

Enseigner SQL en NSI

Emmanuel Desmontils

Introduction

- Bases de données au programme de terminale (lycée), spécialité NSI
- Par rapport à l'université
 - Modélisation ?
 - MCD EAP ou UML
 - Sous-requêtes ?
 - Algèbre relationnel ?
 - Programmation ?
 - ~~Group by / Having~~
- Mais présence importante des projets !
- Demandé / Autorisé / Interdit / Non-dit

Contenus	Capacités attendues	Commentaires
Modèle relationnel : relation, attribut, domaine, clef primaire, clef étrangère, schéma relationnel.	Identifier les concepts définissant le modèle relationnel.	Ces concepts permettent d'exprimer les contraintes d'intégrité (domaine, relation et référence).
Base de données relationnelle.	Savoir distinguer la structure d'une base de données de son contenu. Repérer des anomalies dans le schéma d'une base de données.	La structure est un ensemble de schémas relationnels qui respecte les contraintes du modèle relationnel. Les anomalies peuvent être des redondances de données ou des anomalies d'insertion, de suppression, de mise à jour. On privilégie la manipulation de données nombreuses et réalistes.
Système de gestion de bases de données relationnelles.	Identifier les services rendus par un système de gestion de bases de données relationnelles : persistance des données, gestion des accès concurrents, efficacité de traitement des requêtes, sécurisation des accès.	Il s'agit de comprendre le rôle et les enjeux des différents services sans en détailler le fonctionnement.
Langage SQL : requêtes d'interrogation et de mise à jour d'une base de données.	Identifier les composants d'une requête. Construire des requêtes d'interrogation à l'aide des clauses du langage SQL : SELECT, FROM, WHERE, JOIN. Construire des requêtes d'insertion et de mise à jour à l'aide de : UPDATE, INSERT, DELETE.	On peut utiliser DISTINCT, ORDER BY ou les fonctions d'agrégation sans utiliser les clauses GROUP BY et HAVING.

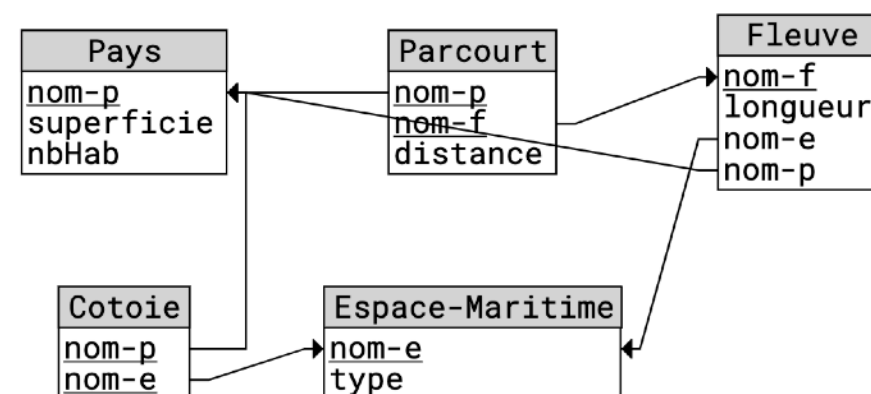
Difficultés à enseigner SQL

- SQL : nouveau paradigme
 - ~~procédural, fonctionnel, objet~~
 - Langage de requête
- Difficultés sur les opérateurs
- Difficulté à comprendre le modèle
 - Surtout sans connaissance des techniques de modélisation

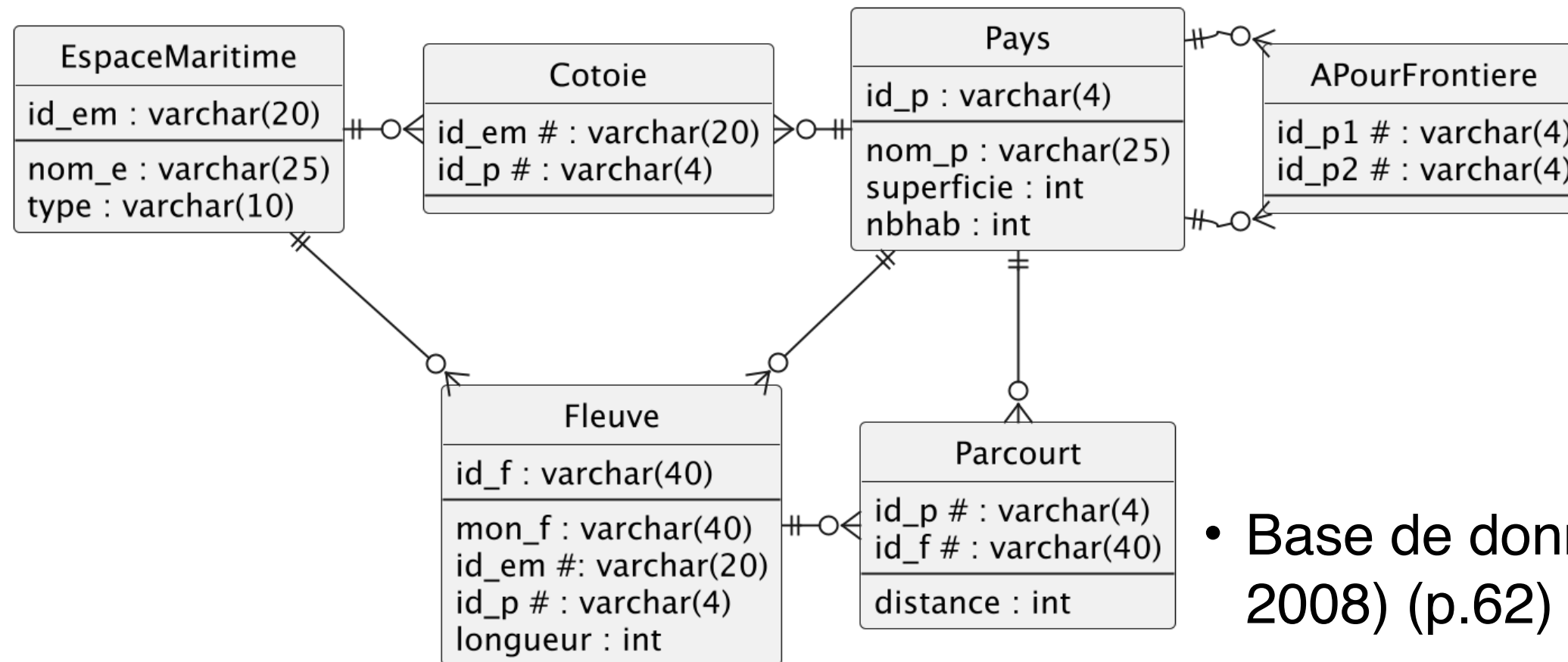
Opération	Poids
projection, tri	1
distinct	2
fonct-aggrégation, sélection simple	3
jointure simple	4
sélection complexe	5
jointures multiples, auto-jointure	6
aggrégation	10

(Kearns et al., 1997; Kleerekoper and Schofield, 2018; Mitrovic, 1998; Renaud and Van Biljon, 2004; Ahadi et al., 2015; Miedema et al., 2021)

Pays (nom-p, superficie, nbHab)
Fleuve (nom-f, longueur, #nom-e, #nom-p)
Espace-Maritime (nom-e, type)
Cotoie (#nom-p, #nom-e)
Parcours (#nom-p, #nom-f, distance)



Base exemple : EM



- Base de données (Chrisment et al., 2008) (p.62)
- Permet de recenser les fleuves et les espaces maritimes

Plan de la présentation

1. Proposition d'une organisation des situations didactiques pour enseigner SQL
2. Réflexions sur la notion de jointure
3. Vers un mode d'exécution pas à pas de SQL

1. Situations didactiques pour SQL

Emmanuel Desmontils, Laura Monceaux.

Enseigner SQL en NSI.

11ème Conférence Environnements Informatiques pour l'Apprentissage Humain (EIAH 2023) -Atelier "Apprentissage de la pensée informatique de la maternelle à l'Université : mise à l'épreuve des dispositifs et outils", EIAH 2023, Jun 2023, Brest, France. [hal-04108258v2](#)

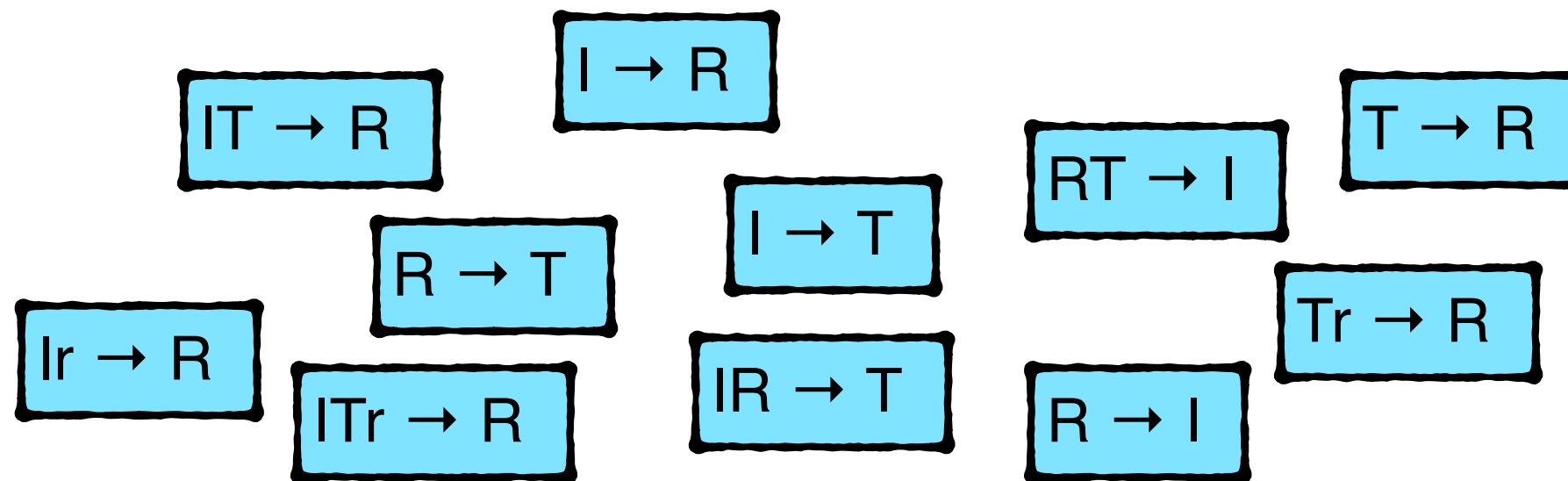
Situations didactiques pour SQL ?

