

SQL – DEVOIR CORRIGÉ

A partir de la correction proposée par math93.com

EXERCICE 1: AU LYCEE

Question 1

seconde	Type
num_eleve (clef primaire)	entier (??)
langue1	CHAR
langue2	CHAR
option	CHAR
classe	CHAR

1. Dans le modèle relationnel, quel est l'intérêt de l'attribut num_eleve?

La clé primaire d'une relation est un attribut qui permet de désigner d'une façon unique un enregistrement.

Par exemple l'attribut **num_eleve** permet d'identifier de façon unique les enregistrements de la table (ou relation) *seconde*. La seule connaissance de la clé primaire permet d'identifier toute ligne de la table. Cet identifiant (souvent auto géré) unique est ici associé à un élève, par essence unique dans l'établissement.

2. Écrire une requête SQL d'insertion permettant d'enregistrer l'élève ACHIR Mussa dans la table seconde. Les informations relatives à cet élève sont données dans la ligne 1 du fichier *seconde_lyc.csv*.

Remarque: Un problème ici car la clé primaire proposée n'est pas un entier. Généralement cette clé est automatiquement attribuée par la BDD. On ne va prendre en compte que la partie de la clé composée de chiffres. On ne complète pas la colonne option car l'élève n'en a pas. On suppose évidemment que cet attribut peut être NULL. Cela doit être spécifié lors de la création de la table.

Réponse:

```
INSERT INTO seconde(num_eleve, langue1, langue2, classe)
VALUES (133310, 'anglais', 'espagnol', '2A') ;
```

3. Lors de l'insertion de l'élève ALTMAYER Yohan (ligne 2 du fichier *seconde_lyc.csv*), une erreur de saisie a été commise sur la première langue, qui devrait être allemand. Écrire une requête SQL de mise à jour corrigeant les données de cet élève.

```
UPDATE seconde
SET langue1 = 'allemand'
WHERE num_eleve = 156929 ;
```

Question 2

On suppose maintenant que la table *seconde* contient les informations issues de la figure 1 (ni plus, ni moins, même si la figure 1 n'est qu'un extrait du fichier *seconde_lyc.csv*).

1. Quel est le résultat de la requête **SELECT num_eleve FROM seconde ;** ?

Le résultat de la requête sera la colonne des clefs primaires *num_eleve* :

num_eleve (clef primaire)
133310
156929
...
666702

2. On rappelle qu'en SQL, la fonction d'agrégation COUNT() permet de compter le nombre d'enregistrements dans une table.

Quel est le résultat de la requête SELECT COUNT(num_eleve) FROM seconde ; ?

Le résultat de la requête sera le nombre de lignes (de clefs primaires) de la table seconde donc ici 30.

3. Écrire la requête permettant de connaître le nombre d'élèves qui font allemand en langue1 ou langue2.

```
SELECT COUNT(num_eleve) FROM seconde
WHERE langue1='allemand' OR langue2='allemand' ;
```

Question 3

Le chef d'établissement souhaite faire évoluer la structure de sa base de données. Pour ce faire, il crée une nouvelle table **eleve** dont la structure est la suivante :

eleve	Type
num_eleve (clef primaire, clef étrangère de la table seconde)	entier (??)
nom	CHAR
prenom	CHAR
datenaissance	CHAR

Là encore, l'attribut num_eleve est un entier, les autres sont des chaînes de caractère (le type CHAR).

1. Expliquer ce qu'apporte l'information clef étrangère pour l'attribut num_eleve de cette table en termes d'intégrité et de cohérence.

Les clés étrangères permettent de gérer des relations entre plusieurs tables, et garantissent la cohérence des données. On peut ainsi modifier des données d'un élève sans avoir à modifier plusieurs tables.

2. On suppose la table **eleve** correctement créée et complétée. Le chef d'établissement aimerait lister les élèves (nom, prénom, date de naissance) de la classe 2A. Écrire la commande qui permet d'établir cette liste à l'aide d'une jointure entre **eleve** et **seconde**.

seconde	Type	eleve	Type
num_eleve (clef primaire)	entier (??)	num_eleve (clef primaire, clef étrangère de la table seconde)	entier (??)
langue1	CHAR	nom	CHAR
langue2	CHAR	prenom	CHAR
option	CHAR	datenaissance	CHAR
classe	CHAR		

```
SELECT nom, prenom ,datenaissance
FROM eleve
INNER JOIN seconde
ON seconde.num_eleve = eleve.num_eleve
WHERE seconde.classe='2A' ;
```

Question 4

Proposer la structure d'une table **coordonnees** dans laquelle on pourra indiquer, pour chaque élève, son adresse, son code postal, sa ville, son adresse mail. Préciser la clef primaire et/ou la clé étrangère en vue de la mise en relation avec les autres tables.

coordonnees	Type
num_eleve (clef primaire, clef étrangère de la table seconde)	entier (??)
adresse	CHAR ou TEXT (pour des données plus longues)
codepostal	INT
ville	CHAR
email	CHAR

