ALGÈBRE RELATIONNELLE

Algèbre relationnelle in a nutshell

- Inventée par Edgar Frank Ted Codd en 1970
- · Langage de requête dans les BDR
 - Expressions algébriques en notation fonctionnelle
 - Variables représentent des tables, instances de relations
 - · Opérateurs sur ces tables
 - Il existe un interprète de ce langage, qui permet de calculer toute expression portant sur des tables.

Algèbre relationnelle in a nutshell

- Opérations sur des relations
- Dont le résultat est une nouvelle relation qui peut à son tour être manipulée
- L'algèbre relationnelle permet de faire des recherches dans les relations

Algèbre relationnelle in a nutshell

- Opérations ensemblistes de la théorie des ensembles
 - Union, Intersection, Différence
 - Produit cartésien
- · Opérations relationnelles
 - Projection
 - Sélection
 - Jointure naturelle
 - Division
 - Renommage

Union, Intersection, Différence

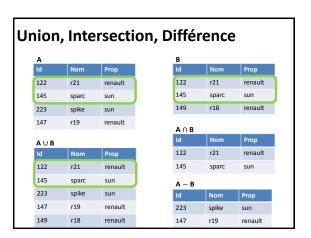
Soient r et s deux instances d'un même schéma de relation R,

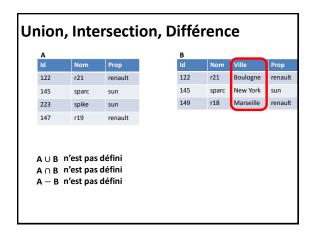
Les trois opérations suivantes définissent chacune une instance du même schéma de relation R:

 $r \cup s = \{t \mid t \in r \text{ ou } t \in s\}$ $r \cap s = \{t \mid t \in r \text{ et } t \in s\}$ $r - s = \{t \mid t \in r \text{ et } t \notin s\}$



Ces opérations sont impossible entre deux instances qui n'ont pas le même schéma de relation





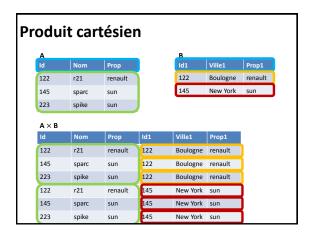
Produit cartésien

Soient r et s deux instances de schémas relations

$$r \times s = \{ (t_1, t_2) \mid t_1 \in r \text{ et } t_2 \in s \}$$



Le produit cartésien entre deux instances de schémas de relation n'est possible que si ces schémas de relation n'ont aucun attribut commun

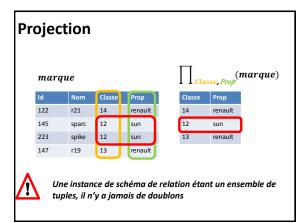


Projection

Soient R un schéma de relation et A = $\{A_1,, A_n\}$ un sous-ensemble d'attributs de R

- Projection sur A d'un tuple t défini sur R: $\prod_{A}(t) = (t.A_1:A_1, ..., t.A_n:A_n)$
- Projection sur A de l'instance de relation r du schéma ${\bf R}$:

$$\prod_{A}(r) = \{ \prod_{A}(t), t \in r \}$$



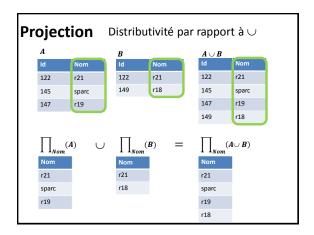
Projection

Propriété

Soient r et s deux instances du schéma de relation R et A un sous-ensemble d'attributs de R

Distributivité par rapport à l'union:

 $\prod_A (r \cup s) = \prod_A (r) \cup \prod_A (s)$





Propriétés

Soient r et s deux instances du schéma de relation R et A un sous-ensemble d'attributs de R

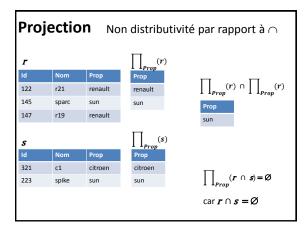
Distributivité par rapport à l'union:

$$\prod_{A}(r \cup s) = \prod_{A}(r) \cup \prod_{A}(s)$$

On n'a pas toujours :

$$\prod_A (r \cap s) = \prod_A (r) \cap \prod_A (s)$$

$$\prod_A (r-s) = \prod_A (r) - \prod_A (s)$$



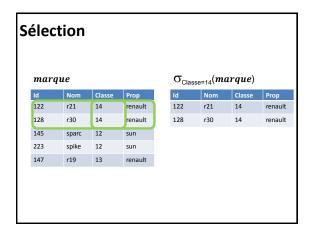
Sélection

Soient

- r une instance d'un schéma de relation R
- A un attribut de R
- a ∈ dom(A) une constante

