



Bases de données

La théorie relationnelle

Département d'informatique
Faculté des sciences

 UNIVERSITÉ DE
SHERBROOKE

Christina.Khnaisser@USherbrooke.ca
<http://info.USherbrooke.ca/ckhnaisser>
Luc.Lavoie@USherbrooke.ca
<http://info.USherbrooke.ca/llavoie>

À faire :

* référence, dénotation, variable : voir

- Scott, Stoy et Strachey : Joe Stoy, Denotational Semantics: The Scott-Strachey Approach to Programming Language Semantics, MIT Press, 1981. ISBN 978-0-262-69076-8.

* intégrer la définition des opérateurs

PLAN

- Préambule
- Modèles et représentations
- Types, valeurs, variables et relations
- Théorie relationnelle
- Opérateurs
- Contraintes
- De la théorie aux modèles
- Absence
- Vocabulaire
- Références
- Les colles du prof



En science, comment fait-on pour résoudre un problème ?

On formule une théorie sur la base d'axiomes (principes), on élabore un modèle qui respecte ces principes.

PRÉAMBULE

- Qu'est-ce qu'une relation
- Quels que mots pour le dire
- Vérification des relations
- Au-delà du binaire
- Une intuition : la logique et les relations

PRÉAMBULE**QU'EST-CE DONC QU'UNE RELATION?**

- La réponse mathématique :
 - Un sous-ensemble d'un produit cartésien d'ensembles.

- Soit les ensembles
 - Nom = {Paul, Éliane, Mohamed, Sergeï, Diego}
 - Ville = {>Δ^ςσ>^{ςb}, Blanc-Sablon, Tadoussac, Chandler, Adélaïde, San Diego}

- Un exemple de **valeur** de relation
 - **v1** = {(Paul, >Δ^ςσ>^{ςb}), (Éliane, Blanc-Sablon), (Mohamed, Blanc-Sablon), (Sergeï, Chandler), (Sergeï, San Diego)}

- Une constante de relation est aussi appelée, dans certains contextes, *constante* (de relation) ou *littéral* (de relation).

La théorie relationnelle est construite à partir de la théorie des ensembles et du calcul des prédicats

PRÉAMBULE**QU'EST-CE DONC QU'UN TYPE DE RELATION?**

- Informellement, un type est un ensemble fini de valeurs.
- Un type de relation est donc un identifiant qui réfère à un ensemble de valeurs d'une même relation.

- Un exemple de **type** de relation
 - **TVN** = *Nom* × *Ville* ;

TVN l'ensemble des valeurs de la relation *Nom* × *Ville*

PRÉAMBULE**QU'EST-CE DONC QU'UNE VARIABLE RELATION?**

- Informellement, une variable est un identifiant qui réfère (désigne, est associé...) à une valeur.
- Une variable de relation est donc un identifiant qui réfère à une valeur de relation.

- Un exemple de **variable** de relation :
 - **Visite** : TVN ;

- À sa déclaration la variable est associée à l'ensemble vide {}.
- Pour en changer l'association, on a recours à l'opérateur d'affection :
 - **Visite** := v1 ;

PRÉAMBULE**QU'EST-CE DONC QU'UN INVARIANT?**

- Un invariant est une condition qui doit être vraie en tout temps.
- À la déclaration d'une variable est toujours associée un invariant qui prescrit que la valeur associée à la variable doit faire partie (être un élément) du type associé à la variable.
- Par exemple, attendu la déclaration **Visite** : **TVN** ;
 - **Visite** \subseteq **TVN**
 - **Visite** \subseteq **Nom** \times **Ville**

PRÉAMBULE**LA VÉRIFICATION DES AFFECTATIONS EST CAPITALE!**

○ Soit les ensembles

- Nom = {Paul, Éliane, Mohamed, Sergeï, Diego}
- Ville = {>Δ^ςσ>^{ςb}, Blanc-Sablon, Tadoussac, Chandler, Adélaïde, San Diego}
- v2 = {(Paul, >Δ^ςσ>^{ςb}), (Éliane, Blanc-Sablon), (Tadoussac, Blanc-Sablon), (Sergeï, Chandler), (Sergeï, Sans Diego)}

○ La constante v2 ne fait pas partie du type TVN :

- v2 \notin Nom \times Ville
- v2 \notin TVN

○ La variable Visite : TVN ne peut donc pas référer à v2.

- L'affectation « Visite := v2 » induit donc une erreur.

PRÉAMBULE**VOCABULAIRE**

- L'expression « variable de relation » est souvent abrégée en *relvar*.
- Par analogie, il y a aussi *reltype* et *relcon*.
- Un identifiant est aussi appelé, dans certains contextes, *identificateur*.

PRÉAMBULE**NOTATION**

○ Par souci d'expressivité, on permet souvent la déclaration d'une *relvar* sous une forme compacte, par exemple

- **RELATION Visite** {*Nom*, *Ville*}

Nous verrons bientôt qu'il est nécessaire de nommer les attributs de la relation (ne serait-ce que pour distinguer deux attributs de même type).

Une syntaxe plus appropriée serait donc

RELATION **Visite** {*n* : *Nom*, *v* : *Ville*}

REMARQUE

- Les ensembles sont des cas particuliers de relations en ce sens qu'il y a une bijection triviale entre une relvar définie sur un un seul ensemble et cet ensemble lui-même.
- Par exemple
 - RELATION $V \{Ville\}$ et Ville

PRÉAMBULE**RELATION N-AIRE**

- **Visite** est une variable de relation binaire, qu'advient-il des relations de degré supérieur?
- Soit les ensembles
 - Nom = {Paul, Éliane, Mohamed, Sergeï, Diego}
 - Ville = {>Δ^ςσ^ςb, Blanc-Sablon, Tadoussac, Chandler, Adélaïde, San Diego}
 - Matricule = {15113150, 15112354, 15113870, 15110132}
- **RELATION Etudiant** (Matricule, Nom, Ville)
 - **Etudiant** \subseteq **Matricule** \times **Nom** \times **Ville**
 - Etudiant = { (15113150, Paul, >Δ^ςσ^ςb),
 (15112354, Éliane, Blanc-Sablon),
 (15113870, Mohamed, Blanc-Sablon),
 (15110132, Sergeï, Chandler),
 (15110132, Sergeï, San Diego) }

PRÉAMBULE**PRÉDICAT ET PROPOSITIONS**

- Une relation est décrite par un prédicat qui représente une partie de la réalité.
- Un prédicat est une fonction avec une ou plusieurs variables. En associant des valeurs aux variables du prédicat, on obtient la proposition.
- Une proposition est un énoncé vrai ou faux à propos d'une partie de la réalité.

