

Master Internet des Objets et Systèmes Mobiles

2021-2022

**Module: S.I et Bases de données**

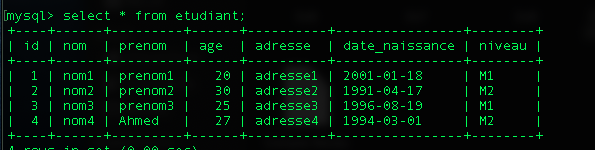
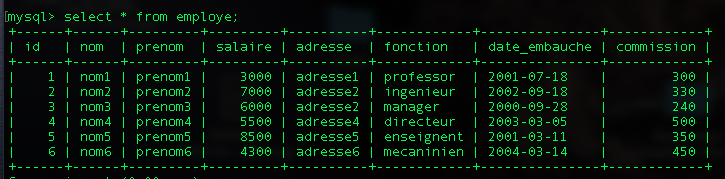
Rapport des Travaux Pratiques

Série n° 1

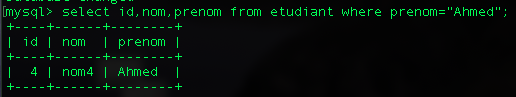
Réalisé par : Encadré par :

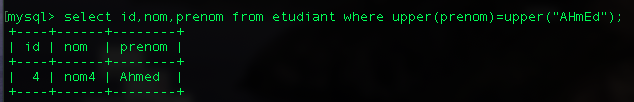
ZAROUAL Mohammed Pr. EL AKKAD Nabil

Les relations utilisées dans les six premiers exercices sont :

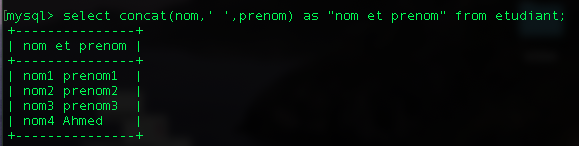
* La relation **etudiant** (id, nom, prénom, age, adresse) :
* La relation **employé** (id, nom, prénom, salaire, adresse, fonction, date\_embauche,commission) :

***Exercice 1 (Fonction de manipulation de caractères):***

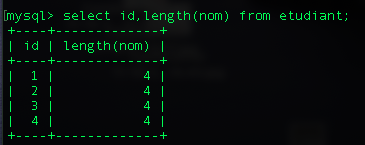
1. Afficher l’identifiant, le nom et le prénom des étudiants dont le prenom est : Ahmed.   
     
   select id,nom,prenom from etudiant where prenom="Ahmed";
2. On supposera qu’aucun enregistrement n’a pas été affiché lors de l’exécution à cause de la casse (lettre en minuscule ou en majuscule). Réécrire la requête pour remédier au problème.

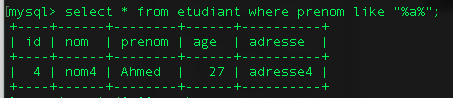
 select id,nom,prenom from etudiant where upper(prenom)=upper("AHmEd");

1. Afficher le nom et le prénom concaténé de tous les étudiants sous le nom ‘’ Nom de l’étudiant ‘’.

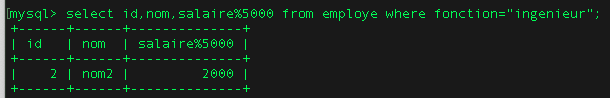
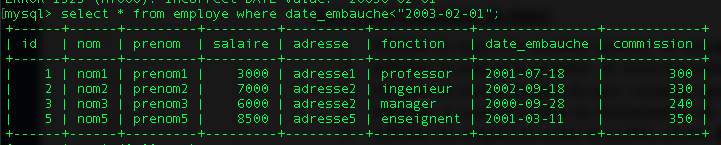
 select concat(nom,' ',prenom) as "nom et prenom" from etudiant;

