

BASE DE DONNÉES TEMPORALISÉES

TEMPORALITÉ

Opérateurs relationnels temporels

PACK, UNPACK et USING

(fold, unfold et normalize)

BD203

v110d

2021-02-09

Christina KHNAISSER et Luc LAVOIE
Département d'informatique
Faculté des sciences



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

PLAN

○ Rappels

- Prédicats et propositions
- Type intervalle

○ Problématique d'unicité temporelle

- Redondance
- Circonlocution
- Contradiction

○ Opérateurs relationnels

- EXPAND et COLLAPSE
- PACK (fold) et UNPACK (unfold)
- USING (normalize)



RAPPELS

PRÉDICAT ET PROPOSITION

○ Prédicat (relation)

- *Toute relation représente un prédicat fermé.*
- Exemple
Le patient ***idPatient*** habite la ville appelée ***nomVille***
- Remarque
La relation détermine la ville d'habitation de tous les patients pour lesquels elle est connue.

○ Proposition (tuple)

- *Toutes les propositions dans la BD sont vraies.*
- Exemple
Le patient ***P258517*** habite la ville de ***Coaticook***
- Remarque
Si la BD est normalisée, aucune autre proposition ne représente le même fait.

RAPPELS

EXEMPLE DE PRÉDICAT

- Le patient *idPatient* (dont le numéro assurance maladie est *NAM*, le nom est *nom*, le prénom est *prenom*, né le *dateNaissance* habitant dans la ville *ville* et dont le nom et le prénom de la mère sont *nom_mere* et *prenom_mere*) possède un dossier à l'hôpital durant la période *periode*.
- Cette modélisation est fautive... nous verrons bientôt pourquoi!

Patient
<u>idPatient</u>
NAM
nom
prenom
nom_mere
prenom_mere
dateNaissance
ville
<u>periode</u>

RAPPELS

TYPE INTERVALLE

- Points : un type ordonné, c'est-à-dire :
 - un ensemble fini
 - de valeurs discrètes
 - muni d'un ordre total.
- Intervalle :
 - ensemble fini non vide de points,
 - noté **[d:f]** avec $\mathbf{d} \leq \mathbf{f}$.
- Un intervalle est unitaire lorsque $|\mathbf{[d:f]}| = 1$
(donc $\mathbf{d} = \mathbf{f}$)

