

IFT 187

Éléments de bases de données

Présentation de l'activité

IFT 187
Hiver 2022

2022-01-03



Luc.Lavoie@USherbrooke.ca

© 2018-2021, Μητίς (<http://info.usherbrooke.ca/llavoie>)
CC BY-NC-SA 4.0 (<https://creativecommons.org/licenses/by-nc-sa/4.0/>)

Plan de la présentation

- Sciences, informatique et bases de données
- Le plan de l'activité
- Les références
- Les colles du prof!



Sciences, informatique et bases de données

- La place de l'informatique en science
- La place des bases de données en informatique
- La place de l'activité dans votre formation

Informatique et science

○ Sciences formelles

- Mathématiques
- Statistiques
- Informatique
- ...

○ Philosophie

- Logique
- Ontologie
- Épistémologie
- Éthique
- ...

○ Sciences de la nature

- Physique
- Chimie
- Géologie
- Biologie
- ...

○ Sciences humaines

- Histoire
- Sociologie
- Psychologie
- Linguistique
- ...

Bases de données et informatique

○ Informatique (fondamentale)

- Calculabilité
- Langages formels
- Algorithmique
 - déterministe
 - heuristique
- **Modélisation**
 - des problèmes
 - des données
- Complexité

○ Mathématiques

- ...

○ Informatique (appliquée)

- Systèmes, réseaux et télématique
- **Bases de données**
- Intelligence artificielle
- Traitement d'images
- Bio-informatique
- ...

○ Génie logiciel

- ...

○ Gestion de projets

- ...

Bases de données et informatique (bis)

○ Informatique (fondamentale)

- Calculabilité
- Langages formels
- Algorithmique
 - déterministe
 - **euristique**
- **Modélisation**
 - des problèmes
 - des données
- Complexité

○ Mathématiques

- ...

○ Informatique (appliquée)

- Systèmes, réseaux et télématique
- **Bases de données**
- Intelligence artificielle
- Traitement d'images
- Bio-informatique
- ...
- **Génie logiciel**
 - ...
- **Gestion de projets**
 - ...



- Études libres 1^{er} cycle - Faculté des sciences
- Certificat en technologies de l'information
- Baccalauréat en génie électrique
- Baccalauréat en informatique
- Baccalauréat en informatique de gestion
- Baccalauréat en microbiologie
- Baccalauréat en sciences de l'image et des médias numériques
- Maîtrise en informatique

Plan de l'activité

- Organisation
- Objectifs
- Contenu
- Méthode pédagogique
- Calendrier
- Évaluation

Organisation

Enseignant

Luc Lavoie

Courriel : luc.lavoie@usherbrooke.ca

Local : D4-2006

Site : <http://info.usherbrooke.ca/llavoie>

Disponibilité : sur rendez-vous

Responsables

Marc Frappier et Luc Lavoie

Horaire :