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

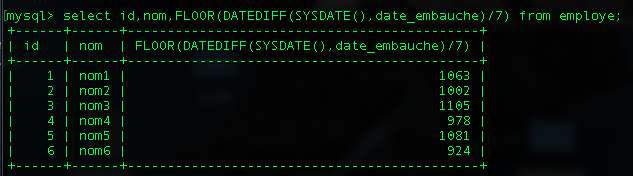
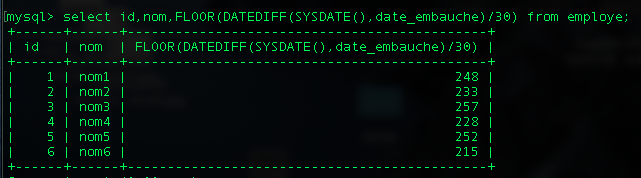
 select id,length(nom) from etudiant;

1. Afficher les étudiants dont le prénom contient la lettre ‘a’.   
     
   select \* from etudiant where prenom like "%a%"

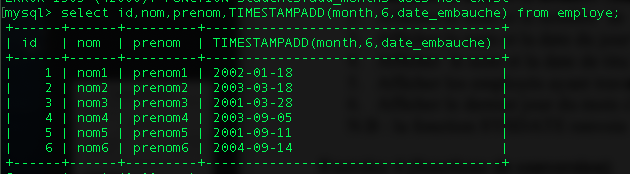
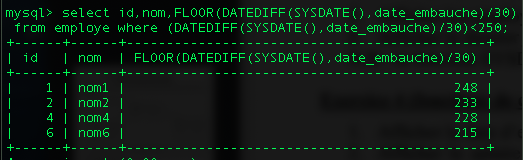
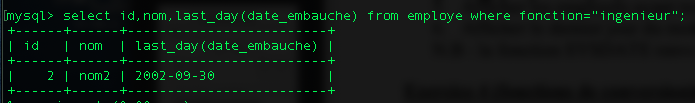
***Exercice 2:***

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";
2. Afficher tous les employés embauchés avant le ’01/02/03’   
     
   select \* from employe where date\_embauche<"2003-02-01";

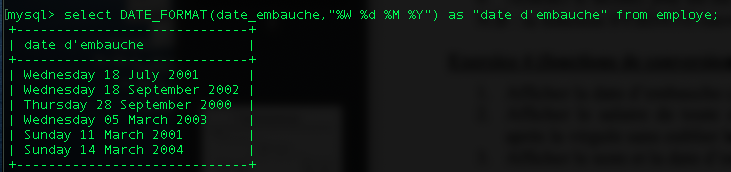
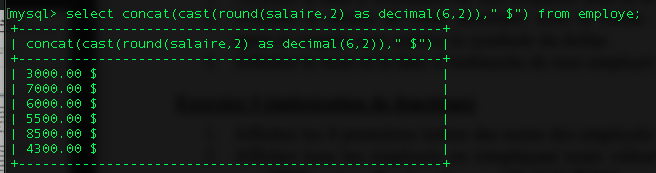
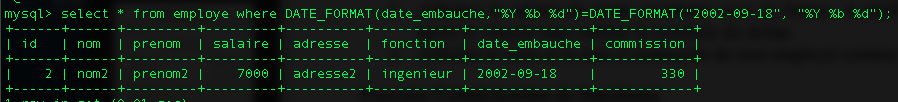
***Exercice 3 (fonctions de date):***

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

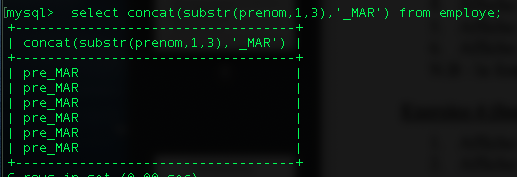
select nom,prenom, next\_day(date\_embauche,"friday") as "premier vendredi" from employe;

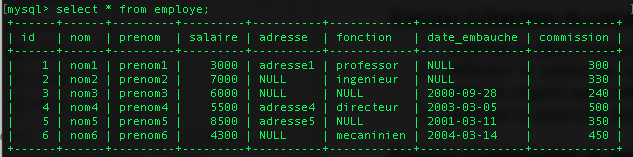
1. Afficher le nom et la date de titularisation de tout employé (la titularisation se fait après 6 mois).   
     
