



Département d'informatique

**ift287 - Exploitation de bases de données relationnelles
et orientées objet****Plan de cours
Hiver 2012**

Enseignant : Marc Frappier
Courriel : marc.frappier@usherbrooke.ca
Téléphone : (819) 821-8000 poste 62096
Site : <http://www.dmi.usherbrooke.ca/~frappier/ift287/ift287.html>
Disponibilité : sur rendez-vous par courriel

Horaire

Cours théoriques :	mardi	13h30 à 15h20	salle D7-2016
	jeudi	13h30 à 15h20	salle D7-2016
Exercices/laboratoires :	jeudi	13h30 à 15h20	salle

Description

Crédits :	3
Organisation :	Cours théoriques : 3 heures/semaine Exercices/laboratoires : 1 heure(s)/semaine Travail personnel : 5 heures/semaine
Préalable :	ift187
Fiche signalétique	http://www.usherbrooke.ca/fiches-cours/ift287
Particularités :	Aucune

1 Présentation

1.1 Mise en contexte

Les bases de données jouent un rôle central dans le développement des systèmes informatiques. Elles permettent de stocker l'information relative à un domaine d'application, d'en préserver l'intégrité, de l'extraire en utilisant un langage de haut niveau, de traiter plusieurs transactions simultanément, de répartir les données, et d'assurer la sécurité et le recouvrement des données.

Le modèle relationnel prédomine dans l'industrie. Il s'agit d'une des plus belles réussites de la recherche en informatique. On y retrouve l'élégance des mathématiques appliquée de manière efficace à un problème concret. Le modèle relationnel mérita à son auteur, E. F. Codd, le ACM Turing Award, l'équivalent du prix Nobel pour les informaticiens. De par sa puissance, sa simplicité, son niveau d'abstraction, ses fondements mathématiques, et son degré de pénétration du marché qui illustre son adéquation, le modèle relationnel constitue un cas de figure intéressant pour l'étude des bases de données.

Le modèle orienté objet est en pleine croissance. D'usage plus simple que le modèle relationnel dans un contexte de programmation orientée objets, il constituera sûrement une alternative efficace et souple dans le futur, avec l'amélioration des SGBD le supportant. Il sera donc notre deuxième cas de figure.

XML est devenu un norme de facto dans l'industrie pour l'échange électronique de données. Nous étudierons la structure du langage XML ainsi que les outils existants en Java pour manipuler des données avec XML. Nous aborderons finalement la conception d'interfaces graphiques simples basées sur des navigateurs web (HTML, JSP, servlet). Le langage Java, largement utilisé dans l'industrie pour le développement d'applications distribuées utilisant internet et des bases de données, sera notre principal outil de développement, avec bien sûr le langage SQL.

Des modèles plus anciens de bases de données, tels les modèles réseaux et hiérarchiques, sont encore en utilisation en industrie. Toutefois, ils sont rarement choisis pour développer de nouveaux systèmes. L'étudiant pourra se référer au livre de base du cours, s'il désire approfondir ses connaissances à leur sujet.

Ce cours fait suite au cours IFT187 – Éléments de bases de données. Il permettra à l'étudiant d'appliquer et de connaître de manière approfondie les concepts de traitement de données et de savoir exploiter une base de données. Dans le cours suivant, IFT487 – Modélisation avancée de bases de données, les concepts de modélisation de données, de concurrence, de recouvrement, de gestion et d'implantation des bases de données seront traités.

1.2 Objectifs généraux

À la fin de ce cours, l'étudiant devrait être capable :

1. de connaître les composantes de base d'un système de gestion de base de données;
2. de savoir utiliser des patrons de conception pour une application interactive de base de données multi-utilisateurs.
3. de savoir exploiter une base de données relationnelle et une base de données orientée objets;
4. de connaître et appliquer les principes de bases du langage XML pour l'échange électronique de données;
5. de développer une interface graphique web simple;
6. de parfaire sa connaissance des bases de données de manière autonome.

