Module BASES DE DONNEES RELATIONNELLES
TD 1
Algèbre relationnelle

Hervé Owsinski
Herve.Owsinski@uphf.fr

2026-2027

1 RAPPELS

Schémas utilisés

```
ETUDIANT (IdEtudiant, Nom, Prenom)
INSCRIPTION (IdEtudiant, IdCours)
```

Convention utilisée

- les relations/tables en MAJUSCULES : **ETUDIANT INSCRIPTION**
- les attributs/colonnes en CamelCase (1er lettre d'un mot en majuscule, le reste en minuscules).
  IdEtudiant Nom Prenom IdEtudiant IdCours

Clé primaire (soulignée par convention)

Attribut ou tuple d'attributs dont la valeur ne peut apparaître qu'une seule fois sur l'ensemble des lignes de la table. Exemple avec le couple de la relation **INSCRIPTION** :

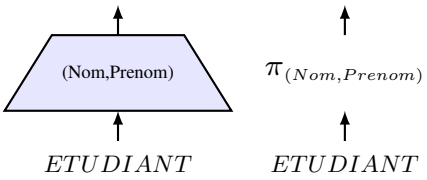
```
IdEtudiant  |  IdCours   |
25          |  9         |   ->  possible
25          |  16        |   ->  possible
25          |  9         |   -> impossible : doublon !
```

Projection : diminuer le nombre de colonnes

En **algèbre relationnelle**, cela correspond à l'opérateur de projection π (pi) :

$$\pi_{(Nom,Prenom)}\ (ETUDIANT) = \pi(ETUDIANT, (Nom, Prenom))$$

L'**arbre algébrique** :



En **SQL** :

```
SELECT nom, pre
FROM Client ;
```

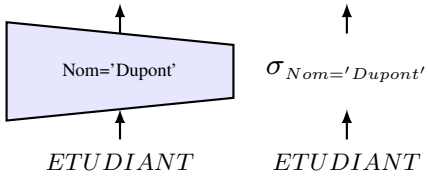
Restriction

Diminuer le nombre de lignes sans toucher au schéma

En **algèbre relationnelle**, cela correspond à l'opérateur de restriction σ (sigma) :

$$\sigma_{Nomp='Dupont'}(ETUDIANT) = \sigma(ETUDIANT/Nom =' Dupont')$$

L'**arbre algébrique** :



En **SQL** :

```sql
SELECT *
FROM ETUDIANT
WHERE Nom = 'Dupont';
```

Jointure : relier deux tables (exemple)

Relation A

id	nom
1	$Alice$
2	Bob

Relation B

id	$ville$
1	$Lille$
2	$Paris$

Jointure $A \bowtie_{A.id=B.id} B$

$A.id$	nom	$B.id$	$ville$
1	$Alice$	1	$Lille$
2	Bob	2	$Paris$

$$A \bowtie_{A.id=B.id} B = \bowtie(A, B\ /A.id = B.id)$$

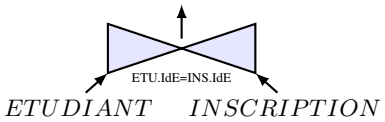
Joins une ligne de A avec une ligne de B lorsque l'id de A de cette ligne correspond au id de B pour l'autre ligne.

Jointure : relier les colonnes de deux tables

En **algèbre relationnelle**, cela correspond à l'opérateur de restriction $\bowtie$:

$$ETUDIANT \bowtie_{ETUDIANT.IdE=INSCRIPTION.IdE} INSCRIPTION$$
$$=$$
$$\bowtie(ETUDIANT, INSCRIPTION/ETUDIANT.IdE = INSCRIPTION.IdE)$$

L'**arbre algébrique** :



En **SQL** :

```sql
SELECT *
FROM ETUDIANT JOIN INSCRIPTION ON ETUDIANT.IdE = INSCRIPTION.IdE;
```

2 Exercice 1 : les bases

Exercice 1 : R1

CJH

IdCours	Jour	Heure
Archi	Lu	9h
Algo	Ma	9h
Algo	Ve	9h
Syst	Ma	14h

CS

IdCours	IdSalle
Archi	S1
Algo	S2
Syst	S1

ENA

IdEtudiant	Nom	Adresse
100	Toto	Nice
200	Tata	Paris
300	Titi	Rome

CEN

IdCours	IdEtudiant	Note
Archi	100	A
Archi	300	A
Syst	100	B
Syst	200	A
Syst	300	B
Algo	100	C
Algo	200	A

A partir des tables ci-dessus, donner les résultats des opérations suivantes :

$$R_1 = \pi\ (\ CJH\ /\ (IdCours)\ )$$

$$R_1 = \pi_{IdCours}\ (\ CJH\ )$$

La relation R_1 est le résultat de la projection de l'attribut `IdCours` obtenu depuis relation CJH.

