

Devoir Surveillé

Semestre : 2

Module : Base de données et Systèmes d'Information

Élément: Base de données

Durée : 1h et 30mn

Date : 03 /06 /2015

Exercice 1 (Questions de cours – 5 Points)

1. A quoi sert une contrainte d'intégrité?

- À obtenir une indépendance logique
- À garder les bases cohérentes
- À intégrer des applications existantes

2. Quand doit-on déclarer la contrainte d'intégrité ?

- À la création du schéma de la base de données
- Lorsqu'on veut faire une mise à jour
- Lorsqu'on veut ajouter un nouvel enregistrement.

3. Une base de donnée est cohérente lorsque:

- Toutes les contraintes d'intégrité sont respectées
- Toutes les contraintes d'intégrité référentielles sont respectées
- Les données saisies ne sont pas contradictoires

4. Que signifie la contrainte "not null" sur un attribut ?

- La colonne doit avoir des valeurs différentes de la chaîne de caractère d'un seul espace
- La colonne doit être renseignée
- La colonne doit avoir des valeurs différentes de zéro

5. Les contraintes "unique" et "primary Key" jouent le même rôle ?

- Vrai
- Faux
- Vrai sauf que la colonne décrite dans une clause "unique" peut ne pas avoir de valeurs

Exercice 2 (15 Points)

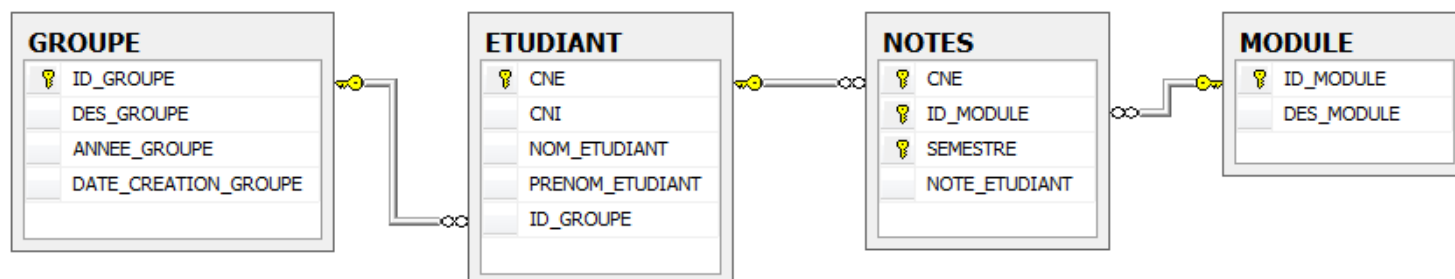
Pour afficher les résultats de leurs étudiants à la fin de chaque semestre, l'EST- Laâyoune utilise la base de données **ESTL\_NOTES** dont le schéma est le suivant :

GRUPE (ID GROUPE, DES GROUPE, ANNEE GROUPE, DATE\_CREATION GROUPE)

ETUDIANT (CNE, CNI, NOM\_ETUDIANT, PRENOM\_ETUDIANT, ID GROUPE#)

MODULE (ID MODULE, DES MODULE)

NOTES (CNE#, ID MODULE#, SEMESTRE, NOTE\_ETUDIANT)



Dont les champs de chaque relation définie dans le tableau ci-dessous :

Nom du champ	Type de données	Possibilité de valeurs nulles	Description
CNE	Nombre Naturel	Non Null	Code national d'étudiant
CNI	Chaîne de caractère (10)	Non Null	N° de la carte nationale d'identité est aussi une clé de la relation
NOM_ETUDIANT	Chaîne de caractère (20)	Non Null	Nom de l'étudiant
PRENOM_ETUDIANT	Chaîne de caractère (20)	Non Null	Prénom de l'étudiant
ID_GROUPE	Numéro Auto	Non Null	Identificateur du groupe
DES_GROUPE	Chaîne de caractère (100)	Non Null	Description de groupe
ANNEE_GROUPE	Nombre Naturel	Non Null	Année de groupe, valeurs possibles 1 pour la première année et 2 pour la deuxième année
DATE_CREATION_GROUPE	Date	Non Null	Date de création de groupe, Valeur par défaut date d'aujourd'hui
ID_MODULE	Chaîne de caractère (8)	Non Null	Identifiant du Module
DES_MODULE	Chaîne de caractère (100)	Non Null	Description du Module
SEMESTRE	Nombre Naturel	Non Null	Semestre, Valeurs possibles 1 pour première semestre 2 pour deuxième semestre 3 pour troisième semestre et 4 pour quatrième semestre.
NOTE_ETUDIANT	Nombre réel	Null	Note du module sur 20

