech | manager    |
| nom4 | rabat     | secretaire |
| nom1 | tanger    | ingenieur  |
| nom2 | marrakech | enseignant |
| nom7 | tanger    | professeur |
+-----+-----+
```

7. Pour chaque type compter combien des habitations de ce type ont été prises en location. Montrer seulement les types dont au moins 3 habitations (pas forcément différentes) ont été prises en location.

```
select h.typeh,count(l.id_location) from location l,habitation h where h.codeh=l.codeh group by h.typeh;
```

```
mysql> select h.typeh,count(l.id_location) from location l,habitation h
where h.codeh=l.codeh group by h.typeh;
+-----+
| typeh | count(l.id_location) |
+-----+
| t1    |          3           |
| t2    |          2           |
| t3    |          2           |
+-----+
```

8. Pour chaque habitation compter combien de mois en totale a été loué.

```
select h.codeh, sum(l.nombMois) as "mois loués" from habitation h,location l where
h.codeh=l.codeh group by h.codeh;
```

```
[mysql]> select h.codeh, sum(l.nombMois) as "mois loués" from habitation h,location l
where h.codeh=l.codeh group by h.codeh;
+-----+
| codeh | mois loués |
+-----+
| ch1   |      37      |
| ch2   |      19      |
| ch3   |      20      |
| ch4   |       9      |
| ch5   |      15      |
| ch6   |       3      |
+-----+
```

9. Pour chaque client calculer combien a dépensé en location en totale.

```
select c.coded,sum(l.nombMois*h.loyerM) from habitation h,client c,location l whered=l.coded
and l.codeh=h.codeh group by c.coded;
```

```
mysql> select c.coded,sum(l.nombMois*h.loyerM) from habitation h,client c,location
l whered=l.coded and l.codeh=h.codeh group by c.coded;
+-----+
| coded | sum(l.nombMois*h.loyerM) |
+-----+
| c3   |      10500     |
| c6   |      45000     |
| c5   |      47500     |
| c4   |      42000     |
| c1   |      10800     |
| c2   |      45000     |
| c7   |       9300     |
+-----+
```

10. Trouver les clients qui n'ont jamais loué une habitation.

```
select c.* from client c,location l where c.coded=l.coded and l.id_location=NULL;
```

2ème méthode : select * from client where coded not in (select distinct(coded) from location);

```
[mysql]> select c.* from client c,location l where c.coded=l.coded and l.id_location=NULL;
Empty set (0.00 sec)
```

11. Trouver les clients qui ont loué à la fois des habitations de type T1 et de type T2.

```
select c.*,h.typeh from client c,habitation h,location l where c.coded=l.coded and
l.codeh=h.codeh and (h.typeh="t1" or h.typeh="t2");
```

```
[mysql]> select c.* ,h.typeh from client c,habitation h,location l where c.coded=l.coded and
l.codeh=h.codeh and (h.typeh="t1" or h.typeh="t2");
+-----+-----+-----+-----+
| coded | nom   | villeResid | profession | typeh |
+-----+-----+-----+-----+
| c3   | nom3  | fes       | manager    | t1      |
| c6   | nom6  | fes       | directeur  | t1      |
| c5   | nom5  | marrakech | manager    | t2      |
| c4   | nom4  | rabat     | secretaire | t2      |
| c1   | nom1  | tanger    | ingenieur  | t1      |
+-----+-----+-----+-----+
```