L3 INFORMATIQUE

Examen de Base de Données **6 janvier 2025**
Durée : 2 heures **Aucun document autorisé**

Exercice 1 : Github (3 points)

GitHub est une plateforme qui permet de gérer des dépôts de code source (*repositories*), de collaborer sur des projets, de gérer des utilisateurs et leurs permissions, et d'organiser les tâches à travers des *issues*.
On vous demande de construire une base de données relationnelles pour assurer la gestion d'une telle plateforme. Vous avez à gérer les informations suivantes :

Chaque utilisateur, identifié par **user_id**, a un nom **user_name**, un mot de passe **password_hash**. Un utilisateur peut posséder/créer plusieurs dépôts et effectuer diverses actions sur ces dépôts.

Un dépôt (*repository*), identifié par **repo_id** de nom **repo_name,** est l'endroit où le code source est stocké. Les dépôts peuvent être publics ou privés (**visibility**). Les collaborateurs sont des utilisateurs qui ont accès à un dépôt, ils disposent d'une **autorisation** en lecture, en écriture ou en administration sur le dépôt.

Une *issue*, identifiée par **issue_id,** est une tâche ou une demande créée par un utilisateur dans un dépôt. Elle peut être utilisée pour signaler des bogues, demander des fonctionnalités ou poser des questions. Une *issue* a une description détaillée (**description**), un **statut** (de valeur soit *open* soit *closed*) et une date de création (**created_at**).

Question 1 : Donner le graphe minimum des dépendances fonctionnelles de la relation R qui regroupe tous les attributs :
R(user_id, user_name, password_hash, repo_id, repo_name, visibility, autorisation, issue_id, description, statut, created_at)

Question 2 : Quel est (sont) le (les) identifiant(s) de R ? En quelle forme normale est R ? Justifiez-le.

Question 3 : Si R n'est pas en troisième forme normale, proposez une décomposition de R sans perte de tuple ni de dépendance fonctionnelle, qui constitue un schéma en troisième forme normale au moins.

Exercice 2 : PFAS (7 points)

Les substances per- et polyfluoroalkyles (PFAS) sont des produits chimiques synthétiques largement utilisés dans l'industrie, l'alimentation et la santé. Les PFAS sont souvent qualifiés de "polluants éternels" en raison de leur persistance dans l'environnement et dans l'organisme. L'exposition aux PFAS peut provoquer des cancers, des perturbations endocriniennes, … Pour ces raisons, plusieurs réglementations ont été mises en place. Des prélèvements sont effectués régulièrement sur différents sites et analysés afin de détecter la présence éventuelle de PFAS.

PFAS(id_PFAS, nom, formule_chimique, date_decouverte)
Par exemple, PFHxA de nom acide perfluorohexanoïque, de formule chimique C6HF13O2 a été découvert le 01 janvier 1980.

Source(id_source, nom_source, description)
Une source possible de PFAS est la « lutte contre les incendies » ayant pour description « mousse anti-incendie ».

PFAS_Source (id_PFAS, id_source)

Effet_Sanitaire(id_effet, nom_effet, description)
Un exemple d'effet sanitaire est « problème de reproduction » avec pour description « effets néfastes sur la fertilité ».

PFAS_Effet(id_PFAS, id_effet)

Reglementation(id_reglementation, nom_reglementation, limite, date_entree_en_vigueur)
Par exemple, la réglementation REACH (*Enregistrement, Évaluation, Autorisation et Restriction des Substances Chimiques*) impose une limite de 0.1 µg/l (microgramme par litre) depuis le 1 juin 2007.

Site(id_site, nom_site, type_site)
L'attribut type_site prend les valeurs industriel, urbain, rural, aquatique.

Resultat_Analyse(id_site, id_PFAS, date_analyse, concentration)

Questions :
Écrire en SQL les requêtes suivantes :

1. Afficher le nom des PFAS avec le nom de leurs sources
2. Afficher les effets sanitaires associés à un PFAS spécifique. On vous demande une requête paramétrée sur l'attribut id_PFAS.
3. Compter le nombre de PFAS par source
4. Afficher les PFAS qui n'ont pas d'effets sanitaires documentés
5. Afficher le nom des sites et l'id des PFAS ayant un/des niveaux pour le PFAS dépassant la concentration réglementaire la plus importante
6. Afficher les PFAS détectés dans tous les sites industriels

Exercice 3 : Avis sur des produits (4 points)
Soit les relations
Produits(produit_id, nom, prix)
Avis_Produits(avis_id, produit_id, client_id, note, commentaire)
où la relation Produits contient des produits et la relation Avis_Produits stocke les avis de clients sur les produits.

Écrire un trigger qui insère un nouvel avis sur un produit dans la relation Avis_Produits. Vous devez vous assurer qu'un client ne puisse pas donner plusieurs avis sur le même produit et gérer le cas où un client tente de mettre un avis sur un produit qui n'existe pas.

Exercice 4 : Transactions SWIFT (6 points)
Les transactions SWIFT (*Society for Worldwide Interbank Financial Telecommunication*) sont des messages utilisés pour effectuer des paiements internationaux entre banques.
SWIFT(id_transaction, devise, montant, date_transaction, id_banque_envoi, id_banque_receveur)
Par exemple le code SWIFT BNPAFRPPXXX correspond à **BNPA** (BNP Paribas) avec **FR** représentant le code du pays (ici la France), **PP** le code de localisation (ici Paris) et **XXX** le code de l'agence (facultatif).
PAYS(Code_pays, Risque)
On vous demande de détecter des schémas de paiements inhabituels dans les transactions SWIFT. Une transaction peut avoir plusieurs schémas de paiements inhabituels.

Pour cela, écrire un programme PL/SQL qui détecte des schémas de paiements inhabituels en analysant les transactions pour identifier :

1. des bénéficiaires inhabituels, c'est-à-dire la banque bénéficiaire (celle qui reçoit la transaction) appartient à un pays classé à haut risque ;
2. des fréquences élevées, c'est-à-dire plusieurs transactions dans un laps de temps court entre les mêmes banques, par exemple plus de 3 transactions en moins de 10 minutes ;
3. des montants répétés, c'est-à-dire le même montant entre les mêmes banques dans un laps de temps court plusieurs fois.

Votre programme doit mettre à jour la relation
Alertes_Fraude(id_alerte, id_transaction, date_alerte, raison)
L'attribut raison prend pour valeur 'Pays à risque', 'Fréquence élévée' ou 'Montant répété'.

Pour rappel :

- La fonction SUBSTR(chaîne, m, n) permet d'extraire de chaîne n caractères depuis la m^{ième} position, par exemple SUBSTR('Bonjour', 3, 2)renvoie nj
- (Date1 – Date2) * 24 * 60 renvoie la différence en minutes entre deux dates