

III) SQL avancé :

- 1) Etablissez le nombre d'emprunts par ouvrage et par exemplaire. Utilisez l'opérateur ROLLUP pour effectuer le calcul d'agrégat sur les critères de regroupement plus généraux. Utilisez la fonction DECODE pour présenter le résultat de façon plus lisible.
- 2) Etablissez la liste des exemplaires qui n'ont jamais été empruntés au cours des trois derniers mois. Pour effectuer les calculs sur les trois derniers mois, c'est la date de retour de l'exemplaire qui est prise en compte.
- 3) Etablissez la liste des ouvrages pour lesquels il n'existe pas d'exemplaires à l'état neuf.
- 4) Extrayez tous les titres qui contiennent le mot « mer » quelque soit sa place dans le titre et la casse avec laquelle il est renseigné.
- 5) Ecrivez une requête qui permet de connaître tous les auteurs dont le nom possède la particule « de ».
- 6) A partir des genres des livres, affichez le public de chaque ouvrage en vous appuyant sur la table des correspondances ci-dessous. L'objectif est de connaître pour chaque titre le public susceptible de lire l'ouvrage. L'instruction CASE peut s'avérer utile pour aboutir rapidement à un tel résultat.

Genre	Public
Bande Dessinée	Jeunesse
Informatique	Professionnel
Policier	Adulte
Récit	Tous
Roman	Tous
Théâtre	Tous

- 7) Pour l'instant, l'objectif de chaque table semble évident. Mais d'ici quelque temps ce ne sera peut-être plus le cas. Aussi est-il judicieux d'associer un commentaire à chaque table, voire à chaque colonne.

Posez les commentaires pour les différentes tables en vous appuyant sur le tableau suivant :

Table	Commentaire
Membres	Descriptifs des membres. Possède le synonymes Abonnes
Genres	Descriptifs des genres possibles des ouvrages
Ouvrages	Descriptifs des ouvrages référencés par la bibliothèque
Exemplaires	Définition précise des livres présents dans la bibliothèque
Emprunts	Fiche d'emprunt de livres, toujours associée à un et un seul membre
Details	Chaque ligne correspond à un livre emprunté

8) Interrogez les commentaires associés aux tables présentes dans le schéma de l'utilisateur courant. La table USER_TAB_COMMENTS du dictionnaire doit être mise à contribution.

9) Lors de la création d'un nouveau membre, on souhaite enregistrer un emprunt dans la même transaction. Comment rendre possible cette nouvelle contrainte de fonctionnement ?

10) Supprimez la table des détails.

11) Annulez cette suppression de table.

13) Les utilisateurs souhaitent une requête qui permette d'afficher un message en fonction du nombre d'exemplaires de chaque ouvrage.

Nombre d'exemplaires	Message
0	Aucun
Moins de 2	Peu
Moins de 5	Normal
Plus de 5	Beaucoup

IV) PL/SQL :

1) Mise à jour conditionnelle : tous les exemplaires ont été enregistrés avec l'état neuf, mais au fur et à mesure des emprunts, leur état s'est dégradé. Il s'agit maintenant de mettre à jour l'état de chacun en fonction du nombre de fois que l'exemplaire a été emprunté. En effet, le nombre d'emprunteurs a plus d'incidence sur l'état général de l'exemplaire que la durée effective des emprunts.

La mise à jour des informations est donc effectuée à l'aide du tableau suivant :

Nombre d'emprunts	Etat
Moins de 10	Neuf
De 11 à 25	Bon
DE 26 à 40	Moyen
Plus de 41	Mauvais

2) Ecrivez un bloc PL/SQL qui permet de supprimer les membres dont l'adhésion a expiré depuis plus de 2 ans.

Si des fiches d'emprunts existent et si les exemplaires empruntés ont été rendus, alors mettre à NULL la valeur présente dans la colonne MEMBRE.

S'il reste des livres empruntés et non rendus, alors ne pas supprimer le membre.

3) Ecrire un bloc PL/SQL qui permet d'éditer la liste des trois membres qui ont emprunté le plus d'ouvrages au cours des dix derniers mois et établissez également la liste des trois membres qui ont emprunté moins.