Mardi 13:30 à 15:20 D3-2040

Jeudi 15:30 à 18:20 D3-2035 ou Laboratoire D4

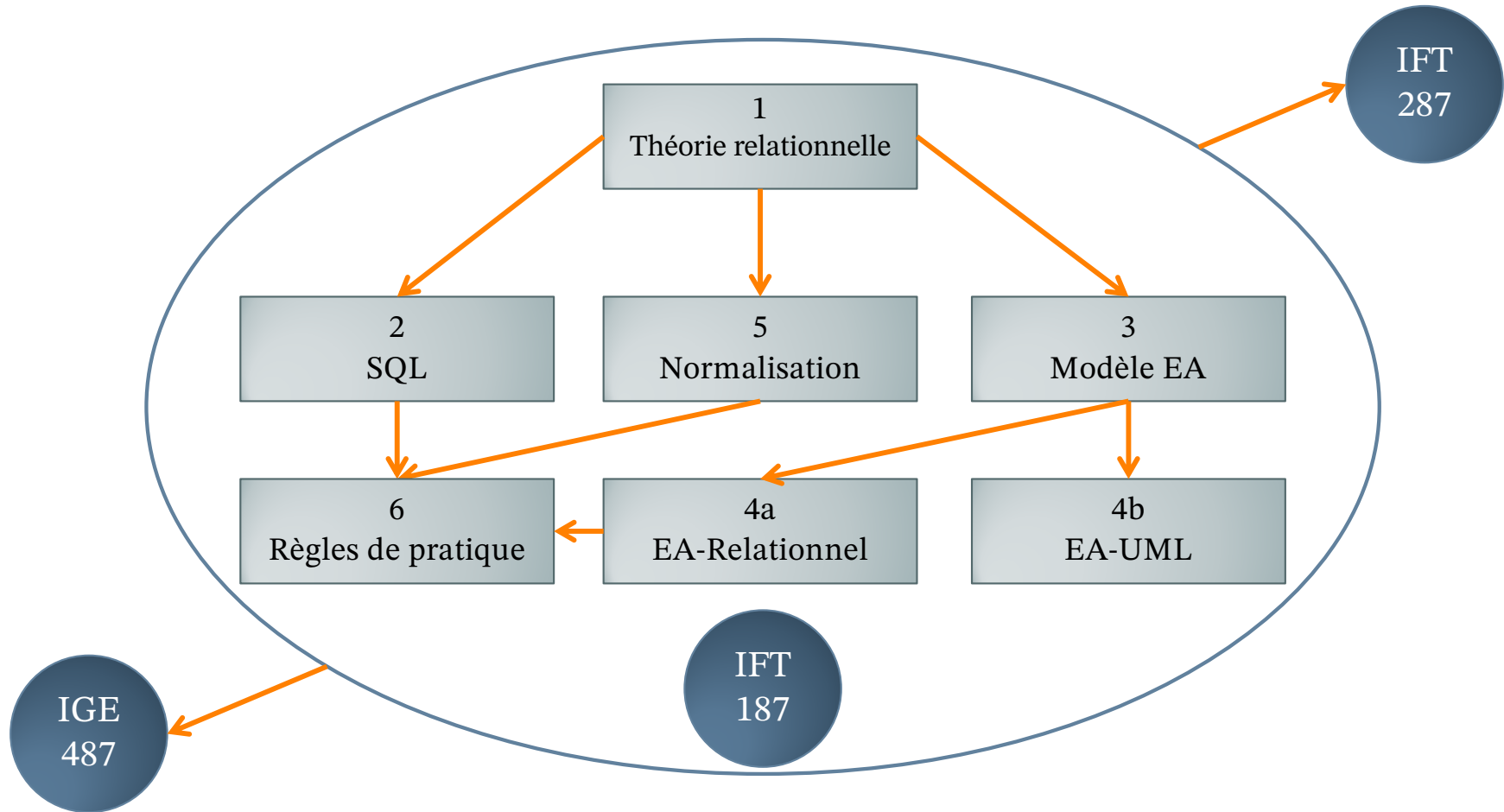
Objectifs spécifiques (A)

- Utiliser la théorie relationnelle pour décrire un problème de modélisation de données.
- Modéliser des données à l'aide du diagramme entité-association (EA), du diagramme de classe UML et du modèle relationnel.
- Transformer un modèle EA et un diagramme de classe UML en un modèle relationnel.
- Normaliser une base de données relationnelle en FNBC et en 5FN.
- Identifier les composantes élémentaires d'un système de gestion de bases de données relationnelles.

Objectifs spécifiques (B)

- Définir les domaines, les types, les relations, les contraintes, les procédures, et les automatismes requis lors de la définition d'une base de données d'envergure moyenne, à l'aide du langage SQL.
- Interroger et mettre à jour une base de données relationnelle, à l'aide du langage SQL.
- Appliquer les bonnes pratiques de définition et de gestion des clés, à l'aide du langage SQL.
- Intégrer l'ensemble des notions présentées pour modéliser un problème de gestion de données et mettre en oeuvre de modèle, à l'aide du langage SQL.

