Année 2016-17
Responsable : Nicolas Lumineau

U.E. BDW1
Bases de Données et programmation WEB

Partie "Algèbre relationnelle"

Fascicule sur l'Algèbre Relationnelle

Fascicule : Algèbre relationnelle

Dans ce fascicule, vous trouverez les rappels sur la syntaxe de l'algèbre relationnelle, ainsi que des exercices corrigés.

Pourquoi ? Comment ?

Pourquoi s'intéresser à l'Algèbre Relationnelle ?
L'**Algèbre Relationnelle** est le langage qui **permet d'exprimer la façon dont une requête va s'exécuter pour obtenir le résultat**. Jusqu'à présent vous avez vu le langage SQL qui est un langage déclaratif qui permet de caractériser le résultat attendu sans pour autant dire comment l'obtenir. L'**Algèbre Relationnelle** est un langage procédural qui permet de décrire l'enchainement des opérateurs qui permettront d'obtenir un résultat.

Pourquoi les SGBD utilise l'Algèbre Relationnelle ?
L'**Algèbre Relationnelle** est principalement utilisé pour faire de l'optimisation de requête. En effet, pour une requête SQL, il existe plusieurs requêtes algébriques équivalentes (*i.e.* plusieurs requêtes en algèbre relationnelle qui retournent les mêmes tuples en résultat). Ces requêtes algébriques sont alors appelés **plans d'exécution**. Autant les tuples retournés sont les mêmes pour tous ces plans d'exécution, autant le temps nécessaire pour obtenir ces tuples, lui peut être très différent. Tout dépend de la façon dont les opérations qui composent le plan d'exécution sont organisés.

Comment exprimer une requête en Algèbre Relationnelle ?
Vous trouverez ci-dessous les différents opérateurs de l'**Algèbre Relationnelle** . Pour illustrer les différents opérateurs nous considèrons les schémas de relations suivants R(<u>A</u>,B,C,#D), S(<u>G</u>,H,I,J), U(<u>D</u>, E, F), V(<u>K</u>,L,<u>M</u>) et W(<u>M</u>) où les clés primaires sont soulignées et la clé étrangère est préfixée par un #.

R

A	B	C	D
1	2	3	4
2	3	1	4
3	4	1	2
4	3	1	2

S

E	F	G	H
1	2	3	4
2	4	1	3
3	4	1	2
4	2	1	3

U

D	I	J
2	3	4
3	1	2
4	2	3

V

K	L	M
1	2	1
1	2	2
2	1	1
2	1	2
2	1	3
3	2	1

W

M
1
2
3

On distingue les opérateurs de base :

— l'union $R \cup S$

— la différence $R - S$

— le produit cartésien $R \times S$

— la projection $\Pi_{A,B}(\mathrm{R})$

— la sélection $\sigma_F(\mathrm{R})$

— le renommage $\rho_{A/A',B/B'}(\mathrm{R})$

et les opérateurs considérés comme des racoucis syntaxiques :

— l'intersection $R \cap S$ qui peut s'exprimer comme $R - (R - S)$

— la θ-jointure $R \bowtie_F S$ qui peut s'exprimer comme $\sigma_F(\mathrm{R} \times \mathrm{S})$

— la semi-jointure $R \ltimes_F S$ qui peut s'exprimer comme $\Pi_{A,B,C,D}(\sigma_F(\mathrm{R} \times \mathrm{S}))$

— la division $V \div W$ qui peut s'exprimer comme $\Pi_{K,L}(\mathrm{V})$ - $\Pi_{K,L}((\Pi_{K,L}(\mathrm{V}) \times \mathrm{W}) - \mathrm{V})$

L'UNION : $R \cup S$

Condition: R et S doivent être compatibles à l'union (même nombre d'attributs, même type d'attributs).

Ce qui est retourné: l'ensemble des tuples qui appatiennent à R ou qui appartiennent à S.

Schéma du résultat : T(A,B,C,D)

Instance du résultat :
T

A	B	C	D
1	2	3	4
2	3	1	4
3	4	1	2
4	3	1	2
1	2	3	4
2	4	1	3
3	4	1	2
4	2	1	3

La DIFFERENCE : $R - S$

Condition: R et S doivent être compatibles à l'union (même nombre d'attributs, même type d'attributs).

Ce qui est retourné: l'ensemble des tuples qui appatiennent à R mais qui n'appartiennent pas à S.