### Travail à faire :

1. Donner les requêtes SQL permettant de créer la base de données et les tables du schéma précédent en respectant les règles d'intégrité indiqué dans le tableau ci-dessus
2. Insérer dans la table ETUDIANT l'étudiant (CNE=1234567890, CIN='SH98765432', NOM= 'Ben Hayane', PRENOM= 'Jaber' , ID\_GROUPE=1)
3. Afficher les étudiants par groupe (dont l'année de création est 2014) ;
4. Afficher la moyenne de chaque étudiant du groupe 1 (i.e. IDGROUPE=1) de l'année 2014 dans le semestre 1 ;
5. Afficher les notes de tous les étudiants du groupe 2 (i.e. IDGROUPE=2) de l'année 2014 dans le Module M121 ;
6. Afficher par semestre la moyenne de l'étudiant dont le CNE=2468013579 ;
7. Afficher les étudiants dont les noms se terminent par la lettre 'e' ;
8. Afficher les étudiants dont les noms contiennent 'a' ;
9. Pour chaque étudiant afficher son CNE, NOM, PRENOM et le nombre des modules a passé (Le résultat de la requête doit inclure aussi les étudiants qui n'ont pas passé aucun module)
10. Afficher les étudiants du groupe 2 (i.e. IDGROUPE=2) de l'année 2014 ayant des notes, du module (ID\_MODULE= 'M121'), comprises entre 10 et 20 ;
11. Ajouter à tous les étudiants du groupe 2 (i.e. IDGROUPE=2) de l'année 2014 un point à la note du module M121 ;
12. Afficher les étudiants du groupe 1 de l'année 2014 qui n'ont pas validé le Module M122.

Bonne Chance... /

**NB :** Vous pouvez utiliser les fonctions SQL : GETDATE() qui retourne la date d'aujourd'hui et YEAR retourne l'année d'une date donnée (**Ex :** GETDATE()= 03/06/2015 et YEAR(03/06/2015)=2015)

Devoir Surveillé  
(Corrigé)

Semestre : 2

Module : Base de données et Systèmes d'Information

Date : 03 /06 /2015

Élément: Base de données

Durée : 1h et 30mn

Exercice 1 (Questions de cours – 5 Points)

1. A quoi sert une contrainte d'intégrité?

- d. À obtenir une indépendance logique
- e. **À garder les bases cohérentes**
- f. À intégrer des applications existantes

2. Quand doit-on déclarer la contrainte d'intégrité ?

- d. **À la création du schéma de la base de données**
- e. Lorsqu'on veut faire une mise à jour
- f. Lorsqu'on veut ajouter un nouvel enregistrement.

3. Une base de donnée est cohérente lorsque:

- d. **Toutes les contraintes d'intégrité sont respectées**
- e. Toutes les contraintes d'intégrité référentielles sont respectées
- f. Les données saisies ne sont pas contradictoires

4. Que signifie la contrainte "not null" sur un attribut ?

- d. La colonne doit avoir des valeurs différentes de la chaîne de caractère d'un seul espace
- e. **La colonne doit être renseignée**
- f. La colonne doit avoir des valeurs différentes de zéro

5. Les contraintes "unique" et "primary Key" jouent le même rôle ?

- d. Vrai
- e. Faux
- f. **Vrai sauf que la colonne décrite dans une clause "unique" peut ne pas avoir de valeurs**

Exercice 2 (15 Points)

