**Série 2 : SELECT - LMD**

**Exercice :**

1. Créer puis Remplir la base de données ci-dessous, en respectant les règles citées, avec les données suivantes :

**Les règles de gestion :**

* Le nom de la classe est unique.
* Le nom de matière est unique.
* Nom et prénom d’étudiant sont uniques
* La note doit être comprise entre 0 et 20
* L’appréciation doit avoir l’un des valeurs suivant :
  1. Faible
  2. Passable
  3. Assez-bien
  4. Bien
  5. Très bien
  6. Excellent
* Les champs suivant doivent être obligatoirement renseignés :
  1. Nom étudiant et prénom étudiant
  2. Nom matière
  3. Nom classe

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Nom stagiaire | prénom Stagiaire | Date Naissance | Ville | classe | Matière | Note | Appréciation |
|  | KHALIL | Anas | 12/01/1990 | CASA | TDI-2A | SGBD1 | 14,5 | Bien |
|  | KHALIL | Anas | 12/01/1990 | CASA | TDI-2A | SGBD2 | 13 | Assez-bien |
|  | KHALIL | Anas | 12/01/1990 | CASA | TDI-2A | HTML-CSS | 15 | Bien |
|  | KHALIL | Anas | 12/01/1990 | CASA | TDI-2A | ASP.NET | 11 | Assez-bien |
|  | KHALIL | Anas | 12/01/1990 | CASA | TDI-2A | Client-Serveur | 12 | Assez-bien |
|  | KHALIL | Anas | 12/01/1990 | CASA | TDI-2A | Gestion Projet | 19 | Excellent |
|  | ALAMI | ZAID | 12/01/1994 | Rabat | TDI-2A | SGBD1 | 16,5 | Très Bien |
|  | ALAMI | ZAID | 12/01/1994 | Rabat | TDI-2A | SGBD2 | 10 | Passable |
|  | ALAMI | ZAID | 12/01/1994 | Rabat | TDI-2A | HTML-CSS | 13 | Assez-bien |
|  | ALAMI | ZAID | 12/01/1994 | Rabat | TDI-2A | ASP.NET | 11,5 | Assez-bien |
|  | ALAMI | ZAID | 12/01/1994 | Rabat | TDI-2A | Client-Serveur | 13 | Assez-bien |
|  | ALAMI | ZAID | 12/01/1994 | Rabat | TDI-2A | Gestion Projet | 13 | Assez-bien |
|  | TALEB | OMAR | 01/01/1993 | TETOUAN | TDI-1A | ALGO | 11,25 | Passable |
|  | TALEB | OMAR | 01/01/1993 | TETOUAN | TDI-1A | Langage C | 12,25 | Assez-bien |
|  | TALEB | OMAR | 01/01/1993 | TETOUAN | TDI-1A | Merise | 13 | Assez-bien |
|  | TALEB | OMAR | 01/01/1993 | TETOUAN | TDI-1A | POO | 10,25 | Passable |
|  | BENNIS | ALI | 12/09/1991 | CASA | TDI-1A | ALGO | 12,25 | Assez-bien |
|  | BENNIS | ALI | 12/09/1991 | CASA | TDI-1A | Langage C | 13 | Assez-bien |
|  | BENNIS | ALI | 12/09/1991 | CASA | TDI-1A | Merise | 14,5 | Bien |
|  | BENNIS | ALI | 12/09/1991 | CASA | TDI-1A | POO | 15,75 | Bien |
|  | SELAMI | LAILA | 11/06/1993 | Rabat | TDI-1A | ALGO | 10,5 | Passable |
|  | SELAMI | LAILA | 11/06/1993 | Rabat | TDI-1A | Langage C | 10 | Passable |
|  | SELAMI | LAILA | 11/06/1993 | Rabat | TDI-1A | Merise | 16 | Très bien |
|  | SELAMI | LAILA | 11/06/1993 | Rabat | TDI-1A | POO | 11,5 | Passable |
|  | MOHA | ZAID | 12/12/1995 | CASA | TDI-2A | SGBD1 | 10 | Passable |
|  | MOHA | ZAID | 12/12/1995 | CASA | TDI-2A | SGBD2 | 7 | Ajourné |
|  | MOHA | ZAID | 12/12/1995 | CASA | TDI-2A | HTML-CSS | 10 | Passable |
|  | MOHA | ZAID | 12/12/1995 | CASA | TDI-2A | ASP.NET | 11 | Passable |
|  | MOHA | ZAID | 12/12/1995 | CASA | TDI-2A | Client-Serveur | 12 | Assez-bien |
|  | MOHA | ZAID | 12/12/1995 | CASA | TDI-2A | Gestion Projet | 8,5 | Ajourné |
|  | FIHRI | HAMZA | 09/09/1995 | TETOUAN | TDI-1A | ALGO | 8 | Ajourné |
|  | FIHRI | HAMZA | 09/09/1995 | TETOUAN | TDI-1A | Langage C | 6 | Ajourné |
|  | FIHRI | HAMZA | 09/09/1995 | TETOUAN | TDI-1A | Merise | 5 | Ajourné |
|  | FIHRI | HAMZA | 09/09/1995 | TETOUAN | TDI-1A | POO | 7 | Ajourné |

