

Production finale d'intégration, la PME SimbaPlus

- Ce travail sera réalisé en équipe de deux maximum;
- À terminer pour le mardi 17 décembre 2019;
- Le travail compte pour 20 % de la note finale;

Objectifs

Ce travail vise principalement à vous faire expérimenter les aspects suivants:

- Concevoir un modèle de base de données;
- Écrire de procédures stockées;
- Écrire de déclencheurs;
- Expérimenter l'aspect sécurité des données;
- Intégration avec un langage de programmation;
- Travailler en équipe;

Mise en situation

La compagnie SimbaPlus est une petite PME qui œuvre dans le développement de jeux vidéo. Sa base de données et une base de données MS SQL Sever

Cette compagnie compte deux départements distincts, « Ressources humaines (**RSH**) » et « Développement (**DEV**) ». Chaque département a sa propre base de données.

Le département RSH compte 3 employés : 1 technicien : Martin St-Louis, un commis Hugo Lapointe et le responsable du département **Saturne Lune**.

Le département DEV compte 4 employés : 3 développeurs (Guy Tares, Bien Thierry, et Martin Lejeune) et le responsable du département **Alain Patoche**.

Authentification au serveur MS SQL

Les tâches spécifiques à la gestion de « MS SQL Server » telles que créer et supprimer une base de données, gérer les rôles et autorisations, créer les connexions et les usagers des bases de données ainsi que de créer les tables dans la base de données **dbrsh** seront réservées uniquement à l'administrateur du serveur qui est Mathieu Powers

Pour vaquer à ses actives, l'**administrateur** s'authentifiera au serveur avec la connexion **vos_initiales_dba**.

Chaque employé de l'entreprise aura sa propre connexion pour s'authentifier au SGBDR. Ces connexions n'auront d'autres rôles que **public**.

Finalement, le jeu **Trivial Pursuit** aura aussi sa propre connexion, **trivial**, qui sera utilisée par l'interface graphique pour se connecter au SGBDR. Cette connexion n'aura d'autres rôles que **public**.

Les bases de données de l'entreprise

Chacun des départements à sa propre base de données. Les données propres au département des « Ressources humaines (**RSH**) » sont conservées dans la base de données **dbrsh** et celles propres au département « Développement (**DEV**) » dans la base de données **dbdev**.

Base de données dbrsh

La base de données **dbrsh** contiendra les 3 tables décrites ci-dessous.

Employes	Departements	Emplois
<u>IdEmploye</u>	<u>Code</u>	<u>Code</u>
NomEmploye	NomDepartement	Description
PrenonEmploye		SalaireMax
AdresseEmploye		SalaireMin
SalaireEmploye		
CodeDepartement		
CodeEmploi		

Le salaire des employés est contrôlé par un trigger CTRLSalaire : Lors de l'ajout d'un nouvel employé, ou lors de la modification du salaire, nous devons vérifier que le salaire de celui-ci se retrouve dans la fourchette des salaires. Les salaires sont en fonction des TypeEmploi et sont dans la table EMPLOIS

Initialement, la table Employes contient tous les employés de l'entreprise.

Initialement, la table Departements contient au moins les départements **RSH** pour ressources humaines et **DEV** pour Développement.

La table **emplois** contient les informations suivantes.

Code	Description	Saliremax	Saliremin
DBA	Data Base Administration	300 000	120 000
RES	Responsable	180 000	80 000
DEV	Développeurs	150 000	40 000
COM	Commis	60 000	25 000

Base de données dbdev

La base de données **dbdev** contiendra les tables du jeu Trival Pursuit (votre TP3) ainsi que la table **Avoirs** décrite ci-dessous.

Les joueurs dans la table joueurs sont les employés du département DEV.

Colonnes	Description
noJoueur	Référence à un joueur de la table joueur
nbOr	Nombre de pièces d'or
nbArgent	Nombre de pièces d'Argent
nbBronze	Nombre de pièces de bronze

Sécurité et intégrité des données au RSH

Pour respecter les règles de sécurité, l'administrateur (DBA) Mathieu Powers décide de ne pas donner tous les droits à tous les employés. Il décide plutôt d'y aller par attribution de rôles. Pour chaque type d'emploi, un rôle est défini.

Voici un tableau qui résume un peu la structure de la PME concernant les droits et les privilèges.

