

Modélisation, conception et exploitation de données

Temporalisation des relations

Quelles sont les contraintes nécessaires à la temporalisation des relations?

TEM_02a
120a
2023-09-27

Christina.Khnaisser@USherbrooke.ca
Luc.Lavoie@USherbrooke.ca

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

1

Plan

○ Rappels

- Modèle relationnel
- Normalisation
- Opérateurs de Allen

○ Exigences

- non-Redondance
- non-Circonlocution
- non-Contradiction
- Compacité

○ Temps

- de domaine
- de référence
 - de transaction
 - de validation

2

Rappels

- Modèle relationnel
- Normalisation
- Opérateurs de Allen

Rappels

Type, valeur et variable de relation

- Relation (variable de relation) est une représentation d'un prédicat
 - Attribut est une propriété non annulable :
 $a_i : D_i$
 - Entête est un ensemble d'attributs typés :
 $\{a_1 : D_1, a_2 : D_2, \dots, a_n : D_n\}$
 - Tuple une proposition vraie relativement à un prédicat :
 $\{(a_1, v_1), (a_2, v_2), \dots, (a_n, v_n)\}$

Rappels

Invariants, vues et schémas

○ Invariant :

- Contrainte : applicable à une relation;
 - cas particulier : unicité
- Assertion : applicable à plusieurs relations;
 - cas particulier : intégrité référentielle
- Dépendance : DF, DJ

○ Vue (*relation virtuelle*):

- relation définie à partir d'une expression relationnelle

○ Schéma $S (\{r_1, \dots, r_n\}, \{c_1, \dots, c_m\}, \{w_1, \dots, w_t\})$

- ensemble de déclarations de relation $R = \{r_i, \dots, r_n\}$
- ensemble d'assertions $C = \{c_i, \dots, c_m\}$
- ensemble de vues $W = \{w_i, \dots, w_t\}$.

5

Rappels

Normalisation, requête et transaction

○ Forme normale :

- 2FN, 3FN, FNBC, 4FN, 5FN et 6FN

○ Base de données : un ensemble de valeurs conformes à un schéma ssi :

- elle comprend exactement une valeur d'une relation pour chacune des déclarations de relation R_i du schéma et
- tous les invariants sont satisfaits.

○ Requête :

- De consultation
- De modification

○ Transaction :

- liste ordonnée de requêtes exécutées séquentiellement et sans interruption.
- ACID (atomicité, cohérence, indépendance, durabilité)

6

Rappels Opérateurs de Allen

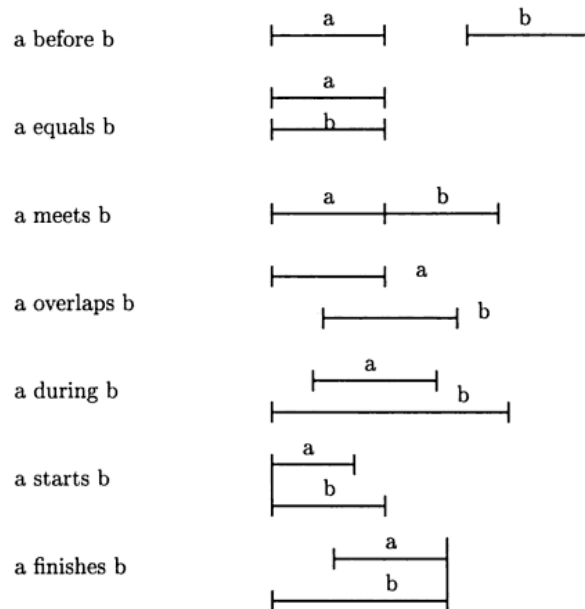


Figure 13.1 Allen's interval comparison operators

Exigences

- Exemple de référence
- Non-redondance
- Non-circonlocution
- Non-contradiction
- Compacité
- Généralisation

Exemple de référence

- Soit **S** une relation ne comportant que des attributs de type scalaire non intervalle telle que **K** est l'ensemble des attributs clés de **S** et **A** l'ensemble des attributs non clés de **S**. Soit **R**, une relation comprenant les attributs **K ∪ A ∪ {d}** et de clé **K ∪ {d}** (une relation construite en ajoutant un seul attribut, **d**, de type intervalle aux attributs de **S**).
- Pour les exemples, la relation **R** est illustrée grâce à la relation **Séjour {noPatient, noUnite, noChambre, d}** dont le prédicat est « le patient *noPatient* était sous la responsabilité de l'unité *noUnite* et occupait la chambre *noChambre* durant la période *d* » et la clé **{noPatient, d}**.