4) Ecrivez un bloc PL/SQL qui permet de connaître les cinq ouvrages les plus empruntés.

5) Etablissez la liste des membres dont l'adhésion a expiré, ou bien qui va expirer dans les 30 prochains jours. Affichez la liste à l'écran.

6) Les exemplaires sont tous achetés à l'état neuf. Pour calculer leur état actuel, il faut être capable de connaître le nombre de fois où ils ont été empruntés. Mais les membres sont nombreux et il est impossible de conserver de nombreuses années en ligne tout ce qui concerne le détail des locations.

Un exemplaire est considéré comme emprunté à partir du moment où il est présent sur une fiche d'emprunt. C'est donc la date de création de la fiche qui permet de savoir quand le livre a été emprunté.

Au niveau des exemplaires, une colonne de type date va être ajoutée afin de connaître la date du dernier calcul de mise à jour de l'état. Lors de l'exécution du bloc PL/SQL, seuls les emprunts effectués, depuis cette date, seront pris en compte. Afin que la mise à jour de l'état de l'exemplaire soit effectuée de la façon la plus juste, une seconde colonne va être ajoutée afin de mémoriser le nombre d'emprunts pour cet exemplaire.

a) Ecrivez un script pour effectuer les modifications de structure demandées. La nouvelle structure de la table des exemplaires est :

ISBN	not null	Char (10)
NUMERO	not null	Number(3)

ETAT	Char(2)
NOMBREEMPRUNTS	Number(3)
DATECALCULEMPRUNTS	Date.

Pour chaque exemplaire, la valeur par défaut au moment de la création dans la colonne DATECALCULEMPRUNTS doit correspondre à la date de premier emprunt de cet exemplaire par l'un des membres, ou bien la date du jour si cet exemplaire n'a pas encore été emprunté.

b) Ecrivez le bloc PL/SQL qui permet de mettre à jour les informations sur la table des exemplaires.

7) Si plus de la moitié des exemplaires sont dans l'état Moyen ou Mauvais alors modifiez la contrainte d'intégrité afin que les différents états possibles d'un exemplaire soient : Neuf, Bon, Moyen, Douteux ou Mauvais.

Un exemplaire est dans l'état Douteux lorsqu'il a été emprunté entre 40 et 60 fois. Il est dans l'état Mauvais lorsqu'il a été emprunté plus de 60 fois.

8) Supprimez tous les membres qui n'ont pas effectué d'emprunt d'ouvrage depuis trois ans.

9) Comme cela a été constaté précédemment, les membres possèdent tous un numéro de téléphone mobile mais ce numéro n'est pas bien formaté et la nouvelle contrainte d'intégrité ne peut être posée.

Ecrivez un bloc PL/SQL qui permet de s'assurer que tous les numéros de téléphone mobile des membres respectent le format 06 xx xx xx xx. Puis posez une contrainte d'intégrité pour vous assurer que tous les numéros posséderont toujours ce format.

V) PL/SQL Procédures et fonctions :

1) Ecrire la fonction **FinValidite** qui calcule la date de fin de validité de l'adhésion d'un membre dont le numéro est passé en paramètre.

2) Ecrire la fonction **AdhesionAjour** qui retourne une valeur booléenne afin de savoir si un membre peut ou non effectuer des locations.

3) Ecrire la procédure **RetourExemplaire** qui accepte en paramètres un numéro d'ISBN et un numéro d'exemplaire afin d'enregistrer la restitution de l'exemplaire de l'ouvrage emprunté.

4) Ecrire la procédure **PurgeMembres** qui permet de supprimer tous les membres dont l'adhésion n'a pas été renouvelée depuis trois ans.

5) Ecrire la fonction **MesureActivite** qui permet de connaître le numéro du membre qui a emprunté le plus d'ouvrage pendant une période de temps passée en paramètre de la fonction. Cette période est exprimée en mois.