Opérateur de Projection (PI π) : $\pi(RELATION/(Attributs))$

CJH

IdCours	Jour	Heure
Archi	Lu	9h
Algo	Ma	9h
Algo	Ve	9h
Syst	Ma	14h

Projection

IdCours
Archi
Algo
Algo
Syst

Définition de Relation : ensemble de tuples

IdCours
Archi
Algo
Syst

$$R_1 = \pi\ (\ CJH\ /\ (IdCours)\ )$$

En **algèbre relationnelle**, une relation est un ensemble de tuples.
Les doublons sont donc automatiquement supprimés de la relation R_1.

Par contre, en SQL, les doublons sont possibles !
Sans les doublons en SQL : `SELECT DISTINCT IdCours FROM CJH ;`
Avec les doublons en SQL : `SELECT IdCours FROM CJH ;`

Exercice 1 : R2

CJH

IdCours	Jour	Heure
Archi	Lu	9h
Algo	Ma	9h
Algo	Ve	9h
Syst	Ma	14h

CS

IdCours	IdSalle
Archi	S1
Algo	S2
Syst	S1

ENA

IdEtudiant	Nom	Adresse
100	Toto	Nice
200	Tata	Paris
300	Titi	Rome

CEN

IdCours	IdEtudiant	Note
Archi	100	A
Archi	300	A
Syst	100	B
Syst	200	A
Syst	300	B
Algo	100	C
Algo	200	A

A partir des tables ci-dessus, donner les résultats des opérations suivantes :

$$R_2 = \pi\ (\ ENA\ /\ (IdEtudiant)\ )$$

$$R_2 = \pi_{(IdEtudiant)}\ (\ ENA\ )$$

La relation R_2 est le résultat de la projection de l'attribut `IdEtudiant` obtenu depuis la relation ENS.

Opérateur de Projection (PI π) :
$\pi(RELATION/(Attributs))$ ou $\pi_{(Attributs)}(RELATION)$

ENA

IdEtudiant	Nom	Adresse
100	Toto	Nice
200	Tata	Paris
300	Titi	Rome

Projection

IdEtudiant
100
200
300

$$R_2 = \pi\ (\ ENA\ /\ (IdEtudiant)\ )$$

Encore une fois, on supprimerait les doublons éventuels en algèbre relationnelle.

Exercice 1 : R3

CJH

IdCours	Jour	Heure
Archi	Lu	9h
Algo	Ma	9h
Algo	Ve	9h
Syst	Ma	14h

CS

IdCours	IdSalle
Archi	S1
Algo	S2
Syst	S1

ENA

IdEtudiant	Nom	Adresse
100	Toto	Nice
200	Tata	Paris
300	Titi	Rome

CEN

IdCours	IdEtudiant	Note
Archi	100	A
Archi	300	A
Syst	100	B
Syst	200	A
Syst	300	B
Algo	100	C
Algo	200	A

A partir des tables ci-dessus, donner les résultats des opérations suivantes :

$$R_3 = \sigma\ (\ CEN\ /\ IdCours = \text{'Algo'}\ )$$

$$R_3 = \sigma_{IdCours=\text{'Algo'}}\ (\ CEN\ )$$

La relation R_3 est le résulat d'une restriction qui ne garde que les tuples de `CEN` qui respectent la condition $IdCours =' Algo'$.

Opérateur de Restriction (SIGMA σ) :
$\sigma(RELATION/condition)$ ou $\sigma_{condition}(RELATION)$

CEN

IdCours	IdEtudiant	Note
Archi	100	A
Archi	300	A
Syst	100	B
Syst	200	A
Syst	300	B
Algo	100	C
Algo	200	A

R3

Restriction

IdCours	IdEtudiant	Note
Algo	100	C
Algo	200	A

$$R_3 = \sigma\ (\ CEN\ /\ IdCours = \text{'Algo'}\ )$$

Exercice 1 : R4

CJH

IdCours	Jour	Heure
Archi	Lu	9h
Algo	Ma	9h
Algo	Ve	9h
Syst	Ma	14h

CS

IdCours	IdSalle
Archi	S1
Algo	S2
Syst	S1

ENA

IdEtudiant	Nom	Adresse
100	Toto	Nice
200	Tata	Paris
300	Titi	Rome

CEN

IdCours	IdEtudiant	Note
Archi	100	A
Archi	300	A
Syst	100	B
Syst	200	A
Syst	300	B
Algo	100	C
Algo	200	A

A partir des tables ci-dessus, donner les résultats des opérations suivantes :

$$R_4 = \bowtie(\ CJH, CS\ /\ CJH.IdCours = CS.IdCours\ )$$

$$R_4 = CJH \bowtie_{CJH.IdCours=CS.IdCours} CS$$

La relation R_4 est le résultat de la jointure de `CJH` et `CS` : on crée de nouveaux tuples dont les attributs sont la concaténation de ceux des deux relations et on relie les valeurs en utilisant l'égalité sur les deux clés fournies.