1. Donner les requêtes SQL permettant de créer la base de données et les tables du schéma précédent en respectant les règles d'intégrité indiqué dans le tableau ci-dessus

a. **BASE DE DONNEES ESTL NOTES :**

```
CREATE DATABASE ESTL_NOTES  
USE ESTL_NOTES
```

b. **TABLE GROUPE :**

```
CREATE TABLE GROUPE (  
ID_GROUPE INT IDENTITY(1,1) PRIMARY KEY NOT NULL,  
DES_GROUPE VARCHAR(100) NOT NULL,  
ANNEE_GROUPE INT CHECK (ANNEE_GROUPE IN (1,2)) NOT NULL,  
DATE_CREATION_GROUPE DATE DEFAULT GETDATE() NOT NULL  
);
```

c. **TABLE ETUDIANT:**

```
CREATE TABLE ETUDIANT (  
CNE INT PRIMARY KEY NOT NULL,  
CIN VARCHAR(10) UNIQUE NOT NULL,  
NOM_ETUDIANT VARCHAR(20) NOT NULL,  
PRENOM_ETUDIANT VARCHAR(20) NOT NULL,  
ID_GROUPE INT FOREIGN KEY REFERENCES GROUPE(ID_GROUPE) NOT NULL  
);
```

d. **TABLE MODULE:**

```
CREATE TABLE MODULE (  
ID_MODULE VARCHAR(8) PRIMARY KEY NOT NULL,  
DES_MODULE VARCHAR(100) NOT NULL,  
)
```

e. **TABLE NOTES :**

```
CREATE TABLE NOTES (  
CNE INT FOREIGN KEY REFERENCES ETUDIANT(CNE) NOT NULL,  
ID_MODULE VARCHAR(8) FOREIGN KEY REFERENCES MODULE(ID_MODULE) NOT NULL,  
SEMESTRE INT CHECK (SEMESTRE IN(1,2,3,4)) NOT NULL,  
NOTE_ETUDIANT FLOAT CHECK (NOTE_ETUDIANT BETWEEN 0 AND 20) NOT NULL,  
PRIMARY KEY (CNE, ID_MODULE, SEMESTRE)  
);
```

2. Insérer dans la table ETUDIANT l'étudiant (CNE=1234567890, CIN='SH98765432', NOM= 'Ben Hayane', PRENOM='Jaber' , ID\_GROUPE=1)

```
INSERT INTO ETUDIANT VALUES (1234567890, 'SH98765432', 'Ben Hayane', 'Jaber' , 1);
```

3. Afficher les étudiants par groupe (dont l'année de création est 2014) ;

```
SELECT CNE, NOM_ETUDIANT, PRENOM_ETUDIANT, GROUPE.DATE_CREATION_GROUPE, ETUDIANT.ID_GROUPE,  
YEAR(GROUPE.DATE_CREATION_GROUPE)  
FROM ETUDIANT, GROUPE  
WHERE ETUDIANT.ID_GROUPE=GROUPE.ID_GROUPE AND YEAR(GROUPE.DATE_CREATION_GROUPE)=2014  
GROUP BY ETUDIANT.ID_GROUPE, CNE, NOM_ETUDIANT, PRENOM_ETUDIANT, GROUPE.DATE_CREATION_GROUPE  
ORDER BY ETUDIANT.ID_GROUPE
```

4. Afficher la moyenne de chaque étudiant du groupe 1 (i.e. IDGROUPE=1) de l'année 2014 dans le semestre 1 ;

```
SELECT ETUDIANT.CNE, NOM_ETUDIANT , PRENOM_ETUDIANT , AVG(NOTE_ETUDIANT) As 'Moyenne S1'  
FROM ETUDIANT , NOTES  
WHERE ETUDIANT.CNE =NOTES.CNE  
AND ETUDIANT.ID_GROUPE=1 AND SEMESTRE=1  
GROUP BY ETUDIANT.CNE, NOM_ETUDIANT , PRENOM_ETUDIANT
```

