



**Faculté des Sciences / Université  
Cote d'Azur**

**MBDS**

**UE3 - Immersion BD et SQL Oracle**

**Titre du projet : Gestion d'un cabinet  
médical**

**Etudiants :**

Pierre Bob Charlemagne

Rodely Douilly

Pierre Rubens Milorme

**Professeur : Wedter Jérôme**

## Description du sujet

La description du sujet : ce que doit faire cette application, décrire textuellement ces structures et ses services

# **1 Exercice 1.2 : Spécification, Analyse et conception**

## **1.1 Description textuelles des requêtes de mise à jour**

(2 requêtes impliquant 1 table, 2 requêtes impliquant 2 tables, 2 requêtes impliquant plus de 2 tables)

## **1.2 Description textuelles des requêtes de suppression**

(2 requêtes impliquant 1 table, 2 requêtes impliquant 2 tables, 2 requêtes impliquant plus de 2 tables)

## **1.3 Description textuelles des requêtes de consultation**

(5 requêtes impliquant 1 table dont 1 avec un group By et une avec un Order By, 5 requêtes impliquant 2 tables avec jointures internes dont 1 externe + 1 group by + 1 tri, 5 requêtes impliquant plus de 2 tables avec jointures internes dont 1 externe + 1 group by + 1 tri)

## **1.4 Dictionnaire de données MERISE. Pour chaque entité décrire chacune des propriétés**

Titre / description / format des données / type / Indentifiant / contraintes

## **1.5 Description textuelles des associations**

Décrire textuellement les associations entre entités

## **1.6 Définition du Modèle Entité-Association MERISE**

(en utilisant le logiciel Poweramc de SYBASE/SAP ou manuellement). Vous devez vous limiter à 10 entités maximum et 5 minimum. Vous devez ici prendre en compte les contraintes identifiées lors de la description du dictionnaire de données. Exemple de liens d'association pour deux entités A et B ayant une liaisons 1 : N ou N-M (exemple UN PILOTE ASSURE 0, 1 ou plusieurs VOL, un VOL est assuré par 1 et 1 PILOTE au plus)

## **1.7 Définition du modèle logique de Données ou schéma relationnel**

(en utilisant le logiciel Poweramc de SYBASE/SAP ou manuellement) un schéma de données logique en respectant les contraintes d'intégrités d'entité (PRIMARY KEY), de domaine (CHECK, NOT NULL, ...) et de référence (REFERENCES / foreign key). Générable automatiquement avec POWERAMC si vous avez décrites au niveau MCD

## **1.8 Spécification des traitement avec des packages PLSQL (Modèle de traitements)**

Choisir parmi vos tables deux d'entres (A et B) elles sur lesquelles les fonctions suivantes vont être spécifiées puis implémentées : Sur la table A, définir un package plsqli ayant le nom de la dite table: - ajouter une nouvelle occurrence à A : fonction AInsérer; - supprimer une occurrence à A (Attention : les enregistrements liés dans B doivent aussi être supprimés) : fonction ASupprimer; - modifier des informations sur de A : fonction AmodifierF1, AmodifierF2 (texte requêtes correspondantes plus haut); - lister toutes les occurrences de A: fonction ALister; - fournir le nombre total des occurrences de A : fonction ATotal ; - Proposer aussi 3 fonctions avec des requêtes de consultation impliquant 2 ou 3 tables au moins (jointure, groupe, tri) : fonction Af1, Af2, Af3. f1, f2, f3 sont des noms à définir AmodifierF2 (texte requêtes

correspondantes plus haut); Sur la table B, définir un package plsql ayant le nom de la dite table : - ajouter une nouvelle occurrence à B : fonction BInsérer - supprimer une occurrence à B : fonction BSupprimer;- modifier des informations sur de B : fonction BModifierF1, BModifierF2 (texte requêtes correspondantes plus haut); - lister toutes les occurrences de B pour une occurrence de A donnée: fonction Blister Nota : Seule la partie spécification de chaque package est nécessaire ici. bien choisir les paramètres des méthodes. Bien nommer les méthodes. Remplacer F1 à Fn par des noms appropriés.

## **1.9 Spécification des triggers**

Vous devez définir textuellement aux moins deux triggers. Ces triggers vous permettant devront vous permettre de gérer aux moins deux règles de gestion qui ne peuvent être prises en compte à travers le schéma de données.

## References