Opérateur de Jointure (Noeud papillon $\bowtie$) :
$\bowtie(RA, RB/RA.a = RB.b)$ ou $RA \bowtie_{RA.a=RB.b} RB$

CJH

IdCours	Jour	Heure
Archi	Lu	9h
Algo	Ma	9h
Algo	Ve	9h
Syst	Ma	14h

CS

IdCours	IdSalle
Archi	S1
Algo	S2
Syst	S1

Nouvelle relation qui récupère les attributs concaténés dans l'ordre d'apparition :

R4

IdCours	Jour	Heure	IdSalle
Archi	Lu	9h	S1
Algo	Ma	9h	S2
Algo	Ve	9h	S2
Syst	Ma	14h	S1

$$R_4 = \bowtie(\ CJH, CS\ /\ CJH.IdCours = CS.IdCours\ )$$

Exercice 1 : R5

CJH

IdCours	Jour	Heure
Archi	Lu	9h
Algo	Ma	9h
Algo	Ve	9h
Syst	Ma	14h

CS

IdCours	IdSalle
Archi	S1
Algo	S2
Syst	S1

ENA

IdEtudiant	Nom	Adresse
100	Toto	Nice
200	Tata	Paris
300	Titi	Rome

CEN

IdCours	IdEtudiant	Note
Archi	100	A
Archi	300	A
Syst	100	B
Syst	200	A
Syst	300	B
Algo	100	C
Algo	200	A

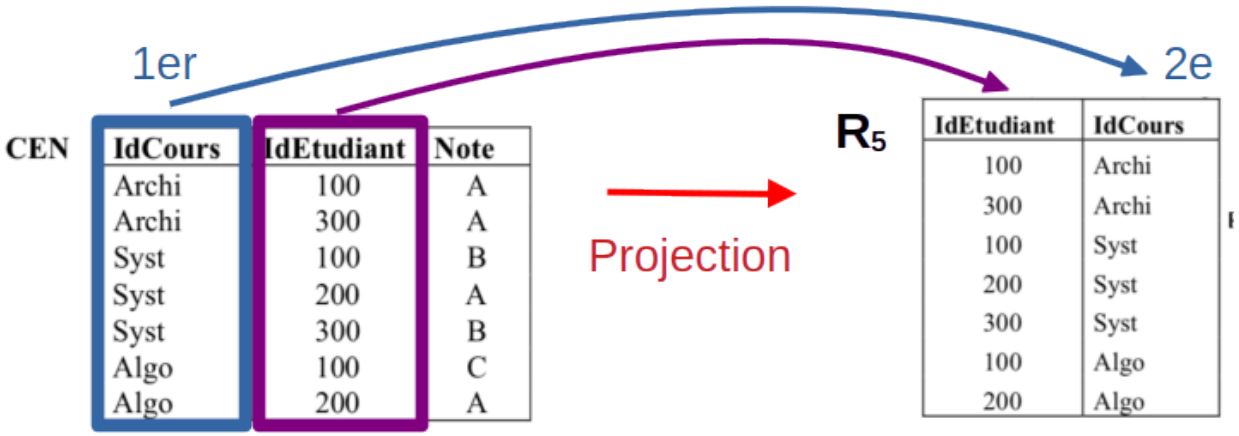
A partir des tables ci-dessus, donner les résultats des opérations suivantes :

$$R_5 = \pi(CEN \ / \ (IdEtudiant, IdCours) \ )$$

$$R_5 = \pi_{(IdEtudiant, IdCours)} \ (CEN \ )$$

La relation R_5 est le résultat de la projection du couple `(IdEtudiant, IdCours)` obtenu depuis la relation CEN.

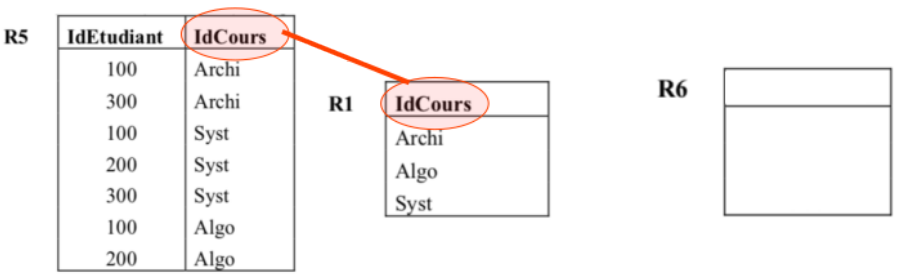
Opérateur de Projection (PI π) :
$\pi(RELATION/(Attributs))$ ou $\pi_{(Attributs)}(RELATION)$



$\pi(RELATION/(attributs))$ ou $\pi_{(Attributs)}(RELATION)$

C'est bien l'ordre dans le tuple fourni pour la projection qui importe : il s'agit bien d'une nouvelle relation.