2020-09-04

BD010 : La théorie relationnelle (V2336) — Christina Khnaisser et Luc Lavoie
Département d'informatique, Faculté des sciences, Université de Sherbrooke, Québec

13

Une relation représente un prédicat qui représente d'une façon formelle (et générique) une portion de la réalité.

Un prédicat est une fonction qui est formée d'une ou de plusieurs variables et qui retourne une valeur booléenne (Vrai ou Faux) selon les valeurs des variables.

Une proposition en logique est un énoncé qui peut être vrai ou faux. Cependant ce qui va nous intéresser, c'est de représenter les propositions vraies (nommé tuple).

Dans nos exemples :

Visite est un prédicat avec 2 variables : Nom et Ville.

Étudiant est un prédicat avec 3 variables : Matricule, Nom et Ville.

Lecture :

[Date et al. 2007] Chapitre 2

<https://www.dcs.warwick.ac.uk/~hugh/TTM/DTATRM.pdf>

PRÉAMBULE**LES RELATIONS ET LA LOGIQUE**

- Quels sont les prédicats représentés par les relations Visite et Étudiant?
- Peut-on les déduire à partir des relations?
- Peut-on déduire les relations à partir des prédicats?
- Y a-t-il un lien entre prédicat et information?

2020-09-04

BDD10 : La théorie relationnelle (V2336) — Christina Khnaisser et Luc Lavoie
Département d'informatique, Faculté des sciences, Université de Sherbrooke, Québec

14

Lorsqu'on rédige le prédicat d'une relation, un usage fréquent veut qu'on place entre guillemets les identifiants des variables du prédicat.

La relation Visite représente le prédicat suivant : la personne nommée « nom » visite « ville ».

La relation Etudiant représente le prédicat suivant : l'étudiant identifié par « matricule » est nommé nom » et né à « ville ».

Remarques :

- on change le prédicat en changeant le temps du verbe « visite », « a visité », « visitera », « a visité ou visite présentement ».
C'est ce qu'on appelle la temporalité (la logique temporelle) une notion avancé que vous pouvez voir dans le cours IGE487.
- le prédicat est particulièrement flou, le sens de nom (prénom, nom de famille, nom à l'état civil, etc.)

Chaque proposition est réputée vraie dans la réalité; C'est-à-dire qu'en substituant les variables du prédicat par des valeurs, la proposition qui en résulte est « vraie ».

La proposition du prédicat Visite énonce que la personne nommée Éliane visite Blanc-Sablon.

La proposition du prédicat Etudiant énonce que l'étudiant identifié par un 15112354 est caractérisé par son Éliane et le lieu de sa naissance Blanc-Sablon.

PRÉAMBULE**QUE CHOISIR ET POURQUOI?**

- Chaque modèle est-il « complet »?
 - Qu'entend-on par complet?
- Chaque modèle facilite certaines opérations
 - Lesquelles?

Toutes les relations sont-elles expressibles dans chacun des modèles ?

Sinon, peut-on exprimer certaines relations dans un modèle qui ne sont pas expressibles dans les autres ?

Examiner les opérations « naturelles » offertes par chaque modèle ?

Toutes les opérations sur les relations sont-elles expressibles dans chacun des modèles ?

Sinon, quelles sont celles qui ne sont pas expressibles dans certains modèles ?

MODÈLES ET REPRÉSENTATIONS

- Avec une matrice ?
- Avec un graphe ?
- Avec un table ?
- Mieux : le modèle (info-)relationnel !

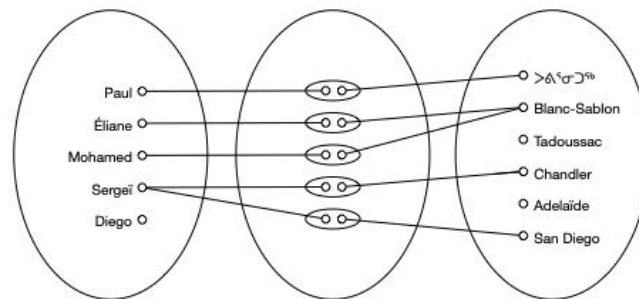
QUEL MODÈLE CHOISIR, LE MODÈLE MATRICIEL?

- Nom = {Paul, Éliane, Mohamed, Sergeï, Diego}
- Ville = { $\text{>}\Delta^{\text{c}}\sigma\text{D}^{\text{cb}}$, Blanc-Sablon, Tadoussac, Chandler, Adlaïde, San Diego}
- Visite = {(Paul, $\text{>}\Delta^{\text{c}}\sigma\text{D}^{\text{cb}}$), (Éliane, Blanc-Sablon), (Mohamed, Blanc-Sablon), (Sergeï, Chandler), (Sergeï, San Diego)}

	$\text{>}\Delta^{\text{c}}\sigma\text{D}^{\text{cb}}$	Blanc-Sablon	Tadoussac	Chandler	Adlaïde	San Diego
Paul	1	0	0	0	0	0
Éliane	0	1	0	0	0	0
Mohamed	0	1	0	0	0	0
Sergeï	0	0	0	1	0	1
Diego	0	0	0	0	0	0

QUEL MODÈLE CHOISIR, LE GRAPHE?

- Nom = {Paul, Éliane, Mohamed, Sergeï, Diego}
- Ville = {>Δ^ςσ>^{ςb}, Blanc-Sablon, Tadoussac, Chandler, Adélaïde, San Diego}
- Visite = {(Paul, >Δ^ςσ>^{ςb}), (Éliane, Blanc-Sablon), (Mohamed, Blanc-Sablon), (Sergeï, Chandler), (Sergeï, San Diego)}



QUEL MODÈLE CHOISIR, LE MODÈLE « TABULAIRE »?

- Nom = {Paul, Éliane, Mohamed, Sergeï, Diego}
- Ville = {>Δ^ςσ>^{ςb}, Blanc-Sablon, Tadoussac, Chandler, Adélaïde, San Diego}
- Visite = {(Paul, >Δ^ςσ>^{ςb}), (Éliane, Blanc-Sablon), (Mohamed, Blanc-Sablon), (Sergeï, Chandler), (Sergeï, San Diego)}

Nom	Ville
Paul	>Δ ^ς σ> ^{ςb}
Éliane	Blanc-Sablon
Mohamed	Blanc-Sablon
Sergeï	Chandler
Sergeï	San Diego

QUE CHOISIR ET POURQUOI?