La sélection de r par filtrage de A sur a est une instance du même schéma de relation R définie par :

$$\sigma_{A=a}(r) = \{t \in r : t.A = a\}$$

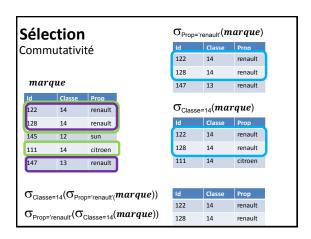


Sélection

Propriétés

Commutativité

$$\sigma_{A=a}(\sigma_{B=b}(r)) = \sigma_{B=b}(\sigma_{A=a}(r)) = \sigma_{A=a, B=b}(r)$$



Sélection

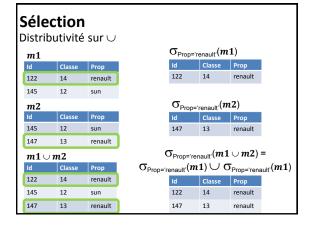
Propriétés

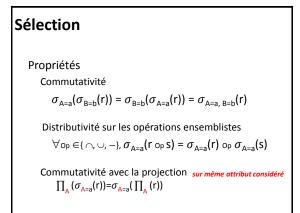
Commutativité

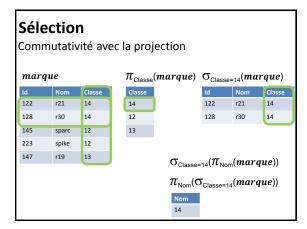
$$\sigma_{A=a}(\sigma_{B=b}(r)) = \sigma_{B=b}(\sigma_{A=a}(r)) = \sigma_{A=a, B=b}(r)$$

Distributivité sur les opérations ensemblistes

$$\forall o_P \in \{ \land, \cup, -\}, \sigma_{A=a}(r \circ_P s) = \sigma_{A=a}(r) \circ_P \sigma_{A=a}(s)$$







Sélection étendue

Soient

- r une instance de relation de schéma R,
- {A₁,...,A_n} un sous-ensemble d'attributs de R,
- f une fonction booléenne définie sur dom(A₁)×...×dom(A_n)

La sélection de r par $f(A_1,...,A_n)$ est une instance de relation de schéma R définie par :

$$\sigma_{f(A1,...,An)}(r) = \{t \in r \mid f(t.A_1,...,t.A_n)\}$$

Sélection étendue

Exemples:

Soient r une instance de relation de schéma R et A et B deux attributs de R

 $\sigma_{\text{A<13}}(r)$

 $\sigma_{A\neq 13}(r)$

 $\sigma_{A<13 \vee A>67}(r)$

 $\sigma_{A=13 \wedge B>12}$ (r)

Jointure naturelle

Contrairement aux autres opérations, elle est définie pour les tuples avant d'être définie pour les relations.

Jointure naturelle de tuples

Si t_1 et t_2 sont deux tuples n'ayant aucun attribut de même nom, alors on peut les joindre et le résultat est la concaténation de t_1 et t_2 .

Sinon, on ne peut les joindre que s'ils coïncident sur leurs attributs de même nom.

Jointure naturelle de tuples

Quand on peut joindre deux tuples, l'ensemble des attributs du tuple résultat est l'union des ensembles d'attributs des deux tuples joints.

On ne duplique pas les attributs de même nom.

Jointure naturelle de tuples

Deux tuples $t_r(R)$ et $t_s(S)$ sont **joignables** ssi il existe un tuple $t(R \cup S)$ tel que :

$$\prod_{R}(t) = t_r$$
 et $\prod_{S}(t) = t_s$

Ce tuple unique est noté:

$$t = t_r \bowtie t_s$$

Jointure naturelle de tuples

$$(t_r \bowtie t_s)$$
 existe $\iff \prod_{R \cap S} (t_r) = \prod_{R \cap S} (t_s)$

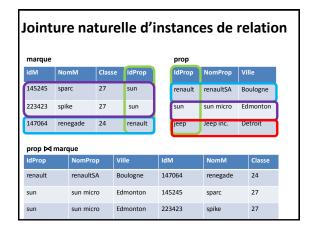
Si
$$(t_r \bowtie t_s)$$
 existe, alors
 $\prod_R ((t_r \bowtie t_s) = t_r \text{ et } \prod_S ((t_r \bowtie t_s) = t_s)$