Structure



Contenu (1-3)

Thème	Contenu	Nbr. d'heures	Objectifs	Travaux	Lectures
1	<p>Introduction :</p> <ul style="list-style-type: none"> • L'information, les données et leur traitement 	1	1 à 9		Chapitre 1 [6]
2	<p>Théorie et modèles relationnels :</p> <ul style="list-style-type: none"> • Fondements théoriques (Rel) : <ul style="list-style-type: none"> – Attributs, types, domaines, valeurs et représentations – Tuples et relations – Schémas et bases de données • Problématique des données manquantes et modèles associés 	4	1 à 9		Chapitres 3, 5 et 8 [6]
3	<p>SQL - Tables et schéma :</p> <ul style="list-style-type: none"> • Schéma • Définition des tables et utilisation des types de base • Insertion, modification et retrait des données (INSERT, UPDATE et DELETE) 	3	5	✓	Chapitre 6 [6]

Contenu (4-6)

4	<p>SQL - Contraintes d'intégrité :</p> <ul style="list-style-type: none"> • Colonne, domaine, type, table (clés primaire, unique et référentielle) 	3	6	✓	Chapitre 6 [6]
5	<p>SQL - Sélection - Concepts de base :</p> <ul style="list-style-type: none"> • Sélection simple (SELECT, FROM, WHERE) • Jointures internes et externes (JOIN) • Agrégation (GROUP et HAVING) • Tri (ORDER) • Vue 	3	6	✓	Chapitres 6 et 7 [6]
6	<p>SQL - Sélection - Concepts avancés :</p> <ul style="list-style-type: none"> • Définition de portée (WITH) et imbrication (des SELECT) • Quantificateurs et opérateurs ensemblistes • Application à INSERT, DELETE et UPDATE • Automatisation et fonctions 	6	6	✓	Chapitres 6 et 7 [6]

Contenu (7-9)

7	<p>Modélisation conceptuelle :</p> <ul style="list-style-type: none">• Modèle entité-association (EA)• Entités, attributs, clés, associations fortes et faibles• Héritage, dérivation disjointe et conjointe, union• Notations UML, de Chen, d'Abrial et d'Elmasri	6	2	✓	Chapitres 3 et 4 [6]
8	<p>Correspondance et traduction entre modèles :</p> <ul style="list-style-type: none">• Relationnel ↔ EA• Relationnel ↔ UML• EA ↔ UML	4	3	✓	Chapitre 9 [6]
9	<p>Normalisation de données :</p> <ul style="list-style-type: none">• 1FN, 2FN, 3FN, BCNF, 4FN, 5FN• Comparaison entre normalisation et traduction	3	4	✓	Chapitres 14 et 15 [6]

Contenu (10-11)

10	<p>SQL - Clés - Règles de pratique :</p> <ul style="list-style-type: none">• Clés primaires et secondaires• Clés naturelles et artificielles• Clés relatives et absolues• Clés référentielles et circularité	3	5, 7 et 6	✓	Chapitre 14 [6]
11	<p>Intégration des concepts à l'aide d'une étude de cas :</p> <ul style="list-style-type: none">• Cas comportant la réalisation d'un modèle conceptuel, sa traduction en modèle relationnel et sa réalisation en SQL avec les contraintes d'intégrité et les types appropriés	3	1 à 9	✓	

Méthode pédagogique

- Cours magistraux
 - dirigés, en classe
- Exercices
 - dirigés, en classe
- Travaux dirigés
 - dirigés ou supervisés, en laboratoire
- Travaux pratiques
 - autonomes, en laboratoire ou autre
- Projet
 - tutorat - dirigé, en classe ou en laboratoire
 - réalisation - autonome, en laboratoire ou autre
- Examens
 - individuels, récapitulatifs ; dates et lieux déterminés par la Faculté

Calendrier

Voir table et annotations ci-après

Semaine	Date	Thème	Échéance des travaux	Lectures dans Elmasri, selon année de publication
1	2022-01-03	1		2004 (1-2, 4-5) ; 2006 (1-3, 6) ; 2016 (1-3, 5, 8).
2	2022-01-10	2		2004 (1-2, 4-5) ; 2006 (1-3, 6) ; 2016 (1-3, 5, 8).
3	2022-01-17	3	TP0	2004 (7) ; 2006 (4) ; 2016 (6).
4	2022-01-24	4	TP1	2004 (7-8) ; 2006 (4-5) ; 2016 (6-7).
5	2022-01-31	5	TP2	2004 (7-8) ; 2006 (4-5) ; 2016 (6-7).
6	2022-02-07	6	TP3	2004 (7-8) ; 2006 (4-5) ; 2016 (6-7).
7	2022-02-14	6	TP4	2004 (19-20) ; 2006 (13.4) ; 2016 (10.4).
8	2022-02-21	Examen périodique		
9	2022-02-28	Relâche		
10	2022-03-07	7 et 11	Projet - jalon 1	2004 (3) ; 2006 (7-8) ; 2016 (3-4).
11	2022-03-14	7 et 11		2004 (3) ; 2006 (7-8) ; 2016 (3-4).
12	2022-03-21	8 et 11	Projet - jalon 2	2004 (3) ; 2006 (7-9) ; 2016 (9).
13	2022-03-28	9 et 11		2004 (6, 9) ; 2006 (10-15) ; 2016 (33, 14-15).
14	2022-04-04	10 et 11		2004 (6, 9) ; 2006 (10-15) ; 2016 (33, 14-15).
15	2022-04-11	Révision	Projet - jalon 3	
16	2022-04-18	Examen final		