Exercice 1 : R6



A partir des tables ci-dessus, donner les résultats des opérations suivantes :

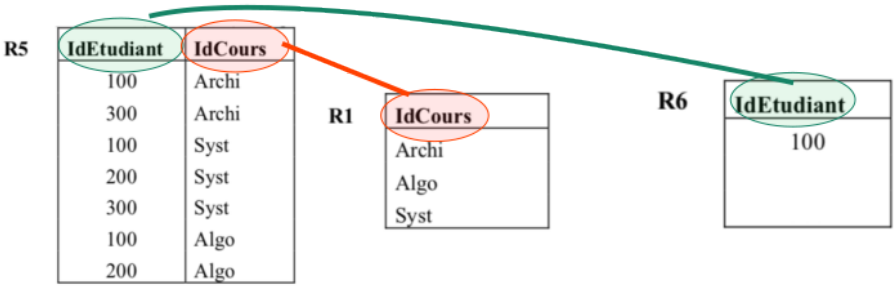
$$R_6 = R_5 \div R_1$$

La relation R_6 est le résultat de la division de la relation R_5 par la relation R_1.

Le schéma de R_1 doit être inclus dans le schéma de R_5 (mais pas égal).

L'association d'un tuple de R_6 avec n'importe quel tuple de R_1
doit donner un tuple présent dans R_5.

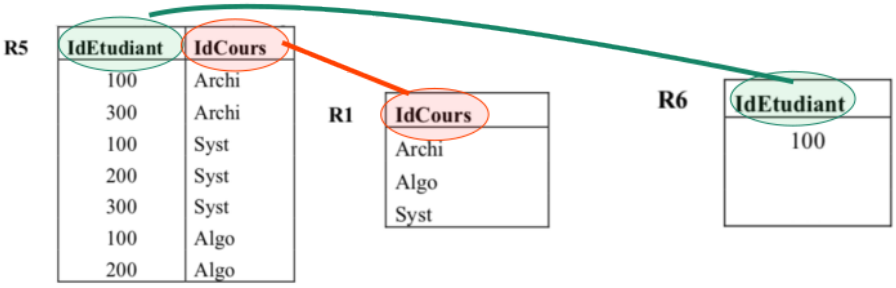
Attention : $R_6 \times R_1 \subseteq R_5$ et pas nécessairement $R_6 \times R_1 = R_5$



Si $(100) \in R_6$ alors on peut reformer :

- $(100, \text{'Archi'})$ qui appartient bien à R_5
- $(100, \text{'Algo'})$ qui appartient bien à R_5
- $(100, \text{'Syst'})$ qui appartient bien à R_5

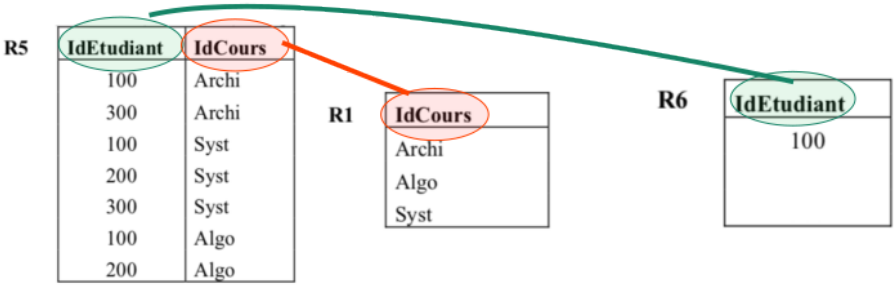
Conclusion : $(100) \in R_6$



Si $(200) \in R_6$ alors on peut reformer :

- $(200, \text{'Archi'})$ qui n'appartient pas à R_5
- ... pas la peine de continuer
- ...

Conclusion : $(200) \notin R_6$

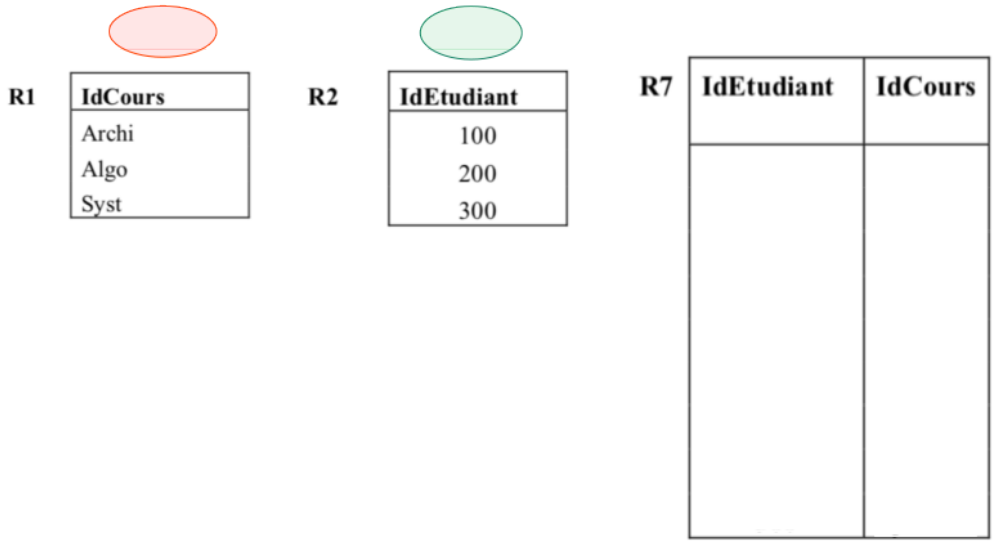


Si $(300) \in R_6$ alors on peut reformer :

