

Ucac-Icam et Ulc-Icam

INFO221 (mai 2025)

Travail dirigé et pratique 4

Enseignant

Luc Lavoie (luc.lavoie@usherbrooke.ca)

Version 1.0.0 (en vigueur)

2025-05-08

Contexte

Le présent travail est divisé en deux parties : le *travail dirigé* réalisé en laboratoire sous la supervision des enseignants sur une période de deux heures et un *travail pratique* réalisé par la suite de façon autonome. Le travail pratique représente l'achèvement du travail dirigé. Le travail peut être réalisé seul ou en tandem.

L'énoncé du problème révisé [EPP] de même que le présent document sont disponibles dans le répertoire public du cours.

Objectifs

Le travail a pour but de mettre en pratique la création d'un schéma de bases de données à l'aide du langage SQL. Il répond aux objectifs spécifiques suivants :

- ◇ définir des routines et des vues ;
- ◇ modifier la structure d'un modèle logique existant ;
- ◇ appliquer les principes de gestion des informations manquantes ;
- ◇ procéder à une alimentation simple et à sa vérification ;
- ◇ poursuivre l'apprentissage d'un ensemble d'outils prescrits (l'atelier DataGrip et le SGBDR PostgreSQL).

Matériel fourni

Le matériel suivant est disponible dans le dépôt du cours :

1. énoncé du travail pratique 1, le présent document (IFT187_2025-1_LAB4.pdf),
2. énoncé préliminaire de portée du travail (Herbivorie_EPP_v022a.pdf),
3. création du schéma (Herbivorie_cre.sql),
4. suppression du schéma (Herbivorie_drop.sql),
5. suppression des données du schéma (Herbivorie_del.sql),
6. jeux de données comportant des insertions *valides* (Herbivorie_jeu1.sql, Herbivorie_jeu2.sql).

Résultat attendu

Les scripts `Herbivorie_req4.sql` et `Herbivorie_jeu4.sql` **complétés, documentés et présentés** conformément au standard [STD-SQL-01_NT]. Le premier script doit comprendre chacune des requêtes demandées en annexe. Le second script doit comprendre les données d'essai valides utiles à la démonstration des requêtes.

Critères d'évaluation

La correction est basée, entre autres, sur le fait que les livrables soient clairs (c'est-à-dire lisibles et compréhensibles), exacts (c'est-à-dire sans erreurs et sans ambiguïtés), concis (c'est-à-dire sans éléments superflus), complets (c'est-à-dire comprenant tous les éléments requis).

Démarche

Dans le cadre du travail dirigé en laboratoire, chaque équipe doit :

- ◇ réaliser une première ébauche des programmes demandés en répétant les cinq étapes d'un développement en mode itératif, à savoir : (a) fixer un objectif restreint, (b) rédiger des cas de test mettant en évidence l'effet de la modification (cas positifs et négatifs), (c) écrire ou modifier le code SQL, (d) tester ;
- ◇ passer en revue les programmes afin de les annoter pour y inclure les tâches encore à accomplir.

Après le travail dirigé, chaque équipe doit, dans le cadre de son travail pratique :

- ◇ compléter les programmes ébauchés en travail dirigé ;
- ◇ réviser les programmes pour en retirer les erreurs ;
- ◇ commenter les programmes.

Durant le travail dirigé, les étudiants utiliseront les postes de travail du laboratoire. Les outils nécessaires au travail demandé y sont mis à disposition. L'accès aux ressources est contrôlé par le CIP et le mot de passe associé.

Dans le cadre du travail pratique, l'étudiant peut utiliser les ressources du laboratoire durant les heures prévues à cet effet. Il peut aussi utiliser ses ressources propres. Dans ce dernier cas, il est libre d'utiliser la plateforme et les outils de son choix, dans la mesure où les programmes livrés sont exécutables sans modifications dans l'environnement du laboratoire.

