

## 1. CORRECTION D'ERREURS D'INSERTION

a. `INSERT INTO formateur VALUES(1, 'Jacques' , 'Mesrine');`

**Erreur SQL :**

ERROR: duplicate key value violates unique constraint "formateur\_pkey"  
DETAIL: Key (id\_formateur)=(1) already exists.

La clé primaire existe déjà, il faut changer sa valeur :

```
INSERT INTO formateur VALUES(DEFAULT, 'Jacques' , 'Mesrine');
```

L'utilisation du **DEFAULT** génère une valeur auto incrémentée, puisque la colonne `id_formateur` est de de type **SERIAL**

b. `INSERT INTO inscrire VALUES(10, 'X00005',1);`

**Erreur SQL :**

ERROR: insert or update on table "inscrire" violates foreign key constraint "inscrire\_id\_formation\_fkey"  
DETAIL: Key (id\_formation)=(10) is not present in table "formation".

La formation d'identifiant N°10 n'existe pas, violation de la clé étrangère, il faut mettre une valeur dans colonne clé étrangère `id_formation` qui correspond à une valeur existante dans la colonne clé primaire (`id_formation`) de la table `formation` :

```
INSERT INTO inscrire VALUES(42, 'X00005',1);
```

c. `INSERT INTO inscrire VALUES(42, 1234556,1);`

**Erreur SQL :**

ERROR: insert or update on table "inscrire" violates foreign key constraint "inscrire\_id\_stagiaire\_fkey"  
DETAIL: Key (id\_stagiaire)=(1234556) is not present in table "stagiaires".

Le stagiaire d'identifiant N°1234556 n'existe pas, violation de la clé étrangère, il faut mettre une valeur dans colonne clé étrangère `id_stagiaire` qui correspond à une valeur existante dans la colonne clé primaire (`id_stagiaire`) de la table `stagiaire`

```
INSERT INTO inscrire VALUES(42, 13167,1);
```

d. `INSERT INTO planifier VALUES(42, 6 , '2019-01-23', NULL, 2, 1, 'Matin', 'E410');`

**Erreur SQL :**

ERROR: null value in column "groupe" violates not-null constraint  
DETAIL: Failing row contains (42, 6, 2019-01-23, null, 2, 1, Matin, E410).

La colonne `groupe` ne peut être vide (NULL) il faut mettre une valeur dans cette colonne.

```
INSERT INTO planifier VALUES(42, 6 , '2019-01-23', 1, 2, 1, 'Matin', 'E410');
```

## 2. MODIFICATION DE LA DATE D'UNE FORMATION

Toutes les formations intitulées 'Initiation a Linux' sont repoussées de 9 jours

--ajouter 9 jours à la date des formations

```
UPDATE planifier
```

```
SET Dateform = Dateform + INTERVAL '9 DAY'
```

```
WHERE id_formation=(SELECT id_formation
```

```
FROM formation
```

```
WHERE intitule_formation like '%Initiation a Linux%');
```

*La fonction **INTERVAL** permet d'ajouter ou retrancher un intervalle à une date, intervalle en jour (DAY), mois (MONTH), année (YEAR) et bien d'autre, voir la documentation pour connaître l'ensemble des intervalles possibles*

### 3. SUPPRESSION DE LA FORMATION "BASES DE DONNEES (ACCESS)"

La formation intitulée "Bases de données (Access)" est supprimée.

Réaliser les suppressions nécessaires pour laisser la base de données dans un état cohérent.

**Avant de supprimer la formation "Bases de données (Access)" il faut supprimer toutes les lignes des tables inscrire et planifier qui y font références**

```
-- suppression de toutes les inscriptions à cette formation de la table inscrire
DELETE FROM inscrire
WHERE id_formation = (SELECT id_formation
                      FROM formation
                      WHERE intitule_formation Like '%Bases de donnees (Access)%');

-- suppression de la formation du planning (table planifier)
DELETE FROM PLANIFIER
WHERE ID_FORMATION = (SELECT ID_FORMATION
                     FROM formation
                     WHERE intitule_formation Like '%Bases de donnees (Access)%');

-- suppression de la formation de la liste des formations (table formation)
delete from formation
where intitule_formation Like '%Bases de donnees (Access)%' ;
```

### 4. CONVERSION EN MINUSCULE

Mettre en minuscule le nom et le prénom des formateurs.

```
UPDATE formateur
SET nom_formateur = Lower(nom_formateur),
    prenom_formateur =Lower(prenom_formateur);
```