- Chaque modèle est-il « complet »?
 - Qu'entend-on par complet?
- Chaque modèle facilite certaines opérations
 - Lesquelles?
- **Notre choix : le modèle « info-relationnel » !**

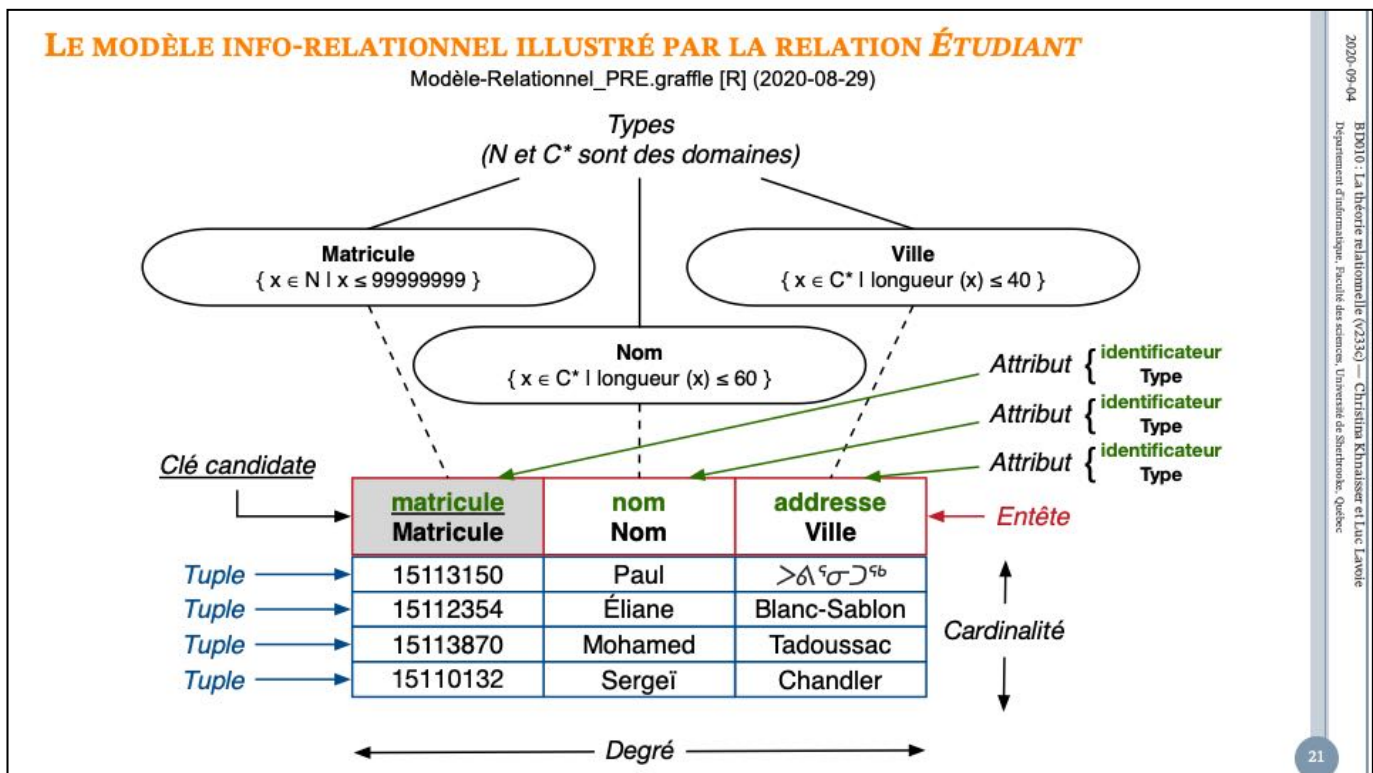
Toutes les relations sont-elles expressibles dans chacun des modèles ?

Sinon, peut-on exprimer certaines relations dans un modèle qui ne sont pas expressibles dans les autres ?

Examiner les opérations « naturelles » offertes par chaque modèle ?

Toutes les opérations sur les relations sont-elles expressibles dans chacun des modèles ?

Sinon, quelles sont celles qui ne sont pas expressibles dans certains modèles ?



Serait-il utile :

- * d'introduire la relation Étudiant au préalable ?
- * d'en donner le prédicat ?
- * de faire la distinction entre proposition et prédicat ?

TYPES, VALEURS, VARIABLES ET RELATIONS

- Domaine, type et relation (bis)
- Référence, dénotation et variable
- Logique et relations (bis)

RAPPELS A

- Un **domaine** est un ensemble *fini* de valeurs *propres*.
- Un **type** est un sous-ensemble d'un **domaine** déterminé par une contrainte (qui en restreint les valeurs).
- Une **relation** de degré n est un **sous-ensemble** du produit cartésien de n domaines (de n types) :

$$D_1 \times D_2 \times \dots \times D_n$$

Pourquoi fini ?

Parce qu'en informatique, nous considérons uniquement les structures finies... tant que nous n'aurons pas de dispositifs de calcul ayant des ressources infinies.

De la définition du type, on tire le corolaire que l'ensemble de valeurs qu'il représente est une sous-ensemble du domaine et conséquemment qu'il est lui-même fini.

RAPPELS B

- Une **référenciation** est une **fonction** associant une référence (adresse) à une valeur.
- Une **dénotation** est une **fonction** associant un nom (identifiant) à une référence.
- Une **variable** est une **dénotation**.
- Une variable est dite **typée** si, par construction, elle réfère toujours à une valeur d'un même type.

Élaborer et motiver les définitions (voir Stoy et Strachey)

Une **dénotation** est une **fonction** associant un nom (identifiant) à une référence.

Une **variable** est une **dénotation** référant à une valeur d'un type, la référence associée peut être modifiée grâce à une opération appelée affectation.

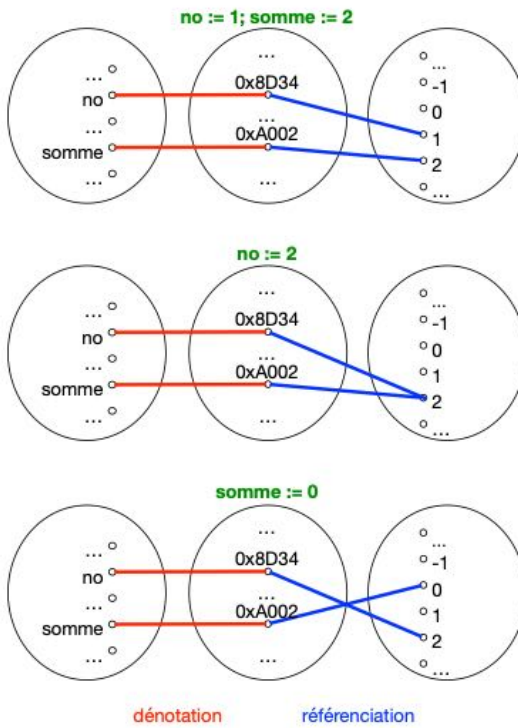
L'admission du domaine des références dans dans celui des types départagent deux classes de modèles : les modèles « à pointeur » des autres.

Entre les modèles d'une même classe, la gestion des références (les opérateurs disponibles) est souvent un facteur de différenciation.

