



Université F. Rabelais de Tours

Département informatique

Master 1

Bases de Données Avancées

TRAVAUX PRATIQUES

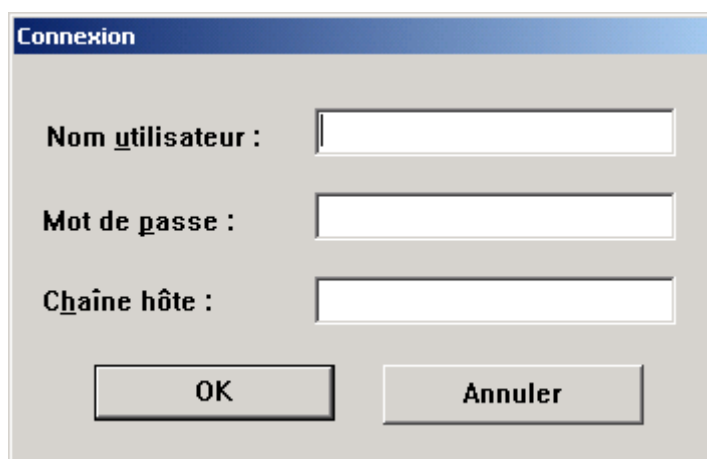
Enseignants **Jean-Yves ANTOINE**

Filière **Master 1 SIAD**

Connexion sur le SGBD Oracle (installation IUP Blois)

Travail sous l'utilitaire SQLplus

1. Logez vous sous votre compte étudiant habituel
2. Connectez-vous sous l'utilitaire SQL+ d'Oracle à partir de la barre Démarrer : menu Programmes → Oracle OraClient → Development Application → SQLPlus



Nom utilisateur votre nom de compte
Mot de passe votre nom de compte
Chaîne hôte db001_Oracle

- Il existe une interface graphique (SQLPlus Worksheet) pour utiliser Oracle de manière plus conviviale. Vous pouvez également l'utiliser..

Travail sous l'utilitaire SQL*Loader

- Ouvrir une fenêtre de commande DOS sous Windows (utilitaire cmd)
- Lancer la commande de lancement du SQL*Loader : sqlldr
Syntaxe : sqlldr userid=nom_userBD/password_userBD@chaine_hote fic_ctl
- Rappel : chaîne hôte à l'IUP = db001_Oracle

TP 1 & 2 : Etude du dictionnaire de données Oracle

1. Présentation

ORACLE dispose d'une structure centralisée (le **dictionnaire de données**) contenant la description de tous les objets (tables, vues, utilisateurs, ...) gérés par ce SGBD. Ce dictionnaire regroupe donc toutes les informations nécessaires au fonctionnement du SGBD. Il présente la particularité d'être lui-même organisé comme une base de données (on parle de **meta-base**) et d'être par conséquent accessible directement à travers SQL. Pour des raisons d'intégrité aisément compréhensibles, la plupart des tables du dictionnaire ne sont pas modifiables par les utilisateurs (seul l'administrateur de la base SYS, a accès à ces dernières). A l'opposé, l'accès à leur contenu est généralement autorisé, ce qui va nous permettre d'explorer quelque peu le dictionnaire afin de mieux appréhender cette structure de donnée essentielle au SGBD.

D'une manière plus précise, les tables du dictionnaire sont généralement cryptées. Leur contenu est par contre accessible par l'intermédiaire de vues. Elles sont de quatre grands types (cf. document annexe) :

- Vues relatives aux objets d'un utilisateur USER_*
- Vues relatives aux objets accessibles à un utilisateur ALL_*
- Vues relatives aux administrateurs DBA_*
- Vues relatives aux suivi des performances V\$*

Au cours de ce TP, nous allons nous intéresser plus particulièrement aux vues de type ALL et USER. Pour vous guider dans vos travaux, vous pouvez consulter la documentation Oracle disponible sur le serveur EMERY.

2. Travail demandé

Ce TP ne donne pas lieu à compte-rendu. Par contre, la suite des questions que vous avez à faire vous sera utile pour le mini-projet « dictionnaire de données » qui sera à rendre.

