
PROJET de Base de Données

UE LSIN513 – Année 2023-2024

Le but de ce projet est de construire une base de données sous Oracle afin de mettre en pratique les différentes fonctionnalités standard d'un SGBD et mettre en application les différents concepts vus en cours.

Le projet sera **réalisé par groupe de 3 étudiants**, dont les étudiants appartiennent tous au même groupe de TD.

Le projet comprend 2 phases:

- la *première phase* correspond à la rédaction d'un cahier des charges sur une application de votre choix, la modélisation de votre problème en Entité/Association et la réaction des contraintes d'intégrité indispensables au bon fonctionnement de votre base de données. Une fois le modèle E/A écrit, donner une transformation de ce modèle vers le relationnel (listes des relations avec leurs colonnes). On vous demandera également d'écrire en langage naturel toutes les questions pertinentes à poser sur votre base de données. **Vous devez rendre à votre chargé de TD, le rapport imprimé de quelques pages (contenant tout ce qui est décrit au dessus) pour le lundi 6 novembre.**
- la *deuxième phase* correspond à l'implémentation de votre base de données en SQL sous Oracle, c'est à dire, tous les scripts SQL correspondants à la création de la base de données, aux chargements des jeux de données, aux requêtes d'interrogation, et à la rédaction des contraintes d'intégrité en PL/SQL. Le rapport final doit contenir à la fois le rapport de la phase 1, qui a pu être mis à jour suite au retour de vos enseignants, puis tous les scripts SQL et PL/SQL. Le tout intégré dans un seul fichier .pdf. **Ce rapport sera remis le mercredi 3 janvier au plus tard par email, chaque groupe aura une date de soutenance durant la période d'examen de janvier.**

Veuillez maintenant lire la suite de l'énoncé pour avoir plus de détail sur le travail demandé et les contraintes imposées.

Email : Beatrice.Finance@uvsq.fr

Partie 1 : Choix de votre application et Modélisation

Dans la première partie on vous demande de construire une base de données relationnelle sur la thématique de votre choix. Par exemple, vous pouvez choisir de modéliser les informations relatives à une université, une compagnie aérienne, une gestion de commandes, une DB cinéphile ou sportive, l'assurance, etc...

Il s'agit ici de faire une conception E/A détaillée mais ni trop compliquée. Après avoir décrit le modèle vous pouvez le traduire en relationnel.

Votre application doit respecter les contraintes suivantes :

- ◆ contenir au moins 6 tables (pas plus de 10). Les types de données gérés doivent être variés. Les chaînes de caractères, les dates et les nombres à virgule sont conseillés, plus si vous le désirez.
- ◆ la sémantique des champs de données doit impliquer la définition de contraintes d'intégrité. Au moins deux contraintes par table doivent être définies et doivent couvrir le spectre suivant des contraintes (NOT NULL, PRIMARY KEY, FOREIGN KEY, CHECK, etc..) et la rédaction en langage naturelle de celles plus complexes.
- ◆ permettre la définition de requêtes SQL relativement complexes (calculs d'agrégats, sélection, projection, jointure, requêtes imbriquées, intersection, union, différence et division, jointure externe etc.). Certaines requêtes devront faire apparaître plus de 3 jointures.

Partie 2 : Implémentation de votre BD sous Oracle

Cette partie comprend 6 sections décrites ci-dessous.

A/ Création du schéma de la base de données

Implémenter le schéma de votre base données, obtenu à partir de la phase 1, sous Oracle.

A rendre : tous les scripts SQL permettant de le générer.

B/ Jeu de données

*Le jeu de données doit être **soigneusement** préparé et permettre la validation des requêtes complexes qui seront posées par la suite. Il doit y avoir au moins 30 n-uplets par table. Les valeurs choisies pour les attributs doivent être cohérentes avec le schéma de la base.*

Pour créer vos jeux de données, vous utiliserez tout d'abord des requêtes *SQL (insert)*, puis l'outil de chargement d'Oracle *SQL*LOAD* pour un chargement massif des données. Pour ce faire, créer un fichier *nomfichier.ctl* qui contient les définitions suivantes:

```
LOAD DATA
INFILE *
APPEND
INTO TABLE NomTable
FIELDS TERMINATED BY "," OPTIONALLY ENCLOSED BY ""
(NomAtt1, NomAtt2, ..., NomAttn)
BEGIN DATA
1,"AAAA",.....,"BBBB "
2,"CCCC", ....., "DDDD"
```

Puis taper la commande suivante sous Unix :

```
sqlldr userid=username/passwd control = client.ctl log = client.log
```

Visualiser le résultat du chargement en consultant le fichier `.log`

Si vous le souhaitez vous pouvez utiliser des séquences (`create sequence`, `currval`, `nextval`). Lire la documentation pour de plus amples détails.

A rendre : Le jeu de données résultant en justifiant sa pertinence par rapport au schéma et aux requêtes. Donner les deux types de script qui ont permis de le générer : 1) à partir de SQL, 2) à partir de l'utilitaire SQL*LOAD.

C/ Manipulation des données

Après avoir chargé vos tables, vous reprendrez la liste des questions que vous vous posiez à la phase 1 et vous les écrirez en SQL.

A rendre : Pour chaque requête, mettre la question en langage naturel, suivi du script SQL.

D/ Vues

Ensuite afin de garantir un bon niveau de confidentialité, on définira des vues sur lesquelles vous allez placer des droits d'accès. Vous définirez au moins 5 vues. Les vues devront contenir soit une jointure, soit une fonction d'agrégat, soit les deux en même temps.

A rendre : Les scripts de création des vues, ainsi qu'une description en langage naturel de chacune, ainsi que la définition des droits d'accès : groupes utilisateurs, droits...

E/ Intégrité des données : les triggers

Oracle fournit plusieurs moyens pour le maintien de l'intégrité des données comme les contraintes d'intégrité, les assertions et les triggers. Un trigger est une procédure PL/SQL associée à un événement sur une table. On peut définir un trigger Oracle à l'aide de la commande CREATE OR REPLACE TRIGGER ...

Vous devez spécifier au moins six règles de gestion de votre application en langage naturel qui donneront lieu à l'écriture de triggers.

A rendre : les scripts PL/SQL de création des triggers, ainsi que leur formulation en langage naturel.

F/ Méta-données

Vous écrirez un script SQL nommé liste_ora_constraints qui donne une fois exécuté la liste de toutes les contraintes d'intégrité définies sur votre BD que vous classerez par table et par type de contrainte, le corps de chaque contrainte devra être récupéré.

Vous écrirez un script SQL nommé liste_ora_triggers qui donne une fois exécuté la liste de tous les triggers que vous avez définis, classés par nom de table.

Vous écrirez deux autres scripts d'interrogation de la méta-base qui vous semblent pertinentes.