- Comment enseigner SQL à des lycéens (et même des étudiants) ?
 - Constat : tous les exercices (quasiment) de la forme « donner la requête qui recherche... »
 - Pas très varié
 - Pas progressif
- Proposition de Reisner 1981 pour comparer les capacités des langages de l'époque :
 - "query writing",
 - "query reading",
 - "query interpretation",
 - "question comprehension ».

Situations didactiques pour SQL ?

- Quelques définitions :
 - Intention (I) : texte en langage naturel décrivant ce que l'on cherche dans la base de données,
 - Table (T) : ensemble de tuples,
 - Requête partielle (r) : requête dont certaines parties sont absentes
 - Requête complète (R) : requête exécutable
- Notre approche : recensement des situations didactiques possibles pour des exercices sur SQL : combinaison des éléments I, T, r & R
- Convention syntaxique
 - Entrée → Sortie
 - éléments optionnels : [x]
 - $R \Rightarrow T$: exécuter une requête sur une instance de base de données

Situations didactiques pour SQL ?



- Remarque : dans la suite, les notions de complexité des situations ne prennent pas en compte la complexité intrinsèque de la requête (des opérateurs utilisés).

Situations : $I[T] \rightarrow R$

- Objectif : être capable de transformer un besoin fonctionnel en une requête
- Forme habituelle : « Donner la requête SQL qui permet de trouver ... »
- Compliquée à concevoir
 - Maîtrise du modèle relationnel
 - Tables (situation des informations)
 - Liens entre tables (clés)
 - Comprendre le contexte fonctionnel
 - Maîtrise des opérateurs relationnels et de leur composition
- Compliquée à vérifier
 - Présence de la table : permet de vérifier que la requête donne le résultat attendu (pas nécessairement la bonne requête)
 - $IT1 \rightarrow R \Rightarrow T1$
 - Pas toujours facile à corriger (syntaxes différentes, mais équivalentes)

*En algo/prog :
« écrire un programme qui... »*

Situations : I[T] → R

APPRENDRE-SQL
Version 0.1
2023-06-07

BIENVENUE TAE1
DECONNEXION

ACTIONS

BASES DE DONNÉES
cours
tp-cinema
tp-vols
tp-em
tp-Lutins

CONTACT

MENTIONS

QUESTION 1 / 1 [refaire](#)

Intention :
Les fleuves de longueur supérieure à 1000 Km.
Donnez la requête correspondant à cette intention :

SELECT DISTINCT
FROM matieres;
 WHERE

[Executer](#) [Valider](#) [Enregistrer](#)

Copyright © 2023 E. Desmontils,
Nantes Université, France.
Designed by Free CSS Templates
Valid XHTML
Other legal mentions

- Remarques :
 - Situation la plus courante dans la littérature, les outils ou sur le Web,
 - Version « I → R » : Query Writing (Reisner, 1981)
 - Faire vérifier la requête !
- Exemple :
 - I = « Les fleuves de longueur supérieure à 1000 km »
 - R = « Select nom_f From Fleuve Where longueur > 1000; »

Situation : I → T

- Objectif : traduire une intention en table
- Forme habituelle : « Trouver le résultat de... »
- Complexité de conception : moyenne
 - Pas nécessité de maîtriser le langage SQL et ces opérateurs
 - Maîtrise du modèle relationnel
 - Tables (situation des informations)
 - Liens entre tables (clés)
 - Comprendre le contexte fonctionnel
- Très compliquée à vérifier :
 - Pas d'exécution possible pour l'élève
 - Par contre, facile à corriger

*« écrire ce qui se produit quand... »
En algo/prog :*

Situation : $I \rightarrow T$

- Remarques :
 - « Question comprehension » de Reisner 1981
 - Situation assez rare
 - Peut être combinée avec la situation précédente :
 - $I \rightarrow T \rightarrow R \Rightarrow T ?$
- Exemple :
 - $I =$ « Les fleuves de longueur supérieure à 1000 km »
 -

T :

nom_f
Loire
Tajo
Elbe
Donau
Rhein
Weichsel
Dnister
Petschora
Western Dwina
Dnepr
Don
Ob
Jenissej
Chatanga
Lena
Kolyma
Amur

Situations : [I]R \rightarrow T

- Objectif : comprendre le fonctionnement des opérateurs utilisés par une requête
- Forme habituelle : « Soit la base de données suivante, donner le résultat de la requête... [qui recherche...] »
- Complexité de conception : assez facile, surtout si l'intention est présente
 - Compréhension des opérateurs relationnels et de leur composition
- Facile à vérifier :
 - exécution possible pour l'élève, il suffit d'exécuter la requête
 - facile à corriger

« Ecrire ce que produit ce programme... »
En algo/prog :

Situations : [I]R → T

- Remarques :
 - "query interpretation" de (Reisner, 1981)
 - Situation assez rare (!)
 - variante : QCM avec plusieurs tables possibles
- Exemple :
 - Donner la table résultat de la requête suivante qui recherche les fleuves de longueur supérieure à 1000 Km
 - `Select nom_f From Fleuve Where longueur > 1000;`

APPRENDRE-SQL
Version 0.1
2023-06-07

BIENVENUE TAE1
DECONNEXION

ACTIONS

BASES DE DONNÉES
cours
tp-cinema
tp-vols
tp-em
tp-Lutins

CONTACT

MENTIONS

Copyright © 2023 E. Desmontils,
Nantes Université, France.
Designed by Free CSS Templates
Valid XHTML
Other legal mentions

tp-em / tp-em-01 / questions

Précédent

Suivant

QUESTION 2 / 4

Intention :
Les fleuves de longueur supérieure à 1000 Km.

Requête :
Select nom_f
From Fleuve
Where longueur>1000

Donnez la table correspondant à cette requête et cette intention :

NOM_F
Amur
Loire
Tajo
Elbe
Donau
Rhein
Weichsel
Dnister
Petschora
Western Dwina
Dnepr
Don
Ob
Jenissej
Chatanga
Lena
Kolyma

Proposer Valider Enregistrer

Félicitation vous as réussi à trouver la bonne table validez votre réponse!

Situations : $I[T]r \rightarrow R$ & $Tr \rightarrow R$

- Objectif : découvrir les différents opérateurs disponibles dans une requête SQL
- Forme habituelle : « Compléter la requête suivante :... »
- Complexité de conception : assez facile (surtout avec l'intention)
 - Maîtrise du modèle relationnel
 - Tables (situation des informations)
 - Liens entre tables (clés)
 - Compréhension des opérateurs relationnels et de leur composition
- Assez facile à vérifier
 - exécution possible pour l'élève
 - pas toujours facile à corriger

*« Compléter ce programme qui... »
En algo/prog :*

Situations : $I[T]r \rightarrow R$ & $Tr \rightarrow R$

APPRENDRE-SQL
Version 0.1
2023-06-07

BIENVENUE TAE1
DECONNEXION

tp-em / tp-em-01 / questions

QUESTION 1 / 4 en cours

Intention :
Les fleuves de longueur supérieure à 1000 Km.
Donnez la requête correspondant à cette intention :

SELECT
FROM Fleuve
WHERE longueur>1000;

DISTINCT

Executer Valider Enregistrer

Copyright © 2023 E. Desmontils,
Nantes Université, France.
Designed by Free CSS Templates
Valid XHTML
Other legal mentions

- Remarques

- curieusement, quasiment jamais présente dans les ressources consultées
- variante : mettre tous les parties d'une requête, pas seulement celle cachées.

- Exemple

- « Select ? ? ? ? ? From Fleuve
Where longueur > 1000 ; «

Situation : $R[T] \rightarrow I$

- Objectif : amener les élèves à comprendre le besoin fonctionnel d'une requête pour éventuellement la corriger
- Forme habituelle : « Indiquer ce que fait cette requête... »
- Complexité de conception : difficile (surtout sans table)
 - Maîtrise des opérateurs relationnels et de leur composition
 - Maîtrise du modèle relationnel
- Très difficile à vérifier
 - Pas facile de trouver des éléments pour vérifier
 - Pas toujours facile à corriger (langage naturel)

« En algo/prog :
« Que fait ce programme... ? »

! « Que retourne cette requête ? »
 $R \rightarrow T$ ou $R \rightarrow I$?

