

Projet Conception d'une base de données SQLite et intégration avec Python



1 Parties et livrables

Le projet se décompose en deux parties :

1. **Première partie (voir TDTP7)** : description d'un problème en langage naturel et création d'un modèle conceptuel en UML avec une traduction en modèle relationnel. **Le sujet est libre, mais dans le cadre "responsabilité sociétale et environnementale"**. Le modèle conceptuel doit avoir **entre 5 et 6 classes**. Le modèle doit contenir au moins une association *Many-to-many* et une association *One-to-many*. Voici quelques idées de domaines : *un système pour la gestion de...*
 - une société à mission
 - téléphones reconditionnés
 - un centre de recyclage
 - bornes électriques de charge
 - trottinettes et vélos électriques
 - cooperative d'énergie verte
 - ...
 - pour d'autres idées voici le lien de l'Agence de la transition écologique (ADEME)

Livrables :

- Description en langage naturel de pas plus d'une demie page (1 point)
 - Modèle UML complet (classes, associations, cardinalités, contraintes, etc.) (1,5 points)
 - Modèle relationnel (relations, spécifications et contraintes) (1 point)
 - Règles de traduction de noms (par exemple, lors qu'on traduit un attribut de classe en modèle logique on introduit comme suffixe le nom de la classe) (0,5 points)
2. **Deuxième partie (voir TDTP7bis)** : mettre en œuvre quelques fonctionnalités simplifiées d'une application Python + SQLite en s'appuyant sur l'exemple fourni.
 - Mise à jour du modèle UML et relationnel avec bon alignement (2 points)
 - Implémentation du modèle logique en SQL (CREATE) (2 points)
 - Script d'un jeu de données SQL d'insertions (INSERT) qui marche (0,5 points)
 - Script d'un jeu de données SQL d'insertions (INSERT) qui produit des erreurs à cause de contraintes implémentés (0,5 points)
 - Interaction utilisateur via la console (2 points). Aucune interface graphique n'est demandée, chaque fonctionnalité peut être implémentée à l'aide de lecture/écriture sur un terminal.
 - Écrire une requête pour chaque niveau (sélection-projection / opérateurs ensemblistes / jointure-agrégation) en utilisant SQL, à partir d'un (ou plusieurs) programme(s) en Python (voir modèle fourni) (3 points)
 - Paramétrage de requêtes en fonction des entrées de l'utilisateur via la console (2 points)
 - Possibilité d'insérer, éliminer des données et mettre à jours certains données (1 point)
 - Création d'une vue ou un trigger pertinent (1 point)
 - Respect de consignes, qualité du code, commentaires et documentation (2 points)

Livrable Final :

- Mise à jour des éléments de la première partie dans **un seul document pdf, à l'intérieur d'un dossier doc** faisant partie du projet Python.
- Dossier compressé (zip ou tar.gz) ayant comme nom : INF403_BINOMEX_NOM1_NOM2 et contenant : le code Python (.py) + scripts SQL + dossier doc contenant le pdf avec les modèles.

2 Instructions de rendu

Le rendu se réalisera via un Quiz Caséine par **binômes obligatoirement**. Les échéances sont les suivantes :

1. Partie 1 (4 points) : Lundi 17 Avril à 8h
2. Partie 2 + Partie 1 (mis à jour) (16 points) : Vendredi 5 Mai à 19h

Le **nom du fichier compressé déposé sur Caséine** sera la forme : INFX_BINOMEX_NOM1_NOM2.zip (exemple : INF2_BINOME1_PARENT_CORTES.zip)

Attention :

- le non-respect des consignes de rendus et des dates, entrainera une pénalisation sur la note
- le style du code ainsi que la rigueur de la documentation seront évalués (orthographe, mise en forme, etc.)
- la rédaction en anglais est possible
- le projet est censé pouvoir se faire lors des séances de TD/TP.
- nous allons privilégier la réponse aux questions dans les séances TD/TP plutôt que par mail
- ce projet a un poids de 12.5% de la note de l'UE.