Schéma du résultat : T(A,B,C,D)

Instance du résultat :
T

A	B	C	D
2	3	1	4
4	3	1	2

La PROJECTION : $\Pi_{listeAttributs}(\mathbf{R})$

Condition: Les attributs en paramètre de la projection (*i.e.* listeAttributs) sont bien des attributs présents dans la relation interrogée (*i.e.* R).

Ce qui est retourné: l'ensemble des tuples composés uniquement des attributs présents en paramètre de la projection. Les doublons des tuples sont automatiquement supprimés.

Schéma du résultat : T(listeAttributs)

Instance du résultat :
Si on considère $\Pi_{B,C}(\mathrm{R})$:
(*i.e.* listeAttributs : {B,C})
alors on aura :
Schéma du résultat : T(B,C)
T

B	C
2	3
3	1
4	1

La SELECTION : $\sigma_F(R)$

Condition: Les attributs intervenant dans la formule logique F sont bien des attributs présents dans la relation interrogée (R).

Ce qui est retourné: l'ensemble des tuples de R qui rendent vraie la formule logique F.

Schéma du résultat : T(A,B,C,D)

Instance du résultat :
Si on considère :
F = A > 2 ,
alors on aura :
T

A	B	C	D
3	4	1	2
4	3	1	2

Si on considère :
F = B > 3 ∨ C = 3,
alors on aura :
T

A	B	C	D
1	2	3	4
3	4	1	2

Si on considère :
F = B > 3 ∧ D < 4,
alors on aura :
T

A	B	C	D
3	4	1	2

Si on considère :
F = B > 3 ∧ D != 2,
alors on aura :
T

A	B	C	D

Le résultat est vide, mais cela reste une relation.

Le RENOMMAGE : $\rho_{mappings}$**(R)**

Condition: Les attributs intervenant dans les mappings sont bien des attributs présents dans la relation interrogée (R).

Ce qui est retourné: l'ensemble des tuples de R avec un schéma correspondant aux transformations définies dans les *mappings*.

Instance du résultat :
Si on considère $\rho_{A/A',B/B'}$(R) :
(*i.e.* mappings : A/A',B/B')
alors on aura :
Schéma du résultat : T(A',B',C,D)
T

A'	B'	C	D
1	2	3	4
2	3	1	4
3	4	1	2
4	3	1	2

Le PRODUIT CARTESIEN : $U \times W$

Condition: U et W doivent avoir des noms d'attributs différents (si ce n'est pas le cas, un renommage pourra être effectué au préalable).

Ce qui est retourné: l'ensemble des tuples composé des attributs réunis des relations U et W et où chaque tuple de U a été associé à chaque tuple de W.

Schéma du résultat : T(D,I,J,M)

Instance du résultat :
T

D	I	J	M
2	3	4	1
2	3	4	2
2	3	4	3
3	1	2	1
3	1	2	2
3	1	2	3
4	2	3	1
4	2	3	2
4	2	3	3

L'INTERSECTION : $R \cap S$

Condition: R et S doivent être compatibles à l'union (même nombre d'attributs, même type d'attributs).

Ce qui est retourné: l'ensemble des tuples qui appatiennent à R et qui appartiennent également à S.

Schéma du résultat : T(A,B,C,D)

Instance du résultat :
T

A	B	C	D
1	2	3	4
3	4	1	2

La θ-JOINTURE : $R \rhd\lhd_F S$

Condition: La formule logique F en paramètre de la jointure ne doit contenir que des comparaisons (*i.e.* $=, !=, <, >, \leq, \geq$) entre des attributs de R et de S.

Ce qui est retourné: l'ensemble des tuples composés des attributs réunis de R et de S et pour lesquels l'association des tuples de R et de S rendent vraie la formule logique F.

Schéma du résultat : T(A,B,C,D,F,G,H)
car dans le cas d'équi-jointure, on ne répète pas l'attribut.

Instance du résultat :
Si on considère :
F : A = E ,
alors on aura :
T

A	B	C	D	F	G	H
1	2	3	4	2	3	4
2	3	1	4	4	1	3
3	4	1	2	4	1	2
4	3	1	2	2	1	3

Si on considère :
F : C > F,
alors on aura :
T

A	B	C	D	E	F	G	H
1	2	3	4	1	2	3	4
1	2	3	4	4	2	1	3

Si on considère :
F : C > F $\wedge$ D != H,
alors on aura :
T

A	B	C	D	E	F	G	H
1	2	3	4	4	2	1	3

Remarque : On parle d'ÉQUI-JOINTURE quand la formule F, ne contient que des égalités.