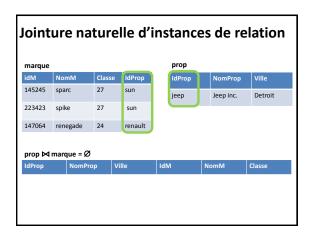
Jointure naturelle de tuples ('Chocolat':NomProd, 302:CodeProd, 4.5:Prix) ('StéX':NomDistrib, 'Chocolat':NomProd) = ('Chocolat':NomProd, 302:CodeProd, 4.5:Prix, 'StéX':NomDistrib)

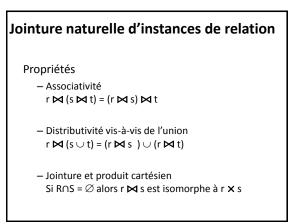
Jointure naturelle de tuples ('Chocolat':NomProd, 302:CodeProd,4.5:Prix) ('StéX':NomDistrib, 'CremeDeMarron':NomProd) n'existe pas

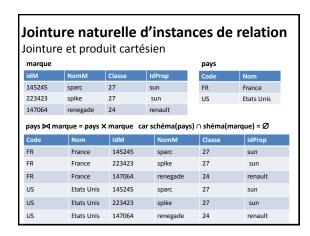
Jointure naturelle d'instances de relation

- Toujours possible
- Ensemble éventuellement vide de toutes les jointures de tuples joignables









Division

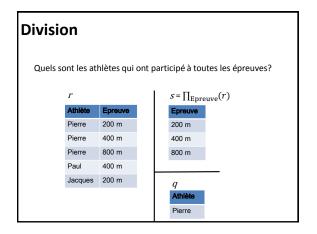
Soient r(R) et s(S) deux instances de relation, avec $S \subseteq R$,

r ÷ s est la relation de schéma Q=R—S définie par:

$$r \div s = \{ t_q \in \pi_Q(r) \mid \forall t_s \in s, t_q \bowtie t_s \in r \}$$

 $r \div s$ regroupe tous les éléments qui dans r sont reliés à tous les éléments de s :

 $r \div s~$ est le plus grand ensemble q de $\pi_Q(r)$ tel que (q $\bowtie s) \subset r$



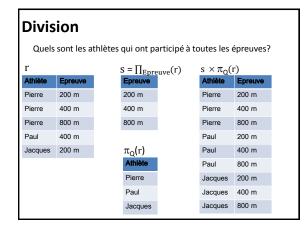
Division

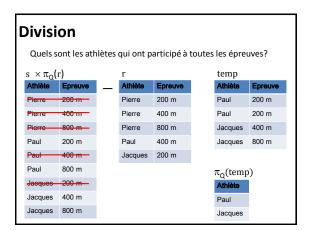
Propriété

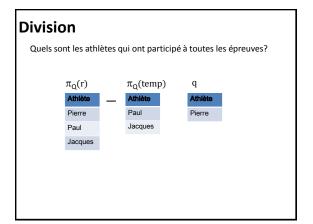
La division s'exprime en fonction des opérateurs précédents :

$$r \div s = \pi_O(r) - \pi_O((\pi_O(r) \bowtie s) - r)$$

$$r \div s = \pi_O(r) - \pi_O((\pi_O(r) \times s) - r)$$







Division

```
Existe-t-il une société qui possède \underline{toutes} les marques de la classe 14?  \prod_{\text{IdProp, IdM}} (\text{marque}) \div \prod_{\text{IdM}} (\sigma_{\text{Classe}=14}(\text{marque})) = \prod_{\text{IdProp}} (\text{marque}) - \prod_{\text{IdProp}} (\text{marque}) \bowtie \prod_{\text{IdM}} (\sigma_{\text{Classe}=14}(\text{marque})) - \prod_{\text{IdProp,IdM}} (\sigma_{\text{Classe}=14}(\text{marque}))
```

Renommage

Le renommage de A en B dans une instance r(R) est une instance du schéma $R'=R-\{A\}\cup\{B\}$ notée :

$$\delta_{A\leftarrow B}(r)$$

On peut étendre le renommage à plusieurs attributs, sous réserve de ne pas créer de collision de noms :

$$\delta_{A1...An\leftarrow B1....Bn}(r)$$

Renommage

Pourquoi renommer?