- $(300, \text{'Archi'})$ qui appartient bien à R_5
- $(300, \text{'Algo'})$ qui n'appartient pas à R_5
- ... pas la peine de continuer

Conclusion : $(300) \notin R_6$

Exercice 1 : R7



R1

IdCours
Archi
Algo
Syst

R2

IdEtudiant
100
200
300

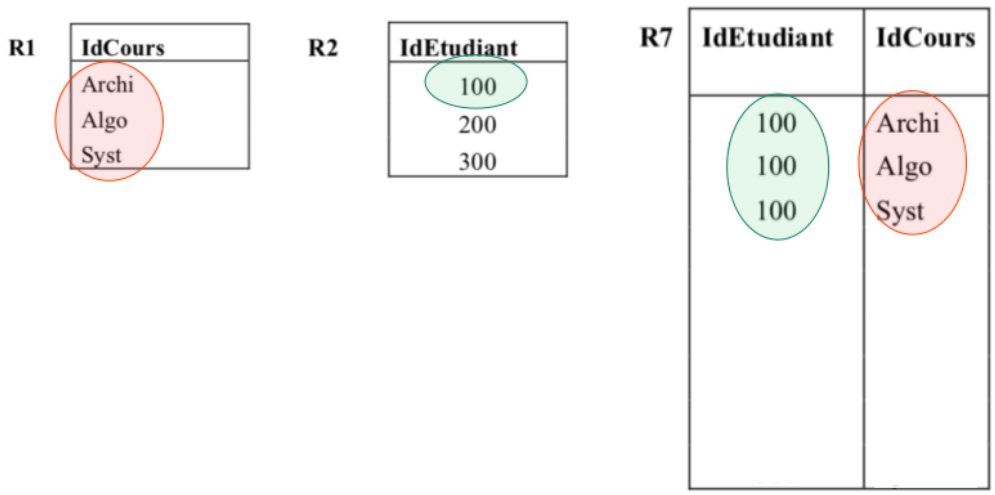
R7

IdEtudiant	IdCours

A partir des tables ci-dessus, donner les résultats des opérations suivantes :

$$R_7 = R_2 \times R_1$$

R_7 est le produit cartésien de la relation R_2 par la relation R_1.
Le schéma de R_7 est donc la concaténation de celui de R_2 puis de R_1.
On doit donc associer tout tuple de R_2 avec chacun des tuples de R_1.
On a $R_7 = \{ (x, y) \mid x \in R_2 \wedge y \in R_1 \}$



