

Semestre : 1 ☐ 2 ☒

 Session : Principale ☒ Rattrapage ☐

Module : Système de Gestion des Bases de Données		Documents autorisés : OUI <input type="checkbox"/> NON <input checked="" type="checkbox"/>
Enseignants : Équipe SGBD		Nombre de pages : 02
Date : 25/05/2022	Heure : 12h30	Nom :
Durée : 1h30		Prénom :
		Classes : 3A-3B

Étude de cas :

Nous nous intéressons à la gestion des données d'une agence de voyages. Cette agence propose des circuits touristiques, composés d'une ou plusieurs journées de voyage. Les clients peuvent réserver ces circuits selon plusieurs critères : nombre de places disponibles, paiement de la réservation, etc. Le schéma relationnel de l'application est donné ci-dessous :

Circuits (idCircuit, intitule, description, prixCircuit, nbPlaceDisponible, idVol#)
 Clients (CodeClient, nom, prenom, dateNaissance, fonction)
 Vols (IdVol, villeDepart, villeDestination, dateDepart, heureDepart, dateArrivee, heureArrivee)
 Reservations (idReservation, codeClient#, idCircuit#, dateReservation, nbPlaceReserve, Acompte, EtatReservation)

Remarques :

- EtatReservation peut être (*confirmé, annulé*).

Travail demandé :

Écrire les requêtes SQL et les blocs PL/SQL permettant de :

1. Créer une séquence **SEQ_RES** qui génère des numéros de 1 à 300 avec un pas de 1. Cette séquence ne génère pas la même valeur. **(1pt)**
2. Créer une vue en mode consultation nommée **V_DETAILS_RES** permettant de lister les noms, les prénoms des clients ainsi que l'intitulé des circuits pour les réservations **confirmées**. **(2pts)**
3. **A.** Créer une fonction stockée **FN_NBCIRCUITS** permettant de retourner le nombre des circuits n'ayant aucune réservation. **(2pts)**
B. Tester cette fonction. **(0.5pt)**
4. Créer un bloc anonyme permettant d'afficher les villes de départ dont le nombre total de vol est supérieur à 50. Afficher les villes et le nombre de vol. **(2.5pts)**

5. Créer une procédure stockée **PROC_INSERTION_RES** qui permet d'ajouter une nouvelle réservation. La procédure doit gérer les exceptions suivantes : **(4pts)**
- Le client n'existe pas.
 - Le circuit n'existe pas.
 - Le montant de l'acompte ne doit pas être inférieur à $\frac{1}{3}$ du prix du circuit.

Remarques :

- La colonne « **idReservation** » est alimentée par la séquence **SEQ_RES**.
 - La colonne « **dateResevation** » est alimentée par la date système.
 - La colonne « **EtatReservation** » reste nulle.
 - Les colonnes « **codeClient** » et « **idCircuit** » sont de type numérique.
6. A. Créer une procédure stockée **PROC_TOP3_CLIENTS** qui affiche le top des 3 clients qui ont passé le plus de réservation. Afficher leurs noms et prénoms. Cette procédure retourne en plus le nombre total des réservations effectuées par ces 3 clients. **(3pts)**.
- B. Tester la procédure. **(0.5pt)**
7. Créer un trigger **TRIG_NBRPLACE** qui permet de mettre à jour la colonne « **nbPlaceReserve** » du circuit après chaque nouvelle affectation ou retrait d'une réservation. **(2.5pts)**
8. Créer un trigger **TRIG_CONTROL** qui empêche toute modification sur la table « **Vols** » dont la date de début dépasse la date de fin prévue et pendant les jours fériés suivants : le 1er janvier et le 1er mai. **(2pts)**

BON TRAVAIL