Calendrier - annotations

- Les périodes d'enseignement des jeudis 6 et 13 janvier se tiendront au D3-2035 ou à distance, selon les directives sanitaires en vigueur.
- Les périodes d'enseignement des jeudis du 20 janvier au 17 février se tiendront aux laboratoires du D4.
- Les modalités relatives aux périodes d'enseignement des jeudis du 10 mars au 14 avril seront communiquées ultérieurement.
- Les remises des travaux sont les dimanches (TP et jalons de projet), jusqu'à 23 h 59.
- Les dates d'examen seront fixées ultérieurement par la Faculté des sciences.
- Pour les lectures, l'année indique la version du manuel [Elmasri2004], [Elmasri2011] ou [Elmasri2016], les chapitres étant indiqués à la suite, entre parenthèses.

Évaluation

Évaluation	Valeur	Commentaire
examen 1	30 %	individuel
examen 2	40 %	individuel et récapitulatif
travaux	16 %	seul ou en tandem
projet	14 %	en équipe de deux à quatre personnes
Total	100 %	

Références

- Matériel des enseignants
- Références essentielles
- Références importantes
- Références aux SGBD (SQL)
- Panorama

Matériel des enseignants

○ Consultez le site de l'activité!

- <http://info.usherbrooke.ca/llavoie/enseignement/IFT187/>

Références essentielles – une parmi celles-ci

- [Elmasri2016]
Ramez ELMASRI, Shamkant B. NAVATHE;
Fundamentals of database systems.
7th edition, Pearson, 2016.
ISBN 978-0-13-397077-7.
- [Elmasri2011]
Ramez ELMASRI, Shamkant B. NAVATHE;
Fundamentals of database systems
6th edition, Pearson Addison Wesley, 2011.
ISBN 978-0-13-608620-8.
- [Elmasri2004]
Ramez ELMASRI, Shamkant B. NAVATHE;
Conception et architecture des bases de données.
4^e édition, Pearson Éducation, 2004.
ISBN 2-7440-7055-6.

Les lectures sont **nécessaires**, les cours et les notes ne les remplacent pas!

Références importantes - Date

- [Date2001]
Chris J. DATE;
Introduction aux bases de données.
7^e édition, Vuibert, Paris, 2001, 912 p.
ISBN 2-7117-8664-1.
- [Date2004]
Chris J. DATE;
Introduction to database systems.
8th edition, Pearson Addison Wesley, 2004.
ISBN 0-321-19784-4.
- [Date2012]
Chris J. DATE;
SQL and Relational Theory: How to Write Accurate SQL Code.
2nd edition, O'Reilly, 2012.
ISBN 978-1-449-31640-2.
- [Date 2016]
Chris J. DATE;
The New Relational Database Dictionary of Terms, Concepts, and Examples.
O'Reilly, 2016.

Références importantes - autres

- [Ullman2008]
Jeffrey D. ULLMAN, Jennifer WIDOM;
A First Course in Database Systems.
3rd edition, Prentice-Hall, 2008, 564 p.
ISBN 978-0-13-600637-4.
- [DoDAF]
Deputy Chief Information Officer;
DoDAF – Department of Defence Architecture Framework.
Version 2.02.
<http://dodcio.defense.gov/Library/DoD-Architecture-Framework/>

Références – SGBD (SQL)

- PostgreSQL (en français, 2021-12-29)
 - <https://docs.postgresqlfr.org>
- PostgreSQL (en anglais, 2021-12-29)
 - <https://www.postgresql.org/docs/current/index.html>
- MariaDB (en anglais, 2020-08-18)
 - <https://mariadb.com/kb/en/library/documentation/>
- MariaDB (en anglais, 2020-08-18)
 - <https://mariadb.com/kb/fr/documentation-de-mariadb/>
- Oracle (en anglais, 2020-08-18)
 - https://docs.oracle.com/cd/E11882_01/index.htm