Situation : $R[T] \rightarrow I$

Apprendre SQL

APPRENDRE-SQL

Version 0.1
2023-06-07

BIENVENUE TAE1
DECONNEXION

tp-em / tp-em-01 / questions

← Précédent Suivant →

QUESTION 3 / 4 en cours

Requête :
Select nom_f
From Fleuve
Where longueur > 1000

Donnez l'intention correspondante à cette requête :

On sélectionne dans les fleuves, ceux qui ont l'attribut longueur plus grand que 1000 et on projette nom_f

Valider Enregistrer

BASES DE DONNÉES
cours
tp-cinema
tp-vols
tp-em
tp-Lutins

CONTACT

MENTIONS

Copyright © 2023 E. Desmontils,
Nantes Université, France.
Designed by Free CSS Templates
Valid XHTML
Other legal mentions

- Remarques :
 - $R \rightarrow I$: "query reading" de (Reisner, 1981)
 - Assez rare dans les ouvrages
 - Attention, souvent une formulation en français de la requête et non l'intention !
- Exemple :
 - « Select nom_f From Fleuve Natural Join Espace-Maritime Where type = 'mer' »
 - « On fait la jointure entre Fleuve et Espace-maritime, on ne garde que les mers et on projette le nom du fleuve. »
 - « Le nom des fleuves qui se jettent dans une mer. »

Situation : $T \rightarrow R$

- Objectif : comprendre des données et le besoin sous-jacent
- Forme habituelle : « Trouver la requête qui permet d'obtenir ces données... »
- Complexité de conception : très difficile
 - Maîtrise du modèle bien sûr
 - Bonne anticipation sur le calcul d'une requête
- Très facile à vérifier
 - il suffit d'exécuter la requête
 - Facile à corriger (selon les attentes)

*En algo/prog :
« écrire un programme
qui produit cette trace... »*

Situation : $T \rightarrow R$

- Remarques :
 - Permet de faire prendre conscience que plusieurs requêtes donnent un même résultat sur une instance.
 - pas facile d'anticiper les solutions possibles
- Exemple :
 - Donner la requête permettant d'obtenir le résultat ci-contre.
 - « `Select nom_f From Fleuve Where longueur > 1000; »`

nom_f
Loire
Tajo
Elbe
Donau
Rhein
Weichsel
Dnister
Petschora
Western Dwina
Dnepr
Don
Ob
Jenissej
Chatanga
Lena
Kolyma
Amur

Outils

AsseSQL (Prior and Lister, 2004)

BlocklySQL (Pöhner et al., 2019)

SQL-Tutor (Mitrovic, 1998)

SQheLper (Jacobs and Jaschke, 2021)

SQLator (Sadiq et al., 2004)

SQLSandbox (Desmontils, 2010)

SQL-KnoT (Brusilovsky et al., 2008, 2010)

- I → R ou requêtes « libres »
- Organisation d'exercices et de tests avec traces
- Construction de requêtes (parfois par blocks)

Ouvrages

Ouvrage / Situation	R-T *	R-I *	IT-R	I-R *
(Connan et al., 2020)		1		9
(Bonneyoy and Petit, 2020)			1	10
(Balabonski et al., 2020)	5	4		20
(Bays, 2020)				68

Types de questions dans les ouvrages à destination des élèves

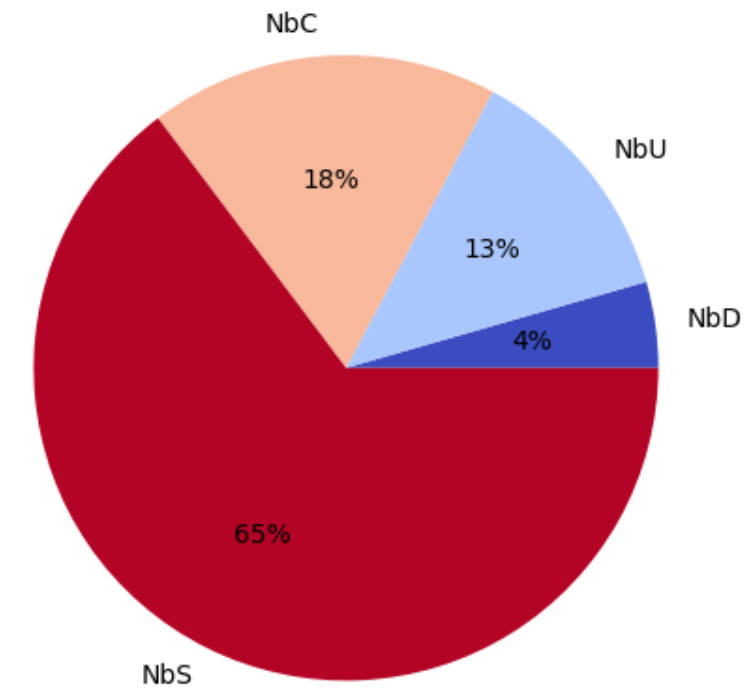
* Reisner 1981

Et au bac ?

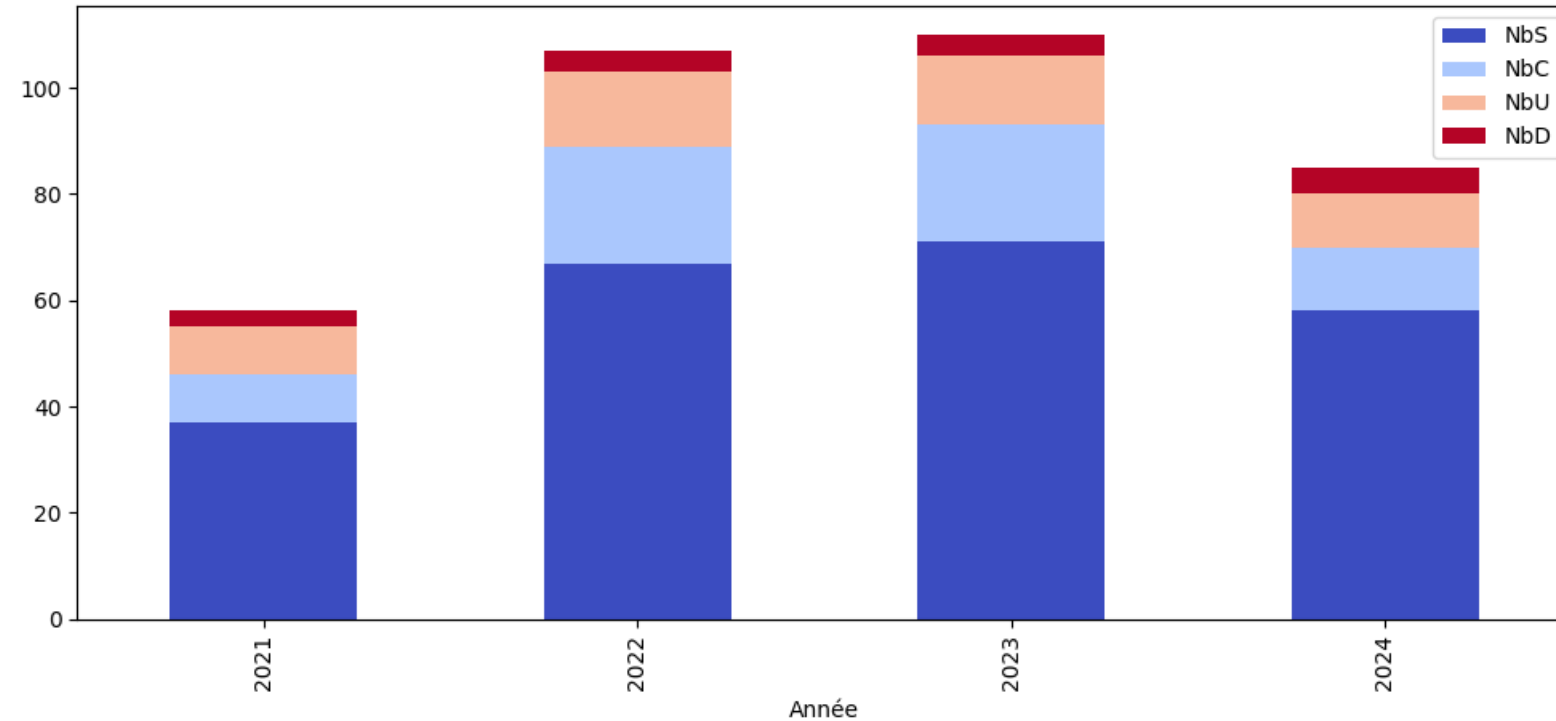
- 2021-2024
- 63 sujets
- 63 exercices
- 360 requêtes
- 5.7 par exercice
- Entre 1 et 11 requêtes par exercice
- Analyse uniquement sur SQL et sur les jointures.