Les modèles relationnel de Codd et de Date sont sans pointeur. Le modèle SQL est avec pointeur.

**VARIABLE ET
AFFECTATION**

Domaine des noms Domaine des références Domaine des entiers



LES RELATIONS ET LA LOGIQUE

relation	prédicat modélisé par un ensemble de tuples représentée par un tableau (une table).
tuple	proposition modélisée par un ensemble d'attributs représenté par une ligne (un enregistrement).
attribut	variable typée représentée par une cellule (un champ).
contrainte	expression logique.
relvar	variable référant une (valeur de) relation.
schéma	un ensemble de définitions de relvar et un ensemble de définitions de contraintes.
base de données	un ensemble de relvars conformes à un schéma (donc aux définitions des relvars et des contraintes).

Note : dans le but de faciliter la gestion de la complexité, on permet généralement qu'une base de données puisse être définie à l'aide de plusieurs schémas; chaque identifiant défini par un schéma est alors qualifié (préfixé) par le nom du schéma.

Tuple : une proposition avérée qui représente un fait dans la réalité

https://fr.wikipedia.org/wiki/Calcul_des_prédicats

https://fr.wikipedia.org/wiki/Calcul_des_propositions

Attention! il y a variable (logique) et variable (informatique)...

Les variables dont on parle dans cette diapositive sont des variables informatiques.

INFORMATION ET RELVAR, RELVAR ET PRÉDICAT, TUPLE ET PROPOSITION

- À chaque relvar on associe un prédicat qui en est « l'interprétation » informationnelle.
- En substituant les valeurs des attributs d'un tuple aux variables correspondantes du prédicat de la relvar, on obtient une proposition.
- À chaque tuple est alors associée une proposition qui en est « l'interprétation » informationnelle.

Lorsqu'on rédige le prédicat d'une relation, un usage fréquent veut qu'on place entre guillemets les identifiants des variables du prédicat.

Les variables dont on parle ici sont des variables « logiques ».

REMARQUES

- L'interprétation informationnelle d'un tuple est aussi appelée, dans certains contextes, « fait ».
- L'association d'une valeur à un attribut passe par la donnée... et c'est très important!
 - C'est la donnée qui permet de rendre l'attribut « manipulable » informatiquement et, corolairement, le tuple, la relvar, le prédicat.
 - À son tour, le prédicat rendu ainsi manipulable permet la représentation et la transmission de l'information.
 - ... et bien d'autres choses encore !

On pense à des systèmes d'inférence, de déduction, d'induction, etc. peuvent être construits sur la base de règles discrètes, probabilistes ou statistiques – notamment.

PRÉAMBULE**DE LA THÉORIE DES TYPES**

- La définition du type présentée ici est minimale. La plupart des auteurs s'entendent pour associer au type un ensemble d'opérateurs. La notion de type peut être développée encore plus.
- La théorie fondamentale des types a été établie dans l'article suivant :
 - Luca Cardelli and Peter Wegner. 1985.
On understanding types, data abstraction, and polymorphism.
ACM Computing Surveys 17, 4 (December 1985), 471–523.
DOI:<http://dx.doi.org/10.1145/6041.6042>
- Le cours IFT 232 approfondira cette question par la présentation de la théorie des types abstraits algébriques (qui est à l'origine de la méthode de conception par classes, aussi appelée programmation orientée-objet).

FONDEMENTS

- Attributs
- Tuples
- Relations
- Opérateurs
- Contraintes

FONDEMENTS — ATTRIBUTS

- Un attribut est un couple formé d'un identifiant **a** et d'un type **D**, noté **a:D**.
- Par abus de langage, lorsque le contexte le permet, il est usuel de désigner l'attribut par son seul identifiant; ainsi écrit-on l'attribut **a**.
- Rappels :
 - Un domaine est un ensemble *fini* de valeurs (distinctes, dans le sens où une même valeur ne peut appartenir à deux domaines distincts).
 - Un type est un ensemble de valeurs défini par un domaine et une contrainte (l'union et l'héritage de types ne sont pas pris en compte ici).

Un tuple correspond à une observation, un « fait ».

Un attribut correspond à une dimension de l'observation, à un « aspect » d'un fait.

La relation est un ensemble d'observations de même type, de même « nature ».

Deux tuples sont de même type si leurs entêtes sont les mêmes.

Transposé dans le domaine de la logique

- un tuple est une proposition (un énoncé vrai sur le monde);
- une relation est un prédicat (...).

Un prédicat peut être défini

- par énumération (l'ensemble de tous les énoncés vrais et eux seuls);
- par compréhension (la caractérisation nécessaire et suffisante des relations entre les variables).

Une base de données est un ensemble de variable de relation définies par leurs valeurs (donc par énumération) et leurs contraintes (donc par compréhension).

En conclusion, une base de données est la représentation d'un système logique.

Dans la pratique, il est souvent difficile d'établir un ensemble de contraintes nécessaires et suffisantes.

Mais on tente de s'en approcher le plus possible.

Pour une théorie des types plus complète, voir (entre autres) IFT 232, IFT 339 et IGE 487.

FONDEMENTS – TUPLES

- Soit a_i des identifiants distincts et D_j des types, un tuple t est défini comme suit :
 - $t \triangleq (\{a_1:D_1, a_2:D_2, \dots, a_n:D_n\}; \{(a_1,v_1), (a_2,v_2), \dots, (a_n,v_n)\})$
 - avec $\forall i : 1 \leq i \leq \text{deg}(t) \Rightarrow \text{val}(t, a_i) \in \text{def}(t, a_i)$
- Où
 - $\text{def}(t) = \{a_1:D_1, a_2:D_2, \dots, a_n:D_n\}$ entête de t
 - $\text{def}(t, a_i) = D_i$ type de a_i
 - $\text{val}(t) = \{(a_1,v_1), (a_2,v_2), \dots, (a_n,v_n)\}$ valeur de t
 - $\text{val}(t, a_i) = v_i$ valeur de a_i
 - $\text{deg}(t) = n$ degré de t
 - $\text{id}(t) = \{a_1, a_2, \dots, a_n\}$ les identifiants d'attributs de t

Le cas $\text{deg}(t) = 0$, est important.

Il existe un seul tuple possible (pourquoi?) :

- $t_0 = (\{\}, \{\})$

Notation simplifiée fondée sur l'ordre d'énumération des attributs

(les identifiants d'attributs et leurs domaines étant déterminés par ailleurs) :

- $t = \langle v_1, v_2, \dots, v_n \rangle$

Finalement, la notation $t.a_i$ désigne l'attribut $a_i:D_i$ dans $\text{def}(t)$.