R1

IdCours
Archi
Algo
Syst

R2

IdEtudiant
100
200
300

R7

IdEtudiant	IdCours
100	Archi
100	Algo
100	Syst

R1

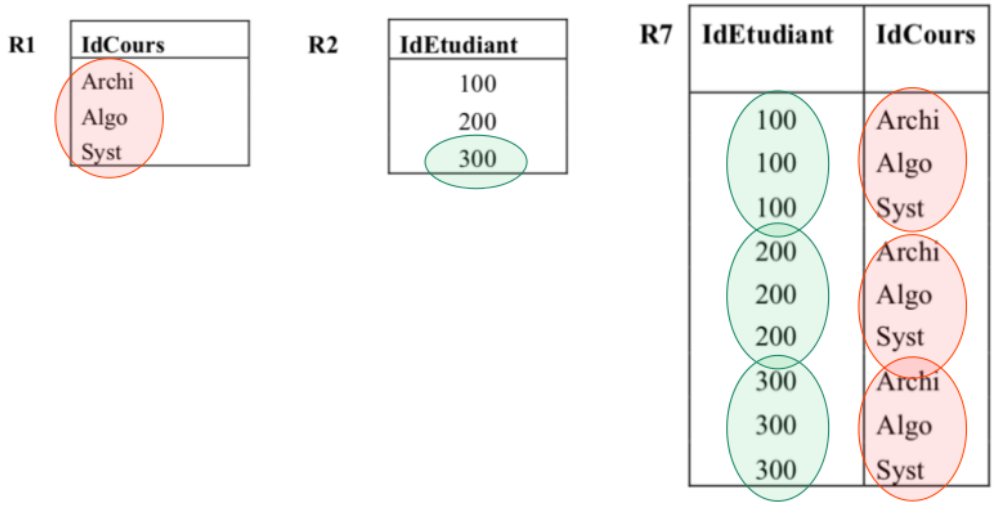
IdCours
Archi
Algo
Syst

R2

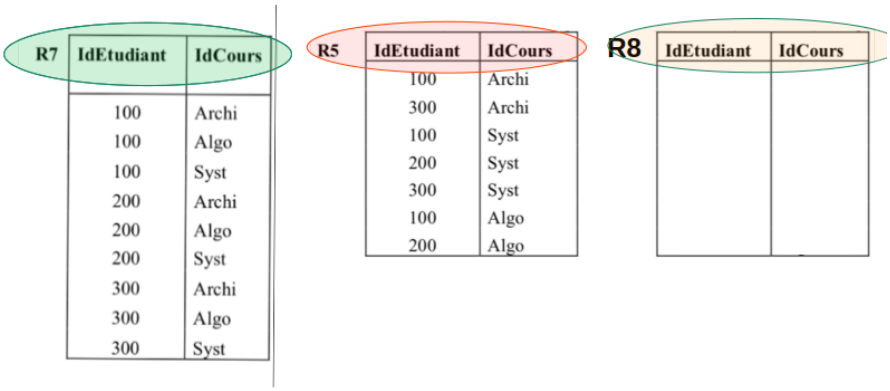
IdEtudiant
100
200
300

R7

IdEtudiant	IdCours
100	Archi
100	Algo
100	Syst
200	Archi
200	Algo
200	Syst



Exercice 1 : R8



A partir des tables ci-dessus, donner les résultats des opérations suivantes :

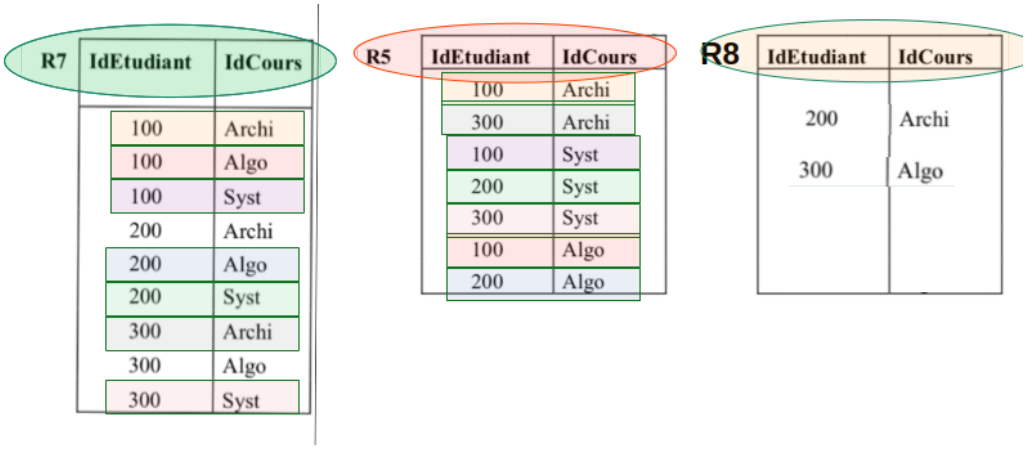
$$R_8 = R_7 - R_5$$

La relation R_8 est le résultat de la différence (ensembliste) entre R_7 et R_5.

Le schéma de R_7 et de R_5 doivent être identique (comme pour les autres opérateurs ensemblistes : union, intersection, différence).

On a $R_8 = \{ x \mid x \in R_7 \wedge x \notin R_5 \}$

On prend le tuple qui sont dans R_7 mais qui ne sont pas dans R_5



R7	IdEtudiant	IdCours
	100	Archi
	100	Algo
	100	Syst
	200	Archi
	200	Algo
	200	Syst
	300	Archi
	300	Algo
	300	Syst

R5	IdEtudiant	IdCours
	100	Archi
	300	Archi
	100	Syst
	200	Syst
	300	Syst
	100	Algo
	200	Algo

R8	IdEtudiant	IdCours
	200	Archi
	300	Algo

$$R_8 = R_7 - R_5$$

On prend chaque tuple de R_7 et on ne garde que ceux qui ne sont pas dans R_5

Exercice 1 : partie 2

Pour les opérations 1 à 5, écrire la requête SQL correspondante.

$R_1 = \pi$ (CJH / $(IdCours)$)
$R_2 = \pi$ (ENA / $(IdEtudiant)$)
$R_3 = \sigma$ (CEN / $IdCours =$ 'Algo')
$R_4 = \bowtie$(CJH, CS / $CJH.IdCours = CS.IdCours$)
$R_5 = \pi$(CEN / $(IdEtudiant, IdCours)$)

Exercice 1 : partie 2

Pour les opérations 1 à 5, écrire la requête SQL correspondante.

$R_1 = \pi$ (CJH / $(IdCours)$)

En **SQL** :

```
SELECT IdCours
FROM CJH ;
```

Exercice 1 : partie 2

Pour les opérations 1 à 5, écrire la requête SQL correspondante.

$R_2 = \pi$ (ENA / $(IdEtudiant)$)

En **SQL** :

```sql
SELECT IdEtudiant
FROM ENA ;
```

Exercice 1 : partie 2

Pour les opérations 1 à 5, écrire la requête SQL correspondante.

