

Série de TD N 02

Exercice 1:

1. Que donne l'exécution des requêtes SQL suivantes:

CREATE TABLE Segment(indIP VARCHAR(11), nomSegment VARCHAR(20) NOT NULL, etage INTEGER, CONSTRAINT pk_Segment PRIMARY KEY (indIP));	CREATE TABLE Salle(nSalle VARCHAR(7), nomSalle VARCHAR(20) NOT NULL, nbPoste INTEGER, indIP VARCHAR(11), CONSTRAINT pk_salle PRIMARY KEY (nSalle));	CREATE TABLE Types (typeLP VARCHAR(9), nomType VARCHAR(20), CONSTRAINT pk_types PRIMARY KEY(typeLP));
CREATE TABLE Logiciel (nLog VARCHAR(5), nomLog VARCHAR(20) NOT NULL, dateAch DATETIME, versionLog VARCHAR(7), typeLog VARCHAR(9), prix DECIMAL(6,2), CONSTRAINT pk_Logiciel PRIMARY KEY (nLog), CONSTRAINT ck_prix CHECK (prix >= 0));	CREATE TABLE Installer (nPoste VARCHAR(7), nLog VARCHAR(5), numIns INTEGER, dateIns DATE, delai DECIMAL(8,2), CONSTRAINT pk_Installer PRIMARY KEY(numIns));	CREATE TABLE Poste(nPoste VARCHAR(7), nomPoste VARCHAR(20) NOT NULL, indIP VARCHAR(11), ad VARCHAR(3), typePoste VARCHAR(9), nSalle VARCHAR(7), CONSTRAINT pk_Poste PRIMARY KEY (nPoste), CONSTRAINT ck_ad CHECK (ad BETWEEN '000' AND '255'));

2. Qu'est-ce qu'une contrainte d'intégrité?
3. Donner des exemples de contraintes locales et de contraintes globales.
4. Donner la requête pour insérer la ligne ('130.120.80','Brin RDC', NULL) dans la table Segment ?
5. Donner la requête pour insérer les valeurs (nPoste='p2', nLog='log1', numIns=1, dateIns='2017-04-15', delai= NULL) dans la table installer.
6. Mettre à jour etage = 0 dans la table Segment pour les lignes ayant indIP = '130.120.80'.
7. Augmenter le prix des logiciels de type: 'PCNT' de 0.9 (Table logiciel).
8. Ajouter deux colonnes, nbSalle et nbPoste (de type integer), dans la table Segment avec 0 comme valeur par défaut.
9. Ajouter une colonne nbInstall (type : Integer, et valeur par défaut: 0) à la table logiciel.
10. Ajouter une colonne nbLog (type : Integer, et valeur par défaut: 0) à la table Poste.
11. Ajouter une contrainte d'unicité sur les deux colonnes (nPoste,nLog) de la table Installer.
12. Ajouter les contraintes de clés étrangères suivantes :
 - a. indIP (sur la table Poste), qui fait référence au champ indIP de la table Segment.
 - b. nSalle (sur la table Poste), qui fait référence au champ nSalle de la table Salle.
 - c. typePoste (sur la table Poste), qui fait référence au champ typeLP de la table Types.
 - d. nPoste (sur la table Installer), qui fait référence au champ nPoste de la table Poste.
 - e. nLog (sur la table Installer), qui fait référence au champ nLog de la table Logiciel.
13. Supprimer les salles ayant l'indIP qui n'existe pas dans la table segment.
14. Donner le nombre de logiciels installés sur chaque poste, et grouper les résultats par nPoste
15. Donner le nombre de poste par salle.
16. Donner le nombre de poste par logiciel.
17. Donner la moyenne des prix des logiciels de type : UNIX

Série de TD N 02

18. Sélectionner les postes ayant deux (02) logiciels installés.
19. Sélectionner les types de logiciels qui existent dans l'ensemble des types de postes.
20. Sélectionner les salles dans lesquelles le logiciels nomLog = 'Oracle 6' est installé.
21. Sélectionner les noms des logiciels ayant la date d'achat la plus récente.
22. Sélectionner les Noms des segments possédant exactement trois postes de travail de type 'TX'

Exercice 2:

Soit la base de données de l'exercice 1.

1. Créer un index unique sur les colonnes nomLog et versionLog de la table logiciel.
2. Créer une vue *LogicielsUnix* regroupant l'ensemble des logiciels de type : UNIX
3. Créer une vue Poste0 (nPos0, nomPoste0, nSalle0, TypePoste0, indIP, ad0) sur les postes ayant indIP dans un segment de l'étage 0.
4. Sélectionner toutes les lignes de la vue *Poste0*.
5. Insérer les valeurs : ('p15','Bidon15', 's01','UNIX','130.120.80','20') dans la vue *Poste0*.
6. Sélectionner la liste des salles en indiquant : salle vide si nbPoste = 0, salle pleine si nbPoste = 30 et place disponible sinon.

Exercice 3:

1. Accorder le droit de SELECT sur la table GENRE à un utilisateur *user*.
2. Accorder les droits de INSERT et UPDATE sur GENRE à un utilisateur *user*.
3. Accorder tous les droits sur la table DISQUE à un utilisateur *user*.
4. Créer une vue relative au titre, à l'artiste (prénom+nom+groupe) et à l'année des disques. Puis accorder les droits de SELECT sur la vue à un utilisateur *user*
5. Créer une vue relative à toutes les informations (titre de la chanson, titre du disque, artiste du disque, position sur le disque, année du disque) des chansons interprétées par David BOWIE. Puis accorder tous les droits sur la vue à un utilisateur *user*.
6. Retirer le droit INSERT à l'utilisateur *user*, sur la table GENRE.
7. Créer un rôle *DroitEnseignant*
8. Accorder le privilège de création de table et de vue pour ce rôle.
9. Créer un utilisateur *user1* et lui octroyer le rôle *DroitEnseignant*.

Exercice 4:

1. Soit le schéma: produits (np integer, nf integer, coût float, auteur varchar, dateMaj date). Définir un trigger qui positionne les attributs auteur et dateMaj aux valeurs courantes de la transaction pour laquelle il est exécuté (auteur= utilisateur courant, dateMaj = Date courante).
2. Définir un trigger pour créer automatiquement la clé (np) lors de l'insertion d'un tuple dans la table *produits*.
3. Définir un trigger (associé à la table *employe* (*Id integer, Nom varchar(20), salaire float*)) permettant de répercuter les mises à jour du salaire sur la table *cumul* des augmentations, du schéma: *CUMUL (ID int, Augmentation float)*.