2021 : 58
2022 : 107
2023 : 110
2024 : 85

Répartition globale des requêtes



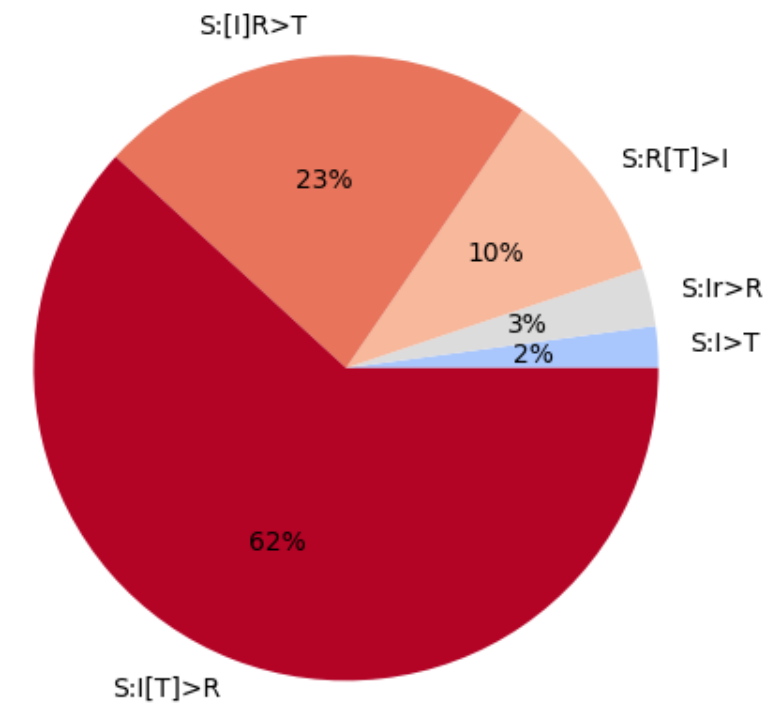
Nombre des types de requêtes par année



Et au bac ?

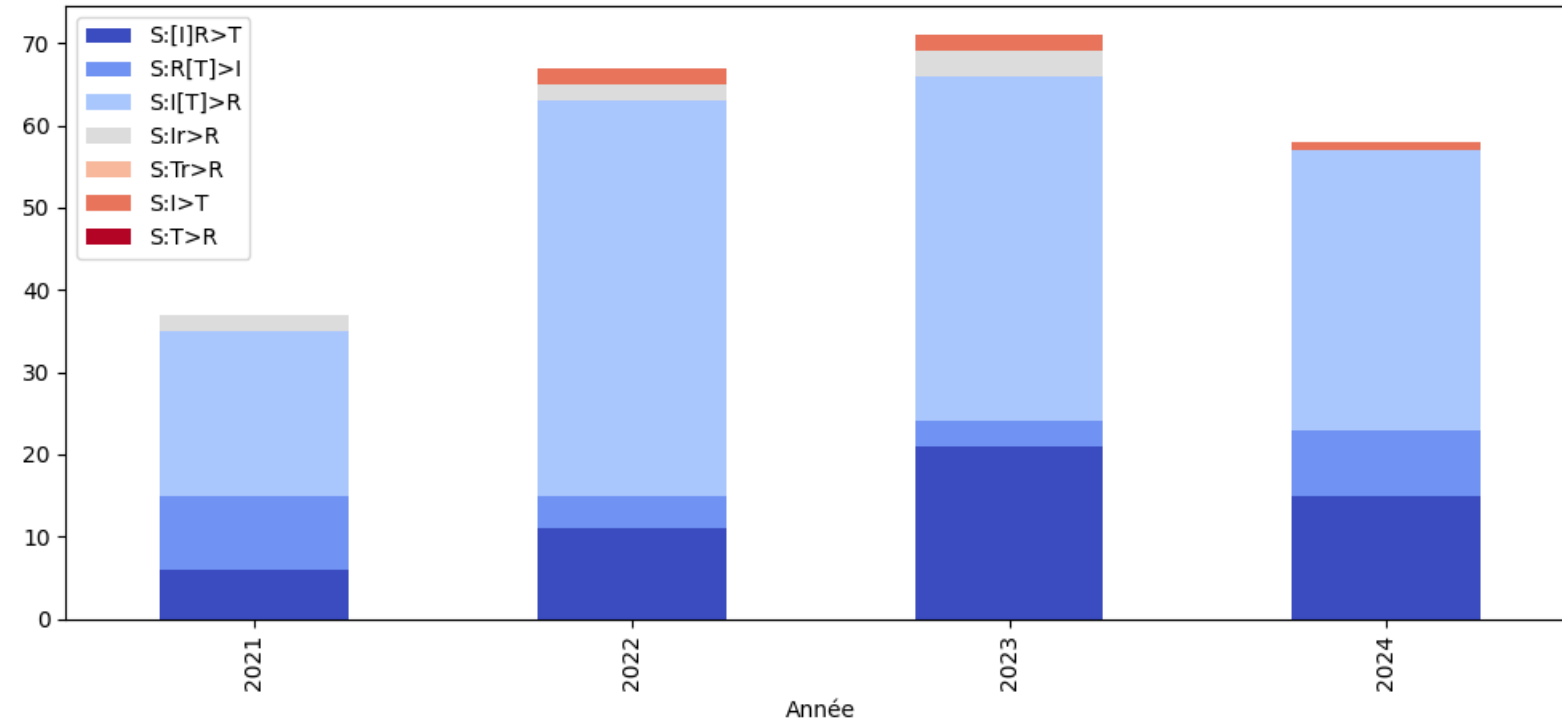
- Requêtes de recherche
- 233 requêtes (65%)
- Très grande majorité:
 $I \rightarrow R$
- Assez souvent : $R \rightarrow T$

Répartition globale des requêtes de type S



2021 : 37
2022 : 67
2023 : 71
2024 : 58

Nombre des types de requêtes S par année

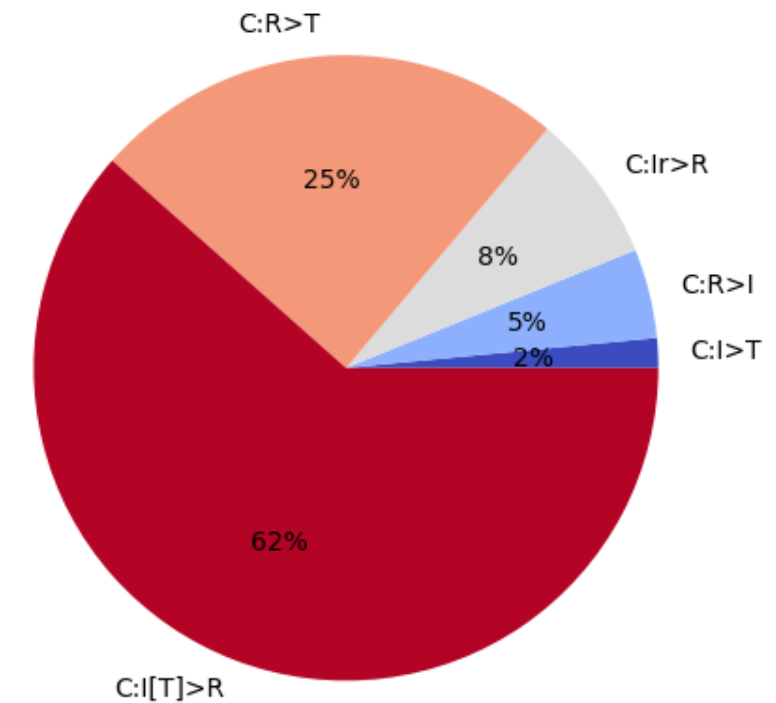


Et au bac ?

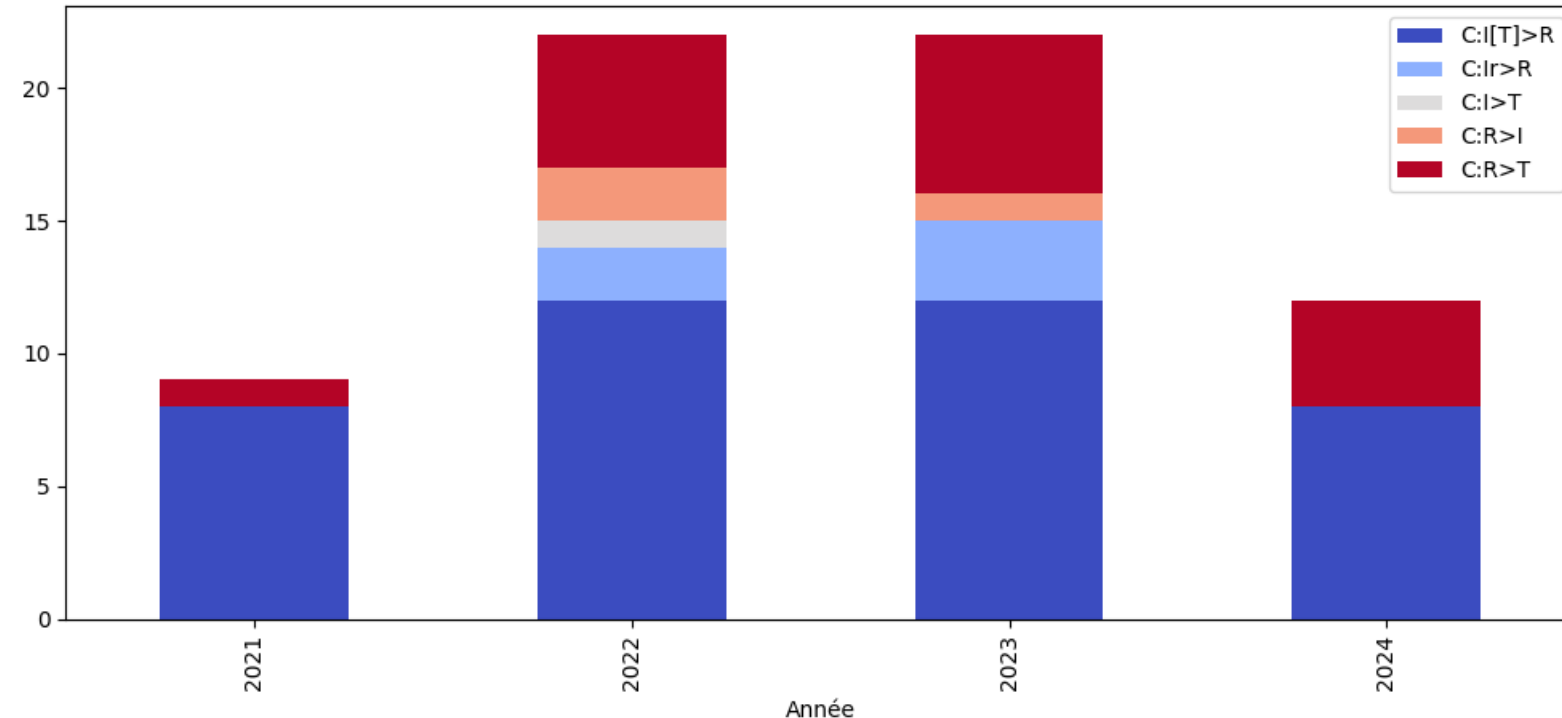
- Requêtes d'ajout
- 65 requêtes (18%)
- Très grande majorité de $I \rightarrow R$
- Assez souvent : $R \rightarrow T$
 - « Pourquoi la requête ne fonctionne pas ? »
- Requêtes à trous aussi