3. Organisation générale du dictionnaire

Dans un premier temps, nous allons étudier l'organisation du dictionnaire de données Oracle. L'ensemble des tables du dictionnaire de données est recensé dans la table principale DICT.

3.1. Vues relatives aux objets accessibles à tous les utilisateurs

- 1 — Donner le schéma de la relation correspondant à DICT et préciser le rôle de chacun de ses attributs. On utilisera pour cela la commande SQL de description `DESCRIBE`.
- 2 — Visualiser le contenu complet du dictionnaire DICT à l'aide de la commande de sélection `SELECT`.
- 3 — Donner le rôle et la structure des tables (ou vues) suivantes : `ALL_CATALOG`, `ALL_USERS`, `ALL_TAB_COMMENTS`, `ALL_COL_COMMENTS`, et `ALL_TAB_PRIVS`, `ALL_TRIGGERS`. En vous aidant le cas échéant du contenu de la table `ALL_COL_COMMENTS` ou de la documentation Oracle; donner succinctement le rôle de chaque attribut de ces tables.
- 4 — Donner le rôle et la structure des tables (ou vues) suivantes : `ALL_CONSTRAINTS`, `ALL_CONS_COLUMNS`. Quels sont les types de contraintes reconnues par Oracle ? Savez-vous expliquer comment sont décrites les contraintes d'intégrité référentielles dans `ALL_CONSTRAINTS` ,.
- 5 — Quels sont les différents types d'objets standards (attribut `TABLE_TYPE`) reconnus par Oracle dans son catalogue ? A quelles vues du dictionnaire correspondent ces différents types ? Les vues concrètes sont-elles considérées comme des objets de base ?
- 6 — Combien y-a-t-il d'objets référencés dans la table `ALL_CATALOG`. Comparer ce résultat avec la somme du nombre d'enregistrements dans les tables `ALL_TABLES`, `ALL_VIEWS`, `ALL_SYNONYMS`, `ALL_SEQUENCES`. Observez-vous une différence ? Les vues concrètes peuvent-elle expliquer

cette différence ?

- 7 — Y-a-t-il des triggers déjà définis parmi les bases qui vous sont accessibles ? Si oui, quel est le type de déclenchement ? Allez voir également leur définition : les actions sont elles des ordres SQL ... ou un langage plus évolué (PL/SQL pour ne pas le nommer) ? Vous étudierez ultérieurement ce langage.

3.2. Vues relatives aux objets d'un utilisateur

- 1 — Donner le rôle et le schéma de relation correspondant à la table `USER_USERS`. Quel est votre nom d'utilisateur Oracle ?
- 2 — Comparer le contenu des tables `ALL_CATALOG` et `USER_CATALOG`. Commentaires ?

4. Organisation générale du dictionnaire

Nous allons maintenant utiliser le dictionnaire de données pour retrouver la structure d'une base de données existante.

- 1 — Existe-t-il des objets de nom `SONDAGES` dans la base ? Quel est leur type ?
- 2 — Nous considérerons maintenant uniquement l'objet de type table. Quel est le propriétaire de la table `SONDAGES` ?
- 3 — Chercher l'ensemble des tables ayant même propriétaire que `SONDAGES` : vous devriez récupérer, entre autres, l'ensemble des tables de la base de données que nous étudions.
- 4 — Quelles sont les contraintes d'intégrité (`CONSTRAINT_NAME`) associées à la table `SONDAGES`. Quel est le type (clé primaire, contrainte d'intégrité référentielle...) de chacune d'elles. Pour les contraintes d'intégrité référentielle, donner le nom de la contrainte correspondante (`R_CONSTRAINT_NAME`) dans la table référence.
- 5 — En analysant de même les contraintes associées aux autres tables, retrouver la structure de la base de données (schémas de relation, clés, intégrités référentielles) formée par les tables trouvées à la question 3.
- 6 — Y a-t-il des contraintes d'autre types associés aux tables de cette base de données ? Listez-les. Sont-elles activées ou désactivées ?