FONDEMENTS — RELATIONS

○ Soit a_i des identifiants distincts, D_j des types et t_k des tuples, une relation R est définie comme suit :

- $R \triangleq (\{a_1:D_1, a_2:D_2, \dots, a_n:D_n\}; \{t_1, t_2, \dots, t_m\})$
- avec $\forall i : 1 \leq i \leq \text{card}(R) \Rightarrow \text{def}(R) = \text{def}(t_i)$

○ Où

- $\text{def}(R) = \{a_1:D_1, a_2:D_2, \dots, a_n:D_n\}$ entête de R
- $\text{def}(R, a_i) = D_i$ type de a_i
- $\text{val}(R) = \{t_1, t_2, \dots, t_m\}$ valeur de R
- $\text{deg}(R) = n$ degré de R
- $\text{card}(R) = m$ cardinalité de R
- $\text{id}(R) = \{a_1, a_2, \dots, a_n\}$ identifiants d'attributs de R

Un prédicat peut être défini

- par énumération (l'ensemble de tous les énoncés vrais et eux seuls);
- par compréhension (la caractérisation nécessaire et suffisante des relations entre les variables).

Le cas $\text{deg}(R) = 0$ est important.

Il existe deux relations possibles (pourquoi?) :

- $R_0 = (\{\}, \{\})$
- $R_1 = (\{\}, \{t_0\})$

et elles sont très importantes, comme le zéro et le un pour les entiers!

Finalement, la notation $R.a_i$ désigne l'attribut $a_i:D_i$ dans $\text{def}(R)$.

FONDEMENTS — ILLUSTRATION

Grâce aux
contraintes sur les
tuples et les
relations, la
représentation
tabulaire initiale
est donc bien
fondée.

Quatre tuples (ayant le même entête)

matricule : Matricule	nom : Nom	adresse : Ville	
matricule : 15113150	nom : Paul	adresse : >Δ ^σ σ ^{5b}	t1
matricule : 15112354	nom : Éliane	adresse : Blanc-Sablon	t2
matricule : 15113870	nom : Mohamed	adresse : Tadoussac	t3
matricule : 15110132	nom : Sergeï	adresse : Chandler	t4

Une relation comprenant quatre tuples

matricule : Matricule	nom : Nom	adresse : Ville	
matricule : 15113150	nom : Paul	adresse : >Δ ^σ σ ^{5b}	t1
matricule : 15112354	nom : Éliane	adresse : Blanc-Sablon	t2
matricule : 15113870	nom : Mohamed	adresse : Tadoussac	t3
matricule : 15110132	nom : Sergeï	adresse : Chandler	t4

La représentation compacte usuelle de cette même relation

matricule : Matricule	nom : Nom	adresse : Ville
15113150	Paul	>Δ ^σ σ ^{5b}
15112354	Éliane	Blanc-Sablon
15113870	Mohamed	Tadoussac
15110132	Sergeï	Chandler

Si on se reporte aux définitions de domaine et de types de la figure du préambule.

Si on considère que l'ensemble des caractères C contient 256 caractères différents.

Combien y a-t-il de valeurs de tuple différentes ?

$$n = (99999999 - 10000000 + 1) \times 256^{60} \times 256^{40} = 9 \times 10^7 \times 2^{800} \approx 6 \times 10^{248}$$

Combien y a-t-il de valeurs de relation différentes ?

toute tuple peut être présent ou non, donc

$$m = 2^n \approx 2^{(6 \times 10^{248})} \approx 10^{(2 \times 10^{248})}$$

Combien y a-t-il d'atomes dans l'Univers ?

Plusieurs scientifiques estiment qu'il est de l'ordre de $10^{80} \approx 10^{(8 \times 10^1)}$

(voir par exemple <https://www.science-et-vie.com/galerie/sait-on-combien-il-y-a-d-atomes-dans-l-univers-6154>)

**FONDEMENTS –
RACCOURCIS ET
NOTATIONS**

- Notations équivalentes à $\text{val}(\mathbf{t}, \mathbf{a}_i)$
 - $\mathbf{t}.\mathbf{a}_i$
 - $\mathbf{a}_i(\mathbf{t})$
 - $\mathbf{t}(\mathbf{a}_i)$
 - \mathbf{a}_i from \mathbf{t}
- Nous utiliserons fréquemment
 - $\mathbf{t}.\mathbf{a}_i$

OPÉRATEURS

- Premier essai
- Renommage
- Deuxième essai
- Et encore plus !

FONDEMENTS — OPÉRATEURS NATURELS

Restriction $R \sigma_{\text{cond}}$

A	B	C
a1	b1	c1
a2	b2	c2
a3	b3	c3

Projection $R \pi \{A, C\}$

A	B	C
a1	b1	c1
a2	b2	c2
a3	b3	c3

Jointure naturelle $R \bowtie S$

A	B
a1	b1
a2	b1
a3	b3
a4	b4

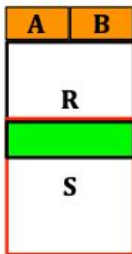
 \bowtie

B	C
b1	c1
b2	c2
b3	c3
b3	c4

 $=$

A	B	C
a1	b1	c1
a2	b1	c1
a3	b3	c3
a3	b3	c4

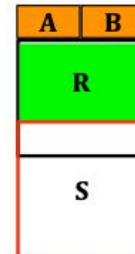
Intersection $R \cap S$



Union $R \cup S$



Différence $R - S$



Note : Le symbole de projection π est souvent omis, à l'instar de la multiplication dans les polynômes.

Opérateurs relationnels propres : 3

Opérateurs ensemblistes : 4

Opérateur structurel : 1 (renommage)

IGE 487

Le produit est-il vraiment nécessaire ?

Et le renommage ?

Quel est l'ensemble de base minimal (nécessaire et suffisant) ?

Cet ensemble est-il unique ?

Voir

DATE, C. J. ; DARWEN, H.

Databases, types, and the relational model: The third manifesto.

3rd ed., Addison-Wesley Inc., 2008.

ISBN 0-321-39942-0

FONDEMENTS — OPÉRATEUR DE RENOMMAGE

- La présence de l'entête dans chacun des tuples et chacune des relations permet de définir un opérateur de renommage.
- L'entête d'une relation est conservé dans le catalogue du SGBDR.
- Le catalogue est la description des schémas relationnels du SGBDR sous la forme d'une BD dont le schéma est lui-même dans le catalogue, comme tous les autres schémas de toutes les autres BD du SGBDR.

Renommage
R ρ A:C

A	B
a1	b1
a2	b2
a3	b3

ρ A:C =

C	B
a1	b1
a2	b2
a3	b3

Pourquoi, et surtout comment, les opérateurs structurels (aussi appelés meta-opérateurs, tels que le renommage) peuvent-ils considérés redondants ?

Astuce : prendre le catalogue en considération.

Esquisser un schéma rudimentaire du catalogue et illustrer votre réponse.

