

Département d'informatique
IFT 287 — Exploitation de BD relationnelles et OO
Plan de cours

Enseignant :

Courriel :
Site du cours :
Disponibilité :

Professeur responsable :

Horaire :

Description officielle de l'activité pédagogique ¹

Objectif(s) Développer une application Web simple exploitant des bases de données relationnelles et orientées objet.

Contenu Exploitation en mode client-serveur d'une base de données relationnelle et d'une base de données objet. Développement d'un système d'information simple. Traitement de transactions. Accès concurrent aux données et préservation de l'intégrité des données. Développement d'une application Web simple avec servlet et JSP. Échange électronique de données avec XML. Utilisation du langage Java comme environnement de programmation.

Crédits 3

Organisation 3 heures d'exposé magistral par semaine

1 heure d'exercice par semaine

5 heures de travail personnel

Préalable IFT 187

Antérieure IFT 159

¹ <http://www.usherbrooke.ca/fiches-cours/ift287>

1 Présentation

Cette section présente les objectifs spécifiques et le contenu détaillé de l'activité pédagogique. Cette section, non modifiable sans l'approbation d'un comité de programme du Département d'informatique, constitue la version officielle.

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 objet, 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 une norme de facto dans l'industrie pour l'échange électronique de données. Dans un troisième temps, 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, IGE487 – *Modélisation 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 spécifiques

À la fin de cette activité pédagogique, l'étudiante ou l'étudiant sera capable de:

1. Manipuler des bases de données relationnelles
 - a. Créer, modifier et interroger une base de données relationnelle en mode client-serveur;
 - b. Développer une application simple, dans un langage de programmation orienté-objet, exploitant une base de données relationnelle.
2. Manipuler des bases de données orientées objets
 - a. Créer, modifier et interroger une base de données orientée objets;
 - b. Utiliser une base de données orientée objets;
 - c. Développer une application simple, dans un langage de programmation orienté objets, exploitant une base de données orientée objets.
3. Manipuler des interfaces
 - a. Créer une description de fichier XML;
 - b. Créer et d'interroger un fichier XML à partir d'une application;
 - c. Développer une interface graphique web simple pour une application.

1.3 Contenu détaillé

Thème	Contenu	Heures	Objectifs	Travaux
1	Rappel sur SQL	3	1.a	TP1
2	Introduction à Java. Différences avec C++, types primitifs, vecteurs, classes, visibilité, exceptions, représentation des classes en UML.	3	1.b et 2.c	TP2
3	L'approche client-serveur avec JDBC et Java : Connexion, Statement, PreparedStatement, ResultSet, transaction, étude d'une architecture Orientée Objets d'une application, gestion des dates et des heures	12	1.a et 1.b	TP3
4	Les bases de données orientées objet : Modélisation Orienté Objet des données, 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	2.a, 2.b et 2.c	TP4
5	XML Introduction au langage (balise, element, attribut), DTD, XML schema, SAXP, DOM, JAXB	6	3.a et 3.b	TP5
6	Interface web HTML, JSP, Servlet, Application, Session, Request, Traitement de la concurrence	6	3.c	TP5

2 Organisation

Cette section propre à l'approche pédagogique de chaque enseignante ou enseignant présente la méthode pédagogique, le calendrier, le barème et la procédure d'évaluation ainsi que l'échéancier des travaux. Cette section doit être cohérente avec le contenu de la section précédente.

2.1 Méthode pédagogique

Une semaine comprend trois heures de présence en classe de cours dit théorique, une heure de laboratoire pour faire des exercices et cinq heures de travail personnel.

Le cours est divisé en quatre activités principales :

- a) **Préparation du cours.** Les étudiantes et les étudiants doivent préparer préalablement la séance en analysant les lectures préparatoires recommandées.

- b) **L'apprentissage en classe.** De manière générale, l'apprentissage en classe est composé d'un exposé du chargé de cours suivi d'une discussion avec les étudiantes et les étudiants, et quelques exercices théoriques pour faciliter la compréhension des concepts traités.

- c) **Exercices au laboratoire.** Chaque semaine il y aura une période d'exercices au laboratoire pour appliquer les connaissances théoriques apprises.

- d) **La réalisation des travaux pratiques.** Les étudiantes et étudiants doivent montrer leurs connaissances sur les sujets du cours en réalisant cinq travaux pratiques.

2.2 Calendrier du cours

	Date	Thème	Lecture
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15-16			

2.3 Évaluation

a) Examens.

Il y aura deux examens pendant le cours :

- Un examen périodique à la mi- session.
- Un examen final.

b) Travaux pratiques.

Il y aura cinq travaux pratiques qui seront faits en équipe.