RAPPELS

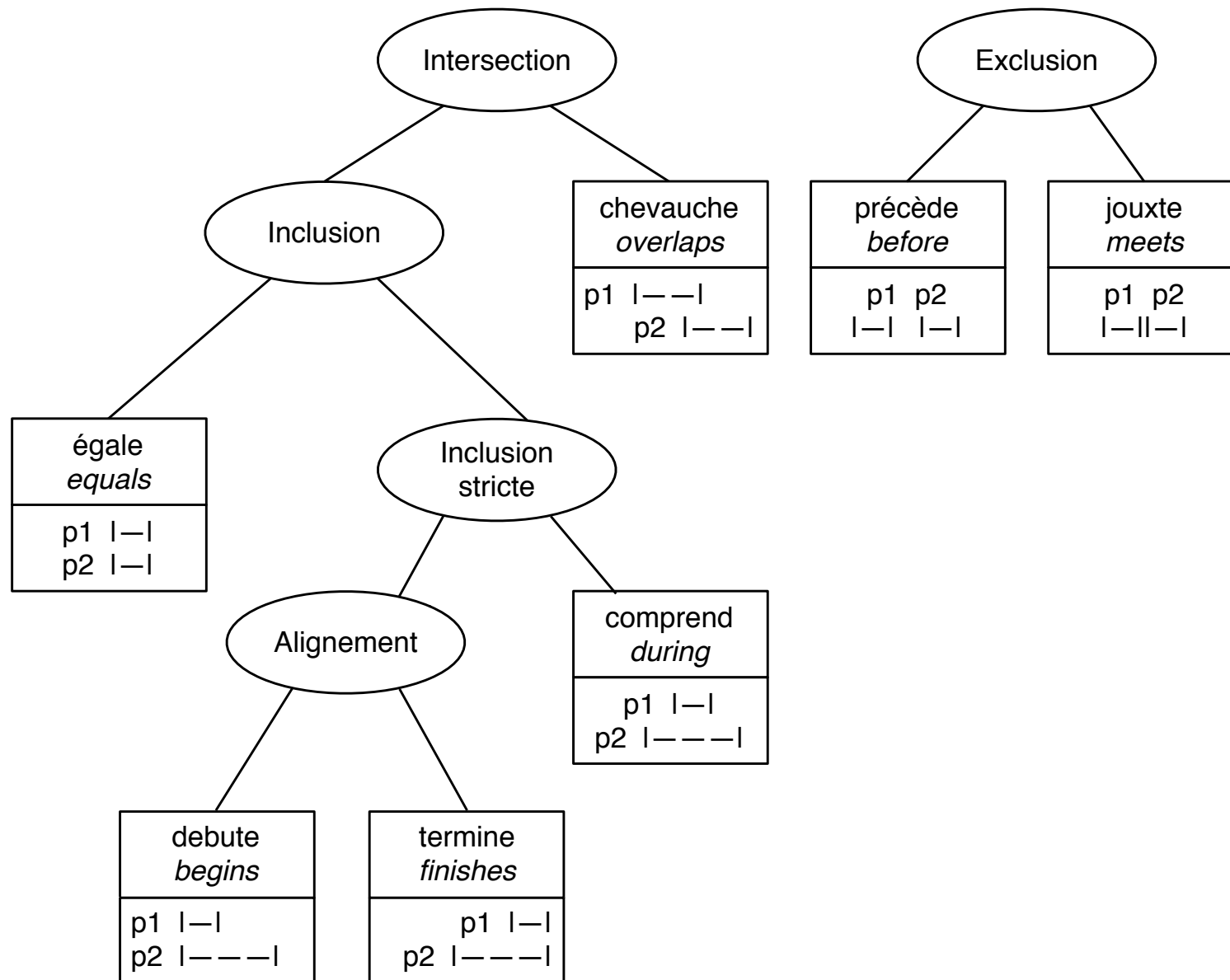
EXEMPLES D'INTERVALLE

- Exemple de type d'intervalles d'entiers :
 - INTERVAL [INTEGER]
- Exemples de types d'intervalles temporels :
 - INTERVAL[DATE]
 - INTERVAL[TIMESTAMP]
- Exemple de valeurs d'intervalles temporels :
 - INTERVAL[d01:d01]
 - INTERVAL[d06:d13]
 - INTERVAL[d20:d99]

Aux fins de la présentation, par souci de simplicité, les dates sont notées « d_i », avec $01 \leq i \leq 99$.

RAPPELS

OPÉRATEURS SUR LES INTERVALLES (DITS DE ALLEN)



RAPPELS

PROBLÉMATIQUE D'UNICITÉ TEMPORELLE

- La proposition :
 - le patient *idPatient* possède un dossier à l'hôpital durant la période *periode* ***inclusivement***.

VillePatient_periodique
<u>idPatient</u>
ville
<u>periode</u>

idPatient	ville	periode
P1	Montréal	[d02:d04]
P1	Montréal	[d03:d05]
P2	Québec	[d02:d03]
P2	Québec	[d04:d06]
P3	Ottawa	[d04:d06]
P3	Sherbrooke	[d03:d08]



RAPPELS

PROBLÉMATIQUE D'UNICITÉ TEMPORELLE

- Redondance
 - Deux tuples contiennent des mêmes données et que leurs périodes de validité se chevauchent (voir P1).
- Circonlocution
 - Deux tuples contiennent les mêmes données et leurs périodes de validité se jouxtent (voir P2).
- Contradiction
 - Deux tuples contiennent des données différentes pour la même période de validité (voir P3).

idPatient	ville	periode
P1	Montréal	[d02:d04]
P1	Montréal	[d03:d05]
P2	Québec	[d02:d03]
P2	Québec	[d04:d06]
P3	Ottawa	[d04:d06]
P3	Sherbrooke	[d03:d08]

OPÉRATEURS

- EXPAND et COLLAPSE
- PACK et UNPACK
- USING

OPÉRATEURS RELATIONNELS

EXPAND ET COLLAPSE

- Prend en entrée un ensemble d'intervalles.
- Retourne un ensemble d'intervalles.

- Exemple

$x : \text{INTERVAL}[\text{DATE}]$

$x := \text{EXPAND}(x)$

$x := \text{COLLAPSE}(x)$

OPÉRATEURS RELATIONNELS

EXPAND ET COLLAPSE

<u>periode</u>
[d02:d04]
[d03:d05]

- **EXPAND** : transforme un ensemble d'intervalles en un ensemble d'intervalles unitaires.

<u>periode</u>
[d02:d02]
[d03:d03]
[d04:d04]
[d05:d05]

- **COLLAPSE** : groupe les intervalles qui se chevauchent ou qui se jouxtent.