2021 : 9
2022 : 22
2023 : 22
2024 : 12

Répartition globale des requêtes de type C



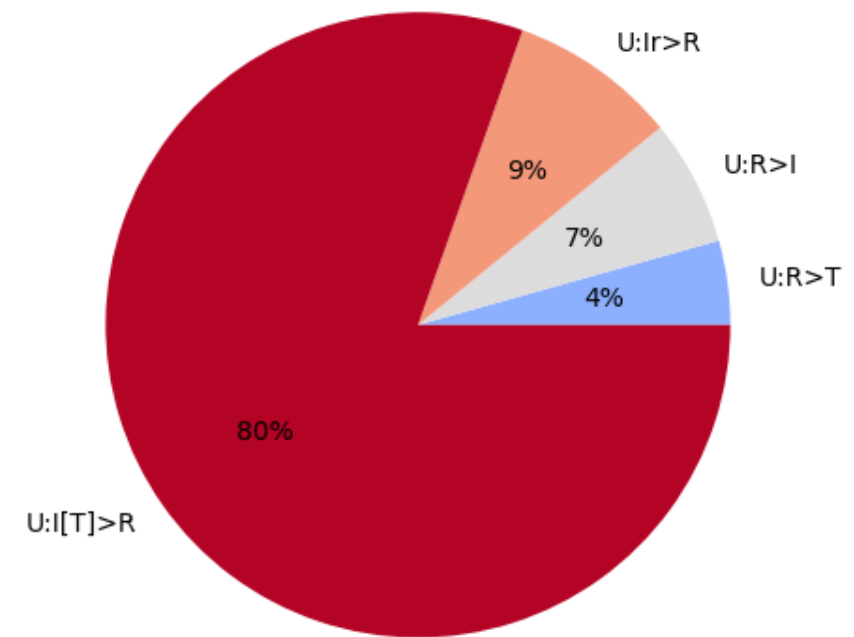
Nombre des types de requêtes C par année



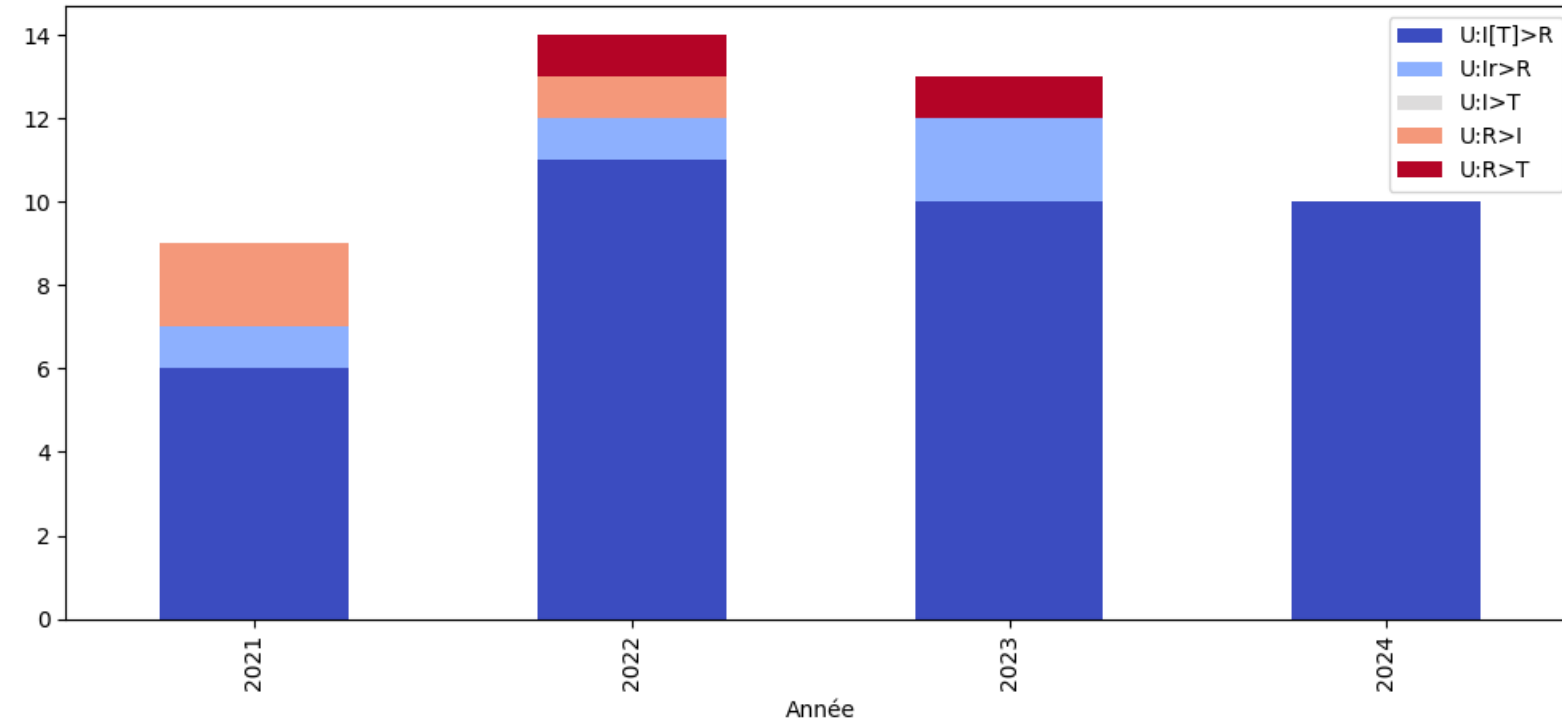
Et au bac ?

- Requêtes de mise à jour
- 46 requêtes (13%)
- Grande majorité de $I \rightarrow R$
- Curieusement très peu de $R \rightarrow T$!
- Requêtes à trous aussi

Répartition globale des requêtes de type U



Nombre des types de requêtes U par année



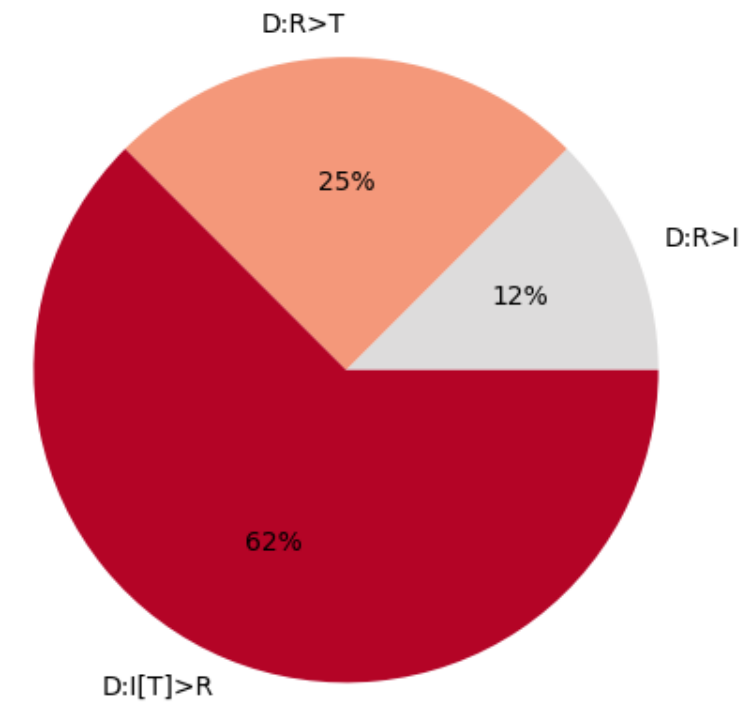
2021 : 9
2022 : 14
2023 : 13
2024 : 10

Et au bac ?