Deux personnes pour le premier et le deuxième, et quatre personnes pour les travaux suivants.

La date de remise des travaux qui apparaît sur le calendrier du cours doit être respectée.

Les remises se feront avec la commande turnin sous UNIX (ou par SSH). Si vous n'êtes pas familier avec turnin, visitez :

<http://www.usherbrooke.ca/informatique/ressources/documentation/data/turnin/turnin.html>

À chaque fois que vous allez faire une remise d'un travail, s'il y a plus d'un fichier à remettre, ils doivent se trouver dans un répertoire nommé tpN. Le répertoire de remise doit toujours contenir un fichier `etudiants.xml` avec le nom et la matricule des membres de l'équipe dans le format suivant :

```
<etudiants>
  < membre nom="Nom Prénom 1" matricule="123456789" />
  < membre nom="Nom Prénom 2" matricule="123456789" />
</etudiants>
```

Des indications particulières à chaque remise seront fournies avec l'énoncé de chaque TP.

Toute remise qui ne respecte pas cette forme ne sera pas corrigée, et la note attribuée sera zéro (0).

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.

À moins d'avis contraire, les travaux soumis seront corrigés avec les logiciels du Département d'informatique (Java, Postgres, ObjectStore, Tomcat, XML) sur les serveurs du département. Il est de votre responsabilité de vous assurer que vos travaux fonctionnent sur ces serveurs.

L'évaluation des travaux pratiques sera faite en deux étapes :

- i) L'évaluation de l'exécution de votre travail. Le jour de remise c'est toujours une journée qui a une période au laboratoire, lors de cette période il faut montrer l'exécution de votre TP à l'évaluateur.
- ii) La révision de votre travail. Une fois validée l'exécution de votre travail, il faut procéder au dépôt avec le turning pour évaluer les points liés à la programmation, la documentation et la structure de votre travail.

Si le TP n'est pas présenté pour l'exécution la date prévue, il sera possible de le présenter la semaine suivante pendant la période de laboratoire, mais il y aura une pénalité de 30% de la note du TP. Aucune autre date ne sera prévue pour présenter les TP en retard.

- c) **Commentaires du cours.** Chaque semaine, après le deuxième cours, l'étudiante ou l'étudiant doit envoyer un courriel au chargé de cours avec un petit résumé de la semaine (quelques lignes), ainsi que ses observations, ses remarques et les questions qu'elle ou il se pose après les cours.

Ces commentaires sont importants pour avoir une idée rapide de ce que les étudiantes et les étudiants ont compris pendant la semaine.

L'envoi de ce courriel ajoute aux étudiantes et aux étudiants des points bonis sur la note finale. Le courriel doit être envoyé au plus tard un jour après le deuxième cours. Si le courriel arrive en retard, il ne sera pas noté.

Pour envoyer ce courriel, il faut écrire dans l'objet du message :

IFT287 – Commentaire de la semaine XX

d) **Attribution des points.**

Éléments de notation	Points
Examen périodique	
Examen final	
Travaux pratiques	
TOTAL	
POINTS EXTRA	
Commentaires après chaque séance. 0.5 par commentaire. Un maximum de 5 points sera alloué.	
TOTAL	

Les points extra permettent aux étudiantes et étudiants de récupérer quelques points perdus lors d'une évaluation. La note maximale attribuée à une étudiante ou un étudiant sera toujours 100 points.

N.B. Les professeurs doivent évaluer la qualité du français dans tous les travaux des étudiantes et étudiants à l'exception des examens. À cet effet, la Faculté des sciences a établi deux modalités :

- a) 5% des points peuvent être soustraits pour la qualité du français;
- b) la reprise du travail peut être demandée jusqu'à un niveau acceptable.

Ces deux modalités s'appliquent à toutes les activités pédagogiques, sauf à celles bien entendu dont l'objet même est la langue ou la communication.

2.4 Échéancier des travaux pratiques

TP	Réception du problème	Thème	Points	Remise
1				
2				
3				
4				
5				

2.5 La communication dans le cours

Pour faciliter la communication pendant le cours, il y a un site Web comme support du cours.

<http://www.dmi.usherb.ca/~palao/cours/Hiver2010/IFT287/siteWeb/index.html>

Le site Web montre une section de nouvelles qu'il faut consulter fréquemment.