****

**Travail à faire :**

Réaliser les requêtes suivantes :

1. Afficher le nom et prénom des étudiants sans redondance. (distinct)
2. Afficher le nom et prénom des étudiants sans redondance par ordre alphabétique concaténés dans une seule colonne. (Concat et distinct et order by)
3. Afficher la liste des étudiants qui sont en TDI-1A et TDI-2A. (Or)
4. Afficher la liste des étudiants dont la taille de leurs noms dépasse 10 caractères ; (len ou datalength)
5. Afficher le nombre de classes, mettre comme nom de colonne ‘Nombre de classes’. (Count et as)
6. Afficher le nombre de matières, mettre comme nom de colonne ‘Nombre de matières’. (count et as)
7. Afficher les étudiants qui ont un nom qui commence par A et se termine par I ou le prénom commence par H et se termine par A. (like et utiliser les parenthèses)

select \* from etudiant where nom like 'A%I' or prenom like 'H%A'

select \* from etudiant where (nom like 'A%' and nom like '%I')or

( prenom like 'H%' and prenom like '%A')

1. Afficher les étudiants de casa qui ont un prénom qui ne commence pas par la lettre K. (not like)
2. Afficher l’année ainsi que le mois(en caractères) de naissance de chaque étudiant ; (datename mm et year)

select cast(year(daten) as varchar)+ ' ' + DATENAME(mm,daten) from etudiant

select FORMAT(daten, 'MMMM/yyyy') from etudiant

1. Afficher les notes de l’étudiant SELAMI LAILA. (where)
2. Afficher la moyenne des notes de chaque étudiant. (Avg et group by)
3. Afficher la moyenne des notes de chaque matière. (Avg et group by)
4. Afficher les classes avec les matières en remplaçant ‘C’ dans ‘langage C’ par ‘C++’ ;

select nomclasse, replace(nommatiere,'Langage C','Langage C++')

from matiere inner join note on matiere.idMatiere = note.idMatiere

inner join etudiant on note.numEtu = etudiant.numEtu

inner join classe on etudiant.idClasse = classe.idClasse

1. Afficher la moyenne des notes de chaque classe.
2. Pour chaque ville, afficher la moyenne des notes de chaque étudiant.

select nomVille, nom, prenom, AVG(note)

from etudiant inner join note on note.numEtu = etudiant.numEtu

inner join ville on etudiant.idville = ville.idVille

group by nomVille, nom, prenom

order by nomVille, nom, prenom

1. Pour chaque appréciation, afficher la moyenne des notes.
2. Pour chaque ville et chaque classe, afficher la moyenne des notes.

select nomVille,nomClasse, AVG(note)

from etudiant inner join note on note.numEtu = etudiant.numEtu

inner join ville on etudiant.idville = ville.idVille

inner join classe on etudiant.idClasse = classe.idClasse

group by nomVille, nomClasse

order by nomVille, nomClasse

1. Afficher les villes qui ont un nombre d’étudiants supérieur à 2.

select nomville, count(numetu)

from etudiant inner join ville on etudiant.idVille = ville.idVille

group by nomVille

having COUNT(numetu) > 2

1. Afficher le nombre d’étudiants pour chaque appréciation.
2. Afficher les étudiants qui n’ont pas de note.

select \* from etudiant where numetu not in (select numetu from note)

1. Afficher les étudiants qui ont la date de naissance entre 01-01-1993 et 05-06-1994 (between)
2. Afficher la liste des étudiants qui ont la moyenne supérieure à 10(having)
3. Afficher les étudiants qui ont la moyenne inférieur à 10 et habitent à Tétouan. (having et where)
4. Afficher l’étudiant qui a la moyenne supérieure.

select nom, prenom, AVG(note) as moy

from etudiant inner join note on note.numEtu = etudiant.numEtu

group by nom, prenom

having AVG(note) in

(

select max(moy) from

(select AVG(note) as moy

from etudiant inner join note on note.numEtu = etudiant.numEtu

group by nom, prenom

) as r1

)

1. Afficher la liste des classes qui contiennent moins de 2 étudiants.
2. Afficher la ville qui a le maximum d’étudiants.
3. Afficher la première et dernière note par matière.
4. Afficher les quatre premiers étudiants et leurs moyennes ;
5. Afficher les noms et prénoms des étudiants en majuscule concaténés dans une colonne ;
6. Afficher l’âge de chaque étudiant de cette façon : l’âge de « nom » « prénom » est « âge »;
7. On veut affecter à chaque étudiant un code, le code est formé des 2 premiers caractères du nom, des 3 derniers caractères du prénom et de l’âge, afficher le nom, prénom et code de chaque étudiant ;
8. Afficher le nombre d’étudiants nés entre les années 1990 et 1995 ;