Plan de travail durant la période du travail dirigé

	Début	Durée	Étape	Mode	Description
1	00:00	00:10	Introduction	Exposé	Présentation du laboratoire par l'enseignant
2	00:10	00:05	Poste de travail	Tutorat	Mise en route du poste de travail
3	00:15	00:05	Dossiers	Tutorat	Repérer les dossiers de travail et copier les fichiers
4	00:20	00:05	Environnement	Tutorat	Établir la connexion, fixer les paramètres d'environnement
5	00:25	00:20	Élaboration A	Tutorat	Élaboration itérative et progressive du schéma
6	00:45	00:10	Pause		Mise au point par l'enseignant
7	00:55	00:25	Élaboration B	Exercice	Élaboration itérative et progressive du schéma
8	01:20	00:10	Revue	Tutorat	Revue du travail et amélioration des commentaires
9	01:30	00:20	Diffusion	Tutorat	Transmission par courriel aux membres de l'équipe
10	01:50		Fin		

Modalités de remise

Le travail pratique est à remettre selon les modalités prescrites par le tuteur.

Références

[STD-SQL-01_NT]

LUC LAVOIE. *Standard de programmation SQL, niveau 1.*

Version 044c ; CoLOED-GRIIS-MAD, septembre 2024 ;

https://github.com/llavoie-qc/IFT187/blob/main/STD-SQL-01_NT.pdf

[EPP]

LUC LAVOIE. *Analyse de données écologiques relatives à l'herbivorie du trille.*

Version 022a ; CoLOED, février 2025 ;

https://github.com/llavoie-qc/IFT187/blob/main/S06/LAB4/Herbivorie_EPP.pdf

Annexe – Demandes à réaliser pour le LAB4

- Y1. Ajouter les types et les tables requises pour consigner les conditions météorologiques quotidiennes.
Pour chaque journée consignée, donner la température minimale, la température maximale, le taux d'humidité minimal, le taux d'humidité maximal, le nombre de millimètres de précipitation et la nature de la précipitation. Le cas échéant, choisir les unités appropriées. **Recommandation** : appliquer les principes de gestion des informations absentes vues en cours.
- Y2. Ajouter des données météorologiques vraisemblables et représentatives pour le mois de juin 2016.
Définir une table CarnetMeteo afin de contenir des données météorologiques brutes (pouvant contenir des erreurs) en provenance du terrain. Un suggestion relative à la strcture de CarnetMeteo est donnée ci-après. Alimenter ensuite les tables créées en Y1 en faisant les vérifications requises. N'insérer que les données valides et intègres. Définir les fonctions requises pour ce faire. Proposer un jeu de données qui illustre l'adéquation de votre alimentation.
- Y3. Définir une vue donnant les conditions météorologiques complètes hors précipitation.
Maintenir les mêmes identifiants d'attributs qu'en Y1.
- Y4. Retirer les données météorologiques pour une période donnée (date de début, date de fin) si la température minimale rapportée est en deçà d'une température donnée. Définir une procédure.
- Y5. Augmenter les températures rapportées d'un pourcentage donné durant une période donnée (date de début, date de fin). Définir une procédure.
- Y6. Définir l'assertion requise de la table Taux.
Dans le script de création, on suggère de vérifier que les intervalles associés aux catégories ne se chevauchent pas. Mettre en oeuvre l'assertion requise à l'aide d'une fonction.
- Y7. Modifier la table ObsFloraison.
Celle-ci ne doit refléter que la date de la première constatation que le plant est porteur d'une fleur ou d'un fruit ; en conséquence, il faut retirer l'attribut « fleur », modifier la contrainte de clé candidate et épurer les données. Faire cette opération sans perdre les données pertinentes à l'aide, notamment, de commandes ALTER.
- Y8. Modifier le nom d'un attribut (ou refuser de le faire en donnant une argumentation motivée).
Changer le nom de l'attribut « peup » de la table Peuplement pour « peuplement ». Faire cette opération sans perdre les données pertinentes à l'aide, notamment, de commandes ALTER.

Suggestion

```
CREATE TABLE CarnetMeteo
-- Carnet de terrain contenant les données brutes des observations météorologiques.
-- La table est utilisée afin de vérifier les données en vue de leur insertion
-- dans le modèle de données.
(
temp_min  text,      -- la température minimale,
temp_max  text,      -- la température maximale,
hum_min   text,      -- le taux d'humidité absolue minimal (en pourcentage),
hum_max   text,      -- le taux d'humidité absolue maximal (en pourcentage),
prec_tot  text,      -- les précipitations totales (en mm),
prec_nat  text,      -- la nature des précipitations (un texte codifié à définir),
date      text,      -- date de la prise de données
note      text       -- note supplémentaire à propos des conditions du jour
);
```