   SQL : select id, nom, prenom, add\_months(date\_embauche,6) from employe;  
   MySQL : select id, nom, prenom, TIMESTAMPADD(month,6,date\_embauche) from employe;
2. Afficher les employés ayant travaillés moins de 250 mois.   
     
   select id,nom,DATEDIFF(SYSDATE(),date\_embauche)/30 from employe where (DATEDIFF(SYSDATE(),date\_embauche)/30)<250;
3. 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";

***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');
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;
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");

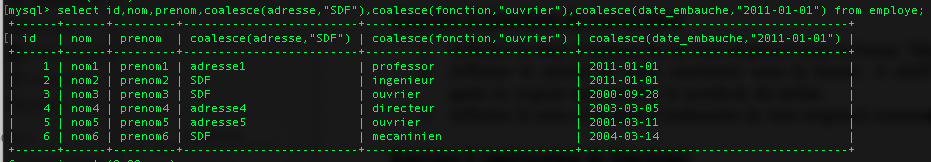
***Exercice 5 (imbrication de fonctions ):***

1. Afficher les 3 premières lettres des noms des employés concaténés à ‘\_MAR’ en majuscule.   
     
   select concat(substr(prenom,1,3),'\_MAR') from employe;
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’.

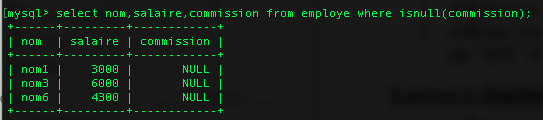
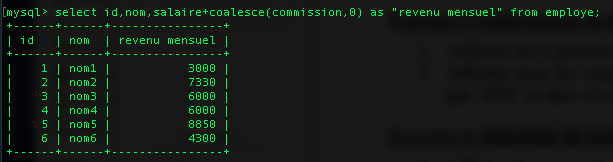
Le tableau "employe" avant la modification (avec les valeurs NULL) :

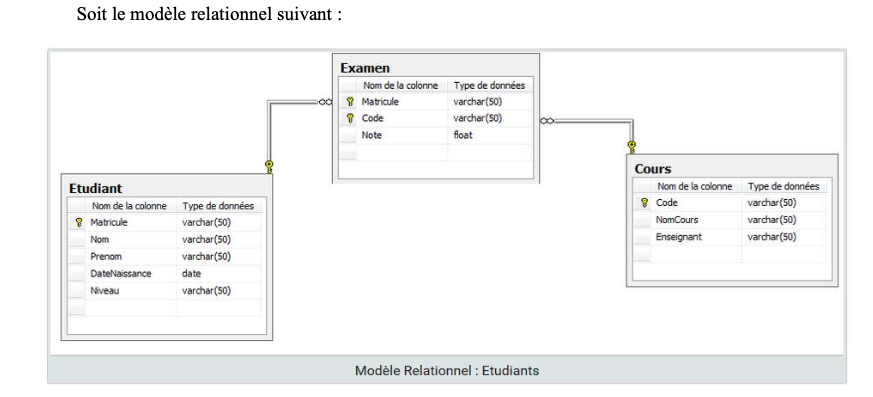
Apres la modification :

SQL : select id, nom, prenom, nvl(adresse,"SDF"), nvl(fonction,"ouvrier"), nvl (date\_embauche, "2011-01-01" ) from employe;

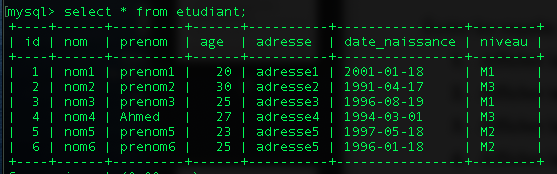
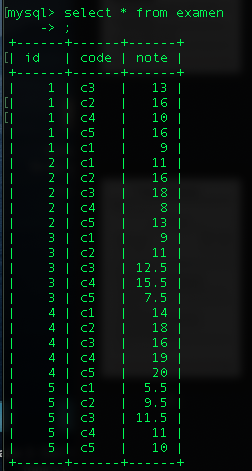
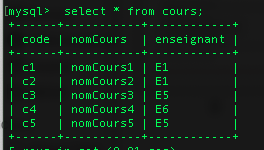
 MySQL : select id, nom, prenom, coalesce(adresse,"SDF"), coalesce(fonction,"ouvrier"), coalesce(date\_embauche,"2011-01-01") from employe;