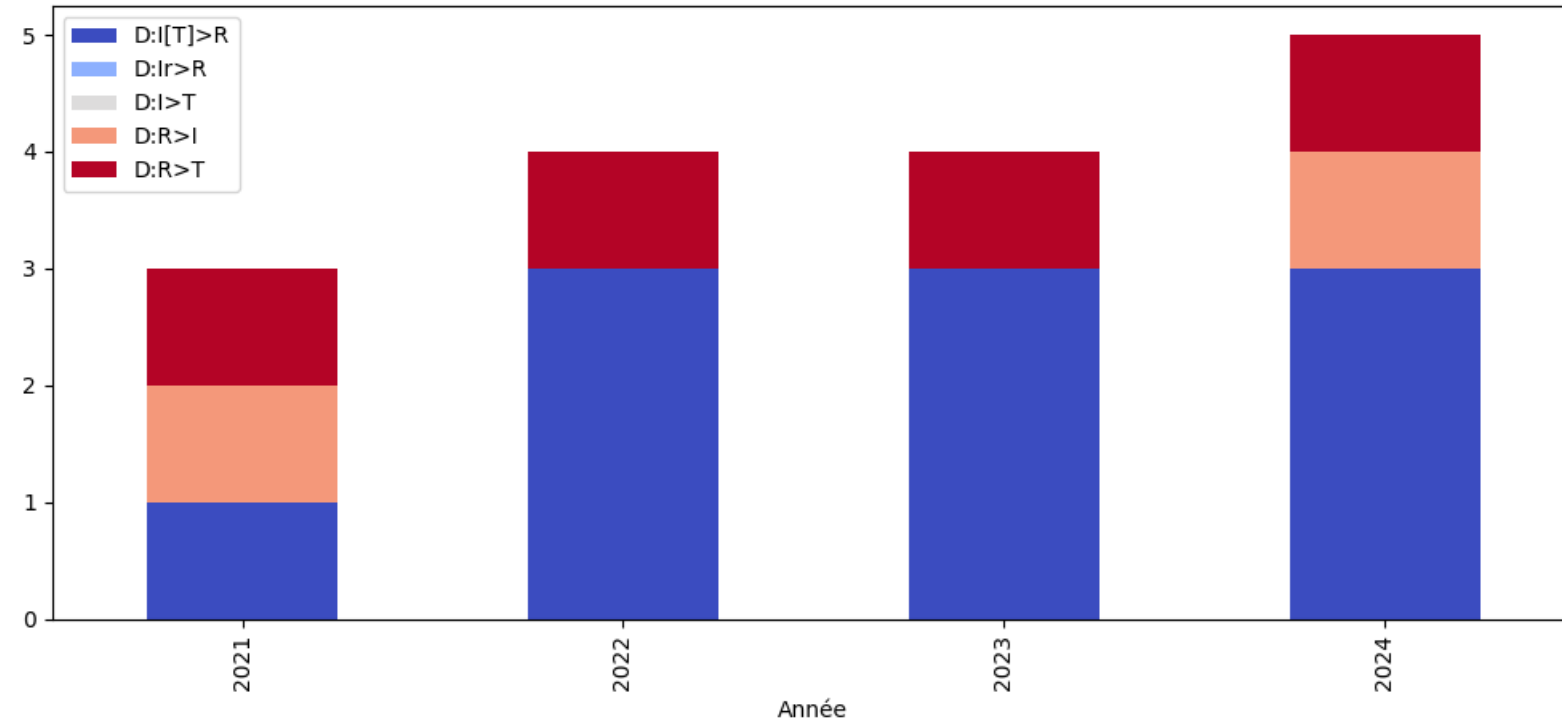
- Requêtes de suppression
- 16 requêtes (4%)
- Très grande majorité de $I \rightarrow R$
- Assez souvent : $R \rightarrow T$
 - « Pourquoi la requête ne fonctionne pas ? »

2021 : 3
2022 : 4
2023 : 4
2024 : 5

Répartition globale des requêtes de type D



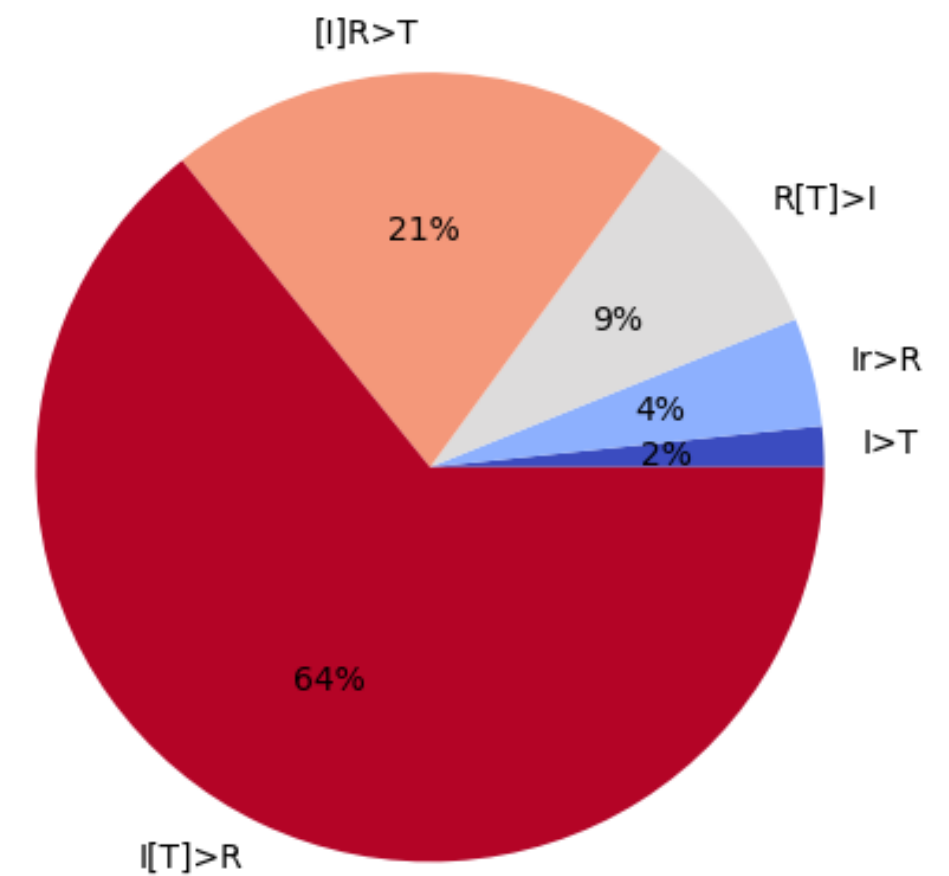
Nombre des types de requêtes D par année



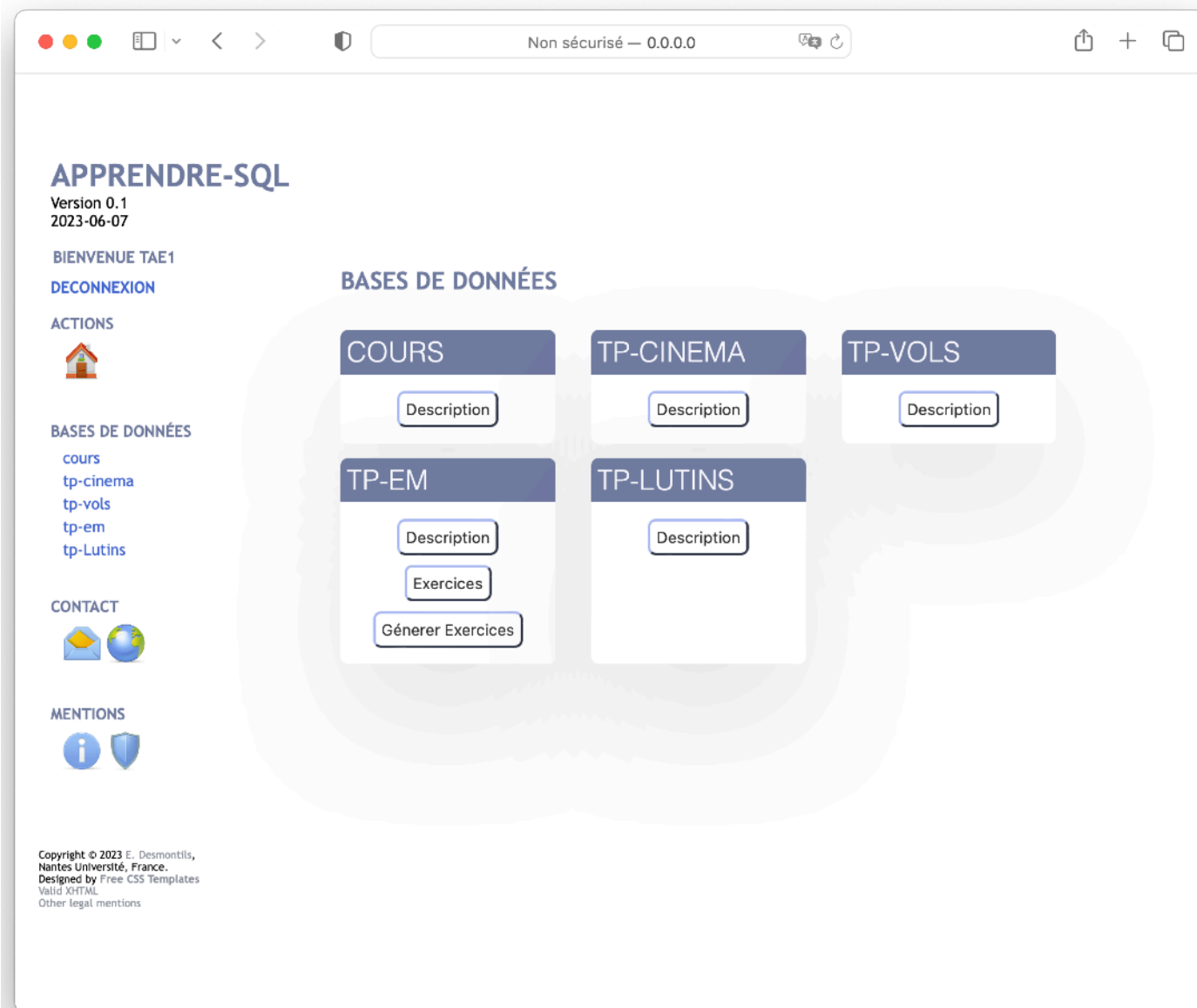
Et au bac ?