<u>periode</u>
[d02:d05]

OPÉRATEURS RELATIONNELS

PACK ET UNPACK

- Opérateurs de relation
 - entrée : une relation et un attribut de type intervalle.
 - sortie : une relation.
- Syntaxe
 - $s := \text{PACK } r \text{ ON } (x)$
 - $t := \text{UNPACK } r \text{ ON } (x)$
- Exemple
 - $\text{patient1} := \text{PACK patient ON (periode)}$
 - $\text{patient2} := \text{UNPACK patient ON (periode)}$

OPÉRATEURS RELATIONNELS

UNPACK ET PACK

<u>idPatient</u>	<u>ville</u>	<u>periode</u>
P1	Montréal	[d02:d04]
P1	Montréal	[d03:d05]
P2	Québec	[d02:d05]
P2	Québec	[d04:d06]
P3	Ottawa	[d04:d06]
P3	Sherbrooke	[d03:d08]

- **UNPACK** : décompose les tuples en regard d'intervalles unitaires (uniquement)

- UNPACK
r WHERE idPatient = P1
ON (periode)

<u>idPatient</u>	<u>ville</u>	<u>periode</u>
P1	Montréal	[d02:d02]
P1	Montréal	[d03:d03]
P1	Montréal	[d04:d04]
P1	Montréal	[d05:d05]

- **PACK** : groupe les tuples ayant des périodes qui se chevauchent ou se jouxtent :

- PACK r ON (periode)

<u>idPatient</u>	<u>ville</u>	<u>periode</u>
P1	Montréal	[d02:d05]
P2	Québec	[d02:d06]
P3	Ottawa	[d04:d06]
P3	Sherbrooke	[d03:d08]

OPÉRATEURS RELATIONNELS

UNPACK ET PACK (DÉFINITION FORMELLE)

○ UNPACK r ON $(a) \equiv$

WITH (

$r1 := r$ GROUP $\{a\}$ AS x ,

$r2 :=$ EXTEND $r1 : \{x := \text{EXPAND}(x)\}$

):

$r2$ UNGROUP x ;

○ PACK r ON $(a) \equiv$

WITH (

$r1 := r$ GROUP $\{a\}$ AS x ,

$r2 :=$ EXTEND $r1 : \{x := \text{COLLAPSE}(x)\}$

):

$r2$ UNGROUP x ;

OPÉRATEURS RELATIONNELS

EXEMPLE - UNPACK R ON (PERIODE)

relation d'origine r

idPatient	ville	periode
P1	Montreal	[d02:d04]
P1	Montréal	[d03:d05]
P2	Québec	[d02:d05]
P2	Québec	[d04:d06]
P3	Ottawa	[d04:d06]
P3	Sherbrooke	[d03:d08]

r1 := r group {periode} as px

idPatient	ville	px
P1	Montreal	[d02:d04]
		[d03:d05]
P2	Québec	[d02:d05]
		[d04:d06]
P3	Ottawa	[d04:d06]
P3	Sherbrooke	[d03:d08]

r2 := EXTEND r1 : {px := EXPAND(px)}

idPatient	ville	px
P1	Montreal	[d02:d02]
		[d03:d03]
		[d04:d04]
		[d05:d05]
P2	Québec	[d02:d02]
		[d03:d03]
		[d04:d04]
		[d05:d05]
P3	Ottawa	[d04:d04]
		[d05:d05]
P3	Sherbrooke	[d06:d06]
		[d03:d03]
		[d04:d04]
		[d05:d05]
P3	Sherbrooke	[d06:d06]
		[d07:d07]
P3	Sherbrooke	[d08:d08]

r := r2 UNGROUP px

idPatient	ville	periode
P1	Montreal	[d02:d02]
P1	Montreal	[d03:d03]
P1	Montreal	[d04:d04]
P1	Montreal	[d05:d05]
P2	Québec	[d02:d02]
P2	Québec	[d03:d03]
P2	Québec	[d04:d04]
P2	Québec	[d05:d05]
P2	Québec	[d06:d06]
P3	Ottawa	[d04:d04]
P3	Ottawa	[d05:d05]
P3	Ottawa	[d06:d06]
P3	Sherbrooke	[d03:d03]
P3	Sherbrooke	[d04:d04]
P3	Sherbrooke	[d05:d05]
P3	Sherbrooke	[d06:d06]
P3	Sherbrooke	[d07:d07]
P3	Sherbrooke	[d08:d08]

OPÉRATEURS RELATIONNELS

EXERCICE - PACK R ON (PERIODE)

- Exercice !