1.3 Objectifs spécifiques

À la fin de l'étude de chacun des thèmes suivants —et pour atteindre les objectifs spécifiques— l'étudiant devrait être capable:

1. manipulation de bases de données relationnelles
 - de créer, de modifier et d'interroger une base de données relationnelle en mode client-serveur;
 - de développer une application simple, dans un langage de programmation orienté objets, exploitant une base de données relationnelle, en utilisant les patrons de conception présentés.

2. manipulation de bases de données orientées objets
 - de créer, de modifier et d'interroger une base de données orientée objets;
 - d'utiliser une base de données orientée objets;
 - de développer une application simple, dans un langage de programmation orienté objets, exploitant une base de données orientée objets, en utilisant les patrons de conception présentés.
3. interface
 - de créer une description de fichier XML;
 - de créer et d'interroger un fichier XML à partir d'une application;
 - de développer une interface graphique web simple pour une application.

1.4 Contenu détaillé

Thème	Contenu	Heures	Chapitre	Travaux
1	Rappel sur SQL (au besoin)	3	1	
1	Introduction à Java : Différences avec C++, types primitifs, vecteurs, classes, visibilité, exception, représentation des classes en UML	3	1	tp1
2	L'approche client-serveur avec JDBC et Java : Connexion, Statement, PreparedStatement, ResultSet, transaction, étude d'une architecture OO d'une application, gestion des dates et heures	12	9, 25.6, notes	tp2, tp3
3	Les bases de données orientées objet : Modélisation OO des données, BD OO, persistance, persistance transitive, postprocesseur, Collection, Set, List, Map, Iterator, Query, survol des complexités algorithmiques des collections, brève introduction au polymorphisme avec le tri, étude d'une architecture d'une application	9	20, 21, notes	tp4
4	XML Introduction au langage (balise, element, attribut), DTD, XML schema, SAXP, DOM, JAXB	6	27, notes	tp5
5	Interface web HTML, JSP, Servlet, Application, Session, Request, Traitement de la concurrence	6	notes	tp6
6	Conclusion			

2 Organisation

2.1 Méthode pédagogique

- Une semaine comprend 4 heures de présence en classe: 3 heures de cours dit théorique constituées d'un exposé magistral et une heure d'exercices.
- La séance d'exercices aura lieu soit en laboratoire, soit en classe, en fonction du sujet à voir.
- Les étudiants doivent lire les chapitres du livre ou les notes avant le cours.
- Les étudiants travaillent en équipe de 2 (premier travail) à 4 (travaux suivants);
- Le cours comporte 6 travaux pratiques.
- Les travaux seront remis électroniquement le vendredi à 16h.

2.2 Calendrier du cours

semaine	au		remise tps	
du lundi	vendredi	Sujet	vendredi 16h	
2012-01-09	2012-01-13	Introduction à Java & Revue de SQL		
2012-01-16	2012-01-20	Introduction à Java & Revue de SQL	tp1	java
2012-01-23	2012-01-27	Client-serveur and Java et JDBC		
2012-01-30	2012-02-03	Client-serveur and Java et JDBC	tp2	jdbc-1
2012-02-06	2012-02-10	Client-serveur and Java et JDBC		
2012-02-13	2012-02-17	BD OO	tp3	jdbc-2
2012-02-20	2012-02-24	intra		
2012-02-27	2012-03-02	BD OO		
2012-03-05	2012-03-09	relâche		
2012-03-12	2012-03-16	XML	tp4	OS
2012-03-19	2012-03-23	XML		
2012-03-26	2012-03-30	interface web avec JSP-Servlet	tp5	XML
2012-04-02	2012-04-06	interface web avec JSP-Servlet		
2012-04-09	2012-04-13	révision / finalisation en classe du tp6	tp6	servlet
2012-04-16	2012-04-20	final		
2012-04-23	2012-04-27	final		

2.3 Évaluation

travaux pratiques	30 %
examen intra	30 %
examen final	40 %

2.3.1 Pénalité pour retard

Les travaux pratiques remis en retard sont sujets à une pénalité. Les travaux seront remis électroniquement. La note sera réduite de 33% pour chaque tranche de 24h de retard. En conséquence, la note attribuée après 2 jours de retard est zéro. Si votre travail n'est pas terminé à temps, vous devrez le soumettre par courrier électronique à l'adresse marc.frappier@usherbrooke.ca sous format d'un fichier .zip contenant tous les fichiers à soumettre en respectant la structure de répertoire déterminée dans l'énoncé du travail.