5. Synonymes

- 1 — Renommez indirectement la table `SONDAGES` en lui créant un synonyme privé du nom de votre choix. Vérifiez dans le dictionnaire de données que ce synonyme a bien été créé.

TP 3, 4 & 5 : Création BD Oracle, SQL*Loader, Transactions

1. Présentation

Plus fort que de savoir qui de Zidane ou Materazzi était le plus fautif en finale de Coupe du Monde de football, ce qui a passionné le microcosme journalistique en 2007 était l'affrontement entre Ségolène Royal et Nicolas Sarkozy pour devenir calife à la place du calife (l'avez-vous oublié ? Jacques Chirac était président de la République, si si je vous assure !). Cette surmédiation de deux candidats est la conséquence de la peopolisation de la politique française, quinze ans après son émergence aux Etats Unis d'Amérique. C'est ainsi que tous les jours, la ménagère de 50 ans put suivre sur TF1 les amours tumultueux de Cécilia et Nicolas ... Après l'élection, elle n'a bien sûr pas manqué d'acheter le numéro de *Voici* où l'on pouvait tout savoir sur la séparation de Ségolène et de François Hollande et le remariage de Nicolas avec Carla. Passionnant tout cela..



Les autres candidats, à commencer par François Bayrou qui parla de collusion entre les puissances médiatico-financières et les deux partis dominants de la vie politique (ce qui existe effectivement aux USA), vivent dans cette surmédiation les raisons de leurs échecs. Ont-ils pourtant raison ? De même que les Français se sont progressivement détournés de la *trash-TV* et de ses pires *reality shows*, cette surmédiation semble commencer à lasser les électeurs en ces temps de crise. C'est ainsi que la courbe de popularité Nicolas-tu-vois-ma-Rolox plonge tandis que celle du chômage grimpe ... De même, Ségolène Royal a sérieusement décroché dans les sondages et il n'y a guère plus qu'elle qui croie encore en ces chances de devenir un jour la première femme présidente de la République (enfin, si tant est qu'elle soit au courant qu'elle ne l'est pas, puisqu'elle a toujours qualifié de « de formidable succès » son second tour à l'élection présidentielle !).

Alors, Nicolas Besancenot ou François Bayrou arriveront-ils à conquérir la plus haute fonction de l'Etat à la prochaine occasion ? Voilà une question de politique-fiction à laquelle nous ne répondrons pas dans ce TP. Nous nous contenterons de prendre ce thème comme prétexte à une étude de la définition de bases de données sous Oracle, à l'aide du langage SQL. On considère donc une base de données regroupant différentes informations sur les hommes politiques de notre pays, et en particulier leur cote de popularité estimée par différents instituts de sondages. Soient donc les relations suivantes :

- personne(num_h, nom, prenom, parti, fonction) liste des hommes politiques,
- institut(sigle,courant), liste des instituts de sondages,
- parti(num_p, sigle, nom, courant) liste des partis,
- sondage(institut,n_hom,cote,date) listes des côtes de popularité.

Ces relations sont définies dans les tableaux pages suivantes. A vous de déterminer les attributs jouant le rôle de clés primaires ou à les rajouter le cas échéant, de même que les relations entre relations donnant lieu à intégrité référentielle. Pour cela, on donne la définition des dépendances fonctionnelles correspondant à cette BD :

- relation PERSONNE num_h à nom, prenom, parti, fonction
- relation INSTITUT sigle à courant
- relation PARTI num_p à sigle, nom, courant
- relation SONDAGES institut, n_hom, date, cote

On donne de même les dépendances d'inclusion correspondantes:

- $\Pi_{\text{parti}}(\text{PERSONNE}) \subseteq \Pi_{\text{num_p}}(\text{PARTI})$
- $\Pi_{\text{institut}}(\text{SONDAGES}) \subseteq \Pi_{\text{sigle}}(\text{INSTITUT})$
- $\tilde{O}_{\text{n_hom}}(\text{SONDAGES}) \dot{\bar{I}} \tilde{O}_{\text{num_h}}(\text{PERSONNE})$

personne	attribut	définition
	num_h	référence de l'homme politique, entier
	nom	nom de la personne, chaîne de 30 caractères, renseigné
	prenom	prénom de la personne, chaîne de 20 caractères, renseigné
	parti	référence au parti auquel appartient la personne
	fonction	fonction (ministre, etc...) assurée, chaîne de 20 caractères

institut	attribut	définition
	sigle	sigle de l'institut, chaîne de 10 caractères, renseigné
	courant	orientation politique de l'institut, chaîne de 20 caractères

parti	attribut	définition
	num_p	référence de parti, entier positif
	sigle	sigle du parti, chaîne de 10 caractères, renseigné
	nom	nom du parti, chaîne de 30 caractères, renseigné
	courant	orientation politique du parti, chaîne de 20 caractères, renseigné

sondages	attribut	définition
	institut	référence à l'institut de sondage
	n_hom	référence à la personne concernée
	cote	cote de popularité (% compris entre 0 et 100), entier, renseigné
	date	date du sondage, date, renseigné

2. Création de la base de données

- 1 — Déterminez l'ensemble des clés primaires et étrangères de la base de données. Dans quel ordre devront nécessairement être créées vos relations ?
- 2 — Créez sous Oracle l'ensemble des relations de votre base de données, en tenant compte de l'ensemble des contraintes d'intégrités.
- 3 — A l'aide de l'ordre SQL COMMENT ON, documentez les différents objets de votre base de données.
- 4 — Vérifiez dans le dictionnaire Oracle que les relations de la base de données ont bien été créées dans leur ensemble (attributs) et que les commentaires que vous avez défini ont bien été associés aux éléments concernés de la base de données. Etudiez de même les vues du dictionnaires associées à la définition de contraintes (par exemple, ALL_CONSTRAINTS mais aussi ALL_CONS_COLUMNS). Pouvez-vous me dire si les contraintes associées aux différentes tables sont actives, différables et si elles ont été mises en mode différé à leur création. Ces résultats sont-ils cohérents avec vos requêtes SQL de création de table ?

3. Chargement de la base de données : SQL*Loader

1 — A l'aide de la clause SQL INSERT INTO, remplissez la table INSTITUT avec les données suivantes :

sigle	courant
IFOP	
IPSOS	droite
BVA	
SOFRES	gauche

Il est possible de remplir une relation en important des données à partir d'un fichier externe. On utilise pour cela l'utilitaire Oracle SQL*Loader (commande sqlload) présenté en cours.

2 — En utilisant SQL*Loader, remplissez la table PARTI avec les données du fichier parti.don qui est présent sur la page web de cet enseignement (<http://www.info.univ-tours.fr/~antoine>). Créez pour cela le fichier de contrôle parti.ctl correspondant.

3 — Tous les éléments du fichier de données ont-ils pu être chargés ? Le cas échéant, quels sont les éléments qui ont posé problème et pourquoi ?

Dans un premier temps, on ne cherchera pas à récupérer ces données non importées.

4 — Les données correspondant à la table PERSONNE sont dans le fichier pers.don situé sur la page Web du cours. Créez un fichier de contrôle pour permettre l'insertion de ces données à l'aide de SQL*Loader.

5 — Cette fois, quels sont les éléments qui n'ont pu être chargés ? Quel est la cause des problèmes rencontrés ?

6 — Nous allons cette fois nous occuper des problèmes rencontrés, en suivant le protocole suivant :

- désactiver les contraintes d'intégrité ayant posé problème lors de l'importation dans la table PERSONNE
- réaliser l'importation des données erronées dans la table PERSONNE
- modifier la définition de la structure de la table PARTI pour permettre l'importation des données erronées
- réaliser l'importation des données erronées dans la table PARTI
- réactiver les contraintes mises précédemment en stand-by. Il serait intéressant ici d'avoir prévu la création d'une table d'erreur (cf cours) récupérant les tuples non intégrables avec les contraintes.

Quelles observations pouvez-vous faire au cours de la réactivation de la contrainte ? Pourquoi n'aurait-il pas été suffisant de différer les contraintes d'intégrité au lieu de les désactiver ?

