CY TECH	**Examen Traitement et Administration des données** **Session normale**	
	ING2-GIA	**Année 2024-25**

Modalités :

- Durée : 2 heures.
- Aucun document n'est autorisé.
- Toutes vos affaires (sacs, vestes, trousse, etc.) doivent être placées à l'avant de la salle.
- Aucune question ne peut être posée aux enseignants, posez des hypothèses en cas de doute.
- Aucune machine électronique ne doit se trouver sur vous ou à proximité, même éteinte.
- Aucune sortie n'est autorisée avant une durée incompressible d'une heure.
- Les déplacements et les échanges ne sont pas possibles.

QCM (4 points) :

1. **Les index Bitmap sont particulièrement adaptés pour :**
   a) Des colonnes avec peu de valeurs distinctes
   b) Des colonnes fortement mises à jour
   c) Des requêtes OLTP avec un grand nombre d'inserts
   d) Des tables stockant des images et documents volumineux

2. **Quel facteur impacte directement la performance d'une requête SQL ?** *(Plusieurs réponses possibles)*
   a) L'utilisation d'index appropriés
   b) La structure du plan d'exécution
   c) Le nombre d'arguments dans une fonction PL/SQL
   d) La fragmentation de la table

3. **Quels sont les rôles prédéfinis en Oracle ?** *(Plusieurs réponses possibles)*
   a) DBA
   b) CONNECT
   c) SYSTEM
   d) RESOURCE

4. **Quelle affirmation est vraie concernant les transactions ?**
   a) Une transaction commence automatiquement après un COMMIT
   b) Une transaction est un ensemble d'opérations traitées comme une seule unité
   c) ROLLBACK valide les modifications
   d) SAVEPOINT termine la transaction

5. **Quelle est la principale différence entre une vue matérialisée avec REFRESH ON COMMIT et REFRESH ON DEMAND ?**
   a) REFRESH ON COMMIT met à jour la vue manuellement
   b) REFRESH ON COMMIT met automatiquement à jour la vue lorsqu'une modification est effectuée sur les tables sous-jacentes
   c) REFRESH ON DEMAND nécessite une mise à jour manuelle avec DBMS_MVIEW.REFRESH
   d) REFRESH ON DEMAND est uniquement utilisable avec les tables temporaires

6. **À quoi sert la commande EXPLAIN PLAN en SQL ?**
   a) Insérer un plan d'exécution dans la base de données
   b) Afficher le résultat d'une requête SQL
   c) Analyser et afficher le plan d'exécution d'une requête SQL
   d) Exécuter une requête plus rapidement

7. **Une table contenant des factures subit 20 000 modifications et plusieurs milliers d'interrogations par minute. Sans plus d'informations détaillées sur les requêtes, quels types d'index pourraient être les plus adaptés ? Choisissez la ou les bonnes réponses :**
   a) Un index B-tree sur la colonne num_facture
   b) Un index bitmap sur la colonne total
   c) Un index B-tree sur la colonne tva
   d) Un index bitmap sur la colonne num_facture
   e) Un index bitmap sur la colonne tva
   f) Un index B-tree sur la colonne total
   g) Aucune de ces réponses n'est correcte

8. **À propos du langage PL/SQL, quelles affirmations sont correctes ?**

   a) Un bloc PL/SQL se compose des sections DECLARE, BEGIN, EXCEPTION et END.
   b) Les exceptions personnalisées ne peuvent pas être définies dans PL/SQL.
   c) Les curseurs explicites doivent être ouverts, traités, puis fermés manuellement.
   d) Une procédure stockée peut retourner une valeur comme une fonction.
   e) Le mot-clé RAISE permet de déclencher une exception.

