

Contraintes d'intégrité et dictionnaire de données - TP2

Nettoyage de son espace de travail ou son compte

1° Supprimer les tables et les vues de votre compte, si par mégarde il en restait encore. Vérifier le résultat en utilisant les éléments du dictionnaire de données **USER_TABLES**, **USER_VIEWS** et plus généralement **USER_OBJECTS** définissant respectivement les tables, les vues et les objets de votre compte.

On rappelle qu'une vue ou une table a souvent beaucoup de colonnes, et que son contenu est alors illisible s'il est constitué de plus d'une ligne. En conséquence on projettera une vue ou une table en choisissant une ou quelques colonnes seulement : on obtiendra les noms des colonnes en sélectionnant une seule ligne de la table ou vue, ou en utilisant la commande `DESC table/vue` de l'éditeur d'Oracle SQL*Plus (- SQL).

Nombre d'objets accessibles à son compte, notamment ceux du dictionnaire de données et du compte (ou de la base de données) *Cirque*

2° Visualiser le nombre des objets qui vous sont accessibles en utilisant la vue **ALL_OBJECTS**, puis le nombre des objets pour chaque type d'objet (les valeurs de l'attribut **OBJECT_TYPE**), puis sélectionner le nombre des objets du dictionnaire de données (de propriétaire SYS) et le nombre des objets par type d'objet de ce dictionnaire des données, enfin le nombre des objets par type du compte *Cirque* (ou **CIRQUE**).

Visualiser les tables et vues de *Cirque* à partir de **ALL_TABLES** et **ALL_VIEWS**.

NB : pour avoir des tables lisibles dans cette question et les suivantes, on diminuera la largeur de leurs colonnes affichées à l'écran en utilisant l'instruction `SQL*Plus COL[UMN]`, notamment :

```
COL OWNER FORMAT A20
COL TABLE_NAME FORMAT A15
COL CONSTRAINT_NAME FORMAT A25
COL R_CONSTRAINT_NAME FORMAT A20
COL SEARCH_CONDITION FORMAT A30
COL COLUMN_NAME FORMAT A18
```

et on augmentera la taille des lignes affichées à l'écran à 120 par exemple (*Options/Environnement/linesize*).

Lister les contraintes d'intégrité visibles de la base de données *Cirque*

3° Pour cela on utilisera les vues du dictionnaire de données

```
ALL_CONSTRAINTS(OWNER TABLE_NAME CONSTRAINT_NAME CONSTRAINT_TYPE[1]
R_CONSTRAINT_NAME[2] DELETE_RULE[3] SEARCH_CONDITION[4]) et
ALL_CONS_COLUMNS(OWNER TABLE_NAME CONSTRAINT_NAME COLUMN_NAME
POSITION[5]).
```

En quoi l'attribut **POSITION** de la vue **ALL_CONS_COLUMNS** est utile ?

4° Lister les contraintes d'intégrité **NOT NULL** obtenues par la commande `"DESC[RIBE] Table/Vue"`. Retrouve-t-on les contraintes affichées dans la question précédente ?

5° On ne voit pas beaucoup de contraintes d'intégrité concernant la vue *Cirque.Accessoires* et les tables *Cirque.AccessoiresI* et *Cirque.Rangement*. Pourquoi ?

- en sachant que les tables *Cirque.AccessoiresI* et *Cirque.Rangement*, à l'origine de la vue *Cirque.Accessoires*, avaient des contraintes NOT NULL sur *Accessoire*, *Couleur* et *NoCamion*

Copie de la base de données *Cirque*

6° Créer dans votre compte les tables *Personnel*, *Numéros*, *Accessoires*, *Utilisation* à partir respectivement des tables ou vue *Personnel*, *Numéros*, *Accessoires*, *Utilisation* du compte *Cirque*.

7° Que constater sur la vue **USER_CONSTRAINTS**, sachant notamment que les tables *Cirque.AccessoiresI* sur *Couleur* et *Cirque.Rangement* sur *NoCamion* avaient des contraintes de type NOT NULL ?

Contraintes d'intégrité de type CHECK implantant des contraintes d'intégrité NOT NULL

8° Essayer de créer une nouvelle contrainte d'intégrité de type NOT NULL sur l'attribut *NoCamion* de votre table *Accessoires*, puis la contrainte d'intégrité CHECK(*NoCamion* IS NOT NULL), puis une autre contrainte d'intégrité de type CHECK CHECK(*NoCamion* IS NOT NULL), enfin la contrainte d'intégrité CHECK(*NoRatelier* IS NOT NULL). Conclure et supprimer les trois contraintes qui viennent d'être créées.

Reconstruction des contraintes d'intégrité non recopiées

9° Recréer (avec des noms significatifs bien sûr) dans votre schéma, les contraintes d'intégrité trouvées dans la question 3 dans le schéma *Cirque* :

PRIMARY KEY de *Personnel*, *Numéros*, *Utilisation*, et contraintes d'intégrité référentielles *Utilisateur* \hat{I}_1 *Nom*, *Responsable* \hat{I}_2 *Nom* et *Utilisation.TitreDeNumero* \hat{I}_3 *Numéros.TitreDeNumero* sans qu'il soit nécessaire d'explicitier les membres droits puisque la PRIMARY KEY est spécifiée par défaut.