2.6 Note sur le plagiat

Un document dont le texte et la structure se rapporte à des textes intégraux tirés d'un livre, d'une publication scientifique ou même d'un site Internet, doit être référencé adéquatement. Lors de la correction de tout travail individuel ou de groupe une attention spéciale sera portée au plagiat, défini dans le Règlement des études comme « le fait, dans une activité pédagogique évaluée, de faire passer indûment pour siens des passages ou des idées tirés de l'œuvre d'autrui. ». Le cas échéant, le plagiat est un délit qui contrevient à l'article 8.1.2 du Règlement des études² : « tout acte ou manœuvre visant à tromper quant au rendement scolaire ou quant à la réussite d'une exigence relative à une activité pédagogique. » À titre de sanction disciplinaire, les mesures suivantes peuvent être imposées : a) l'obligation de reprendre un travail, un examen ou une activité pédagogique et b) l'attribution de la note E ou de la note 0 pour un travail, un examen ou une activité évaluée. Tout travail suspecté de plagiat sera référé au Secrétaire de la Faculté des sciences.

Ceci n'indique pas qu'une étudiante ou qu'un étudiant n'a pas le droit de coopérer dans la réalisation d'un travail d'une autre personne tant que la rédaction finale des documents et la création du programme reste la sienne.

De la même manière, l'utilisation de morceaux de code source ainsi que de documentation provenant d'Internet est autorisée. Par contre, il ne faut pas que le travail soit une copie complète ou partielle du document de référence, mais que l'étudiante ou l'étudiant démontre la compréhension de la partie utilisée. Le chargé de cours peut, en cas de doute, demander à l'étudiante ou à l'étudiant d'expliquer les notions ou le fonctionnement du code qu'il considère comme étant plagié. De plus, il est important de penser à ajouter explicitement dans le travail pratique la référence (auteur, adresse Internet, etc.) du document original.

En cas de doute, ne pas hésiter à demander conseil et assistance au chargé de cours afin d'éviter toute situation délicate par la suite.

3 Matériel pour le cours

Les notes de cours [2] sont disponibles sur la page WEB. Le manuel de base obligatoire pour le cours est celui de

Les normes de programmation du Département d'informatique sont décrites dans [??]. Vous devez absolument vous procurer ce document et le lire. De même, toutes les informations nécessaires pour vous connecter à distance sur nos serveurs sont présentée dans [??].

4 Documentation et références

1. G. Ricardi. Principles of Database Systems with Internet and Java Applications. Addison-Wesley, 2001.
2. Abiteboul, S., Hull, R., Vianu, V. Foundations of Databases. Addison-Wesley, 1995.
3. Silberschatz, A., Korth, H., Sudarshan, Database Systems Concepts S.T. McGraw-Hill, 1997.
4. H. Bergsten. JavaServer Pages, 3^e édition. O'Reilly & Associates, 2003.
5. C. Date, Introduction aux bases de données, Vuilbert, 1998
6. M. CAMPIONE, K. WALRATH et A. HUML : The Java Tutorial : A Short Course on the Basic. Addison-Wesley, 2000. <http://java.sun.com/docs/books/tutorial/>.
7. C. CAVANESS : Programming Jakarta Struts. O'Reilly, 2004.
8. C. CAVANESS et B. KEETON : Jakarta Struts Pocket Reference. O'Reilly & Associates, 2003.
9. Elmasri, Ramez. Fundamentals of database systems. Pearson Addison Wesley, fifth edition.
10. M.C. DACONTA et A. SAGANICH : D'éveloppement XML avec Java 2. CampusPres, 2001.
11. R. ELMASRI et S.B. NAVATHE : Fundamentals of Database Systems, Sieme édition. Addison-Wesley, 2006.
12. R. FLEURY, C. VASSON, F. BAUDEQUIN et G. THOMAS : Java/XML. Eyrolles, 2005.
13. Benoit FRAIKIN : Connexion par SSH et la soumission par turnin. <http://www.dmi.usherb.ca/~fraikin/cours/SSH-turnin>, 2005.
14. Marc FRAPPIER : Normes de rédaction et de programmation du département. <http://www.dmi.usherb.ca/~fraikin/cours/Normes/normes-de-programmation.pdf>, 2005.
15. Marc FRAPPIER : Notes de cours. <http://www.dmi.usherb.ca/~frappier/ift287/ift287.html>, 2007.
16. M. HALL : Servlets and javaserver pages (jsp) 1.0 : A tutorial. <http://www.apl.jhu.edu/~hall/java/Servlet-Tutorial/>.
17. M. HALL et L. BROWN : Core Servlets and JavaServer Pages, Volume 1. Prentice Hall, 2003.
18. Cay Horstman. Au cœur de Java. Pearson Education, 2008, 8^e édition.
19. Cay HORSTMAN : Big Java, 3^{ieme} édition. John Wiley, 2008.
20. Cay HORSTMAN : Java Concepts, 5^{ieme} édition. John Wiley, 2008.
21. Edward SCIORE : Database Design and Implementation. Wiley, 2008.