5. Afficher les notes de tous les étudiants du groupe 2 (i.e. IDGROUPE=2) de l'année 2014 dans le Module M121 ;

```
SELECT ETUDIANT.CNE , ETUDIANT.NOM_ETUDIANT  
, ETUDIANT.PRENOM_ETUDIANT, NOTES.NOTE_ETUDIANT, NOTES.SEMESTRE, GROUPE.ID_GROUPE  
, NOTES.ID_MODULE  
FROM ETUDIANT, NOTES, GROUPE  
WHERE ETUDIANT.CNE=NOTES.CNE  
AND ETUDIANT.ID_GROUPE=GROUPE.ID_GROUPE  
AND ETUDIANT.ID_GROUPE=2 AND YEAR(GROUPE.DATE_CREATION_GROUPE)=2014  
AND NOTES.ID_MODULE = 'M121';
```

6. Afficher par semestre la moyenne de l'étudiant dont le CNE=2468013579 ;

```
SELECT CNE, SEMESTRE, AVG(NOTE_ETUDIANT)  
FROM NOTES  
WHERE CNE=2468013579  
GROUP BY SEMESTRE, CNE
```

7. Afficher les étudiants dont les noms se terminent par la lettre 'e' ;

```
SELECT * FROM ETUDIANT  
WHERE NOM_ETUDIANT LIKE '%e'
```

8. Afficher les étudiants dont les noms contiennent 'a' ;

```
SELECT * FROM ETUDIANT  
WHERE NOM_ETUDIANT LIKE '%a%'
```

9. Pour chaque étudiant afficher son CNE, NOM, PRENOM et le nombre des modules a passé (Le résultat de la requête doit inclure aussi les étudiants qui n'ont pas passé aucun module)

```
SELECT ETUDIANT.CNE, NOM_ETUDIANT, PRENOM_ETUDIANT, COUNT(NOTES.NOTE_ETUDIANT)  
FROM ETUDIANT LEFT OUTER JOIN NOTES  
ON NOTES.CNE = ETUDIANT.CNE  
GROUP BY ETUDIANT.CNE, NOM_ETUDIANT, PRENOM_ETUDIANT
```

10. Afficher les étudiants du groupe 2 (i.e. IDGROUPE=2) de l'année 2014 ayant des notes, du module (ID\_MODULE='M121'), comprises entre 10 et 20 ;

```
SELECT ETUDIANT.CNE, NOM_ETUDIANT, PRENOM_ETUDIANT, NOTES.NOTE_ETUDIANT  
FROM ETUDIANT, NOTES, GROUPE  
WHERE ETUDIANT.ID_GROUPE=GROUPE.ID_GROUPE  
AND ETUDIANT.CNE =NOTES.CNE  
AND ETUDIANT.ID_GROUPE=2  
AND NOTES.ID_MODULE = 'M121'  
AND NOTE_ETUDIANT BETWEEN 10 AND 20  
AND YEAR(GROUPE.DATE_CREATION_GROUPE)=2014
```

11. Ajouter à tous les étudiants du groupe 2 (i.e. IDGROUPE=2) de l'année 2014 un point à la note du module M121 ;

```
UPDATE NOTES
SET NOTE_ETUDIANT=NOTE_ETUDIANT +1
WHERE ID_MODULE='M121'
AND CNE IN(SELECT CNE
            FROM ETUDIANT
            WHERE ID_GROUPE IN ( SELECT ID_GROUPE
                                FROM GROUPE
                                WHERE
                                YEAR (DATE_CREATION_GROUPE) =2014
                                AND ID_GROUPE=2))
AND NOTE_ETUDIANT <=19
```

12. Afficher les étudiants du groupe 1 de l'année 2014 qui n'ont pas validé le Module M122.

```
SELECT ETUDIANT.CNE,NOM_ETUDIANT,PRENOM_ETUDIANT, NOTES.NOTE_ETUDIANT
FROM ETUDIANT,NOTES,GROUPE
WHERE ETUDIANT.ID_GROUPE=GROUPE.ID_GROUPE
AND ETUDIANT.CNE =NOTES.CNE
AND ETUDIANT.ID_GROUPE=1
AND NOTES.ID_MODULE = 'M122'
AND NOTE_ETUDIANT < 12
AND YEAR(GROUPE.DATE_CREATION_GROUPE)=2014
```