7 — La dernière solution consiste à réaliser un fichier de commande regroupant un ensemble de requêtes SQL d'insertion de tuples (celles-ci peuvent avoir été générées par un script). Le fichier de commande sondage.sql (même répertoire) regroupe les commandes nécessaires au remplissage de la dernière table. Après l'avoir consulté, exécuter ce fichier à l'aide de la commande start sous SQLPlus. Avant le chargement des données, on aura pris soin de désactiver les contraintes d'intégrité référentielle associées à la table SONDAGES. Réactivez ensuite ces contraintes en sauvegardant dans une table de récupération des erreurs les tuples ayant violé ces contraintes. Consultez cette table pour donner la liste des erreurs (enregistrements complets) observées. On ne cherchera pas à corriger ces erreurs.

8 — Les dernières élections régionales ont vu l'apparition de partis plus ou moins folkloriques qu'il faut bien rentrer dans notre base de données. Pour cela, on dispose d'un fichier nouveau_parti.don. A l'aide de SQL*Loader, rajoutez les données correspondantes à votre base de données (table parti), sans écraser les données précédentes.

4. Vues : définition, manipulation

1 — Créer une vue V_GVT qui garde tous les attributs de la table PERSONNE et qui donne la liste des ministres de la base de données. On définira la vue avec l'option de vérification des contraintes

sur modification. Vérifiez dans le dictionnaire Oracle que cette vue a correctement été créée.

- 2 — Jouez avec cette vue `V_GVT` : modifiez, ajoutez et supprimez des tuples dans la vue, puis allez voir le contenu de la table entrant dans la définition de la vue. Commentaires ?
- 3 — Si vous ne l'avez pas encore fait, essayer d'insérer dans votre vue un tuple correspondant à un homme politique qui n'est pas ministre. Commentaires ?
- 4 — Modifiez la vue pour la définir sans l'option de vérification et rejouer le cas précédent : que se passe-t-il ?
- 5 — Les informations nécessaires à la cohérence de la base de données (clé primaire par exemple) n'ont pas à être connues d'un utilisateur lambda. Les vues permettent précisément de n'afficher que l'information utile aux utilisateurs. Supprimez donc votre vue `V_GVT` puis recréez-la gardant tous les attributs de la table `PERSONNE` à l'exception de la clé primaire `num_h.`, et sans définir l'option de vérification des contraintes de définition de la revue. Jouez à nouveau avec votre vue : quelles observations faites-vous ?
- 6 — Modifiez une nouvelle fois votre vue `V_GVT` puis recréez-la en gardant cette fois, outre la clé primaire, uniquement les attributs `nom` et `prenom` de la table `PERSONNE`. On ne posera pas, cette fois non plus, d'option de vérification des contraintes de définition de vue. Quelles observations faites-vous cette fois ?
- 7 — Afin d'éviter à avoir à préciser des relations de jointures dans les requêtes de sélection multitable, il peut sembler intéressant de créer une vue réalisant la jointure des tables concernées. Créer ainsi une vue `V_COTES` qui donne directement la cote de popularité (cote, institut, date) de chaque homme politique (nom, prenom, sigle du parti).
- 8 — De même, jouez avec cette vue `V_COTES`. Quelles observations pouvez-vous faire ?

5. Vues concrètes

- 9 — Créez maintenant deux vues matérialisées `CLICHE_HEBDO_GVT` et `CLICHE_LIVE_GVT` répondant à la même définition que la vue `V_GVT` et dont la mise à jour se fait respectivement de manière hebdomadaire ou sur une validation (`COMMIT`). La mise à jour de ces vues concrètes se fera par chargement complet des données. Vérifiez la bonne création de ces vues concrètes dans le dictionnaire Oracle.
- 10 — Ajoutez le tuple (`<valeur_cle>`, "Delanoë", "Bertrand", `<id_parti>`, 'ministre') à la table `PERSONNE`. Consultez le contenu de la vue `V_GVT` et des vues concrètes `CLICHE_HEBDO_GVT` et `CLICHE_LIVE_GVT`. Quels commentaires pouvez-vous faire ?
- 11 — Validez maintenant l'insertion précédente (`COMMIT`). Réitérez l'exploration du contenu des différentes vues : conclusion ?
- 12 — Tentez maintenant de créer une vue concrète équivalente à renouvellement journalier suivant un mode incrémental de mise à jour. Quelle est la réaction d'Oracle ? Savez-vous expliquer cette situation ?

