

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. **Le sujet est libre**. Le modèle conceptuel doit avoir **entre 4 et 6 classes**. Le modèle doit contenir au moins une association *Many-to-many*. Voici quelques idées de domaines : *Un système pour la gestion de...*

- *artistes et ses albums*
- *étudiants et ses cours*
- *organisation de sorties de randonnées*
- *organisation d'événements sportifs*
- ...

Livrables :

- Description en langage naturel de pas plus d'une demi page.
 - Modèle UML complet (classes, associations, cardinalités, contraintes, etc.).
 - Modèle relationnel (relations, spécifications et contraintes).
 - 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).
2. **Deuxième partie (voir TDTP7bis)** : mettre en œuvre quelques fonctionnalités simplifiées d'une application Python + SQLite en s'appuyant sur l'exemple fourni.
 - Implémentation du modèle logique en SQL (**CREATES**)
 - Script d'un jeu de données SQL d'insertions (**INSERT**) qui marche.
 - Script d'un jeu de données SQL d'insertions (**INSERT**) qui produit des erreurs à cause de contraintes implémentés.
 - 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*). Proposer aussi la possibilité d'insérer des données et mettre à jours certains données.

Livraison Final :

- Modèles et description correspondants aux parties précédentes (avec des mise à jours si besoin) **dans un dossier doc**, à l'intérieur du projet Python.
- Dossier compressé (zip ou tar.gz) avec le code Python (.py) + scripts SQLite + modèles + description, **suivant le format de l'archive fournie**.

2 Instructions de rendu

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

1. Partie 1 (5 points) : Vendredi 15 Avril à 18h
2. Partie 2 - COMPLET (15 points) : Vendredi 13 Mai à 18h

Le **nom du fichier compressé déposé sur Caséine** sera la forme : INFX_BINOMEX_NOM1_NOM2.zip (exemple : INF3_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
- nous allons privilégier la réponse aux questions en séance de TD/TP plutôt que par mail

Note : le projet est censé pouvoir se faire lors des séances de TD/TP. Il a un poids de 12.5% de la note de l'UE.