- Résultat

idPatient	ville	periode
P1	Montreal	[d02:d05]
P2	Québec	[d02:d06]
P3	Ottawa	[d04:d06]
P3	Sherbrooke	[d03:d08]

- On remarque que les problèmes de redondance et de circonlocution sont résolus, mais pas celui de contradiction.

OPÉRATEURS RELATIONNELS USING

○ Objectifs

- Faciliter la manipulation et l'interrogation des données temporalisées.
- Étendre la définition des opérateurs relationnels de façon à traiter les attributs de type intervalles.

○ Idée générale

- USING x : r1 <op> r2 ≡
 PACK
 UNPACK r1 ON (x) <op> UNPACK r2 ON (x)
 ON (x)

○ Notes

- Ceci s'étend aussi (naturellement) aux clés...
- Pourquoi ?
- Comment ?

OPÉRATEURS RELATIONNELS

EXEMPLE DE REQUÊTE

- Trouver tous les patients qui ont habité dans la même ville en même temps et déterminer leur période de cohabitation.

R

noPatient	ville	@d
P1	Montréal	[d02:d05]
P2	Québec	[d02:d06]
P3	Montréal	[d09:d11]
P4	Montréal	[d02:d03]
P5	Québec	[d03:d07]
P6	Montréal	[d04:d06]



Patient1	Patient2	ville	@d
P1	P4	Montréal	[d02:d03]
P1	P6	Montréal	[d04:d05]
P2	P5	Québec	[d03:d06]

WITH (

t1 := R RENAME {noPatient as Patient1},

t2 := R RENAME {noPatient as Patient2},

t3 := USING(@d) : (t1 JOIN t2) WHERE (Patient1 < Patient2)

) : t3 PROJECT {Patient1, Patient2, ville, @d}

- où, formulé de façon « compacte »,

$(\text{USING}(@d) : (t \bowtie (t \rho \text{noPatient: noP2})) \sigma (\text{noPatient} < \text{noP2}))$
 $\pi \{\text{noPatient}, \text{noP2}, \text{ville}, @d\}$

1

t1 := R RENAME {noPatient as Patient1}

Patient1	ville	@d
P1	Montréal	[d02:d05]
P2	Québec	[d02:d06]
P3	Montréal	[d09:d11]
P4	Montréal	[d02:d03]
P5	Québec	[d03:d07]
P6	Montréal	[d04:d06]

t1' := UNPACK t1 ON (@d)

Patient1	ville	@d
P1	Montréal	[d02:d02]
P1	Montréal	[d03:d03]
P1	Montréal	[d04:d04]
P1	Montréal	[d05:d05]
P2	Québec	[d02:d02]
P2	Québec	[d03:d03]
P2	Québec	[d04:d04]
P2	Québec	[d05:d05]
P2	Québec	[d06:d06]
P3	Montréal	[d09:d09]
P3	Montréal	[d10:d10]
P3	Montréal	[d11:d11]
P4	Montréal	[d02:d02]
P4	Montréal	[d03:d03]
P5	Québec	[d03:d03]
P5	Québec	[d04:d04]
P5	Québec	[d05:d05]
P5	Québec	[d06:d06]
P5	Québec	[d07:d07]
P6	Montréal	[d04:d04]
P6	Montréal	[d05:d05]
P6	Montréal	[d06:d06]

t2' := UNPACK t2 ON (@d)

Patient2	ville	@d
P1	Montréal	[d02:d02]
P1	Montréal	[d03:d03]
P1	Montréal	[d04:d04]
P1	Montréal	[d05:d05]
P2	Québec	[d02:d02]
P2	Québec	[d03:d03]
P2	Québec	[d04:d04]
P2	Québec	[d05:d05]
P2	Québec	[d06:d06]
P3	Montréal	[d09:d09]
P3	Montréal	[d10:d10]
P3	Montréal	[d11:d11]
P4	Montréal	[d02:d02]
P4	Montréal	[d03:d03]
P5	Québec	[d03:d03]
P5	Québec	[d04:d04]
P5	Québec	[d05:d05]
P5	Québec	[d06:d06]
P5	Québec	[d07:d07]
P6	Montréal	[d04:d04]
P6	Montréal	[d05:d05]
P6	Montréal	[d06:d06]

2

t2 := R RENAME {noPatient as Patient1}

Patient2	ville	periode
P1	Montréal	[d02:d05]
P2	Québec	[d02:d06]
P3	Montréal	[d09:d11]
P4	Montréal	[d02:d03]
P5	Québec	[d03:d07]
P6	Montréal	[d04:d06]