Roles	Privilèges	Membres
RoleTrivial	SELECT, UPDATE, INSERT sur toutes les table de la base de données devdb .	trivial
RespRSH	SELECT, UPDATE, INSERT sur toutes les table du Département RSH. Également le privilège EXECUTE.	Responsable des RSH
RoleTech	SELECT sur toutes les tables du département RSH UPDATE de la colonne Adresse de la table Employés, UPDATE de description de typeemplois, INSERT et UPDATE de la table Départements	Techniciens
RoleCommis	SELECT dur toutes les table du département RSH	Les commis
db_owner	Propriétaire de la base de données du département DEV	Responsable du département Dev
db_datawriter	Peuvent écrire sur la BD du département DEV	Les développeurs
db_datareader	Peuvent lire sur la BD du département DEV	Les développeurs
db_ddladmin	Peuvent exécuter les commande DDL su la bd du département DEV	Les développeurs
En plus, Les développeurs doivent pouvoir exécuter les procédures stockées de la base de données de leur département		

Tableau 1:Tableau 1: Rôles et autorisations

Ce qu'on vous demande :

1. Créer les bases de données des deux départements, RSH et DEV
2. Créer les tables pour le département RSH. Puis insérer les données.
3. Créer les logins avec les utilisateurs mappés sur les logins et la base de données correspondante
4. Attribuer les rôles serveur et bases de données s'il y a lieu
5. Créer les RÔLES non prédéfinis du tableau1
6. Ajouter les membres aux rôles en tenant compte des autorisations.
7. Faites-en sorte que les développeurs puissent exécuter les procédures stockées.
8. Programmer le trigger qui contrôle le salaire des employés. Lorsqu'un employé est inséré ou lorsque son salaire est modifié, on doit garantir qu'il est toujours à l'intérieur de la fourchette de son emploi.

Développement et sécurité des données au département DEV

Les employés du département DEV voulant se détendre durant leurs heures libres jouent au jeu **Trivial Pursuit** que vous avez développé dans le tp3.

A- Système de récompense, et questions réponses

Vous devez modifier le fonctionnement du jeu pour qu'il incorpore un système de récompense basé sur le niveau de difficulté des questions. Chaque question se verra attribuer un niveau de difficulté de 1 à 3; 1 pour facile, 2 moyen et 3 difficile. La quantité de pièces obtenues en fonction du niveau de difficulté de la question est affichée dans le tableau ci-dessous.

Niveau de difficulté	Nombre de pièces obtenus
1	5 pièces de bronze
2	10 pièces d'argent
3	15 pièces d'or

Pour mettre en place cette nouvelle fonctionnalité vous devez ajouter une colonne à la table **questions** pour y conserver le niveau de difficulté de la question.

Les récompenses ainsi obtenues seront comptabilisées dans la table **avoirs** (ci-dessous) par la procédure **validerRéponse**.

Ce qu'on vous demande :

1. Modifier la table questions pour ajouter la colonne **difficulte**. Les valeurs possibles pour cette colonne sont uniquement 1 ou 2 ou 3
2. Modifiez votre procédure **ValiderRéponse** pour que celle-ci comptabilise les pièces d'or, d'argent ou de bronze associées à la question. Les récompenses sont comptabilisées dans la table **avoirs**.
3. Programmer la procédure **AjouterQuestionRéponses** qui permet d'ajouter une question avec ses réponses associées;
4. Pour garantir l'intégrité des données de la table **réponses** et s'assurer qu'il existe en tout temps qu'une seule bonne réponse par question, vous devez programmer le trigger **CTRLRéponse**.
5. L'information de la table **Avoirs** est confidentielle. Il est donc d'une importance capitale qu'un usager de la base de données du département DEV ait accès uniquement aux données qui le concernent. Ici on suppose qu'un joueur correspond à un usager. Vous devez mettre en place un mécanisme qui garantira:
 - a. Qu'un joueur puisse consulter ses avoirs et uniquement les siens
 - b. Qu'un joueur puisse mettre à jour ses avoirs et uniquement les siens.

B- Authentification des joueurs et information de crédit

Des projets sont en discussion pour implémenter un système d'achats en ligne pour les joueurs inscrits au jeu. L'aspect sécurité des données devient donc un enjeu important.

Dans un premier temps vous devez implémenter un mécanisme d'authentification des joueurs à l'aide d'un mot de passe. Le mot de passe des joueurs sera conservé dans la table **joueur** et sera chiffré à l'aide d'une fonction de hachage. Vous aurez donc à ajouter une nouvelle colonne à la table **joueurs** pour conserver ce mot de passe.

De plus les joueurs auront la possibilité d'associer un numéro de carte de crédit à leur profil dans la table **joueurs**. Votre travail est d'ajouter la fonctionnalité pour pouvoir conserver en toute sécurité le numéro de carte de crédit des joueurs.

Le numéro de carte de crédit sera crypté à l'aide d'un algorithme de chiffrement symétrique. La clé utilisée pour crypter le numéro sera le mot de passe du joueur. Cette façon de procéder garantit qu'uniquement le joueur en possession du mot de passe aura accès à l'information de crédit.

Pour mettre en place cette nouvelle fonctionnalité, vous devez programmer les procédures et fonctions suivantes.

1. Programmez la procédure **AjouterJoueur** qui ajoute un joueur à la table **joueurs** avec toutes ses informations, incluant son mot de passe.
2. Programmez la procédure **ModifierMotPasse** qui permet à un joueur de modifier son mot de passe. L'ancien mot de passe doit être fourni pour valider son identité.
3. Programmez la fonction scalaire **ValiderIdentité** pour valider l'identité d'un joueur basé sur son mot de passe. La fonction retourne 0 (succès) ou 1 (échec).
4. Programmez la procédure **AjouterInfoCrédit** qui permet d'ajouter un numéro de carte de crédit à un joueur donné. Le mot de passe du joueur doit être fourni.
5. Programmez la fonction table **ObtenirInfoCrédit** qui retourne toute l'information d'un joueur, incluant son numéro de carte de crédit, mais pas son mot de passe.