Références – un panorama

Voir diapositives suivantes

A Relational Model of Data for Large Shared Data Banks

E. F. Codd
IBM Research Laboratory, San Jose, California

Future users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation). A prompting service which supplies such information is not a satisfactory solution. Activities of users at terminals and most application programs should remain unaffected when the internal representation of data is changed and even when some aspects of the external representation are changed. Changes in data representation will often be needed as a result of changes in query, update, and report traffic and natural growth in the types of stored information.

Existing noninferential, formatted data systems provide users with tree-structured files or slightly more general network models of the data. In Section 1, inadequacies of these models are discussed. A model based on n -ary relations, a normal form for data base relations, and the concept of a universal data sublanguage are introduced. In Section 2, certain operations on relations (other than logical inference) are discussed and applied to the problems of redundancy and consistency in the user's model.

KEY WORDS AND PHRASES: data bank, data base, data structure, data organization, hierarchies of data, networks of data, relations, derivability, redundancy, consistency, composition, join, retrieval language, predicate calculus, security, data integrity

CR CATEGORIES: 3.70, 3.73, 3.75, 4.20, 4.22, 4.29

1. Relational Model and Normal Form

1.1. INTRODUCTION

This paper is concerned with the application of elementary relation theory to systems which provide shared access to large banks of formatted data. Except for a paper by Childs [1], the principal application of relations to data systems has been to deductive question-answering systems. Levein and Maron [2] provide numerous references to work in this area.

In contrast, the problems treated here are those of *data independence*—the independence of application programs and terminal activities from growth in data types and changes in data representation—and certain kinds of *data inconsistency* which are expected to become troublesome even in nondeductive systems.

The relational view (or model) of data described in Section 1 appears to be superior in several respects to the graph or network model [3, 4] presently in vogue for non-inferential systems. It provides a means of describing data with its natural structure only—that is, without superimposing any additional structure for machine representation purposes. Accordingly, it provides a basis for a high level data language which will yield maximal independence between programs on the one hand and machine representation and organization of data on the other.

A further advantage of the relational view is that it forms a sound basis for treating derivability, redundancy, and consistency of relations—these are discussed in Section 2. The network model, on the other hand, has spawned a number of confusions, not the least of which is mistaking the derivation of connections for the derivation of relations (see remarks in Section 2 on the "connection trap").

Finally, the relational view permits a clearer evaluation of the scope and logical limitations of present formatted data systems, and also the relative merits (from a logical standpoint) of competing representations of data within a single system. Examples of this clearer perspective are cited in various parts of this paper. Implementations of systems to support the relational model are not discussed.

1.2. DATA DEPENDENCIES IN PRESENT SYSTEMS

The provision of data description tables in recently developed information systems represents a major advance toward the goal of data independence [5, 6, 7]. Such tables facilitate changing certain characteristics of the data representation stored in a data bank. However, the variety of data representation characteristics which can be changed without *logically impairing some application programs* is still quite limited. Further, the model of data with which users interact is still cluttered with representational properties, particularly in regard to the representation of collections of data (as opposed to individual items). Three of the principal kinds of data dependencies which still need to be removed are: ordering dependence, indexing dependence, and access path dependence. In some systems these dependencies are not clearly separable from one another.

1.2.1. *Ordering Dependence.* Elements of data in a data bank may be stored in a variety of ways, some involving no concern for ordering, some permitting each element to participate in one ordering only, others permitting each element to participate in several orderings. Let us consider those existing systems which either require or permit data elements to be stored in at least one total ordering which is closely associated with the hardware-determined ordering of addresses. For example, the records of a file concerning parts might be stored in ascending order by part serial number. Such systems normally permit application programs to assume that the order of presentation of records from such a file is identical to (or is a subordering of) the

stored ordering. Those application programs which take advantage of the stored ordering of a file are likely to fail to operate correctly if for some reason it becomes necessary to replace that ordering by a different one. Similar remarks hold for a stored ordering implemented by means of pointers.