Que constate-t-on à partir des vues **USER_INDEXES** et **USER_IND_COLUMNS** ?

Création d'une contrainte d'intégrité référentielle supplémentaire

10° Essayer de créer la contrainte d'intégrité référentielle *Utilisation.Accessoire* \hat{I}_4 *Accessoires.Accessoire*. Créer une clé UNIQUE sur *Accessoires.Accessoire*.

11° Essayer à nouveau de créer la contrainte d'intégrité référentielle \hat{I}_4 .

Implanter avec la clause **DISABLE** infirmant \hat{I}_4 (elle pourra être activée ultérieurement par la clause **ENABLE**). Une autre façon de l'implanter est de remplacer **DISABLE** par **ENABLE NOVALIDATE**.

Que conclure à partir des questions 10 et 11 ?

12° Pour activer la contrainte et pour qu'elle soit complètement vérifiée, on opérera ainsi :

d'abord construire pour lister les violations de la contrainte la table *Exceptions*(ROW_ID ROWID,OWNER VARCHAR2 (30), TABLE_NAME VARCHAR2(30), CONSTRAINT VARCHAR2(30)), puis utiliser la commande ALTER TABLE *Utilisation* ADD CONSTRAINT *rUtilisation_Access* FOREIGN KEY(*Accessoire*) REFERENCES *Accessoires*(*Accessoire*) EXCEPTIONS INTO *Exceptions*, enfin déduire de la table *Exceptions* les lignes de la table référençante violant la contrainte désirée (on n'en trouvera qu'une).

13° Insérer dans *Accessoires* la nouvelle ligne ('étrier', NULL, NULL,0.2, NULL). Est-ce possible? Insérer dans *Accessoires* ('étrier', ' ', NULL, 0.2, NULL). Est-ce possible? Finalement insérer ('étrier', ' ', NULL, 0.2, 0) et créer la contrainte d'intégrité référentielle Í4.

Contrôle par le S.G.B.D. des contraintes d'intégrité de type UNIQUE, PRIMARY KEY ou REFERENCES

14° Essayer de définir une nouvelle contrainte d'intégrité UNIQUE sur *Accessoires.Accessoire*, puis une nouvelle CI PRIMARY KEY sur *Accessoires.Accessoire*, enfin une nouvelle contrainte d'intégrité référentielle Í4. Remplacer la contrainte d'intégrité UNIQUE sur *Accessoires.Accessoire* par une PRIMARY KEY.

15° Renforcer la contrainte d'intégrité NOT NULL sur *Couleur* en créant une contrainte d'intégrité supplémentaire interdisant que le premier caractère soit un blanc. Pour cela on utilisera la fonction SUBSTR(*Couleur*, 1, 1) sélectionnant le premier caractère de *Couleur* et on modifiera la ligne *étrier* en ('étrier', 'or', NULL, 0.2, 0). Afficher les lignes de la table *Accessoires* de volume NULL.

Tous les musiciens ont démissionné du cirque. Peut-on modifier les lignes correspondantes de *Personnel*, en gardant le champ *musicien*? Non à cause de la PRIMARY KEY sur *Personnel.Nom*. La recherche de deux musiciens sera l'objet d'une petite annonce d'emploi. Comment relâcher cette PRIMARY KEY de façon à pouvoir effectuer la modification, tout en interdisant que deux employés ne portent pas le même nom? Appliquer cette contrainte.

16° Essayer de supprimer dans *Accessoires* la ligne *étrier* référencée.

17° On adjoindra à la contrainte d'intégrité référentielle Í4 l'option ON DELETE CASCADE (pour cela détruire Í4, puis créer une nouvelle Í4 avec l'option ON DELETE CASCADE). Afficher l'attribut DELETE_RULE et supprimer dans *Accessoires* la ligne *étrier*. Constater les suppressions à partir des questions 16 et 17

Contraintes d'intégrité et vues

18° Au lieu d'utiliser la table *Accessoires*, on crée la vue *VueAccessoires* (Pourquoi *VueAccessoires* au lieu de *Accessoires*? Au niveau interne, quelle différence entre *Accessoires* et *VueAccessoires*?).

Essayer de créer la contrainte d'intégrité référentielle *Utilisation.Accessoire* Í4 *VueAccessoires.Accessoire*. Expliquer le résultat.

19° N'oubliez pas d'éliminer tout objet de votre compte.

[1] Définition du type de contrainte : C pour CHECK, P pour PRIMARY, U pour UNIQUE, R pour une contrainte d'intégrité référentielle, V pour une vue WITH CHECK OPTION SQL 4-443, O pour une vue WITH READ ONLY.

[2] Nom de la contrainte UNIQUE ou PRIMARY KEY pour la table référencée d'une contrainte d'intégrité référentielle.

[3] Règle de suppression pour une contrainte référentielle : CASCADE ou NO ACTION (voir question 17).

[4] Condition d'une contrainte de type CHECK.

[5] Position de la colonne dans la définition de la table.