(t1' JOIN t2') WHERE Patient1 < Patient2

Patient1	Patient2	ville	@d
P1	P4	Montréal	[d02:d02]
P1	P4	Montréal	[d03:d03]
P1	P6	Montréal	[d04:d04]
P1	P6	Montréal	[d05:d05]
P2	P5	Québec	[d03:d03]
P2	P5	Québec	[d04:d04]
P2	P5	Québec	[d05:d05]
P2	P5	Québec	[d06:d06]

PACK

Patient1	Patient2	ville	@d
P1	P4	Montréal	[d02:d03]
P1	P6	Montréal	[d04:d05]
P2	P5	Québec	[d03:d06]

CONCEPTS DE BASE

LA VRAIE NATURE DE PACK ET UNPACK

- UNPACK permet d'utiliser les opérateurs relationnels conventionnels.
- PACK permet de maintenir l'intégrité des données et retourner des informations cohérentes.
- Ces opérations peuvent être généralisées en regard de plusieurs attributs.
- On remarque que UNPACK est commutatif et PACK ne l'est pas.

CONCEPTS DE BASE

LA FORME ABRÉGÉE USING

- USING est un raccourci pratique pour effectuer les opérateurs relationnels de base :
 - projection, restriction, jointure,
 - union, différence,
 - renommage.
- Conséquemment, tous les opérateurs composés aussi :
 - intersection, semi-jointure, semi-différence,
 - extension, synthèse...

CONCEPTS DE BASE

USING APPLIQUÉ AUX OPÉRATEURS RELATIONNELS

- USING (listeA) : r PROJECT {ai, ..., an} \equiv
PACK((UNPACK r ON (listeA) PROJECT {ai, ..., an})
ON (listeA))
- USING (listeA) : r WHERE condition \equiv
PACK((UNPACK r ON (listeA)) WHERE condition)
ON (listeA))
- USING (listeA) : r1 JOIN r2 \equiv
PACK ((UNPACK r1 ON (listeA)) JOIN (UNPACK r2 ON (listeA)))
ON (listeA))

CONCEPTS DE BASE

USING APPLIQUÉ AUX OPÉRATEURS ENSEMBLISTES

- USING (listeA) : r1 UNION r2 \equiv
PACK((UNPACK r1 ON (listeA)) UNION (UNPACK r2 ON (listeA)))
ON (listeA)
- USING (listeA) : r1 INTERSECT r2 \equiv
PACK((UNPACK r1 ON (listeA)) INTERSECT (UNPACK r2 ON (listeA)))
ON (listeA)
- USING (listeA) : r1 EXCEPT r2 \equiv
PACK ((UNPACK r1 ON (listeA)) EXCEPT (UNPACK r2 ON (listeA)))
ON (listeA)

CONCEPTS DE BASE

USING APPLIQUÉ AUX OPÉRATEURS COMPOSÉS

- USING (listeA) : EXTEND r : {ai, ..., an} \equiv
PACK((UNPACK r ON (listeA) EXTEND {ai, ..., an})
ON (listeA)
- ...

CONCEPTS DE BASE

USING APPLIQUÉ AUX AFFECTATIONS

- USING (listeA) : INSERT t IN r \equiv
USING (listeA) : r1 UNION RELATION[t]
- USING (listeA) : DELETE r WHERE condition \equiv
r := USING (listeA) : r WHERE NOT (condition)
- USING (listeA) : DELETE t IN r \equiv
r := USING (listeA) : r EXCEPT RELATION[t]

CONCEPTS DE BASE

USING APPLIQUÉ À LA MODIFICATION

- USING (listeA) : UPDATE r
WHERE condition { listeAffectations } ≡

WITH (
 t1 := UNPACK R ON (listeA),
 t2 := t1 WHERE NOT (condition),
 t3 := t1 EXCEPT t2,
 t4 := EXTEND t3 : { listeAffectations },
 t5 := t2 UNION t4) :
r := PACK t5 ON (listeA)

CONCEPTS DE BASE

USING APPLIQUÉ AUX CLÉS

- $r \text{ USING } (\text{listeA}) : \text{KEY } \{k\} \equiv$
 $(\text{UNPACK } r \text{ ON } (\text{listeA})) \text{ KEY } (\{k\} \cup \{\text{listeA}\})$
- $r1 \text{ USING } (\text{listeA}) : \text{FOREIGN KEY } \{k\} \text{ REFERENCES } r2 \equiv$
 $(\text{UNPACK } r1 \text{ ON } (\text{listeA}))$
 $\text{FOREIGN KEY } (\{k\} \cup \{\text{listeA}\})$
 $\text{REFERENCES } (\text{UNPACK } r2 \text{ ON } (\text{listeA}))$