### 5. CONVERSION EN MAJUSCULE

Mettre en majuscule le nom et la première lettre du prénom des formateurs.

```
UPDATE formateur
SET nom_formateur = UPPER(nom_formateur),
    prenom_formateur = initcap(prenom_formateur);
```

-- vérification de la modification

```
SELECT * FROM formateur ;
```

### 6. MODIFICATION DE SALLE

Toutes les séances de formation qui ont lieu dans les 30 jours qui suivent le 8 avril 2006 change de salle, elles se dérouleront toutes en salle G333 (une seule requête permet de réaliser cette modification)  
(une seule requête permet de réaliser cette modification)

```
-- Affiche les formations prévues dans les 30 jours qui suivent le 8 avril 2006
SELECT *,(dateform -date '2006-04-08') as "ecart"
from planifier
where (dateform -date '2006-04-08') between 0 and 30;
```

```
-- modifie les dates de formation
Update planifier
Set numsalle='G333'
where (dateform -date '2006-04-08') between 0 and 30;
```

**ou**

```
Update planifier
Set numsalle='G333'
where dateform between date '2006-04-08' and date '2006-04-08'+ 30;
```

```
-- vérifie que les modifications sont réalisées
SELECT *,(dateform - date '2006-04-08') as "ecart"
from planifier
where (dateform - date '2006-04-08') between 0 and 30;
```

## 7. MODIFICATION DE LA DATE DE LA FORMATION

Les formations qui auront lieu le 139<sup>ème</sup> jour de l'année seront repoussées de 3 jours  
(Une seule requête permet de réaliser cette modification)

```
-- Affiche les formations qui ont lieu le 139ème jour de l'année
SELECT *, EXTRACT(DOY FROM dateform ) as "Numéro du jour de L'année 2006"
from planifier
where EXTRACT(DOY FROM dateform ) =139
AND extract (year from dateform)=2006 ;
```

```
-- repousse de 3 jours les formations qui ont lieu le 139ème jour de l'année
update planifier
set dateform=dateform + INTERVAL '3 day'
where EXTRACT(DOY FROM dateform ) =139
AND extract (year from dateform)=2006 ;
```

```
-- vérifie que les modifications sont réalisées
SELECT *, EXTRACT(DOY FROM dateform )as Num_jour_annee
from planifier
where EXTRACT(DOY FROM dateform ) =142
AND extract (year from dateform)=2006 ;
```

## 8. DEPART D'UN FORMATEUR, QUI SERA REMPLACÉ PAR UN NOUVEAU

- A. Affichez le planning des formations, on verra apparaître (en plus des informations de la table "plannifier") le nom et prénom des formateurs ainsi que l'intitulé des formations qu'ils assurent.

```
SELECT intitule_formation, nom_formateur, prenom_formateur,
to_char(dateform,'DD/MM/YYYY')as "DATE", planifier.*
FROM formation NATURAL JOIN planifier NATURAL JOIN formateur;
```

- B. Le formateur Cancel Christophe est malade, il ne peut donc plus assurer les formations qu'il a en charge. Un nouveau formateur est recruté (Durant Pierre) pour remplacer Cancel Christophe le temps de sa convalescence.  
Vous devez :

- a. Insérer Durant Pierre dans la base données; -- default pour l'auto incrémentation de l'id

```
INSERT INTO formateur
VALUES( default , 'Durant', 'Pierre');
```

- b. Modifier le planning pour que les formations assurées par Cancel Christophe soit assurée par Durant Pierre ;

```
-- toutes les formations assurées par Cancel sont maintenant assurées par Durant, il faut donc changer la valeur de la
-- clé étrangère de la table inscrire
```

```
UPDATE planifier
SET id_formateur = (SELECT id_formateur FROM formateur
WHERE nom_formateur LIKE 'Durant'
AND prenom_formateur LIKE 'Pierre')
WHERE id_formateur =(SELECT id_formateur FROM formateur
WHERE nom_formateur LIKE 'CANCEL'
AND prenom_formateur LIKE 'Christophe') ;
```

- C. Afficher le planning des formations afin vérifier que le changement à bien été effectué

**Reprendre la requête du A**

## 9. AJOUT D'UNE FORMATION ET INSCRIPTION

- A. Ajouter une formation intitulée "Javascript" d'une durée de 12 h niveau Confirme  
 -- insertion de la nouvelle formation, l'identifiant et auto incrémenté default permet d'ajouter 1 à la dernière  
 --valeur de la colonne id\_formation

```
INSERT INTO formation
VALUES ( default, 'Javascript', 12, 'Confirme' )
```

- B. Tous les inscrits à la formation "Excel" niveau "Débutant" désirent participer à la formation précédemment créée. Le groupe de chaque professeur est identique. Réaliser les requêtes qui permettent de satisfaire cette demande.