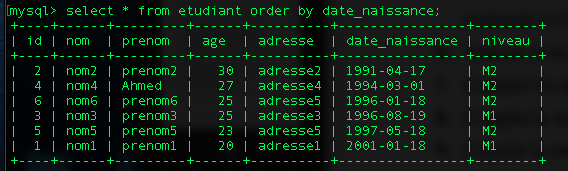
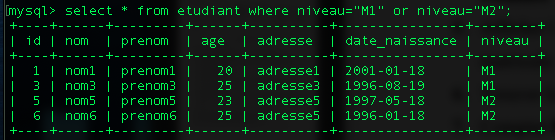
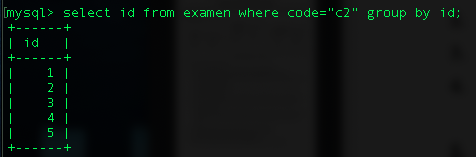
***Exercice 6 (fonctions de restitution des valeurs nulles) :***

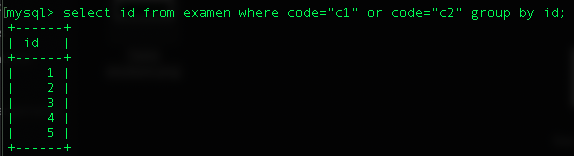
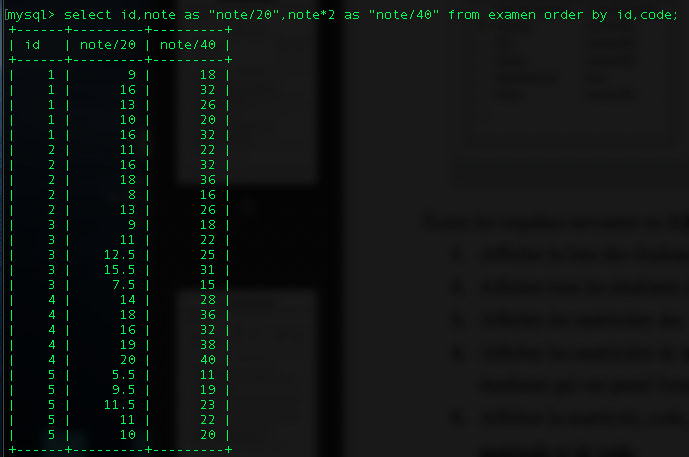
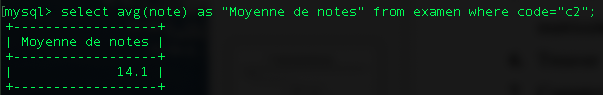
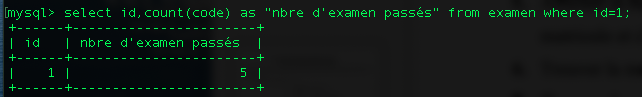
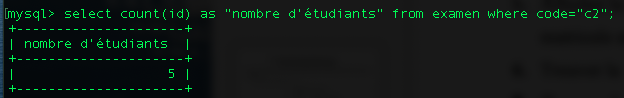
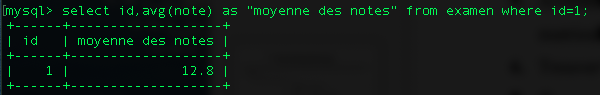
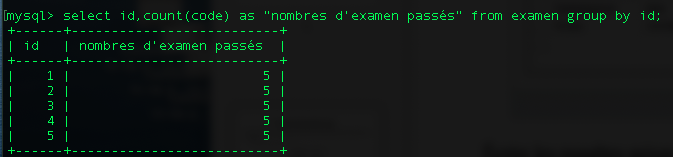
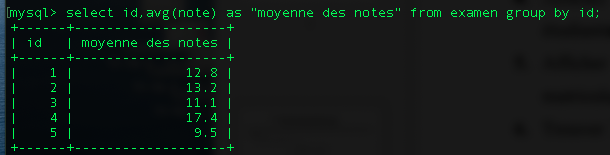
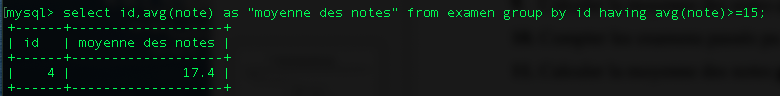
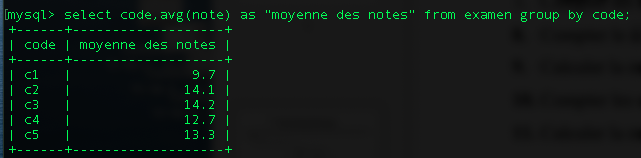
1. Afficher le nom et le salaire annuel de tout employé (la colonne commission contient des valeurs ‘null’) ;   
     