6. Synonymes

- 13 — Créez un synonyme `S_PERSONNE_<VOTRE_NOM>` à partir de la table `PERSONNE`. Vérifiez dans le dictionnaire de données qu'il a bien été créé. Jouez avec ce synonyme : est-il possible de modifier le contenu de la table à travers ce synonyme ?
- 14 — Ce synonyme est-il visible de la part des autres utilisateurs Oracle ? Comment est-il déclaré dans le dictionnaire de données ?

7. Privilèges objets et système

Transparence de la vie politique oblige, on souhaiterait accorder à tous les utilisateurs le droit de consulter le contenu de la table `PERSONNE`, ou plus précisément les informations concernant les ministres.

- 1 — Dans un premier temps, on se contente d'accorder des droits en lecture sur la table `PERSONNE`. Donnez ce privilège à l'ensemble des utilisateurs. Vérifiez dans le dictionnaire que ce droit a bien été accordé. En pratique, les autres utilisateurs peuvent-ils bien accéder au contenu de cette table ?

On voudrait maintenant limiter ce droit de lecture aux seuls hommes politiques occupant une fonction de ministre. L'idée est donc d'accorder un droit de lecture uniquement sur la vue `V_GVT`.

- 2 — Révoquez les droits de lecture que vous venez de définir sur la table `PERSONNE`. Vérifiez que ces droits sont bien supprimés et qu'il n'est plus possible d'accéder à votre table.
- 3 — Limitez maintenant les droits de lecture à la vue `V_GVT` et vérifiez que ce privilège est bien opérationnel.
- 4 — Sans jouer sur les privilèges, une dernière solution consiste à passer par la création d'un synonyme public bien choisi. Vérifiez tout d'abord dans le dictionnaire Oracle que vous disposez du privilège système correspondant. Si oui, créez ce synonyme public et vérifiez qu'ainsi les informations sur tous les ministres sont bien visibles des autres utilisateurs. Qui est le propriétaire de cette vue publique ?

8. Transactions

Nous allons maintenant étudier la gestion de transaction en SQL.

- 1 — Validez tout d'abord vos dernières actions en fermant la transaction courante.
- 2 — Dans la table `PARTI`, ajoutez le tuple ("LDNJ","Ligue de Défense des Nains de Jardin","ExtGauche"). Consultez le contenu la table puis annulez cette transaction. Quelle observation faites-vous en consultant à nouveau le contenu de la table.
- 3 — Refaites la même opération, mais validez la transaction avant de faire une annulation. Conclusion ?
- 4 — Dans la table `PARTI`, ajoutez le tuple ("SS","Sologne aux Solognots","ExtDroite"). Puis ajoutez un attribut `naissance`, de type `DATE`, à cette table (cet attribut correspond à la date de création du parti en question). Annulez cette transaction puis consultez la structure et le contenu de la relation. Quelle observation faites-vous ? Pouvez-vous expliquer ce résultat ?

Lionel Jospin, comme Robert Hue, se sont retirés de la vie politique nationale. On aimerait donc remplacer les enregistrements correspondants de la table `PERSONNE` respectivement par Jack Lang (illustre ancien maire de Blois qui a été ministre) et Marie-George Buffet (qui n'a jamais eu de portefeuille ministériel). Il nous faut donc pour cela ôter les anciens enregistrements et les remplacer par les nouveaux, sans modifier le contenu de la base de données.

- 5 — Réaliser ces opérations en utilisant deux stratégies différentes : la désactivation de contrainte pour Jack Lang et le report de validation en fin de transaction pour Marie-George Buffet.