

## Projet: Développement d'une Application de Base de Données

*Remise: dernier jour des cours (dans la boîte aux devoirs et électroniquement à un AE)*

*Valeur: 15% de la note finale*

*Remarque: à faire en groupe de 3 (ou 4 dans le cas d'étudiants en Gestion)*

*Prix: le meilleur projet recevra un prix – probablement une copie d'un manuel de BD*

---

Dans cet exercice vous appliquerez la matière des chapitres 6 et 7 pour créer une petite application qui interagit avec une base de données. Ce projet vise à étendre les designs conceptuel et logique effectués au Devoir 1 afin de créer un code JDBC qui accède à une base de données en PostgreSQL. Il vise aussi à créer les parties de l'application résidant à l'étage intermédiaire ainsi qu'à l'étage de présentation de l'architecture à trois étages vue au chapitre 7.

**Description des exigences** — Rappelez vous du design de la base de données de la banque *Main Savings Inc.* effectué au Devoir 1. Tous les aspects réalistes du design n'avaient pas été modélisés. Vous aviez seulement pris en compte un certain nombre d'aspects qui illustrent quelques uns des concepts importants du design des diagrammes ER vus en classe. La base de données de *Main Savings Inc.* avait les exigences suivantes:

- Main Savings Inc. est organisée en plusieurs branches. Chaque branche a un nom unique et est localisée dans une cité bien définie. Bien plus, chaque branche a un actif.
- Les clients de Main Savings Inc. ont un identificateur unique, un prénom et un nom de famille, une adresse et une cité de résidence. Chaque client doit avoir un employé de la banque qui lui sert de banquier. Ce dernier supervise à la fois les emprunts effectués par le client et la possession des comptes par ce dernier. Ce rôle de supervision a un type (par exemple "banquier personnel" ou "responsable des prêts").
- Les employés de Main Savings Inc. sont identifiables par un numéro d'identification. De plus, la banque stocke le prénom ainsi que le nom de famille de l'employé, son numéro de téléphone et sa date de début d'emploi. Main Savings Inc. garde aussi les traces de quel employé travaille pour quel manager.
- Main Savings Inc. offre des prêts à ses clients. Un prêt est initié dans une branche bien spécifique et peut être octroyé à un ou plusieurs clients. Chaque prêt a un numéro d'identification unique ainsi qu'un montant. Main Savings Inc. garde les traces des remboursements comme suit: un numéro de remboursement identifie chaque remboursement de chaque prêt; chaque prêt doit avoir un tel numéro de remboursement ainsi qu'une date et le montant du remboursement.
- Main Savings Inc. offre des comptes d'épargne ainsi que des comptes courants à ses clients. Chaque client peut avoir plus qu'un compte de chaque type et chaque compte peut avoir plus d'un propriétaire. Chaque compte est identifié par un numéro de compte et a une balance ainsi que la dernière date à laquelle chacun des propriétaires y a accédé. Finalement, chaque compte d'épargne a un taux d'intérêt et chaque compte courant a une indication de la hauteur du découvert permis.

En plus des exigences ci-dessus, ajoutez les exigences suivantes pour rendre le design un peu plus réaliste:

- Main Savings Inc. maintient de l'information au sujet des activités sur les comptes. Ces activités sont générées par les comptes des clients. Chaque activité au compte indique le numéro du compte concerné, une date une description (p.ex. "liste de paye", "facture", etc.), un montant et la balance en souffrance sur le compte.
- Un client de Main Savings Inc. a un numéro de téléphone, un courriel et le meilleur temps de la journée où il est accessible.
- Main Savings Inc. offre à ses clients la possibilité de payer leur factures en ligne. Un paiement de facture a un compte à partir duquel le paiement est fait, un destinataire, un montant, ainsi qu'une date.

Sur base des informations ci-dessus, faites ce qui suit:

1. **(10 points)** – Étendez le diagramme ER fait pour la base de données de Main Savings Inc. au Devoir 1 en prenant en considération les exigences additionnelles ci-dessus.
2. **(10 points)** – Traduisez votre diagramme ER en un schéma de base de données relationnelles en écrivant un script SQL en PostgreSQL. Utilisez les contraintes appropriées dans votre script SQL. Votre script doit être exécutable au serveur de SITE.

**(Soumettez votre script ensemble avec votre diagramme ER le 28 Octobre.)**

3. **(60 points)** – Écrivez un code JDBC qui réalise l'accès à la base de données ainsi que sa manipulation. Supposez que les employés de la banque ont une interface différente de celle des clients. L'interface de votre système devrait offrir un ensemble raisonnable de richesses fonctionnelles incluant les suivantes:
  - (a) Un utilisateur ouvre une session sur le système.
  - (b) Il y a un menu avec 4 rubriques: Comptes, Paiements, Virements et Profile.
  - (c) Le menu Paiements offre 2 fonctionnalités, à savoir Sommaire et Activités. Le Sommaire donne un résumé de tous les comptes d'un client, c'est-à-dire le type du compte ainsi que la balance. L'Activité donne le numéro du compte, la date, une description (p.ex. "liste de paye", "facture", etc.), un montant ainsi que la balance sur le compte concerné. Le client devrait avoir le choix de montrer les activités de chacun des comptes qu'il possède.
  - (d) Le menu Paiements offre une interface pour payer les factures. L'utilisateur peut spécifier le compte à utiliser, le bénéficiaire, le montant à payer ainsi que la date de paiement.
  - (e) Le menu Virements offre une simple interface pour virer de l'argent du compte du client à un autre compte pouvant appartenir à quelqu'un d'autre.
  - (f) Le menu Profile offre la possibilité de changer le email d'un client, son numéro de téléphone ainsi que son adresse.
  - (g) En plus des fonctionnalités ci-dessus, le personnel de la banque a accès aux données sur les employés. Trouvez quelques requêtes sur les données d'employés avec une interface ad hoc. Une telle interface comprendra des critères raisonnables de recherche.
  - (h) L'interface Java devrait être simple et fonctionnelle.

**(Soumettez cette partie au dernier jour des cours.)**

4. **(20 points)** Faites le design d'une (simple) interface web ainsi qu'une étage intermédiaire utilisant une des technologies existantes, p.ex. HTML, scripts CGI, Java servlets, pages JavaServer ou autre. Partez du code Java utilisé dans la deuxième partie du devoir et offrez les fonctionnalités décrites ci-haut. Lisez les sections 6.6 et 7.8 du manuel afin d'avoir une idée sur la manière de faire le design de l'architecture de votre système.

**(Soumettez cette partie au dernier jour des cours.)**

**Quelques Ressources:** Liens vers du matériel au sujet de JSP, Servlets, XML, JDBC, HTML, etc:

<http://www.cs.wisc.edu/~dbbook/openAccess/>

<http://www.java.sun.com/products/jdbc>

<http://www-db.stanford.edu/~ullman/fcdb/oracle.html>

<http://www.utoronto.ca/ian/books/>

<http://www.postgresql.org/docs>

**A soumettre:**

- Votre diagramme ER, script SQL et code Java.
- Tout code CGI ou autre que vous aurez écrit.
- Sortie imprimée du système qui montre clairement que ce dernier fonctionne.

Vous ferez une courte démonstration de votre système à un des assistants à l'enseignement.