

TD1 : Rappel

R3.07 SQL dans un langage de programmation

Vous devez travailler sur 172.26.82.68 *et pdb1* comme base de données.

A partir du schéma "**basetd**" vous allez en créer une, à votre nom, dont vous serez propriétaire et que vous pourrez modifier.

Le schéma 'BASETd' sur laquelle on travaille est formée des tables suivantes:

EMPLOYE : (**nuempl**,nomempl,hebd, affect)

SERVICE : (**nuserv**,nomserv, chef)

PROJET : (**nuproj**,nomproj, resp)

TRAVAIL : (**nuempl**,**nuproj**, duree)

CONCERNE : (**nuserv**,**nuproj**)

Vous créez une copie de ces tables facilement par des commandes du type:

create table nom_table as select * from basetd.employe;

La table *nom_table* est créée avec les domaines (nom et type) d'employe, toutes les lignes de **employe de basetd** y sont insérées. La table *nom_table* est créée dans un schéma dont le nom est votre **username**, elle a pour nom complet *votre_username.nom_table*, vous pouvez donc si vous le voulez l'appeler **employe**.

Pour vous, elle peut être désignée par son nom court *nom_table*(par exemple **employe**) (elle est dans "votre schéma").

Vous pouvez voir vos tables dans l'arborescence de *SQLdeveloper (Tables)*

Vous constaterez que les contraintes **not null** (qui ne sont pas la conséquence de **primary key**) sont conservées, ce sont les seules.

1 – Modifiez la description de vos tables pour ajouter des contraintes d'intégrité (clés **primaires** et clés **étrangères**) de sorte que l'ensemble de données ainsi décrit soit conforme à la spécification. Vous pouvez les décrire toutes.

Exemples :

alter table employe add constraint **PK_employe** primary key (nuempl) ;

....

alter table employe add constraint **FK_affect** foreign key (**affect**) references service (**nuserv**) ;

Pensez aussi à donner des **noms mnémoniques** aux contraintes. Exemple le nom de la contrainte associée à la clé primaire ou clé étrangère est préfixé par **PK** ou **FK**. Exemples: *PK_employe*, *FK_Affect*, etc...

Complétez le tableau suivant avec les différentes contraintes traitées:

Nom table	Nom contrainte	Type de contrainte	Différée ou non
Employe	PK_Employe	Clé primaire	non

Vérifiez que tout cela fonctionne bien en faisant des insertions/suppressions/modifications interdites

et/ou autorisées. Rappelez vous que l'utilisation "judicieuse" de **rollback** permet, si vous le voulez, de conserver les tables dans l'état initial. Rappelez vous aussi que l'utilisation d'une commande qui modifie le dictionnaire de données est provoqué un **commit**.

Pour chacune des contraintes PK et FK, provoquez une erreur et complétez le tableau ci-dessous : Les numéros d'erreur sont données sous la forme **ORA-0001**

Attention les **PK** sont vérifiées à partir des **insertions** et les **FK** sont vérifiées à partir d'un **insert** et d'un **delete**.

Exemple la contrainte **FK_affect** :

test1 : une **insertion d'employé** avec une valeur de l'attribut *affect* qui n'existe pas dans la table service,

test2 : un **delete d'un service** avec des enregistrements de la table employe qui sont liés à ce service. Suppression du service 1 avec des employés qui sont affectés à 1.

Nom table	requête	Code Erreur	Message d'erreur
PK_Employe	–insertion d'un employé qui existe déjà Insert into employe values(20,'toto',35,1) ;	'-0001'	Violation de la contrainte

Décrivez et complétez les tableaux ci-dessus avec les contraintes suivantes :

- Le chef d'un service est un employé affecté au même service. Attention les contraintes de clé étrangère **Affect** et **Chef** se **croisent**. Dans ce cas, vous devriez créer une contrainte unique (**nuempl, affect**) et une contrainte **différée(initially deferred)** de la table service (**chef,nuserv**) vers cette contrainte unique(**nuempl, affect**).
- Un responsable de projet doit travailler sur le projet. Dans ce cas vous n'avez pas besoin de créer la contrainte **Unique** (nuempl, nuproj) de la table travail, car les deux attributs **forment déjà** la clé primaire. Par contre la contrainte (resp, nuproj) est une clé étrangère différée vers la clé primaire de la table *travail*.
- La durée hebdomadaire d'un employé est inférieure ou égale à 35h.

Attention, les contraintes différées sont vérifiées seulement au moment du **commit**. Si le commit échoue, les insertions sont annulées par un **rollback**.

2 – Mise à jour des données:

a) La table employé contient un attribut salaire de type *number*.

Faites les modifications suivantes, **à chaque fois avec une seule requête**:

- Les responsables de projet gagnent au moins 2500 Euros,
- Les chefs de service gagnent au moins 3500 euros.

b) Certaines données ne respectent pas les contraintes suivantes :

- La somme des durées d'un employé (de la table travail) doit être inférieur à la durée hebdomadaire (Sum(duree) <= hebdo).

Affichez les employés qui ne respectent pas cette contrainte avec la requête suivante(à compléter):

Select * from employe e where

Vous modifiez les données de la table **travail**(*update sur la durée*) ou la table **employé** (*update sur hebdo*) jusqu'à ce que la requête précédente retourne un ensemble vide.

- Un employé ne peut être responsable de plus de 3 projets. Idem que la contrainte précédente, trouvez les employés qui ne respectent pas la contrainte et modifiez les données.

- Le **chef d'un service gagne plus que les employés du service**. Idem que la contrainte précédente, trouvez les employés qui ne respectent pas la contrainte et modifiez les données.

- Un **service** ne peut être concerné par plus de 3 projets.

Ces différentes requêtes seront utilisées lors du développement des Trigger.