

14 novembre 2022

Partie 1 **Analyse & Conception**

Sujet

Suivi pédagogique à l'Université

Il s'agit de concevoir une base de données permettant de faire un suivi pédagogique des étudiants durant plusieurs années universitaires. La base de données doit contenir des informations sur les diplômes, les enseignements ainsi que notes et les résultats des étudiants.

L'Université est divisée en composantes (UFR MIM, IUT Metz, ...). Un diplôme, identifié par un code et décrit par un libellé, est rattaché à une composante. Un diplôme se prépare en plusieurs années (étapes). Une étape est identifiée par code et possède un libellé. Une étape se divise en deux semestres. Les enseignements sont organisés en éléments pédagogiques rattachés à des semestres. Un élément pédagogique, identifié par un code, est caractérisé par un libellé et des volumes horaires de cours, de TD et de TP. Un élément pédagogique de type 'UE' peut se décomposer en éléments pédagogiques de type 'EC'. Chaque élément pédagogique est affecté d'un coefficient. Seuls les éléments pédagogiques de type 'UE' sont affectés de crédits ECTS.

Un étudiant est inscrit, chaque année, à une ou plusieurs étapes. Un étudiant, connu par son nom et son prénom, est identifié par un numéro. Le contrôle des connaissances est organisé sous forme d'épreuves. Une épreuve correspond à une modalité de contrôle ('CC', 'CT', ...). Une épreuve, identifiée par un code, est rattachée à un élément pédagogique et une session ; elle est affectée d'un coefficient et on connaît sa nature ('écrit', 'tp', 'projet', 'oral', ...). Une épreuve a lieu à une date et un créneau horaire donnés, dans une ou plusieurs salles. Les salles sont rattachées à une composante et sont identifiées par un numéro ; on connaît la catégorie ('Amphi', 'TD', 'TP') et la capacité de chaque salle.

On conserve, pour chaque session et chaque année, les notes obtenues par les étudiants dans les épreuves, les notes calculées pour les 'EC' ainsi que les notes et les résultats ('admis', 'ajourné', 'compensé') calculés pour les 'UE', les semestres, les années et les diplômes. Il y a deux sessions par année.

Travail demandé

Il s'agit de faire une analyse du sujet et de concevoir un schéma conceptuel de données. Toutes les hypothèses dans l'analyse sont acceptées à condition qu'elles complètent le sujet, sans le simplifier.

Contraintes

La partie 1 du projet peut être traitée par groupe de 2 à 4 étudiants.

Dossier à remettre

Un fichier au format PDF à transmettre par mail au plus tard le **12 décembre 2022** contenant :

- le rapport d'analyse : les explications de vos choix et de vos hypothèses ;
- un schéma conceptuel de données en **Entité-Association** ;
- le schéma **Relationnel** équivalent obtenu par transformation, normalisé si nécessaire ;
- liste des étudiants formant le groupe.

Barème Note1 /60

- Rapport : 10
- Schéma Entité-Association : 40
- Schéma Relationnel : 10

Partie 2

Réalisation

Sujet : « Gestion de l'occupation des salles d'enseignement »

Il s'agit de réaliser une base de données et une interface en PL/SQL (Oracle ou MySQL) servant à une application permettant à la scolarité d'une composante de gérer l'occupation des salles d'enseignement durant une année universitaire. Le schéma relationnel de la base de données à implémenter est le suivant :

Salle (NoSalle, Categorie, NbPlaces)

ELP (CodeELP, NomELP, Formation, HC, HTD, HTP, HCRes, HTDRes, HTPRes)

Groupe (Groupe, Formation, Effectif)

Reservation (NoReservation, NoSalle, CodeELP, Groupe, Formation, Nature, Debut, Duree)

Categorie : Catégorie de la salle = { 'Amphi' / 'Salle' / 'Salle TP' }.

ELP : Élément Pédagogique.

Groupe = { 'Promo' / 'TDn' / 'TPn' } où n est un numéro d'ordre (si plusieurs groupes).

Formation : { 'BUT Info 1', 'BUT Info 2', 'BUT Info P', 'L1MI', 'L2I', 'L3I', 'M1I', 'M2I', ... }.

HCRes, HTDRes, HTPRes : Heures d'enseignement réservées (initialement à 0), incrémentées lors des réservations.