   select nom,salaire,commission from employe where isnull(commission);
2. Afficher le revenu mensuel de tout employé, qui égale au salaire si la commission est nulle et le salaire + commission dans le cas contraire.   
     
   select id,nom,salaire+coalesce(commission,0) as "revenu mensuel" from employe;

***Exercice7 :***

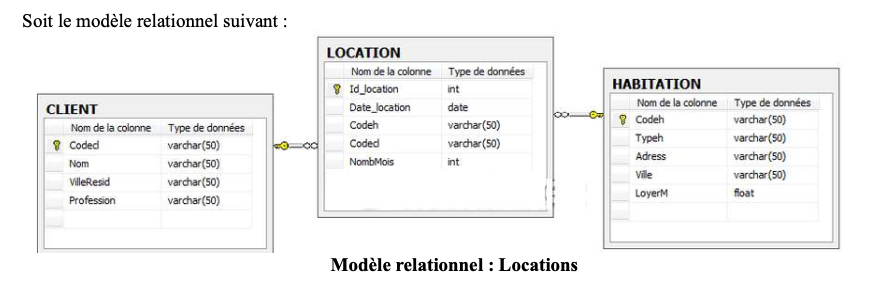
**Les valeurs des Tableaux :**



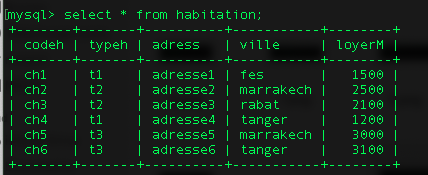
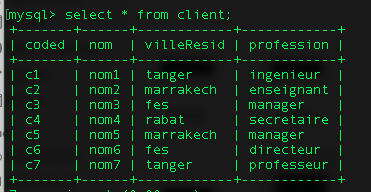
1. Afficher la liste des étudiants triés par ordre croissant de date de naissance.   
     
   select \* from etudiant order by date\_naissance;
2. Afficher tous les étudiants inscrits à M1 et tous les étudiants inscrits à M2.   
     
   select \* from etudiant where niveau="M1" or niveau="M2";
3. Afficher les matricules des étudiants qui ont passé l'examen du cours 002.   
     
   select id from examen where code="c2" group by id;

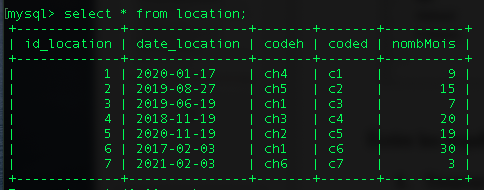
1. Afficher les matricules de tous les étudiants qui ont passé l'examen du cours 001 et de tous les étudiants qui ont passé l'examen du cours 002.   
     
   select id from examen where code="c1" or code="c2" group by id;
2. Afficher la matricule, code, note /20 et note /40 de tous les examens classés 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;
3. Trouver la moyenne de notes de cours 002.   
     
   select avg(note) as "Moyenne de notes" from examen where code="c2";
4. 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;
5. 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";
6. 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;
7. Compter les examens passés par chaque étudiant.   
     
   select id,count(code) as "nbre d'examen passés" from examen group by id;
8. Calculer la moyenne des notes pour chaque étudiant.   
     
   select id,avg(note) as "moyenne des notes" from examen group by id;
9. 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;
10. Trouver la moyenne de notes de chaque cours.   
      