It is unnecessary to single out any system as an example, because all the well-known information systems that are marketed today fail to make a clear distinction between order of presentation on the one hand and stored ordering on the other. Significant implementation problems must be solved to provide this kind of independence.

1.2.2. *Indexing Dependence.* In the context of formatted data, an index is usually thought of as a purely performance-oriented component of the data representation. It tends to improve response to queries and updates and, at the same time, slow down response to insertions and deletions. From an informational standpoint, an index is a redundant component of the data representation. If a system uses indices at all and if it is to perform well in an environment with changing patterns of activity on the data bank, an ability to create and destroy indices from time to time will probably be necessary. The question then arises: Can application programs and terminal activities remain invariant as indices come and go?

Present formatted data systems take widely different approaches to indexing. TDMS [7] unconditionally provides indexing on all attributes. The presently released version of IMS [5] provides the user with a choice for each file: a choice between no indexing at all (the hierarchic sequential organization) or indexing on the primary key only (the hierarchic indexed sequential organization). In neither case is the user's application logic dependent on the existence of the unconditionally provided indices. IDS [8], however, permits the file designers to select attributes to be indexed and to incorporate indices into the file structure by means of additional chains. Application programs taking advantage of the performance benefit of these indexing chains must refer to those chains by name. Such programs do not operate correctly if these chains are later removed.

1.2.3. *Access Path Dependence.* Many of the existing formatted data systems provide users with tree-structured files or slightly more general network models of the data. Application programs developed to work with these systems tend to be logically impaired if the trees or networks are changed in structure. A simple example follows.

Suppose the data bank contains information about parts and projects. For each part, the part number, part name, part description, quantity-on-hand, and quantity-on-order are recorded. For each project, the project number, project name, project description are recorded. Whenever a project makes use of a certain part, the quantity of that part committed to the given project is also recorded. Suppose that the system requires the user or file designer to declare or define the data in terms of tree structures. Then, any one of the hierarchical structures may be adopted for the information mentioned above (see Structures 1-5).

Structure 1. Projects Subordinate to Parts

File	Segment	Fields
F	PART	part # part name part description quantity-on-hand quantity-on-order
	PROJECT	project # project name project description quantity committed

Structure 2. Parts Subordinate to Projects

File	Segment	Fields
F	PROJECT	project # project name project description
	PART	part # part name part description quantity-on-hand quantity committed

Structure 3. Parts and Projects as Peers
Commitment Relationship Subordinate to Projects