La SEMI-JOINTURE : $R \ltimes_F S$

Condition: La formule logique F en paramètre de la jointure ne doit contenir que des comparaisons (*i.e.* $=, !=, <, >, \leq, \geq$) entre des attributs de R et de S.

Ce qui est retourné: l'ensemble des tuples composés uniquement des attributs de R et pour lesquels l'association des tuples de R et de S rendent vraie la formule logique F.

Schéma du résultat : T(A,B,C,D)

Instance du résultat :
Si on considère :
F : A = E ,
alors on aura :
T

A	B	C	D
1	2	3	4
2	3	1	4
3	4	1	2
4	3	1	2

Si on considère :
F : C > F,
alors on aura :
T

A	B	C	D
1	2	3	4

Si on considère :
F : C > F $\wedge$ D != H,
alors on aura :
T

A	B	C	D
1	2	3	4

La DIVISION : $V \div W$

Condition: Le schéma de V est une sur-ensemble du schéma de W (*i.e.* tous les attributs de W sont également des attributs de V). Ici, on a bien M qui est l'unique attribut de W et qui appartient bien au schéma de V.

Ce qui est retourné: l'ensemble des tuples composés des attributs de V qui ne sont pas dans le schéma de W et dont le produit cartésien avec W est un sous-ensemble de tuples de V (*i.e.* ce produit cartésien ne génèrerait pas de tuples qui ne sont pas dans V).

Schéma du résultat : T(K,L)

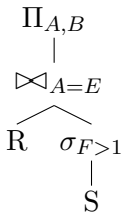
Instance du résultat :
T

K	L
2	1

On remarque que T $\times$ W $\subseteq$ U. On remarque également que si on avait considéré le tuple (1,2) dans le résultat, le produit cartésien nous aurait fait apparaitre le tuple (1,2,3) qui n'est pas dans V. De même, si on avait considéré le tuple (3,2) dans le résultat, le produit cartésien nous aurait fait apparaitre les tuples (3,2,2) et (3,2,3) qui ne sont pas dans V.

Comment représenter une requête en Algèbre Relationnelle ?
Une requête en **Algèbre Relationnelle** peut se représenter de deux manières différentes :

— Sous forme d'expression algébrique : $\Pi_{A,B}(R \ \rhd\lhd_{A=E} \sigma_{F>1}(S))$
— Sous forme d'arbre algébrique :



Exercices applicatifs

Exercice 1 : Quelques requêtes algébriques

Soient les schémas de la relation JOUEUR et PALMARES suivant :

JOUEUR(Nom, Prenom, AnNaiss, Nation, Taille, Poids)
PALMARES(Nom, Annee, Titre)

On suppose que : *AnNaiss* représente l'année de naissance d'un joueur, qu'il n'y a pas d'homonymes dans la base de données, que la taille est exprimée en centimètre et que le poids est exprimé en kilogramme.

Ecrire en algèbre relationnelle, quand cela est possible, les requêtes suivantes :

1. Donner les joueurs nés avant 1987 (inclus) et faisant plus de 78kg (strict).
2. Donner les nationalités présentes dans la table.
3. Donner le nom des joueurs de plus 1m80. Le résultat aura pour nom "NomDesGrands".
4. Donner le nom et le prénom des joueurs ayant eu un palmarès en 2014.
5. Donner le nom des joueurs sans palmarès.
6. Donner le nom des joueurs titré en 2010 et 2014.
7. Donner le nom des joueurs titré tous les ans (i.e. pour toutes les années stockées dans la base)
8. Donner le nom du/des joueurs le(s) plus jeune(s) stocké(s) dans la base.
9. Donner le nombre de titre obtenu par le joueur le plus titré.

Exercice 2 : Trouvez les erreurs

On considère les relations JOUEUR et PALMARES de l'exercice précédent. Expliquer pourquoi les requêtes suivantes ne sont pas correctes ou ne correspondent pas à ce qui est attendu. Pour chaque requête algébrique, proposer une correction afin que l'énoncé proposé soit en adéquation avec la requête.

1. $\Pi_{Nom,Titre}(JOUEUR)$:
   retourne les titres obtenus pour chaque joueur.

2. $\Pi_{Nom}(JOUEUR \cup PALMARES)$:
   retourne l'ensemble des noms de joueurs apparaissant dans JOUEUR et ceux apparaissant dans PALMARES.

