
SAE BD/MATH.
SAE n°S4.03
Au delà du relationnel

1 La SAE

Le but de cette SAE est de vous fournir une vision générale des différents modèles de bases de données et des langages de requêtes utilisés pour les interroger. Cette SAE a pour objectif de vous amener à : (1) faire preuve d'initiative en recherchant des informations et en consultant la documentation technique, (2) appliquer les connaissances acquises dans la matière principale (la ressource R4.03), ainsi que dans d'autres domaines de votre formation, tels que les mathématiques et l'anglais.

La tâche centrale de cette SAE consiste à implémenter des requêtes dans différents modèles de données et les langages de requêtes associés.

2 L'organisation du travail

La SAE-S4.03 sera réalisée en **groupe de trois étudiants**, encadrés par les enseignants lors des séances de discussion. L'évaluation est individuelle. Chaque participant du groupe doit maîtriser toutes les parties du travail.

3 Bases de données

3.1 Spécification générale

Dans chaque section, vous devez créer une petite base de données. Nous allons considérer des données concernant des parcours de randonnées. Selon la section, il suffit de créer une partie de la base.

Base RANDONNEE

Nous considérons une base de données simplifiée qui stocke des informations des tronçons faisant partie des parcours de randonnée. Un tronçon a un identificateur et est caractérisé par un point de départ et un point d'arrivée. Nous pouvons avoir plusieurs tronçons différents entre un point de départ et un point d'arrivée. Pour chaque tronçon, la base stocke, son nom usuel (s'il existe) ; la distance à parcourir, en kilomètre ; le type de sol (terre, bitume, gravier, etc), la dénivellé moyenne entre les points de départ et d'arrivée. Un point d'arrivée ou de départ est considéré comme une étape. La base stocke des informations sur les étapes : codeEtape (unique) nom de l'étape (par exemple, *Prati*, *L'Onda*), latitude, longitude et son niveau de confort. Le niveau de confort d'une étape est défini par un indice de catégorie. Il existe 10 catégories de confort (pour indiquer, par exemple que l'étape correspond à une ville, à un refuge, à un placement de camping, etc). La base stocke une description pour chaque catégorie.

3.2 Modèle relationnel

1. Créer votre base de données relationnelle à partir de la spécification donnée dans la section 3.1. Vous devez créer toutes les tables.
2. Nous souhaitons trouver des parcours de randonnées. Écrire en SQL (Oracle) les requêtes suivantes :
 - (a) Donner les noms des étapes que nous pouvons atteindre directement en suivant les tronçons qui partent d'Orléans.
 - (b) Veuillez fournir la liste des étapes accessibles depuis Orléans avec une seule étape intermédiaire.
 - (c) Veuillez fournir la liste des étapes accessibles depuis Orléans, avec un nombre quelconque d'étapes intermédiaires.

REMARQUE : Afin de permettre le traitement des requêtes récursives, des améliorations ont été apportées au langage SQL. Vous devez étudier ces enrichissements, en vous référant aux sources suivantes :

- <https://oracle-base.com/articles/11g/recursive-subquery-factoring-11gr2#basic>
- <https://www.youtube.com/watch?v=RI59HaceYIc>

3.3 Modèle objet-relationnel

Nous souhaitons modéliser les informations de nos tronçons en utilisant un modèle objet-relationnel. Dans ce cadre, nous envisageons de faire une table qui contient une autre table (informant sur les points d'intérêts sur un tronçon) et une liste des deux indices de qualité, avec leur poids actuel. Chaque tuple de la table représente un tronçon. La figure 1 présente un exemple d'une petite instance du modèle.

idTroncon	nomUsuel	pointDep	pointArr	distance	Points d'interet		IndicesQualite
					Nom	Type	
t345	chemin vert	Orleans	Olivet	10	Jardin de plantes Monument aux morts Eglise XX	jardin public monument lieu de culte	[(dificulte, 3, 4), (securite, 4, 5)]

FIGURE 1 – Exemple d'une instance dans le modèle objet-relationnel

1. Utiliser ORACLE pour implémenter votre base dans le modèle objet-relationnel. Vous devez définir les types complexes comme mentionné dans nos cours. Il suffit d'implémenter la partie de la base concernant les tronçons, comme illustré dans la figure 1 (mais vous pouvez y ajouter d'autres attributs).
2. Écrire les requêtes suivantes
 - (a) Pour chaque tronçon, donner le nombre de points d'intérêt, par type.
 - (b) L'*impact* d'un indice de qualité est donné par le produit de sa valeur et du poids que lui est attribué. Pour chaque tronçon, indiquer l'impact de chaque indice de qualité.

3.4 Modèle Logique

Pour cet exercice, vous pouvez utiliser AbcDatalog, une implémentation de Datalog disponible à l'adresse suivante : <https://abcdatalog.seas.harvard.edu/>. Vous pouvez télécharger

le fichier .jar correspondant depuis CELENE, sur le site de notre SAE. Avant de commencer, nous vous recommandons de lire attentivement les instructions sur la syntaxe de cette implémentation sur le site web d'AbcDatalog.