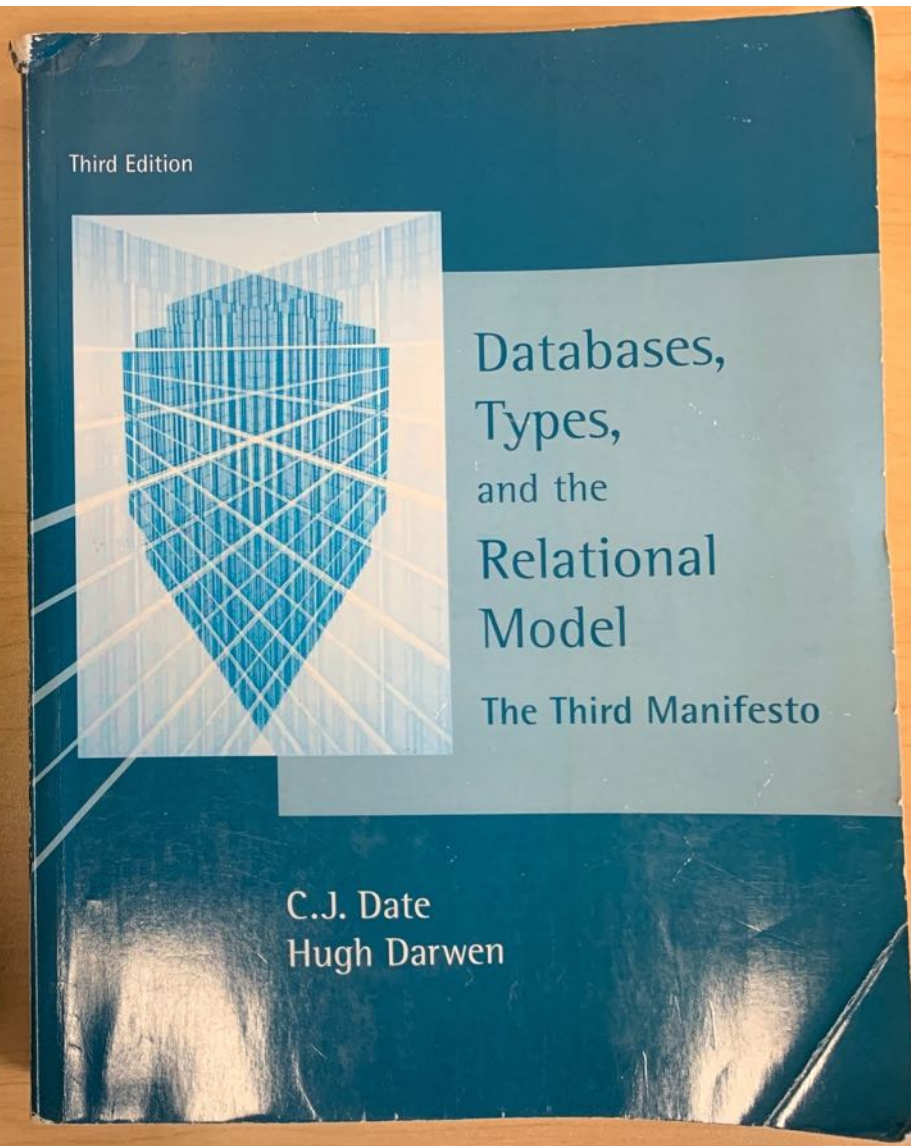
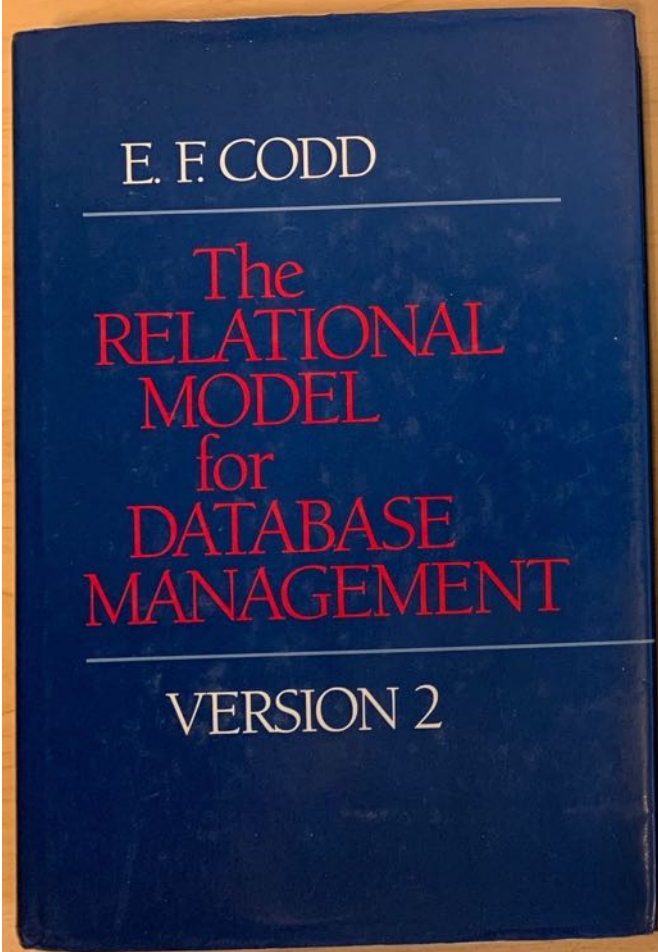
File	Segment	Fields
F	PART	part # part name part description quantity-on-hand quantity-on-order
G	PROJECT	project # project name project description
	PART	part # quantity committed

Structure 4. Parts and Projects as Peers
Commitment Relationship Subordinate to Parts