Conseil : vous pouvez créer une table "tampon" qui sera détruite à la fin

-- recherchons les inscrits à la formation excel niveau debutant

```
SELECT id_formation , id_stagiaire, groupe
FROM inscrire natural join formation
WHERE intitule_formation LIKE '%Formation Excel%'
AND niveau LIKE '%Debutant%' ;
```

L' id\_formation 26 correspond à l'id de la formation Excel, avant d'insérer ces lignes dans la table inscrire, il suffit donc de changer cette id par celui de javascript, nous placerons une sous requête à la place de la colonne id\_formation après le SELECT

```
SELECT (SELECT max(id_formation) from formation) ,
id_stagiaire, groupe
FROM inscrire natural join formation
WHERE intitule_formation LIKE '%Formation Excel%'
AND niveau LIKE '%Debutant%' ;
```

La sous requête `SELECT max(id_formation) from formation` retourne la valeur numérique la plus grande (la dernière entrée)

?column?	id_stagiaire	groupe
4310600		1
4312456		1
4313073		1
4313150		1
4313516		1
4313677		1

Il suffit maintenant d'insérer ces lignes dans la table inscrire

```
INSERT INTO inscrire
SELECT (SELECT max(id_formation) from formation) ,
id_stagiaire, groupe
FROM inscrire natural join formation
WHERE intitule_formation LIKE '%Formation Excel%'
AND niveau LIKE '%Debutant%' ;
```

id_formation	id_stagiaire	groupe
2610600		1
2612456		1
2613073		1
2613150		1
2613516		1
2613677		1
2614186		1
2614487		1
2614899		1
2614918		1
2615118		1
2619362		1
2621435		2
2623248		2
2627053		2
2629239		2
2629428		2
263803		2
2638424		2
2641078		2
2641181		2
2641206		2
264453		2
2644961		1
266564		2
266725		2

26 ligne(s)

## 10. SUPPRESSION DES FORMATIONS NON DISPENSEES AU SEIN DU LYCEE

- A. Afficher toutes les formations (toutes les colonnes) qui ne sont pas assurées au sein du lycée (nombre d'heures égal à 0)

```
SELECT *
from formation
where Nbheures =0;
```

- B. Afficher toutes les colonnes de la table inscrire, mais seulement les lignes qui correspondent aux inscrits aux formations qui ne sont pas assurées au sein du lycée (nombre d'heures égal à 0).

```
SELECT I.*
FROM inscrire I inner join formation F ON F.id_formation=I.id_formation
WHERE nbheures=0;
```

**Ou**

```
SELECT inscrire.*
FROM inscrire natural join formation
WHERE nbheures=0;
```

- C. Vous devez supprimer de la base de données les formations qui ne sont pas assurées au sein du lycée et les inscrits à ces formations, mais avant de les supprimer nous aimerions avoir une sauvegarde de ces données :

- a. En utilisant la syntaxe donnée dans les remarques :

- Créez une table nommée *Formation\_Ext* en y insérant les formations qui ne sont pas assurées au sein du lycée (nombre d'heures égal à 0)

*-- création de la table Formation\_Ext et insertion des formations extérieures*

```
CREATE TABLE Formation_Ext  
AS SELECT * FROM formation WHERE nbheures =0;
```

- Créez une table *Inscrit\_Form\_Ext* en y insérant les inscrits aux formations qui ne sont pas assurées au sein du lycée (nombre d'heures égal à 0)

*-- Création de la sauvegarde des inscriptions aux formations extérieures*

```
CREATE TABLE Inscrit_Form_Ext  
AS SELECT inscrire.*  
FROM inscrire natural join formation  
WHERE nbheures=0;
```

- b. Une fois les sauvegardes réalisées supprimez les données correspondantes, des tables d'origines ;

*-- Suppression des tuples qui correspondent aux inscrits aux formations extérieures*

```
DELETE FROM inscrire  
WHERE id_formation in (SELECT id_formation  
FROM formation  
WHERE nbheures =0);
```

*-- Suppression de 44 lignes*

**Commentaire :** Suppression des tuples de la table "suivreformation" dont l'identifiant est dans la liste (IN) des identifiants des formations qui ne sont pas assurées (Nbheures =0)

*-- Suppression des tuples qui correspondent aux formations non assurées par le lycée*

```
delete from formation  
where nbheures =0;
```

*-- Suppression de 4 lignes*