EXERCICE 2: ORDINATEURS ET IMPRIMANTES

Ordinateur (nom_ordi : String, salle : String, marque_ordi : String, modele_ordi : String, annee : Int, video : Boolean)

Les 5 premières lignes de la relation Ordinateur

nom_ordi	salle	marque_ordi	modele_ordi	annee	video
Gen-24	012	HP	compaq pro 6300	2012	true
Tech-62	114	Lenovo	p300	2015	true
Gen-132	223	Dell	Inspiron Compact	2019	true
Gen-133	223	Dell	Inspiron Compact	2019	false
Gen-134	223	Dell	Inspiron Compact	2019	false

1. a) À l'aide d'un système de gestion de base de données, on envoie au serveur la requête SQL suivante: **SELECT** salle, marque_ordi **FROM** Ordinateur ;
Quel résultat produit cette requête sur l'extrait de la relation Ordinateur donné ci-dessus?

salle	marque_ordi
012	HP
114	Lenovo
223	Dell
223	Dell
223	Dell

On obtient la liste des marques de chaque ordinateur présent dans chaque salle.

b) Quel résultat produit la requête suivante sur l'extrait de la relation Ordinateur donné ci-dessus?
SELECT nom_ordi, salle **FROM** Ordinateur **WHERE** video = true ;

nom_ordi	salle
Gen-24	012
Tech-62	114
Gen-132	223

On obtient la liste des ordinateurs et leur salle associée qui sont connectés à un vidéoprojecteur.

2. Écrire une requête SQL donnant tous les attributs des ordinateurs correspondant aux années supérieures ou égales à 2017 ordonnées par dates croissantes.

Il faut taper la requête SQL ci-dessous :

```
SELECT * FROM ordinateur WHERE annee >= 2017 ORDER BY annee ;
```

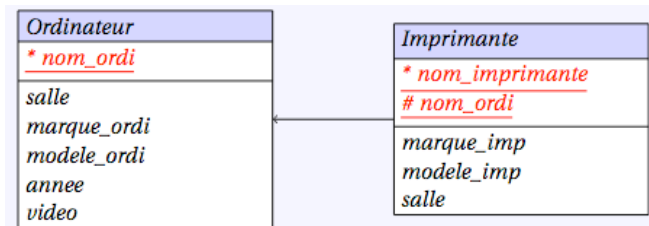
3. a) Pour quelle raison l'attribut salle ne peut-il pas être une clé primaire pour la relation Ordinateur?

Dans une base de données relationnelle, une clé primaire est la donnée qui permet d'identifier de manière unique un enregistrement (une ligne) dans une table. De ce fait, l'attribut salle ne peut pas être une clé primaire pour la relation Ordinateur puisqu'une salle peut contenir plusieurs ordinateurs. Le nom de la salle n'est pas associé à un unique ordinateur, par exemple la salle 223 en contient 3.

b) On prend (nom_imprimante, nom_ordi) comme clé primaire. Écrire le schéma relationnel de la relation Imprimante en précisant les éventuelles clés étrangères pour les autres relations.

nom_imprimante	marque_imp	modele_imp	salle	nom_ordi
imp_BTS_NB	HP	Laserjet pro M15w	114	Tech-62
imp_BTS_Couleur	Canon	Megatank Pixma G5050	114	Tech-62
imp_salle-info1	Brother	2360DN	223	Gen-132
imp_salle-info1	Brother	2360DN	223	Gen-133
imp_salle-info1	Brother	2360DN	223	Gen-134

Réponse:



* : clé primaire
: clé étrangère
La clé primaire est soulignée dans chaque table.

4. Les 4 premières lignes de la relation Videoprojecteur:

salle	marque_video	modele_video	tni
012	Epson	xb27	true
114	Sanyo	PLV-Z3	false
223	Optoma	HD143X	false
225	Optoma	HD143X	true

a) Écrire une requête SQL pour ajouter à la relation Videoprojecteur le vidéoprojecteur nouvellement installé en salle 315 de marque NEC, modèle ME402X et non relié à un TNI.

Il faut écrire la requête suivante :

```
INSERT INTO Videoprojecteur ('salle','marque_video','modele_video','tni')
VALUES ('315','NEC','ME402X',false);
```

Comme on remplit tous les champs, on peut écrire plus simplement :

```
INSERT INTO Videoprojecteur VALUES ('315','NEC','ME402X',false);
```

b) Écrire une requête SQL permettant de récupérer les attributs salle, nom_ordi, marque_video des ordinateurs connectés à un vidéoprojecteur équipé d'un TNI.

On réalise une jointure:

```
SELECT Ordinateur.salle, nom_ordi, marque_video FROM Ordinateur
INNER JOIN Videoprojecteur ON Ordinateur.salle=Videoprojecteur.salle
WHERE Ordinateur.video = true
```

On peut obtenir le même résultat avec une clause WHERE :

```
SELECT Ordinateur.salle, nom_ordi, marque_video
FROM Ordinateur, Videoprojecteur
WHERE Ordinateur.salle=Videoprojecteur.salle
AND Ordinateur.video = true
```