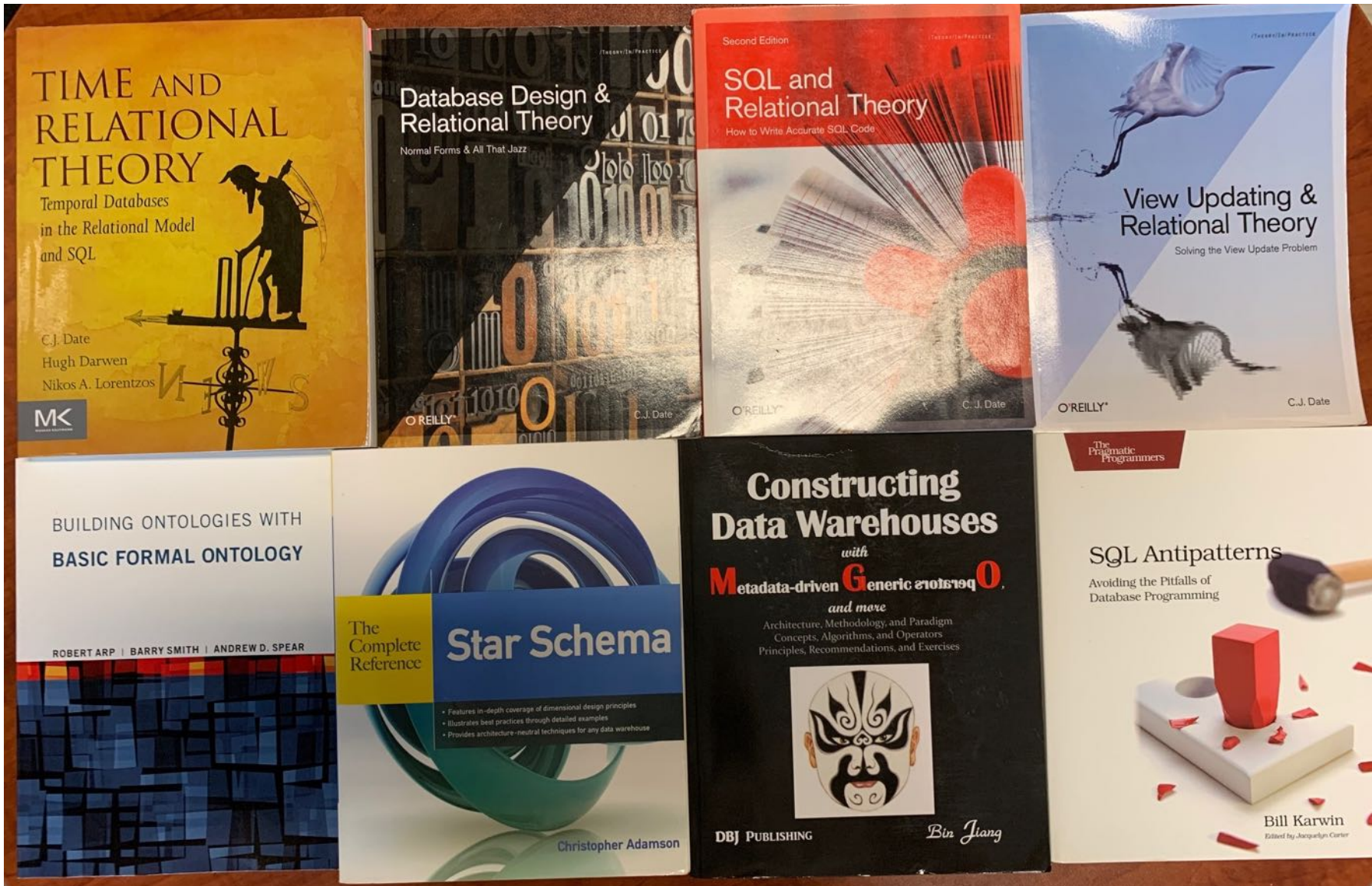
File	Segment	Fields
F	PART	part # part description quantity-on-hand quantity-on-order
	PROJECT	project # quantity committed
G	PROJECT	project # project name project description

Structure 5. Parts, Projects, and
Commitment Relationship as Peers

File	Segment	Fields
F	PART	part # part name part description quantity-on-hand quantity-on-order
G	PROJECT	project # project name project description
H	COMMIT	part # project # quantity committed







Les colles du prof

- ◆ Quel est le lien entre l'activité IFT187 et votre programme d'études ?
- ◆ Faut-il acheter un manuel ou des notes polycopiées ?
- ◆ Quand devez-vous commencer à lire ?
- ◆ Quelles sont les dates d'examen ?
- ◆ Pouvez-vous faire les travaux en équipe ?
- ◆ Comment les travaux sont-ils organisés ?
- ◆ Que représente la photo de droite ?