Nature : { 'Cours' / 'TD' / 'TP' / 'Epreuve' }.

Debut : Date et Heure de début de la réservation.

Duree : Durée de la réservation (en minutes).

Travail demandé

- **Concevoir un schéma** physique de la base de données en tenant compte du maximum des contraintes d'intégrité.
- **Réaliser des opérations en PL/SQL** permettant les traitements suivants :
 - Saisie et mise à jour des salles, des enseignements, des groupes ;
 - Réservation des salles ;
 - Recherche/Calcul et affichage des informations répondant à différents besoins.
- Les traitements demandés nécessitent une analyse détaillée :
 - Vérifier la cohérence des données vis-à-vis des contraintes d'intégrité ;
 - Tenir compte du maximum de situations pour répondre à l'utilisateur ;
 - Un jeu de données et des exemples d'exécution sont proposés dans le fichier test pour expliciter les contraintes d'intégrité et pour tester vos opérations.
- **Développer un jeu de données** permettant de tester vos opérations.
- Les procédures et les fonctions à écrire sont les suivantes (réparties en 4 projets) :

○ **Projet 1 : Gestion des Salles**

MajSalle (Salle in varchar2, Cat in varchar2, Nb in number)

- Tous les paramètres sont obligatoires ;
- Si la Salle n'existe pas elle est ajoutée, *NbPlaces* doit être positif ;
- Si la Salle existe et *Nb* est positif, *NbPlaces* est remplacé par *Nb* ;
- Si la Salle existe et *Nb*=0, la salle est supprimée et toutes ses réservations sont supprimées.

EstDisponible (Salle in varchar2, Debut in date, Duree in number) return boolean

- Tous les paramètres sont obligatoires ;
- La fonction retourne 'Vrai' si la Salle est disponible dans le créneau horaire donné (date et heure de Début et Durée en minutes), sinon la fonction retourne 'Faux' ;
- Si la Salle n'existe pas, on affiche un message d'erreur.

SallesDisponibles (Debut in date, Duree in number, Cat in varchar2, Nb in number)

- Les paramètres *Cat* et *Nb* sont optionnels (NULL) ;
- Recherche et affiche la liste des salles disponibles de catégorie *Cat* et *NbPlaces* $\geq Nb$ en utilisant la fonction *EstDisponible* ;
- La procédure affiche *NoSalle*, *Catégorie* et *NbPlaces*, triés par catégorie ;
- Si la catégorie (*Cat*) est erronée, on affiche un message d'erreur ;
- Si la liste est vide, on affiche le message « Pas de salle disponible dans ce créneau horaire ».

○ **Projet 2 : Gestion des Enseignements**

MajELP (Code in varchar2, Nom in varchar2, Forma in varchar2, C in number, TD in number, TP in number)

- Si l'ELP (donné par le paramètre *Code*) n'existe pas, il est ajouté, sinon l'ELP est modifié : *NomELP*, *Formation*, *HC*, *HTD* et *HTP* sont remplacés par les valeurs des paramètres ;
- *C*, *TD* et *TP* ne peuvent pas être tous à 0 ou null, sinon l'ELP est supprimé et toutes ses réservations sont supprimées.

Reservations (Code in varchar2)

- Recherche et affiche la liste chronologique des réservations d'un ELP ;
- La liste est présentée avec les informations suivantes : *Début*, *Fin* (*Début*+*Durée*), *Groupe*, *Formation*, *Nature*, *NoSalle* ;
- Si l'ELP n'existe pas, on affiche un message d'erreur ;
- Si la liste est vide, on affiche le message « Pas de réservation pour cet ELP ».

ReservationComplete(Code in varchar2) return boolean

- Retourne 'Vrai' si toutes les heures d'enseignement de l'ELP ont été réservées et 'Faux' sinon ;
- Si l'ELP n'existe pas, on affiche un message d'erreur.

○ **Projet 3 : Gestion des Groupes**

MajGroupe (Gpe in varchar2, Forma in varchar2, Eff in number)

- Tous les paramètres sont obligatoires ;
- Si le groupe de la formation (*Gpe, Forma*) n'existe pas il est ajouté, *Effectif* doit être positif ;
- Si le groupe de la formation existe et *Eff* est positif, *Effectif* du groupe est remplacé par *Eff*, sinon le groupe est supprimé ainsi que toutes ses réservations.