$R_3 = \sigma$ (CEN / $IdCours =$ 'Algo')

En **SQL** :

```sql
SELECT *
FROM CEN
WHERE IdCours = 'Algo' ;
```

Exercice 1 : partie 2

Pour les opérations 1 à 5, écrire la requête SQL correspondante.

$R_4 = \bowtie$(CJH, CS / $CJH.IdCours = CS.IdCours$)

En **SQL** :

```sql
SELECT *
FROM CJH
JOIN CS ON CJH.IdCours = CS.IdCours ;
```

Ou encore (mais à éviter car on mélange condition de jointure et condition de restriction classique) :

```sql
SELECT *
FROM CJH, CS
WHERE CJH.IdCours = CS.IdCours ;
```

Exercice 1 : partie 2

Pour les opérations 1 à 5, écrire la requête SQL correspondante.

$R_5 = \pi$(CEN / $(IdEtudiant, IdCours)$)

En **SQL** :

```sql
SELECT IdEtudiant, IdCours
FROM CEN ;
```

Ou encore :

```sql
SELECT CEN.dEtudiant, CEN.IdCours
FROM CEN ;
```

3 Exo 2

Exercice 2

Une maîtresse de maison veut construire une base de données sur les personnes qu'elle invite et les plats qu'elle leur sert. Elle identifie pour cela les trois relations suivantes :

- REPAS(INVITÉ , DATE) qui contient la liste des invités reçus et à quelle date.
- MENU (PLAT, DATE) qui contient le menu servi à chaque date.
- PRÉFÉRENCE (PERSONNE, PLAT) qui contient, pour chaque personne, ses plats préférés.

Sachant que **les attributs PERSONNE et INVITÉ ont le même domaine de valeurs**, réaliser les opérations relationnelles, dont les résultats sont :

...

Exercice 2

1. Les invités du repas du 02/10/2009.
2. Les dates auxquelles un « Bœuf Bourguignon » a été servi.
3. Les plats préférés de « Mme Machine ».
4. Les plats qui ont été servis à « Mr Machin ».
5. Les personnes invités qui ont été servi par leurs plats préférés.
6. Les personnes qui n'ont jamais été invité.
7. Les invités qui ont assisté à tous les repas.

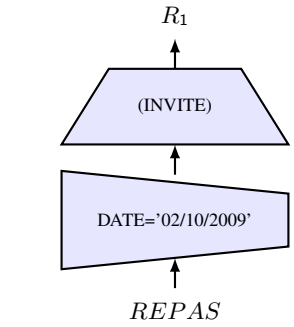
Les schémas :

- REPAS(INVITÉ , DATE) qui contient la liste des invités reçus et à quelle date.
- MENU (PLAT, DATE) qui contient le menu servi à chaque date.
- PRÉFÉRENCE (PERSONNE, PLAT) qui contient, pour chaque personne, ses plats préférés.

1. Les invités du repas du 02/10/2009.

- REPAS(**INVITÉ , DATE**)
- MENU (PLAT, DATE)
- PRÉFÉRENCE (PERSONNE, PLAT)

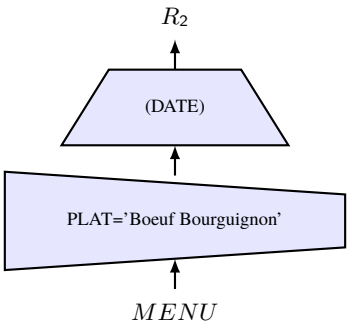
$$R_a = \sigma\ (\ REPAS\ /\ DATE =' 02/10/2009'\ )$$
$$R_1 = \pi\ (\ R_a\ /\ (INVITE\ )$$



2. Les dates auxquelles un « Bœuf Bourguignon » a été servi

- REPAS(INVITÉ , DATE)
- MENU (**PLAT, DATE**)
- PRÉFÉRENCE (PERSONNE, PLAT)

$$R_b = \sigma\ (\ MENU\ /\ PLAT =' Boeuf Bourguignon'\ )$$
$$R_2 = \pi\ (\ R_a\ /\ (DATE\ )$$

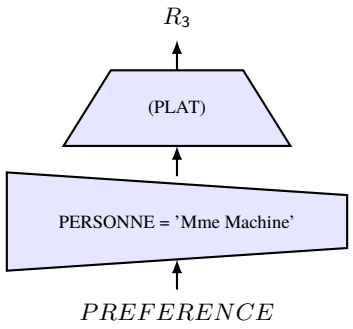


3. Les plats préférés de « Mme Machine ».

- REPAS(INVITÉ , DATE)