Considérons une base de données RANDONNEE qui contient des informations sur les vols directs.

1. Identifiez quels sont les prédicats extensionnels de votre base de données déductive.
2. Écrivez un programme Datalog qui permet de lister toutes les étapes connectées par des trajets de randonnées (directs ou avec étapes intermédiaires) possibles à partir de d'une instance de la base donnée.
3. Écrivez un programme Datalog qui permet de lister les parcours qui passent uniquement par des tronçons en terre.
4. Écrivez un programme Datalog qui permet de lister toutes les étapes connectées par des trajets de randonnées ayant un nombre impair de tronçons.
5. Écrivez un programme Datalog qui permet de lister toutes les étapes connectées par des trajets correspondant à un circuit (c'est-à-dire, qui commence et fini à la même étape).

3.5 Modèle graphe

Pour cet exercice, vous devez utiliser le SGBD Neo4J et le langage Cypher.

1. Proposez une modélisation de la base de données RANDONNEE dans le modèle graphe. Expliquez vos choix de modélisation. Ici, vous devez prendre en compte toute la spécification de la section 3.1.
2. Implémentez votre base de données en Neo4J.
3. Écrivez les requêtes suivantes en Cypher Faites différentes versions :
 - (a) Trouvez des tronçons qui ont les mêmes points de départs et d'arrivée, mais de types de sol différents.
 - (b) Listez les trajets (avec ville de départs et ville d'arrivées) qui partent d'Orléans.
 - (c) Listez les trajets (avec ville de départs et ville d'arrivées) qui arrivent à Orléans.
 - (d) Listez les trajets qui sont des circuits, c'est-à-dire le point d'arrivée est identique au point de départ.
 - (e) Refaites les requêtes 3b-3d ci-dessus, mais en affichant les noms des étapes intermédiaires.

Vous trouverez sur CELENE des instructions pour installer Neo4J.

4 Évaluation

L'évaluation du projet SAE sera effectuée individuellement par un contrôle, dans lequel chaque étudiant devra démontrer une maîtrise complète de toutes les parties du projet. L'examen portera sur les concepts manipulés pendant le projet SAE ainsi que sur les particularités du projet de chaque étudiant. La modalité exacte du contrôle est à définir, mais, il s'agira, très probablement, des questions sur papier.

CALENDRIER : Contrôle le (TBA)

RENDUS :

- *Chaque étudiant* doit fournir un dossier en version papier qui contient toutes les requêtes réalisées, ainsi que les explications demandées dans le projet. Ce dossier doit être annexé à sa copie d'examen et pourra être consulté pendant l'évaluation pour aider l'étudiant à répondre aux questions.
- Chaque groupe doit soumettre les scripts des requêtes réalisées sur CELENE, accompagnés de leur jeu de tests, avant la date limite. Cela permettra éventuellement de les exécuter lors de l'évaluation. *Veuillez noter qu'un seul rendu par groupe est requis.*

5 Rappel : Instructions pour travailler sur Oracle

5.1 La routine de travail

1. Travailler avec un éditeur. Faites vos requêtes dans un fichier que vous nommez, par exemple, *tpX.sql*.
2. Ouvrir un terminal dans le même répertoire de votre fichier *tpX.sql* et lancer la connexion Oracle.
3. Vous pouvez lancer le script de *tpX.sql* en faisant : `@tpX` dans SQLPLUS ou faire copier-coller de chaque requête. Il est important d'avoir votre fichier avec vos requêtes (que vous sauvegardez).
4. Faire des fichiers séparés. Par exemple : un fichier *creationBD1.sql*, pour la création de la base ; un fichier *insertionBD1.sql* pour les insertion dans la base BD1 ; un fichier *reqBD1.sql* pour les requêtes sur la base BD1.

5.2 Des commandes en vrac

- Connectez vous à Oracle. À l'IUT vous devez faire :
`sqlplus <nom>@ora12`
ou mieux
`rlwrap sqlplus <nom>@ora12`
qui donnera plus de souplesse à l'édition en ligne.
- Changez votre mot de passe :
`alter user <login> identified by <nouveauMotPasse>;`
ou utiliser *passwd* comme sur un shell.
- Pour augmenter la taille des lignes : `set LINESIZE 500`
- Pour diminuer la taille d'une colonne : `column cname format A15`
- Pour voir l'ensemble de tables : `select * from tab;`
- Pour effacer les tables qui apparaissent après un drop : `purge recyclebin;`
- Pour voir le schéma d'une table : `describe nomTable;`
- Pour faire la suppression de toutes vos tables (avec les contraintes). La requête
`select 'drop table" ' || table_name || ' " cascade constraints; ' from use_table;`
vous donne comme réponse la liste de tables à supprimer. Lancez ensuite le script généré.
- Database SQL Language Quick Reference
https://docs.oracle.com/cd/E11882_01/server.112/e41085/sqlqr01001.htm#SQLQR110