ReservationsGroupe (Gpe in varchar2, Forma in varchar2)

- Recherche et affiche la liste chronologique des réservations d'un groupe d'une formation (*Gpe, Forma*) ou de tous les groupes d'une formation si *Gpe* est omis. La liste est présentée avec les informations suivantes : *Début*, *Fin* (*Début*+*Durée*), *CodeUE*, *NomUE*, *Nature*, *NoSalle*, *Gpe* ;
- Si le groupe ou la formation n'existe pas, on affiche un message d'erreur ;
- Si la liste est vide, on affiche le message « Pas de réservation pour ce groupe ou cette formation ».

EstLibre (Gpe in varchar2, Forma in varchar2, Debut in date, Duree in number) return boolean

- Retourne 'Vrai' si le groupe est libre au créneau indiqué et 'Faux' sinon ;
- Si le groupe n'existe pas, on affiche un message d'erreur.

○ **Projet 4 : Gestion des Réservations**

Reserver (Salle in varchar2, Code in varchar2, Gpe in varchar2, Forma in varchar2, Nat in varchar2, Deb in date, Dur in number)

- Effectue une réservation de salle à un créneau horaire donné (*Deb, Dur*) pour un enseignement et un groupe donnés. On suppose que la Salle est disponible et que le groupe (*Gpe, Forma*) est libre. *NoReservation* est attribué automatiquement. *Effectif* du groupe doit être inférieur ou égal à *NbPlaces* de la Salle. Selon la nature (*Nat*), *HCRes*, *HTDRes* ou *HTPRes* de l'UE est incrémenté de la durée (*Dur*) ;
- Si la Salle, l'ELP ou le groupe de la formation (*Gpe, Forma*) n'existe pas, on affiche un message d'erreur ;
- La durée de la réservation (*Dur*) doit être positive et inférieure ou égale à 240.

ReservationsSalle (Salle in varchar2)

- Recherche et affiche la liste chronologique des réservations d'une Salle. La liste est présentée avec les informations suivantes : *Debut*, *Fin* (*Debut*+*Durée*), *CodeELP*, *NomELP*, *Nature*, *Groupe*, *Formation*.

TauxOccupationSalle (Salle in varchar2, Deb in date, Fin in date) return number

- Retourne le taux d'occupation (%) d'une Salle sur une période donnée par deux dates (*Deb, Fin*), le taux occupation est calculé sur la base d'une moyenne de 6 heures par jour ;
- Si la Salle n'existe pas, on affiche un message d'erreur ;

Contraintes

- La partie 2 du projet est strictement **individuelle**.
- Répartition des 4 projets entre les étudiants au sein d'un même groupe de la partie 1.

- Serveur de bases de données : Oracle ou MySQL.
- Langage de développement : PL/SQL.
- **Respecter scrupuleusement les noms des objets :**
 - **Tables et attributs**
 - **Procédures et Fonctions ainsi que le nombre et le type des paramètres**

Dossier à remettre

- Le dossier doit être remis au plus tard le **23 janvier 2023** et consiste en trois (3) fichiers :
 - Un rapport d'analyse de 2/3 pages (en **pdf**) décrivant les traitements demandés et les requêtes d'interrogation (SELECT) nécessaires (**PAS de code PL/SQL**) ;
 - Un fichier script (avec l'extension **script.txt**) contenant les commandes de création ET les commandes de suppression des objets demandés (tables, procédures, fonctions et séquence) dans l'ordre imposé par la compilation (**Pas les données**) ;
 - un fichier contenant les tests avec le jeu de données, au format texte (avec l'extension **tests.txt**).
- Les trois fichiers, nommés avec votre **nom**, séparés et non compressés, doivent être **déposés** sur la plateforme Arche - cours « BUT Info 2 Passerelle – BD1 & BD2 » - section « Projet » – « Dépôt du dossier ».

Barème Note2 /140

- Rapport : 20
- Création/Compilation du schéma de la base de données : 20
- Programmation (gestion des erreurs, validation, commentaires, ...) : 10
- Jeu de données et tests : 15
- Evaluation des opérations demandées : 75 (25 par opération)

Note Finale /20 : (Note1 + Note2)/10