Questions de réflexion (4 points):

1. Pourquoi est-il recommandé d'utiliser des rôles plutôt que d'attribuer directement des droits aux utilisateurs ?
2. Quelle est la différence entre un tablespace et une table en base de données ?
3. Expliquez dans quel cas une vue matérialisée est plus performante qu'une vue simple et comment gérer son rafraîchissement de manière efficace dans un environnement transactionnel.
4. Quelles sont les principales stratégies pour la fragmentation des données dans une base distribuée ?
5. Quand peut-on décider de la mise en place d'un cluster ?

Problème étudié : Gestion des locations de véhicules (12 points)

Nous souhaitons gérer les locations de véhicules auprès d'une entreprise de location automobile. Le modèle logique de données associé à une base de données centralisée est le suivant :

- **Client**(no_client, nom, prenom, date_naissance, permis_conduire)
- **Vehicule**(no_vehicule, marque, modele, annee, categorie, tarif_journalier)
- **Agence**(no_agence, nom, adresse, ville, cp)
- **Location**(no_location, date_debut, date_fin, montant_total, statut, #no_client, #no_agence)
- **Vehicule_Location**(#no_location, #no_vehicule)
- **Facture**(no_facture, date_facture, montant, paye, #no_location)
- **Cumul_Depenses_Client**(#no_client, annee, montant_total)

Règles générales :

- Un client peut louer un ou plusieurs véhicules via une agence de location.
- Une location a un statut qui peut être *en cours*, *terminée* ou *annulée*.
- Une facture est générée pour chaque location et peut être payée ou en attente de paiement.
- Le montant total d'une location est calculé en fonction du tarif journalier du véhicule et de la durée de la location.
- Une relation permet de suivre, pour chaque client et chaque année civile, le montant total dépensé en locations.
- Chaque agence gère ses propres locations et véhicules, mais les responsables d'agence peuvent consulter toutes les informations du réseau.

Règle impliquant une répartition verticale :

- Les données clients (table *Client*) et les données de facturation (table *Facture*) sont gérées par un système centralisé au siège de l'entreprise.
- Chaque agence possède uniquement les informations nécessaires sur les véhicules et les locations qu'elle gère, mais ne stocke pas les informations personnelles des clients ni les factures.
- Pour consulter ou modifier les informations d'un client ou générer une facture, une agence doit faire appel à la base centrale.

Règle impliquant une répartition horizontale :

- La table *Location* est répartie en fonction des agences :
  - Chaque agence possède et gère uniquement les locations effectuées dans son établissement.
  - Lorsqu'un client loue un véhicule dans plusieurs agences, ses différentes locations sont stockées dans les bases de données correspondantes à ces agences.
  - Une requête globale doit être mise en place pour permettre au siège de suivre l'historique des locations d'un client à travers toutes les agences.

Partie A – PL/SQL

On suppose que l'on est dans une base de données centralisée.

1. Écrire un trigger qui gère la clé de la location lors de sa création. On suppose qu'une séquence *LocID* existe. (1 pt)
2. Écrire un trigger qui vérifie qu'un client possède un permis de conduire valide avant de lui attribuer une location. (2 pts)
3. Écrire une procédure qui permet d'ajouter un véhicule à une location existante. Cette procédure reçoit le numéro de location et le numéro du véhicule. (2 pts)
4. Écrire une fonction qui reçoit un entier *n* et affiche le *n*-ième et le *(n+1)*-ième client ayant loué le plus de véhicules dans l'année en cours. (2 pts)
5. Pourquoi doit-on associer un trigger à la gestion de la table *Cumul_Depenses_Client* ? Écrivez ce trigger. (2 pts)

Partie B – Base de données répartie

1. Proposer un modèle de bases de données réparties sur différentes agences de location (exemples : Paris, Marseille, Lyon) en prenant en compte la répartition verticale et horizontale définies dans les règles de gestion. Justifier vos choix. (2 pts)

2. On suppose que dans la base de données de l'agence de Paris, il existe une table contenant la relation *Agence*. Que faut-il faire dans les bases de données des autres agences pour optimiser l'accès aux informations de cette table ? (1 pt)