

---

## *PROJET de Base de Données*

*UE LSIN513 – Année 2024-2025*

---

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. Au moins un membre du groupe doit avoir installé la machine virtuelle oracle pour pouvoir tester l'outil de chargement massif, ainsi que les droits d'accès.

### **LIRE LA TOTALITE DU SUJET AVANT DE COMMENCER**

Email : [Beatrice.Finance@uvsq.fr](mailto:Beatrice.Finance@uvsq.fr)

Le projet comprend 2 phases:

- la *première phase* correspond à la phase d'analyse du problème. Il n'y a rien à implémenter ; juste à réfléchir à la rédaction d'un cahier des charges sur une application de votre choix. Voir ci-dessous. **Le document à rendre sera écrit en FRANCAIS et ne contient AUCUNE LIGNE DE CODE SQL/ ou PL/SQL, NI DE CONCEPTS INFORMATIQUES.**

Vous devez me rendre le rapport imprimé de quelques pages (contenant tout ce qui est décrit au dessus) durant vos séances de TD de **la semaine du 21 octobre, c-à-d les 23, 24, 25 octobre**. La semaine du 6 novembre durant vos séances de TD vous aurez rendez-vous avec moi pour un retour de la phase 1, qui donne soit feu vert (passer à la phase 2), soit feu orange (vous devez faire quelques mises à jour mineures, avant de passer à la phase 2), soit feu rouge (vous devez faire de grosses modifications que vous devez revalider avec moi avant de passer à la phase 2).

- 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 aux remarques données, puis tous les scripts SQL et PL/SQL. **Le tout intégré dans un seul fichier .pdf. Ce rapport sera remis le 20 décembre avant minuit au format électronique pdf qui sera à déposer sur e-campus dans le dossier associé à votre TD, un seul dépôt par groupe.** Chaque groupe aura une date de soutenance durant la période d'examen de janvier.

## Phase1 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. Entre dix et quinze concepts est un bon chiffre.

La phase 1 comprend uniquement du langage naturel, et doit contenir les différentes sections suivantes:

- ◆ Décrire le cahier des charges, c'est-à-dire la description des besoins de votre application de bases de données.
- ◆ *Modélisation* : donner le schéma Entité/Association détaillé modélisant les besoins définis dans le cahier des charges. On rappelle qu'un schéma E/A est indépendant d'un modèle de données.
- ◆ *Contraintes d'intégrité* : Définition des contraintes d'intégrité qui ne peuvent pas être exprimées sur le modèle conceptuel. Mettre l'accent sur la rédaction en langage naturel de contraintes d'intégrité **complexes**.
- ◆ *Droits et Vues – confidentialité des données*: Décrire les différents utilisateurs ou groupes d'utilisateurs de votre base de données, ainsi que les droits qu'ils devraient avoir en vous référant aux concepts décrits au-dessus.
- ◆ *Requêtes* : Qu'est-ce qui intéresse les utilisateurs de votre base de données? Définir en langage naturel une vingtaine de questions relativement complexes en langage naturel nécessitant des calculs d'agrégats, sélection, projection, jointure, requêtes imbriquées, intersection, union, différence et division, jointure externe etc..
- ◆ Traduction du schéma E/A en relationnel. Lister le nom des tables avec le nom de leurs colonnes, en indiquant en gras les clés primaires et les clés étrangères soulignées.

## 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.