FONDEMENTS — OPÉRATEURS DE BASE

- On constate toutefois que $R \cap S = R - (R - S) = R \bowtie S$
- L'opérateur \cap n'est donc pas « nécessaire ».

Restriction
 $R \sigma \text{ cond}$

A	B	C
a1	b1	c1
a2	b2	c2
a3	b3	c3

Projection
 $R \pi \{A, C\}$

A	B	C
a1	b1	c1
a2	b2	c2
a3	b3	c3

Jointure (naturelle)
 $R \bowtie S$

A	B
a1	b1
a2	b1
a3	b3
a4	b4

 \times

B	C
b1	c1
b2	c2
b3	c3
b3	c4

 $=$

A	B	C
a1	b1	c1
a2	b1	c1
a3	b3	c3
a3	b3	c4

Différence
 $R - S$

A	B
R	
S	

Union
 $R \cup S$

A	B
R	
S	

Renommage
 $R \rho A:C$

A	B
a1	b1
a2	b2
a3	b3

 $\rho A:C =$

C	B
a1	b1
a2	b2
a3	b3

FONDEMENTS — OPÉRATEURS NATURELS
RÈGLES D'APPLICATION

- Les trois opérateurs ensemblistes ne sont bien définis que si les entêtes des relations sont identiques.
- L'opérateur de projection n'est bien défini que si tous les identifiants d'attributs sont définis dans l'entête de la relation.
- L'opérateur de restriction n'est bien défini que si tous les identifiants d'attributs de la condition sont définis dans l'entête de la relation.
- L'opérateur de renommage n'est bien défini que si l'identifiant à changer est défini dans l'entête de la relation et que le nouvel ne l'est pas.

FONDEMENTS — PRODUIT CARTÉSIEN OU JOINTURE ?

○ Constatons d'abord l'un se calcule à l'aide de l'autre :

- $R \times S = R \bowtie S$
pourvu que R n'ait aucun attribut commun avec S
ce dont on peut s'assurer par renommage;
- $R \bowtie S = (R \times S) \sigma (R.a_1 = S.a_1 \wedge \dots \wedge R.a_n = S.a_n)$
pourvu qu'on ait la notion pointée à disposition.

Préférez-vous le chocolat ou le cacao ?

FONDEMENTS — OPÉRATEURS : ENSEMBLE MINIMAL ?

- En pratique, deux ensembles de six opérateurs de base sont proposés, selon les auteurs :
 - Codd : $\pi \sigma \times \cup - \rho$
 - Date : $\pi \sigma \bowtie \cup - \rho$
- Un ensemble minimal comprend aussi peu que deux opérateurs, voir l'algèbre \mathcal{A} proposée par Date (2007).
- Nous utiliserons toutefois tantôt l'un, tantôt l'autre, car ils proposent un bon équilibre entre minimalisme et expressivité.

Montrer aussi comment on peut retirer les opérateurs de restriction et d'extension en considérant les fonctions comme des cas particulier de relation.

Montrer que l'intersection est aussi un cas particulier de jointure, comme le produit cartésien.

FONDEMENTS — OPÉRATEURS CONSTRUITS

Extension (augmentation)

$$R \bowtie C:f = (R \bowtie F) \rho f':C$$

A	B	A	B	C
a1	b1	a1	b1	f(a1,b1)
a2	b1	a2	b1	f(a2,b1)
a3	b3	a3	b3	f(a3,b3)
a4	b4	a4	b4	f(a4,b4)

Note : Le symbole de l'extension varie beaucoup d'un auteur à l'autre.

Semi-jointure (matching)

$$R \ltimes S = (R \ltimes S) \pi R$$

A	B	B	C	A	B
a1	b1	b1	c1	a1	b1
a2	b1	b2	c2	a2	b1
a3	b3	b3	c3	a3	b3
a4	b4	b3	c4		

Produit $R \times S = R \bowtie S$ (jointure sans attributs partagés)

A	B	A	B
x	a	x	a
y	b	x	b
	c	x	c
		y	a
		y	b
		y	c

Semi-différence (not matching, anti-join...) $R \times S = R - (R \ltimes S)$

A	B	B	C	A	B
a1	b1	b1	c1		
a2	b1	b2	c2		
a3	b3	b3	c3		
a4	b4	b3	c4	a4	b4

Pour l'extension, F est la relation correspondant à la fonction f et f' est l'attribut de F correspondant à l'image de f. Au besoin les opérandes de F pourront être renommés pour correspondre aux attributs idoines de R.

Pour le produit, on suppose que R et S n'ont aucun attribut partagé (sinon ceux-ci doivent être renommés au préalable).

FONDEMENTS — OPÉRATEURS CONSTRUITS

- Se reporter au module BD012 qui comprend une suite d'exemples de requêtes relationnelles utilisant les opérateurs de base et quelques opérateurs construits.

CONTRAINTES

- Contraintes générales
- Contraintes spécifiques
- Vocabulaire

CONTRAINTES

- **Contrainte (générale)**
 - Une contrainte est une expression logique applicable aux objets d'une base de données; l'expression qui doit être maintenue vraie tout au long de l'existence de la base de données.
- **Contrainte (spécifique)**
 - Toute autre contrainte spécifique est un cas particulier de contrainte générale.
 - On désigne souvent une contrainte spécifique selon
 - sa portée par rapport à un objet particulier de la base de données : contrainte d'attribut, contrainte de relation, contrainte de schéma ;
 - son rôle : clé candidate, clé référentielle

Notes :

* si on accepte que les seuls objets d'une BD sont des relations, alors une contrainte est une « expression logique applicable aux relations d'une base de données »

CONTRAINTES SPÉCIFIQUES (PORTÉE)

- **Contrainte de domaine**
 - restreindre l'ensemble des valeurs d'un domaine (pour en faire un type, par exemple).
- **Contrainte d'attribut**
 - limiter la valeur des attributs d'un tuple entre eux.
- **Contrainte de relation**
 - limiter globalement les valeurs de la relation (donc celles des tuples de la relation entre eux);
 - exemple : clés candidates.
- **Contrainte d'intégrité référentielle**
 - limiter globalement les valeurs d'attributs d'une relation relativement à une clé candidate d'une autre;
 - exemple : clés référentielles (étrangères).

CONTRAINTES SPÉCIFIQUES (RÔLE)

○ Clé (d'une relation R)

- sous-ensemble d'attributs déterminant un tuple unique au sein de la relation.