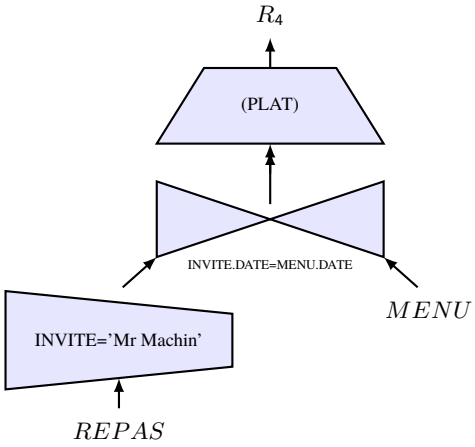
- MENU (PLAT, DATE)
- PRÉFÉRENCE (**PERSONNE, PLAT**)

$$R_c = \sigma\ (\ PREFERENCE\ /\ PERSONNE =' MmeMachine'\ )$$
$$R_3 = \pi\ (\ R_c\ /\ (PLAT\ )$$



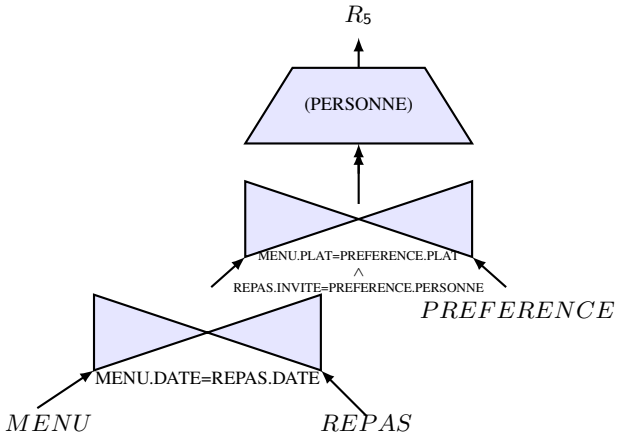
4. Les plats qui ont été servis à « Mr Machin ».

- REPAS(**INVITÉ** , DATE)
- MENU (**PLAT**, DATE)
- PRÉFÉRENCE (PERSONNE, PLAT)



5. Les personnes invités qui ont été servi par leurs plats préférés.

- REPAS(**INVITÉ** , DATE)
- MENU (PLAT, DATE)
- PRÉFÉRENCE (PERSONNE, **PLAT**)



La DIFFÉRENCE de deux relations

ENTREES : les deux relations doivent avoir le **même schéma.**
SORTIE : on récupère un ensemble en sortie : **pas de doublon.**

A − B

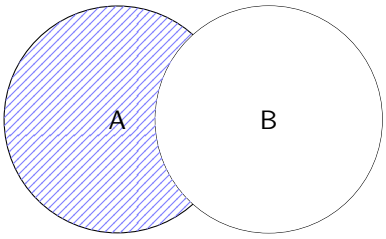
idP	x
1	a
4	a

A

idP	x
1	a
2	a
4	a

B

idP	x
1	b
2	a
4	c



A - B

On garde les éléments de A qui ne sont pas dans B.
L'opérateur n'est PAS commutatif : $\neg(\forall\ A, B\ (A - B = B - A))$.

La DIFFÉRENCE de deux relations

ENTREES : les deux relations doivent avoir le **même schéma.**
SORTIE : on récupère un ensemble en sortie : **pas de doublon.**

A − B

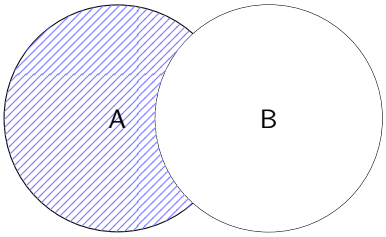
idP	x
1	a
4	a

A

idP	x
1	a
2	a
4	a

B

idP	x
1	b
2	a
4	c

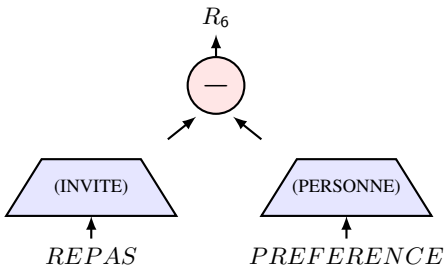


A - B

On garde les éléments de A qui ne sont pas dans B.
L'opérateur n'est PAS commutatif : $\neg(\forall\ A, B\ (A - B = B - A))$.

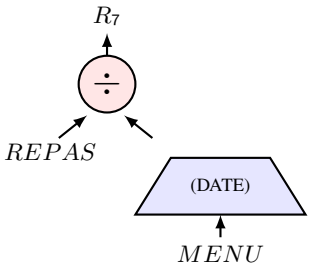
6. Les personnes qui n'ont jamais été invité.

- REPAS(**INVITÉ** , DATE)
- MENU (PLAT, DATE)
- PRÉFÉRENCE (**PERSONNE**, PLAT)



7. Les invités qui ont assisté à tous les repas.

- REPAS(INVITÉ , DATE)
- MENU (PLAT, DATE)
- PRÉFÉRENCE (PERSONNE, PLAT)



On divise donc REPAS(INVITE,DATE) par (DATE) de MENU.

On va donc obtenir les invités qui forment toujours un couple (INVITE, DATE) avec n'importe quelle date présente dans REPAS.