Parce qu'on ne dispose que de la jointure naturelle, qui dépend de l'homonymie des attributs

```
prop = {IdProp, NomProp, Pays, Ville}
enreq = {NumEnr, IdM, Date, Deposant}
```

L'attribut *Deposant* de *enreg* et l'attribut *IdProp* de *prop* correspondent à la même notion, mais ne portent pas le même nom.

Renommage

```
prop = {IdProp, NomProp, Pays, Ville}
enreg = {NumEnr, IdM, Date, Deposant}
```

Quels sont les noms de propriétaires ayant déposé au moins une marque avant le 15 janvier 91?

```
\begin{array}{c} \Pi_{\mathsf{NomProp}} (\\ \delta_{\mathsf{Deposant} \leftarrow \mathsf{IdProp}} (\\ \sigma_{\mathsf{Date} < 910115} (\mathsf{enreg}) \\ \\ \bowtie \mathsf{prop} \end{array})
```

Exemples de requêtes

Schéma de la base de données :

employe (nomEmploye, ville, rue) entreprise (nomEntreprise, ville, rue) travaille (nomEmploye, nomEntreprise, salaire)

Trouver les noms de tous les employés qui travaillent pour 'Banquissimo'

Exemples de requêtes

Trouver les noms de tous les employés qui travaillent pour 'Banquissimo'

```
\bigcap_{\mathrm{nomEmploye}} ( \sigma_{\mathrm{nomEntreprise='Banquissimo'}} (travaille)
```

)

)

Exemples de requêtes

Schéma de la base de données :
employe (nomEmploye, ville, rue)
entreprise (nomEntreprise, ville, rue)
travaille (nomEmploye, nomEntreprise, salaire)

Trouver les villes dans lesquelles résident au moins un des employés qui travaillent pour 'Banquissimo'

Exemples de requêtes

Trouver les villes dans lesquelles résident au moins un des employés qui travaillent pour 'Banquissimo'

```
\prod_{\text{ville}} (
\sigma_{\text{nomEntreprise='Banquissimo'}} (travaille)
\bowtie
employe
```

Exemples de requêtes

Schéma de la base de données :
employe (nomEmploye, ville, rue)
entreprise (nomEntreprise, ville, rue)
travaille (nomEmploye, nomEntreprise, salaire)

Trouver les noms des employés qui travaillent dans leur ville de résidence

Exemples de requêtes

Trouver les noms des employés qui travaillent dans leur ville de résidence

Exemples de requêtes

Schéma de la base de données :
employe (nomEmploye, ville, rue)
entreprise (nomEntreprise, ville, rue)
travaille (nomEmploye, nomEntreprise, salaire)

Trouver les noms des entreprises localisées dans au moins deux villes différentes

Exemples de requêtes

Trouver les noms des entreprises localisées dans au moins deux villes différentes

```
\bigcap_{\text{nomEntreprise}} (
\sigma_{\text{ville} \neq \text{ville2}} (
\text{entreprise}
\bowtie
\delta_{\text{ville} \leftarrow \text{ville2}, \text{rue} \leftarrow \text{rue2}} (\text{entreprise})
```

Exemples de requêtes

Schéma de la base de données :
employes (nomEmploye, ville, rue)
entreprises (nomEntreprise, ville, rue)
travaille (nomEmploye, nomEntreprise, salaire)

Trouver les noms des entreprises localisées dans au moins deux villes différentes

Exemples de requêtes

Trouver les noms des entreprises localisées dans au moins deux villes différentes

```
\begin{split} &\prod_{nomEntreprise}(\\ &\sigma_{ville \neq ville2}(\\ &\prod_{NomEntreprise,ville}(entreprise)\\ &\bowtie\\ &\delta_{ville \leftarrow ville2}(\prod_{NomEntreprise,ville}(entreprise)\\ &))) \end{split}
```

Exemples de requêtes

Schéma de la base de données :
employe (nomEmploye, ville, rue)
entreprise (nomEntreprise, ville, rue)
travaille (nomEmploye, nomEntreprise, salaire)

Trouver les noms des villes dans lesquelles toutes les entreprises sont implantées

Exemples de requêtes

Trouver les noms des villes dans lesquelles toutes les entreprises sont implantées

```
 \prod_{\text{nomEntreprise, ville}} (\text{entreprise}) 
 \vdots 
 \prod_{\text{ville}} (\text{entreprise})
```

Pour en savoir plus

- https://fr.wikipedia.org/wiki/Algèbre_relationnelle
- https://en.wikipedia.org/wiki/Relational_algebra
- http://georges.gardarin.free.fr/Livre_BD_Contenu/XX-TotalBD.pdf (chapitre 7)