○ Soit X une clé de R :

$$\#R = \#(R \pi \{X\})$$

○ Clé référentielle (d'une relation R sur une relation S)

- sous-ensembles d'attributs dont les valeurs sont restreintes par la clé d'une (autre) relation

○ Soit X une clé référentielle de R sur S :

$$(R \pi \{X\}) \subseteq (S \pi \{X\}) \text{ et } (X \text{ est une clé de } S)$$

Dépendance fonctionnelle

Une dépendance fonctionnelle est une association entre deux sous-ensembles d'attributs d'une même relation, le premier déterminant le second. Pour exprimer cette contrainte, on a traditionnellement recours au concept de clé candidate : tout sous-ensemble minimal d'attributs d'une relation déterminant toujours uniquement (tous) les autres attributs. Le concept peut aisément être étendu entre les relations d'une même base de données grâce au concept de dépendance d'inclusion.

Dépendance d'inclusion

Un ensemble d'attributs X de R est en dépendance d'inclusion de (la relation) S

$$\text{ssi } R \pi \{X\} \subseteq S \pi \{X\}$$

Clé référentielle

Si X est une clé candidate de S, alors on dit que X est une clé référentielle de R sur S.

Faut-il parler des à présent des clés candidates primaires et secondaires ?

--

VOCABULAIRE

- Clé irréductible
 - clé dont on ne peut retirer aucun attribut sans qu'elle cesse d'être une clé.
- Clé candidate
 - synonyme : clé irréductible.
- Surclé
 - tout ensemble d'attributs contenant une clé.

CONTRAINTES, UNE SUITE ?

- Nous reprendrons l'étude des clés (et des dépendances fonctionnelles) quand nous aurons mieux maîtrisé la théorie relationnelle.

DE LA THÉORIE AUX MODÈLES

Plusieurs modèles ont été proposés sur la base d'une même théorie

- Modèle de Codd I
- Modèle de Codd II
- Modèle de Date
- Modèle d'Ullman
- Modèles SQL
- ...
- Et la liste n'est pas complète!
- ...
- Que choisir ?

DE LA THÉORIE AUX MODÈLES
THÉORIE, MODÈLES ET LANGAGES

- Il y a **une** théorie relationnelle.
- Il y a **plusieurs** modèles relationnels, par exemple
 - Modèle de Codd I
 - Modèle de Codd II
 - Modèle de Date
 - Modèle d'Ullman
 - Modèles SQL (ANSI:1992, ISO:1999, ISO:2006... ISO:2016...)
- Pour **chacun** de ces modèles, il y a **plusieurs** langages
(et même, plusieurs dialectes pour certains de ces langages)

DE LA THÉORIE AUX MODÈLES

QUE CHOISIR ?

- Pour l'exposé des principes relationnels, nous utiliserons toujours le modèle de Date.
- Pour la programmation SQL, nous présenterons des techniques permettant d'être aussi proche que possible du modèle de Date, en indiquant les écarts possibles en fonction du modèle SQL ISO 9075:2016.
- Certains des autres modèles seront couverts par les activités IFT 287, IGE 487 et IFT 723.
- Le calcul des propositions et des prédicats est couvert par l'activité MAT 115.

DONNÉES ABSENTES

- Pourquoi une donnée serait-elle absente?
- Un modèle simple
- Solutions AVEC annulabilité
- Solutions SANS annulabilité
- Impact sur la logique devant être utilisée dans le modèle
- Et, non, le problème n'est pas résolu!

DONNÉES ABSENTES

POURQUOI UNE DONNÉE SERAIT-ELLE ABSENTE?

- Discussion en classe.

DONNÉES ABSENTES**UN MODÈLE SIMPLE (PROPOSÉ PAR CODD)****○ NA**

- L'information n'est **pas applicable**.
- Dans ce cas, l'annulabilité est à remettre en question; une bonne modélisation permet généralement d'éviter d'y avoir recours.

○ IN

- L'information est **inconnue**.
- Dans ce cas, l'annulabilité pourrait être légitime; la question est de savoir comment la représenter pour que cela pose le moins de problèmes possible.

○ X

- L'information n'est **pas accessible**.
 - **À court terme** : dans ce cas, le gestionnaire transactionnel permet d'éviter l'utilisation de l'annulabilité. L'information peut être partagée entre plusieurs utilisateurs en même temps avec un contrôle des accès concurrents. L'exécution d'une transaction doit préserver la cohérence de la BD.
 - **À long terme** : équivalent à IN.

DONNÉES ABSENTES**IMPACT SUR LA LOGIQUE DES MODÈLES RELATIONNELS**

- On en conclut que
 - le cas **X**, qui se réduit en **IN** ou **NA**, peut être ignoré ;
 - le cas **NA**, pourrait être ignoré, si la modélisation est (toujours) adéquate.
- Que faire du cas **IN** ?
 - Plusieurs théoriciens, dont Date, ont décidé de ne pas intégrer ce cas au modèle relationnel et de le traiter aussi par la modélisation, préservant ainsi la logique **bi-valuée** (faux, vrai) et donc booléenne.
 - Les membres du Comité de normalisation du langage SQL ont décidé d'intégrer ce cas à leur modèle et, conséquemment, d'utiliser une logique **tri-valuée** (faux, vrai, inconnu).

DONNÉES ABSENTES**SOLUTIONS SANS ANNULABILITÉ (DATE ET CIE)****○ Principes**

- Séparer les propositions complètes des incomplètes.
- Conserver les causes d'absence séparément.

○ Pour un inventaire des techniques de modélisation

- <http://www.dcs.warwick.ac.uk/~hugh/TTM/Missing-info-without-nulls.pdf>

DONNÉES ABSENTES**SOLUTIONS AVEC ANNULABILITÉ (SQL ET CIE)****○ Trois étapes à faire dans l'ordre :**

- si possible, corriger cette lacune à la source (dans la réalité);
- si possible, modifier le modèle conceptuel pour en tenir compte;
- sinon, introduire le concept d'annulabilité dans la théorie relationnelle, ce qui induit le recours
 - à l'un des deux artifices suivants :
 - un **marqueur** NUL (une propriété des attributs) ou
 - une **valeur** NULLE ajoutée à tous les domaines.
 - et à une logique **à 3 valeurs**
 - (afin de pouvoir définir l'égalité, essentielle aux opérations d'affectation, de restriction, de jointure, d'union...)

DONNÉES ABSENTES
EXEMPLE DE LOGIQUE À 3 VALEURS (SQL)

V : Vrai
 F : Faux
 I : Inconnu

<i>A</i>	<i>B</i>	$A \vee B$	$A \wedge B$	$\neg A$
V	V	V	V	F
V	I	V	I	F
V	F	V	F	F
I	V	V	I	I
I	I	I	I	I
I	F	I	F	I
F	V	V	F	V
F	I	I	F	V
F	F	F	F	V

REMARQUES

- Codd, dans son deuxième modèle, a proposé d'introduire deux marqueurs (IN et NA), ce qui induit une logique à 4 valeurs.
- D'autres chercheurs ont proposé d'autres systèmes logiques à 4 et même 5 valeurs.
- Humblement, nous nous en tiendrons, le plus souvent, à la logique booléenne dans le cadre du présent cours.