    select code,avg(note) as "moyenne des notes" from examen group by code;

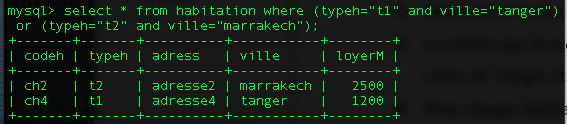
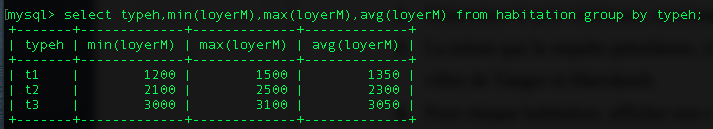
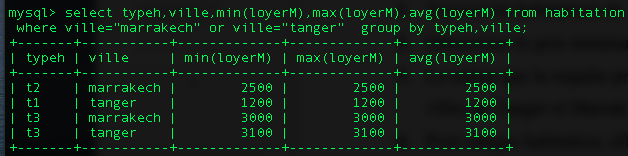
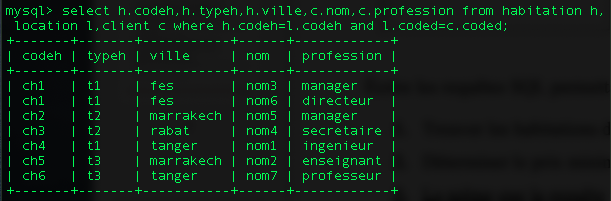
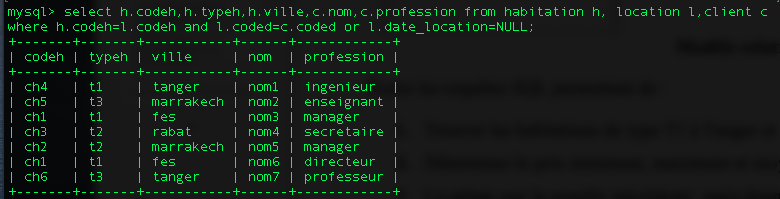
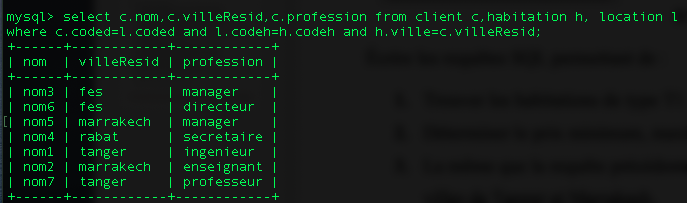
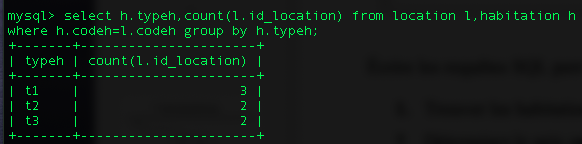
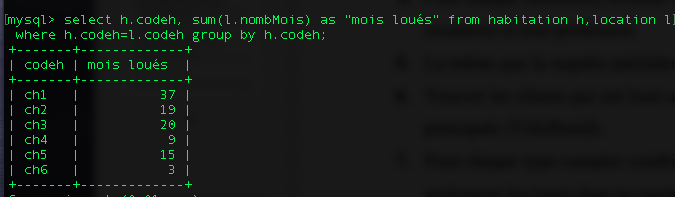
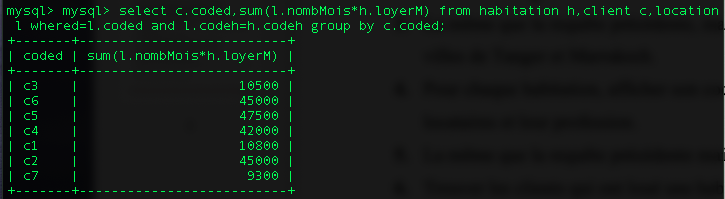
***Exercice 8 :***



**Les valeurs des Tableaux** :





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");
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;
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;
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;
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;
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;
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 forcement 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;
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;
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;
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 methode : select \* from client where coded not in ( select distinct(coded) from location);



1. 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");