Exigence non-Contradiction

La non-contradiction est garantie lorsque les intervalles de deux tuples ayant la même clé dans **S** et des données différentes n'ont aucun point en commun :

$$\forall y_1 \in r, y_2 \in r. y_1 \neq y_2 \wedge y_1 \pi \{K\} = y_2 \pi \{K\} \wedge y_1 \pi \{A\} \neq y_2 \pi \{A\} \Rightarrow \\ \neg(y_1 \pi \{d\} \text{ OVERLAPS } y_2 \pi \{d\})$$

Exemple de contradiction :

Séjour			
noPatient	noUnite	noChambre	d
P1	U1	1200	[j02:j08]
P1	U1	1300	[j04:j10]

Le patient P1 à séjourné à la chambre 1200 et à la chambre 1300 durant [j04:j08].

Exigence non-Redondance

La non-redondance est garantie lorsque les intervalles de deux tuples contenant les mêmes données n'ont aucun point en commun :

[non-redondance]

$$\forall y_1 \in r, y_2 \in r. y_1 \neq y_2 \wedge y_1 \pi (K \cup A) = y_2 \pi (K \cup A) \Rightarrow \\ \neg(y_1 \pi \{d\} \text{ OVERLAPS } y_2 \pi \{d\})$$

Exemple de redondance :

Séjour			
noPatient	noUnite	noChambre	d
P1	U1	1200	[j02;j08]
P1	U1	1200	[j04;j10]

Il existe deux tuples dans la relation Séjour qui montre que le patient P1 a séjourné à la chambre 1200 sous la responsabilité de l'unité U1 durant la période [j04;j08].

11

Exigence non-Circonlocation

La non-circonlocation est garantie lorsque les intervalles de deux tuples contenant les mêmes données ne se jouxtent pas :

[non-circonlocation]

$$\forall y_1 \in r, y_2 \in r. y_1 \neq y_2 \wedge y_1 \pi (K \cup A) = y_2 \pi (K \cup A) \Rightarrow \\ \neg(y_1 \pi \{d\} \text{ MEETS } y_2 \pi \{d\})$$

Exemple de circonlocation :

Séjour			
noPatient	noUnite	noChambre	d
P1	U1	1200	[j02;j08]
P1	U1	1200	[j09;j10]

Deux tuples dans la relation Séjour qui montrent que le patient P1 a séjourné à la chambre 1200 de l'unité U1 alors qu'un seul était suffisant.

12

Exigence non-Redondance et non-Circonlocation

Sachant que $\text{MERGES} \equiv \text{OVERLAPS} \vee \text{MEETS}$, les deux règles (non-redondance et non-circonlocation) peuvent être combinées en une seule :

$$\begin{aligned} & [\text{non-circonlocation_non-redondance}] \\ & \forall y_1 \in r, y_2 \in r. (y_1 \neq y_2 \wedge y_1 \sqcap (K \cup A) = y_2 \sqcap (K \cup A) \Rightarrow \\ & \neg (y_1 \sqcap \{d\} \text{ MERGES } y_2 \sqcap \{d\})) \end{aligned}$$

Exigence Compacité

La compacité garantit la cohérence référentielle temporelle entre deux relations.

Soit $R1(K1, A1, d)$, $R2(K2, A2, d)$ avec
 $(K2 \subseteq K1 \cup A1) \wedge (\text{REF } R1(K2) \rightarrow R2)$

$\forall y1 \in R2, \exists y2 \in R2.$
 $(y1 \sqcap K2 = y2 \sqcap K2) \wedge (y1 \sqcap \{d\} = y2 \sqcap \{d\})$ [stricte \equiv prédicative]

$\forall y1 \in R2, \exists y2 \in R2.$
 $(y1 \sqcap K2 = y2 \sqcap K2) \wedge (y1 \sqcap \{d\} \subseteq y2 \sqcap \{d\})$ [inclusive \equiv référentielle]

Exigence Exemple de non-Compacité

Soit la relation **Hospitalisation** {**noPatient**, **pHospitalisation**} dont le prédicat est « le patient *noPatient* est hospitalisé durant la période *pHospitalisation* » participant à l’assertion référentielle **Séjour (noPatient) → Hospitalisation** :

Hospitalisation	
noPatient	pHospitalisation
P1	[j02;j20]

Séjour			
noPatient	noUnite	noChambre	d
P1	U2	2400	[j02;j23]

Dans l’exemple ci-dessus la relation Séjour ne respecte pas la contrainte de compacité : le patient P1 n’était pas hospitalisé pas durant la période [j21;j23].