2.3.2 Votre programme doit fonctionner

Pour qu'un travail pratique soit comptabilisé, le programme doit réussir les tests utilisés pour la correction. Après la remise d'un travail, le fichier de test utilisé pour la correction sera publié sur le site web du cours ainsi que les résultats attendus (au besoin). L'étudiant aura alors *deux* semaines pour corriger son programme afin qu'il réussisse les tests en faisant une démonstration auprès d'un correcteur *lors d'une séance d'exercices ou sur rendez-vous*. La note attribuée sera celle déterminée sur le programme soumis (et non pas sur la version corrigée).

2.3.3 Soumission des travaux

Il est de votre responsabilité de débiter votre travail le plus tôt possible et de pouvoir le soumettre électroniquement avant l'heure d'échéance pour la soumission du travail. L'incapacité de trouver un poste de travail ou de se connecter à distance quelques minutes avant l'heure d'échéance ne sont pas des raisons valables pour justifier un retard.

2.3.4 Logiciels utilisés pour la correction

Les travaux soumis seront corrigés avec les logiciels du DI fournis dans une machine virtuelle Linux (Java, Postgres, ObjectStore, Tomcat, XML). Il est de votre responsabilité de vous assurer que vos travaux fonctionnent sous cette architecture.

3 Références

Matériel obligatoire

- [1] R. Elmasri and S.B. Navathe. *Fundamentals of Database Systems*. Addison-Wesley, 4e, 5e ou 6e edition.
- [2] Benoit Fraikin. Connexion par SSH et la soumission par turnin, 2005.
<http://www.dmi.usherb.ca/~fraikin/cours/SSH-turnin/>
- [3] Marc Frappier. Norme de documentation des programmes, 2004.
<http://www.dmi.usherb.ca/~frappier/normes/norme-programmation.pdf>
- [4] Marc Frappier. IFT287 : notes de cours, 2011.
<http://www.dmi.usherb.ca/~frappier/ift287/notes-de-cours.pdf>

Bibliographie

- [1] S. Abiteboul, R. Hull, and V. Vianu. *Fondements des bases de données*. Vuibert, 2000.
- [2] H. Bergsten. *JavaServer Pages, 3e édition*. O'Reilly & Associates, 2003.
- [3] F. Berqué, S. Frezefond, and L. Sorriaux. *Java/XML et Oracle : e-commerce, EAI, portails d'entreprise, applications mobiles*. Eyrolles, 2001.
- [4] C. Cavaness. *Programming Jakarta Struts*. O'Reilly, 2004.
- [5] C. date. *Introduction aux bases de données, 6e édition*. Vuibert, 1998.
- [6] S.W. Dietrich and S. D. Urban. *An Advanced Course in Database Systems : Beyond Relational Databases*. Prentice Hall, 2005.
- [7] R. Fleury, C. Vasson, F. Baudequin, and G. Thomas. *Java/XML*. Eyrolles, 2005.
- [8] G. Gardarin. *Bases de données objet & relationnel*. Eyrolles, 1999.
- [9] M. Hall and L. Brown. *Core Servlets and JavaServer Pages*. Prentice Hall, 2003.
- [10] Cay Horstman. *Big Java, 3e édition*. John Wiley, 2008.
- [11] Cay Horstman. *Java Concepts, 5e édition*. John Wiley, 2008.
- [12] G. Reese. *JDBC et Java : guide du programmeur*. O'Reilly & Associates, 2001.
- [13] G. Ricardi. *Principles of Database Systems with Internet and Java Applications*. Addison-Wesley, 2001.
- [14] A. Silberschatz, H. Korth, and S. Sudarshan. *Database Systems Concepts, 3e édition*. McGraw-Hill, 1997.
- [15] J.D. Ullman. *Principles of Database and Knowledge-Base Systems, volume I*. Computer science Press, 1988.
- [16] J.D. Ullman. *Principles of Database and Knowledge-Base Systems, volume II*. Computer science Press, 1989.
- [17] J.D. Ullman and J. Widow. *A First Course in Database Systems*. Prentice-Hall, 1994.