3. $\sigma_{AnNais>1983}(\Pi_{Nom}(JOUEUR))$:
   retourne le nom des joueurs nés après 1983.

4. $\Pi_{Nom}(\sigma_{Titre!='ChampiondeFrance'}(PALMARES))$:
   retourne le nom des joueurs n'ayant jamais obtenu le titre de 'Champion de France'.

5. $\Pi_{Nom,Prenom}(\sigma_{Nation='France' \wedge Nation='Argentine'}(JOUEUR))$:
   retourne le nom et le prénom des joueurs français et des joueurs argentins.

6. $\Pi_{Nom}(\sigma_{Titre='ChampiondeFrance' \wedge Titre='CoupedeFrance' \wedge Annee='2014'}(PALMARES))$:
   retourne le nom des joueurs ayant obtenu à la fois le titre de 'Champion de France' et le titre de 'Coupe de France' en 2014.

Corrections des exercices applicatifs

Exercice 1 : Quelques requêtes algébriques

1. $\sigma_{AnNaiss<=1987 \wedge Poids>78}(JOUEUR)$
2. $\Pi_{Nation}(JOUEUR)$
3. $\rho_{Nom/NomDesGrands}(\Pi_{Nom}(\sigma_{Taille>180}(JOUEUR))$
4. $\Pi_{Nom,Prenom}(JOUEUR \ \rhd\lhd_{Nom} \sigma_{Annee=2014}(PALMARES))$
5. $\Pi_{Nom}(JOUEUR) - \Pi_{Nom}(PALMARES)$
6. $\Pi_{Nom}(\sigma_{Annee=2010}(PALMARES)) \cap \Pi_{Nom}(\sigma_{Annee=2014}(PALMARES))$
   NB : si cela est proposé par un étudiant, insister sur le fait que l'expression
   $\Pi_{Nom}(\sigma_{Annee=2010 \wedge Annee=2014}(PALMARES))$ n'a pas de sens car on ne peut pas avoir deux valeurs différentes pour un attribut.
7. $\Pi_{Nom,Annee}(PALMARES) \div \Pi_{Annee}(PALMARES)$
8. $\Pi_{Nom,Prenom}(JOUEUR) - \Pi_{Nom,Prenom}(\rho_{AnNaiss/A1}(JOUEUR) \ltimes_{A1<A2} \rho_{AnNaiss/A2}(JOUEUR))$
   car la semi-jointure ne retourne que les joueurs pour lesquels il existe au moins un autre joueur plus jeune et en faisant la différence, il ne reste plus que les noms et prénoms des joueurs pour lesquels il n'existe pas d'autre joueurs plus jeune : donc ce sont les plus jeunes !
9. Impossible car pas d'opérateur d'agrégation dans cette version de l'algèbre relationnelle.

Exercice 2 : Trouvez les erreurs !

1. Erreur : Titre n'est pas un attribut de JOUEUR.
   $\Pi_{Nom,Titre}(PALMARES)$
2. Erreur : JOUEUR et PALMARES sont incompatible à l'union.
   $\Pi_{Nom}(\Pi_{Nom}(JOUEUR) \cup \Pi_{Nom}(PALMARES))$
3. Erreur : La projection sur le nom est appliquée avant la sélection qui ne peut pas s'appliquer car AnNaiss n'existe plus.
   $\Pi_{Nom}(\sigma_{AnNaiss>1983}(JOUEUR))$
4. La requête proposée retourne le nom des joueurs ayant obtenu au moins une fois un titre différent de celui de 'Champion de France'.
   $\Pi_{Nom}(JOUEUR) - \Pi_{Nom}(\sigma_{Titre='Champion\ de\ France'}(PALMARES))$
5. La requête proposée ne retourne rien : un attribut ne peut pas avoir deux valeurs différentes en même temps. Dans le modèle relationnel est les attributs sont mono-valués.
   $\Pi_{Nom,Prenom}(\sigma_{Nation='France' \vee Nation='Argentine'}(JOUEUR))$
6. idem, pour les mêmes raisons.
   $\Pi_{Nom}(\sigma_{Titre='Champion\ de\ France' \wedge Annee='2014'}(PALMARES))$
   $\cap$
   $\Pi_{Nom}(\sigma_{Titre='Coupe\ de\ France' \wedge Annee='2014'}(PALMARES))$