DONNÉES ABSENTES

**Et non,
le problème
n'est pas
résolu!**

**Nous y
reviendrons
donc dans le
module BD028**

De: oracle-acct_ww@oracle.com
Objet: Nom d'utilisateur de votre compte Oracle
Date: 2 octobre 2014 19:44
À: luc.lavoie@usherbrooke.ca

ORACLE

Cher/Chère NULL !,

Vous avez demandé à recevoir par email le nom d'utilisateur de votre compte Oracle.

Votre nom d'utilisateur est : **luc.lavoie@usherbrooke.ca**

Merci !

L'équipe de gestion des comptes Oracle

Mettez votre compte à jour :

- > [Abonnez-vous aux communications](#) dédiées aux thèmes qui vous intéressent.
- > [Devenez membre des communautés Oracle.](#)
- > [Pour modifier votre adresse email, votre mot de passe](#) ou toute autre information de votre compte, cliquez sur le lien [Compte](#) en haut des pages Oracle.com.

Obtenir de l'aide

- > Des questions ? [Aide \(page Account Help\)](#)
- > Se connecter
 - [Envoyer une demande d'aide](#)
 - profilehelp_ww@oracle.com

Hardware and Software
ORACLE
Engineered to Work Together



Copyright © 2014, Oracle et/ou ses filiales.
Tous droits réservés.

[Aide \(page Account Help\)](#) | [Ne pas envoyer d'email](#) | [Mentions légales](#) | [Conditions d'utilisation](#) | [Confidentialité](#)

VOCABULAIRE**LANGAGES ET NOTATIONS**

- Un script est la description textuelle d'un modèle de données conforme à une théorie (entité-association, relationnelle, etc.).
- Un diagramme est une représentation graphique (le plus souvent incomplète) d'un script.
- Nous utiliserons souvent en cours un langage textuel inspiré de l'algèbre relationnelle (sous le nom « langage relationnel »).
- SQL comprend un langage de description textuelle de schémas relationnels... c'est l'un des langages de programmation les plus utilisés au monde.
- Nous utilisons en cours deux notations graphiques
 - la notation graphique classique pour les diagrammes relationnels (des rectangles pour les relations, des flèches pour les clés référentielles);
 - la notation graphique classique pour les diagrammes entité-association (des rectangles, des ovales, des losanges et divers types de flèches).
- UML est une autre notation graphique fréquemment utilisée pour les diagrammes de classes or, plusieurs l'utilisent aussi pour représenter des diagrammes entité-association.

À développer.

Définir le langage relationnel et la notation graphique classique dans le modules BD012 – Exemples relationnels

VOCABULAIRE**... ET ARTICLES ENCYCLOPÉDIQUES D'INTRODUCTION**

- Une théorie (mathématique) est un ensemble d'affirmations dont certaines sont des axiomes et les autres des théorèmes démontrables à partir de ces axiomes et au moyen de règles d'inférence (exprimée à l'aide de la) logique.
 - <http://fr.wikipedia.org/wiki/Théorie>
- Un modèle est une représentation conforme à une théorie.
 - <http://fr.wikipedia.org/wiki/Modèle>
- Un langage (formel) est un formalisme permettant de décrire des propositions sémantiquement interprétables en termes d'un modèle.
 - http://fr.wikipedia.org/wiki/Langage_formel
- Une théorie peut être à l'origine de plusieurs modèles, un modèle de plusieurs langages, un langage de plusieurs dialectes.
- Pour en savoir plus sur le calcul des prédicats :
 - http://fr.wikipedia.org/wiki/Calcul_des_prédicats

THÉORÈME, COROLAIRES ET CONSTATS.

- Théorème d'incomplétude de Gödel :
 - *Toute théorie cohérente ayant un nombre fini d'axiomes exprimés dans un langage qui permet de décrire l'arithmétique comprend des propositions indécidables.*
 - Exemples :
 - l'arithmétique de Peano,
 - la théorie des ensembles.
- Par ailleurs, un modèle n'est pas forcément complet relativement à une théorie.
- Par ailleurs, un langage n'est pas forcément complet relativement à un modèle.

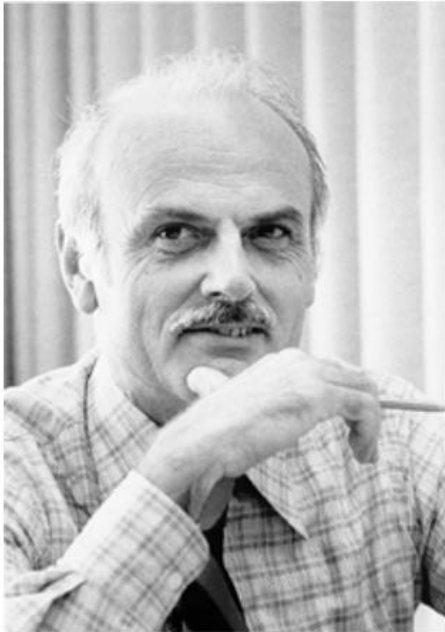
RÉFÉRENCES

- Théorie relationnelle
 - E.F. Codd. 1990.
The Relational Model for Database Management: Version 2.
Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.
 - C.J. Date, H. Darwen. 2007.
Databases, types and the relational model: the third manifesto.
Reading, Mass.: Addison-Wesley.
 - F. de Sainte Marie. 2013.
Bases de données relationnelles et normalisation : de la première à la sixième forme normale.
<https://fsmrel.developpez.com/basesrelationnelles/normalisation/>
 - H. Darwen. 2006.
How To Handle Missing Information Without Using NULL.
<http://www.dcs.warwick.ac.uk/~hugh/TTM/Missing-info-without-nulls.pdf>
- Manuels classiques
 - [C. J. Date 2004], chapitre 3.
 - [Elmasri and Navathe 2004], chapitre 4.
 - [Elmasri and Navathe 2011], chapitre 3.
 - [Elmasri and Navathe 2016], chapitre 8.
 - [Ullman and Widom 2008], chapitre 3.

LES COLLES DU PROF

- Quelles différences existe-t-il entre
 - un domaine et type?
 - un tuple et une relation?
 - une relation et une variable de relation?
 - un schéma et une base de données?
 - une théorie et un modèle?
 - un modèle et un langage?
 - une clé candidate et une superclé?
- Quels sont les opérateurs de base proprement relationnels?
- En quoi se distinguent-ils des opérateurs ensemblistes?

EDGAR FRANK CODD ET CHRISTOPHER J. DATE



https://en.wikipedia.org/wiki/Edgar_F._Codd



Photo of Chris Date by Douglas Robertson, Edinburgh

https://en.wikipedia.org/wiki/Christopher_J._Date