- 6) Ecrire la fonction **EmpruntMoyen** qui accepte en paramètre d'entrée le numéro d'un membre et qui retourne la durée moyenne (en nombre de jours) d'emprunt d'un ouvrage.
- 7) Ecrire la fonction **DureeMoyenne** qui accepte en paramètre un numéro d'ISBN et éventuellement un numéro d'exemplaire et qui retourne, soit la durée moyenne d'emprunt de l'ouvrage (seul le numéro ISBN est connu), soit la durée moyenne d'emprunt de l'exemplaire dans le cas où l'on connaît le numéro d'ISBN et le numéro de l'exemplaire.
- 8) Ecrire la procédure **MajEtatExemplaire** pour mettre à jour l'état des exemplaires et planifier l'exécution de cette procédure toutes les deux semaines.
- 9) Au cours des questions précédentes, la séquence **Seq_Membre** a été définie et est utilisée pour l'ajout d'informations dans la table des membres. Pour faciliter le travail avec cette séquence, il est judicieux de créer la fonction **AjouteMembre**, qui accepte en paramètre les différentes valeurs de chacune des colonnes et qui retourne le numéro de séquence attribué à la ligne d'information nouvellement ajoutée dans la table.
- 10) Ecrire la procédure **SupprimeExemplaire** qui accepte en paramètre l'identification complète d'un exemplaire (ISBN et numéro d'exemplaire) et supprime celui-ci s'il n'est pas emprunté.
- 11) Le plus souvent, les membres n'empruntent qu'un seul ouvrage. Ecrire la procédure **EmpruntExpress** qui accepte en paramètre le numéro du membre et l'identification exacte de l'exemplaire emprunté (ISBN et numéro). La procédure ajoute automatiquement une ligne dans la table des emprunts et une ligne dans la table des détails.
- 12) Regrouper l'ensemble des procédures et des fonctions définies au sein du package **Livre**.

VI) Déclencheurs de bases de données :

- 1) Mettre à jour un déclencheur de base de données afin de vous assurer que lors de la suppression du dernier exemplaire d'un ouvrage, les informations relatives à l'ouvrage sont également supprimées.
- 2) Définir un déclencheur de base de données permettant de garantir que les emprunts sont réalisés uniquement par des membres à jour de leur cotisation.
- 3) Définir un déclencheur qui interdit le changement de membre pour une fiche de location déjà enregistrée.
- 4) Définir un déclencheur qui interdit de modifier la référence d'un ouvrage emprunté, il faut le rendre puis effectuer une nouvelle location.
- 5) Définir un déclencheur qui met automatiquement à jour l'état d'un exemplaire en fonction de la valeur enregistrée dans **NombreEmprunts**. Par exemple, lors de la mise à jour de valeurs représentant le nombre d'emprunts pour un exemplaire, l'état est mis à jour de façon automatique.

6) Lors de la suppression d'un détail, assurer que l'emprunt a bien été pris en compte au niveau de l'exemplaire.

7) Afin d'améliorer le service rendu aux membres, il est souhaitable de savoir quand l'emprunt d'un ouvrage a été enregistré et quel employé a effectué l'opération. Le même genre d'informations doit être disponible pour le retour des exemplaires.

Définir le code nécessaire pour prendre en compte cette nouvelle exigence. Apporter des modifications de structures si nécessaire.

8) Ecrire la fonction **AnalyseActivite** qui accepte en paramètres le nom d'un utilisateur Oracle et une date et calcule le nombre d'opérations (emprunts et détails) réalisées par l'utilisateur, ou bien sur la journée, ou bien pour l'utilisateur sur la journée. La valeur de cette fonction est toujours un nombre entier.

9) Si tous les exemplaires référencés sur une fiche ont été rendus, alors interdire tout nouvel ajout de détails.

Bibliographie :

1. Oracle 10g , Entraînez-vous sur SQL et PL/SQL, Collection « Les TP Informatiques, Editions ENI.
2. Oracle 10g PL/SQL , TechNote Collection, Jérôme GABILLAUD, Editions ENI.
3. Oracle
4. Oracle 9i SQL et PL/SQL, Eyrolles (Collection Tsoft), Razvan BIZOI.
5. Oracle 10g SQL , PL/SQL, SQL*Plus, Jérôme GABILLAUD, Collection Ressources Informatiques, Editions ENI.