- $I[T] \rightarrow R$: entre 60% et 65% tout type confondu
- $[I]R \rightarrow T$: entre 14% et 26% tout type confondu
- Soit 74% à 88% sur ces deux types

Répartition globale des requêtes de type CRUD



Apprendre-SQL : une mise en œuvre des situations



- Mise en œuvre des situations :
 - En fiche de TD sur papier
 - En TP par une application
- Pour les élèves :
 - Exercices composés de questions
 - Question : situation didactique
 - Requêtes libres
- Pour l'enseignant :
 - Correction des exercices
 - Requêtes libres

<https://gitlab.univ-nantes.fr/ls2n-didactique/asql>

Point

- Situations proposées en TD et en TP
- Pour « Select », mais aussi pour « Insert », « Update » et « Delete »
- Application
 - à fiabiliser
 - à peaufiner
 - à mettre en ligne
 - à mettre en oeuvre les traces de conception pour analyse
- Expérimentation commencée en lycée (Machecoul, Loire-Atlantique) sans l'application
 - Pas d'évaluation formelle
- Approfondir l'évaluation des compétences

2. Questions

2.1. Compréhension des tables

Donner la cardinalité et le degré de la relation RESULTATS

Quelle est la clé de la relation epreuve ?

2.2. Compréhension des opérateurs

Quel est le résultat de l'expression : $\pi_{nom,pays}(\sigma_{cat>1}(robots))$?

Est-ce que l'expression $\sigma_{cat>1}(\pi_{nom,pays}(robots))$ donne le même résultat que l'expression précédente ? (justifiez)

Quel est le résultat de l'expression : $\pi_{nom,compet,ville}(\sigma_{pays='France'}(robots) \bowtie \sigma_{montant>200}(inscriptions) \bowtie competitions)$?

Dessinez l'arbre algébrique de cette expression.

Réécrire cette expression sans utiliser de jointure naturelle.

Complétez l'expression suivante (indiquer ce qu'il faut mettre à la place des lettres majuscules) pour obtenir les noms de robots qui ont obtenu plus de 100 points à une compétition française : $\pi_A(B_C(résultats) D_E \text{ epreuves } F G_H(competitions))$

Quel est le résultat de l'expression suivante : $avg_{montant}(\sigma_{compet='IRM'}(inscriptions))$?

2.3. Construction d'expressions

Construire l'expression permettant d'obtenir les épreuves se déroulant le matin pour des robots de catégorie 1.

Construire l'expression permettant d'obtenir les noms et labo des robots de catégorie 1 ayant payé une inscription 200€.

Construire une expression permettant d'obtenir le plus gros gain pour un robot français.

APPRENDRE-SQL
Version 0.1
2023-06-07
BIENVENUE TAE1
DECONNEXION
tp-em / tp-em-01 / questions
Suivant >
QUESTION 1 / 4 en cours
Intention :
Les fleuves de longueur supérieure à 1000 Km.
Donnez la requête correspondant à cette intention :
SELECT [] DISTINCT
FROM Fleuve
WHERE longueur>1000;
Executer Valider Enregistrer
Copyright © 2023 E. Desmontils, Nantes Université, France. Designed by Free CSS Templates. Voir HTML. Other legal mentions.

Question de la jointure

Un opérateur complexe multiforme

- Un BO lacunaire
- Multiforme :
 - [Inner] Join ... On / Using...
 - Left / Right / Full Outer Join
 - Natural Join
 - Cross Join
- Parties plus ou moins implicites
 - Using(att)
 - Natural Join
- Pas de lien « fonctionnel » évident
 - À l'opposé du
 - Select (ce qu'on veut voir/calculer),
 - Where (on filtre)
 - Order By (on trie)
- Et en SQL1 ?

Contenus	Capacités attendues	Commentaires
Langage SQL : requêtes d'interrogation et de mise à jour d'une base de données.	Identifier les composants d'une requête. Construire des requêtes d'interrogation à l'aide des clauses du langage SQL : SELECT, FROM, WHERE, JOIN. Construire des requêtes d'insertion et de mise à jour à l'aide de : UPDATE, INSERT, DELETE.	On peut utiliser DISTINCT, ORDER BY ou les fonctions d'agrégation sans utiliser les clauses GROUP BY et HAVING.

Jointure en algèbre relationnel

- θ -Jointure : opérateur composite de l'algèbre relationnel :

$$R \bowtie_{C(R,S)} S = \sigma_{C(R,S)}(R \times S)$$

- avec $C(R, S)$ une expression logique de comparaisons entre attributs de R et attributs de S
- Composite + présence du produit cartésien : explique en partie la distance fonctionnelle
- Quelques propriétés :
 - Commutativité : $Join(R, S, Q) = R \bowtie_Q S = S \bowtie_Q R$
 - Associativité : $(R \bowtie_{Q_1} S) \bowtie_{Q_2} T = R \bowtie_{Q_1} (S \bowtie_{Q_2} T)$
 - ...

Jointure en algèbre relationnel

$R \bowtie_Q S$

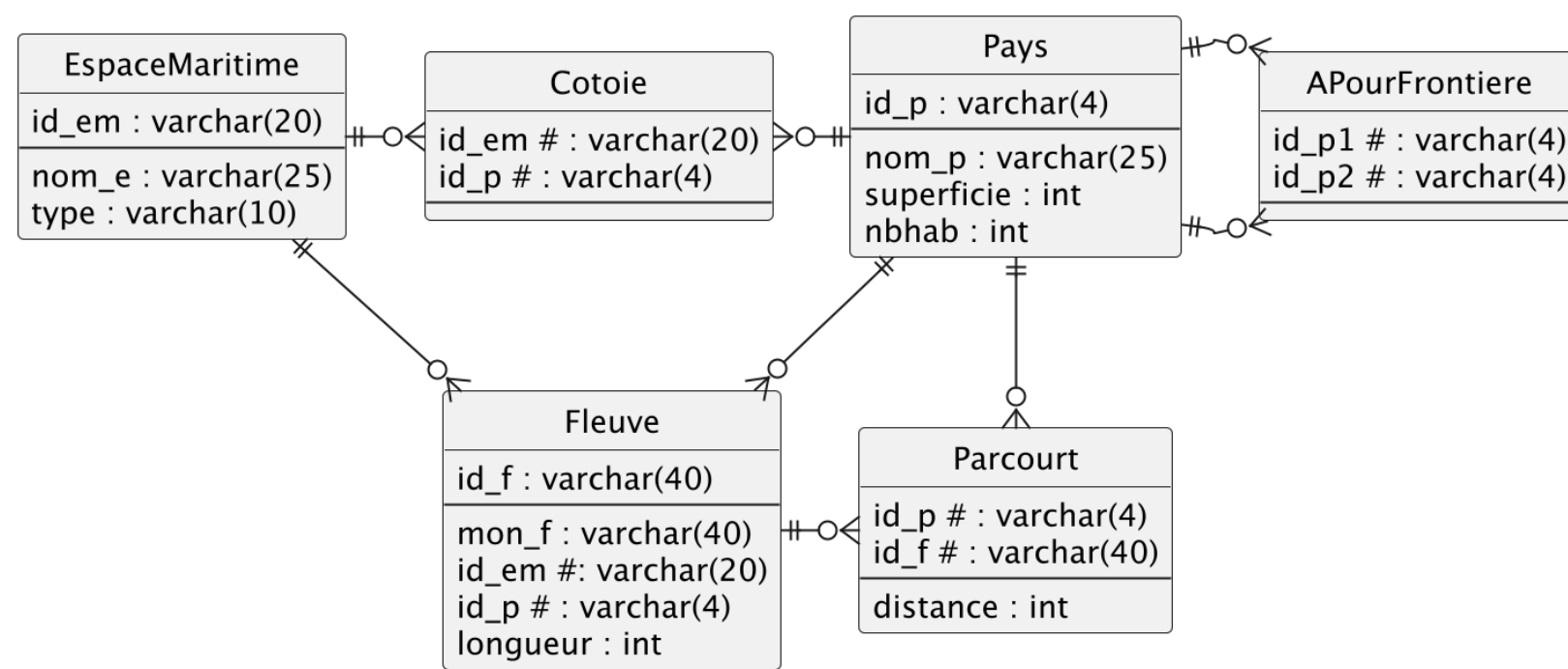
- Cas particuliers :
 - **Auto-jointure** : $R \bowtie_Q R$
 - **Equi-jointure** : Q formée uniquement d'égalités
 - **Inéqui-jointure** : non(Equi-jointure)
 - **Jointure naturelle** : $R \bowtie S$
 - Equi-jointure sur les attributs de même nom
 - **Semi-jointure** : $R \bowtie_Q S = \pi_R(R \bowtie_Q S)$
 - **Jointure externe** : $R \cdot \bowtie_Q S$ ou $R \bowtie_Q \cdot S$ ou $R \cdot \bowtie_Q \cdot S$
 - Tous les tuples de la table « pointée » sont présents dans le résultat (complétés par des Null le cas échéant)

Opérateur ternaire

$R \bowtie_{Q_1} S \bowtie_{Q_2} T$

- 2 tables et une contrainte... pas simple à manipuler en SQL !
 - SELECT * FROM
(R INNER JOIN S ON Q1) INNER JOIN T ON Q2;
 - SELECT * FROM
R INNER JOIN (S INNER JOIN T ON Q2) ON Q1;
 - SELECT * FROM
R INNER JOIN S ON Q1 INNER JOIN T ON Q2;
- Et toutes les variantes sans INNER et en prenant en compte la commutativité !