Manque de généralité des règles

Les règles énoncées précédemment ne tiennent pas compte de l’éventuelle répartition des périodes sur plusieurs tuples. Ainsi la relation suivante répond à l’exigence de compacité, mais pas à la règle telle qu’énoncée.

Séjour			
noPatient	noUnite	noChambre	d
P1	U1	1200	[j02;j18]
P1	U2	2400	[j19;j20]

L’expression adéquate de règles garantissant ces exigences est au coeur de chaque modèle de temporalisation. Nous examinerons les moyens utilisés par BCDM et par TRM.

Temps et référentiels

- Temps de domaine
- Temps de référence
 - Référentiel de transaction
 - Référentiel de validité
- Bitemporalité

17

Les référentiels de temps

Le temps du domaine

Un temps du domaine (*user-defined time*) fait référence à l'horloge associée à l'activité (ou au processus) ayant engendré le fait représenté par la proposition. Il peut en outre y avoir plusieurs référentiels de domaine dans un contexte où les processus ne partagent pas une même horloge.

Exemple : Tom a été hospitalisé du jour 21 au jour 23. La début est obtenu par le processus d'admission, la fin du processus de congé. On suppose que les deux processus font référence à la même horloge.

Hospitalisation	
noPatient	pHospitalisation
P1	[j21;j23]

Une problématique surgit lorsque tous les processus pris en compte ne réfèrent pas à la même horloge. Comme il s'agit d'un problème de modélisation du monde réel, il n'en sera pas traité dans le présent mémoire.

18

Les référentiels de temps

Le temps de transaction

Un temps de transaction (*transaction time*) fait référence à l'horloge du SGBD qui enregistre une proposition (un tuple) dans la BD. Sa valeur est établie par le SGBD sur la base de l'instant de la confirmation (*commit time*) de la transaction (insertion ou retrait des tuples associés). La période de transaction d'un tuple est donc définie par l'instant de la transaction de son insertion jusqu'à l'instant de transaction de son retrait. L'utilisateur ne peut donc la modifier qu'indirectement, par l'entremise d'une transaction. Il n'y a qu'un seul référentiel de transaction.

Exemple : La BD a enregistré le jour 30 que Tom a été hospitalisé du jour 21 au jour 28. Cette information est présente (et enregistrée) depuis ce moment et jusqu'à nouvel ordre.

Hospitalisation		
noPatient	pHospitalisation	pTransaction
P1	[j21;j28]	[j30:ufn]

Les référentiels de temps

Le temps de validation

Un temps de validation (*valid time*) fait référence à l'horloge de l'agent qui a constaté le fait représenté par la proposition. Une période de validation commence à l'instant où l'agent considère le fait comme avéré et le termine lorsqu'il ne le considère plus comme avéré. Sa valeur est donc fournie par l'agent et modifiable par celui-ci. Il peut en outre y avoir plusieurs référentiels de validation dans un contexte où les agents ne partagent pas une même horloge.

Exemple : Le fait que Tom ait été hospitalisé du 21 au 28 mars a été considéré vrai du jour 29 au jour 31 moment auquel l'agent s'est aperçu qu'il y avait eu erreur de saisie et que Tom avait plutôt séjourné du jour 21 au jour 23, ce qui est considéré vrai depuis le jour 32.

Hospitalisation		
noPatient	pHospitalisation	pValidation
P1	[j21;j28]	[j29;j31]
P1	[j21;j23]	[j32:ufn]

Les référentiels de temps

Bitemporalité = validation + transaction

Il est ensuite possible de consigner la période de validation et la période de transaction, c'est ce qui est communément appelé la bitemporalité.

Exemple : En supposant que la première proposition ait été consignée au jour 30 et la seconde au jour 34, nous obtenons la relation suivante :

Hospitalisation			
noPatient	pHospitalisation	pValidation	pTransaction
P1	[j21:j28]	[j21:j31]	[j30:j33]
P1	[j21:j23]	[j32:ufn]	[j34:ufn]