C- ADO.Net

Vous devez concevoir une petite application (C#/ADO.NET/Form) pour permettre l'ajout de questions et réponses dans la base de données du jeu Trivial Pursuit.

L'interface utilisateur doit permettre de saisir un **alias** et un mot de passe. Cette information servira à authentifier le joueur. Pour authentifier le joueur, vous utiliserez les services que vous avez implémentés dans la section précédente

Seulement après avoir été authentifié avec succès que la fonctionnalité d'ajout de questions/réponses sera activée.

Le logiciel devra utiliser la connexion **trivial** pour s'authentifier au SGBDR.

Et pour mieux se cultiver... une petite bibliothèque

Nous voulons schématiser une petite bibliothèque pour les employés de cette entreprise. Les tables de cette bibliothèque sont dans la BD **rshdb**. Les employés pourraient y emprunter des documents : Voici la description de la bibliothèque.

- La bibliothèque contient des documents. Les documents sont soit des périodiques soit des livres;
 - Tous les documents ont en commun un identifiant unique, un titre, un flag de disponibilité, et un type de document.
 - Les livres ont en plus l'année d'édition, une maison d'édition et un auteur (l'auteur principal);
 - Les périodiques ont une périodicité (hebdomadaire, mensuel, journalier) et une date de parution;
 - Les livres ont plusieurs exemplaires, un exemplaire appartient à un seul livre. Les exemplaires sont identifiés par un numéro unique;
 - Les employés peuvent emprunter des livres et uniquement des livres. (Exemplaires);
 - Chaque employé peut emprunter plus d'un livre. Un livre est emprunté par plus d'un employé. Chaque emprunt a une date de prêt, une date de retour et la date réelle de retour;
1. Donner le modèle de la base de données en 3FN.
 2. Créer toutes les tables avec toutes les contraintes d'intégrité. Y insérer quelques enregistrements.
 3. Écrire la procédure stockée **InsertLivres** qui permet d'insérer un livre. On fournit toutes les informations du livre et le **nombre** d'exemplaires. Votre procédure doit permettre de faire les insertions nécessaires dans la table exemplaires.

Le numéro d'exemplaire est la concaténation de la chaîne 'EX' suivie du numéro du livre, suivi du trait suivi d'un numéro séquentiel.

Exemple : si le livre numéro 5 est inséré et que ce livre a deux exemplaires alors les numéros d'exemplaires seront :

EX5-1 et EX5-2

4. Écrire la procédure **InsertEmprunt** qui permet d'ajouter un emprunt. La date prévue pour le retour est égale à la date de prêt plus 15 jours.
5. Écrire le trigger **CTRLDatePret** qui vérifie lors de l'insertion que la date de prêt est antérieure à la date de retour.
6. Écrire une fonction table **LivresPopulaires** qui affiche la liste des livres qui ont été empruntés plus que 3 fois. Afficher le titre, et le nom de l'auteur.

Ce que vous devez remettre

- Le schéma de la BD de la bibliothèque en 3FN, et en format PDF
- Un script (pfi_dba.sql) contenant uniquement les commandes SQL spécifiques à l'administration du serveur (Création des bases de données, connexions, usagers, rôles);
- Un script (pfi_rsh_create_db.sql), spécifique au département RSH, contenant la création des tables, l'insertion des données initiales et le trigger;
- Un script (pfi_dev_create_db.sql), spécifique au département DEV, contenant la création des tables, les procédure/fonctions et insertion des données initiales; vous pouvez partir du script de votre tp3 que vous complétez avec les développements demandés;
- Un script (pfi_biblio_create_db.sql), spécifique au projet de bibliothèque, contenant la création des tables, procédure/fonctions/trigger et l'insertion des données initiales;
- Un projet Visual Studio (ADO.NET/Form)

Modalité de remise : Boite de remise

Vous allez remettre trois fichiers **clairement identifiés par VOS NOMS**

- Le modèle de la BD
- Le projet ADO.NET, zippée
- Un dossier Zippé contenant TOUS vos fichiers SQL

Dates de remise

- 1- Le modèle de la BD au plus tard le lundi 02 décembre avant minuit
- 2- Le reste de la PFI le 17 décembre avant minuit

Barème :

Éléments évalués	Sur
Sécurité et intégrité des données au RSH	25
Développement et sécurité des données au département DEV	40 (total =15+15+10)
A- Système de récompense, et questions réponses	15
B- Authentification des joueurs et information de crédit	15
C- ADO.NET	10
Le modèle de la bibliothèque	10
Les procédures stockées, triggers de la bibliothèque et création de tables	15
Respect des consignes de remise	10
Total	100