Quelques exemples



-- 1 : jointure "simple" à 2 tables

```
Select nom_f, Parcourt.id_p  
From Fleuve Join Parcourt on Fleuve.id_f = Parcourt.id_f;
```

-- 2 : Inéqui-jointure et auto-jointure

```
Select f.nom_f  
From Fleuve f Join Fleuve l on f.id_f <> l.id_f and f.longueur < l. longueur  
Where l.nom_f = 'Loire' ;
```

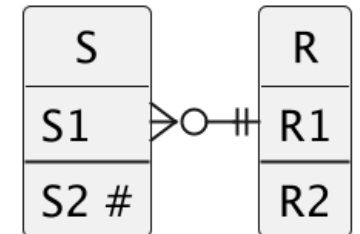
-- 3 : Jointure à 3 tables

```
Select nom_p, nom_e  
From Pays Join Cotoie Join EspaceMaritime Using(id_em) Using (id_p) ;
```

-- 4 : Jointure externe à droite

```
Select id_p, nom_p  
From Fleuve Right Outer Join Pays Using(id_p)  
Where id_f is Null ;
```

Cas très particulier de la jointure de clés

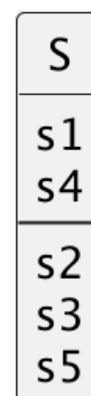
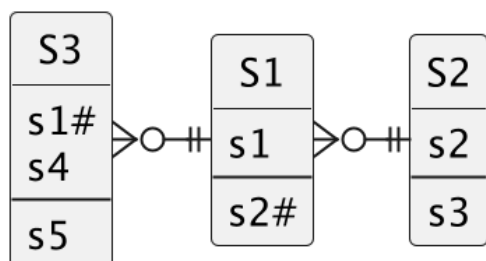
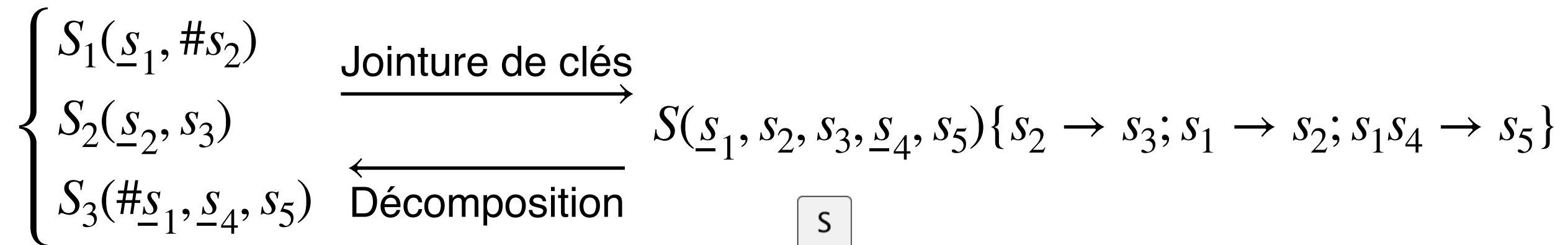


• $R(\underline{R}_1, R_2)$ et $S(\underline{S}_1, \#S_2)$ telles que $S_2 \rightarrow R_1 : R \bowtie_K S = R \bowtie_{R_1=S_2} S$

• Jointure entre les attributs clés d'une table et une clé étrangère de l'autre table (en contrainte d'intégrité référentielle avec la première).

• Remarque : permet de mettre en oeuvre l'inverse du processus de normalisation (passage de la table universelle à des tables sans anomalies, i.e. au moins en 3e forme normale)?

Exemple :



Cas très particulier de la jointure de clés

- Cas de jointure « simple »
- Lisible sur le schéma relationnel : surtout en représentation en graphe

Fleuve(id_f, nom_f, longueur, #id_em, #id_p)

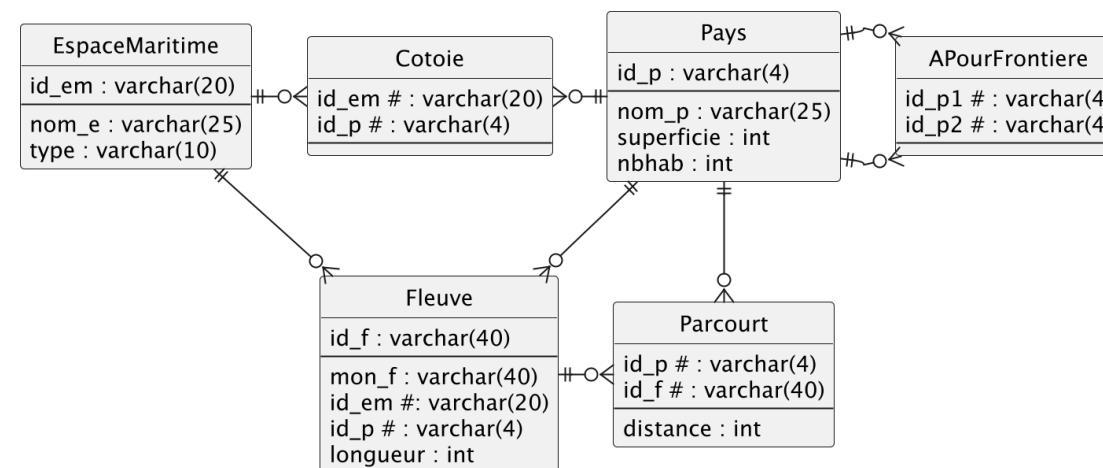
Pays(id_p, nom_p, superficie, nbhab)

EspaceMaritime(id_em, nom_e, type)

Parcours(#id_p, #id_f, distance)

Cotoie(#id_em, #id_p)

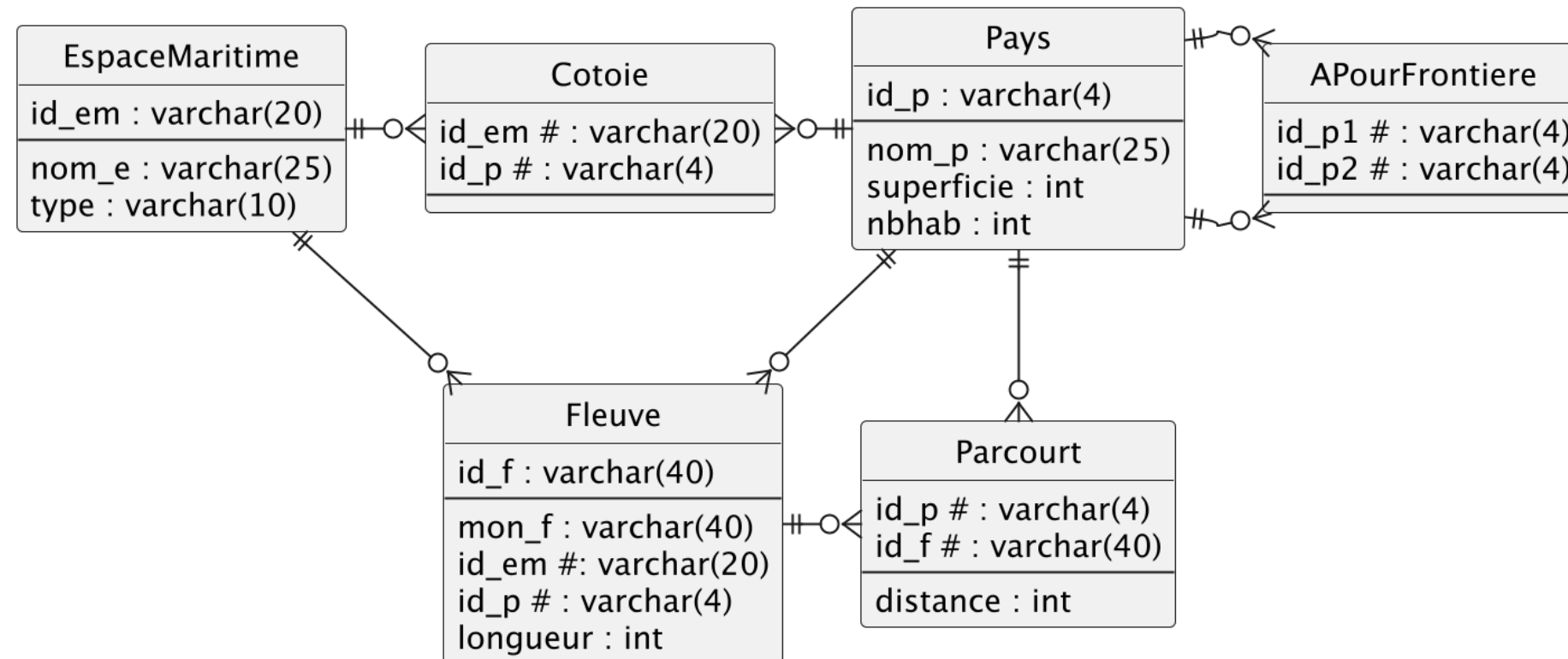
APourFrontiere(#id_p1, #id_p2)



- Concrètement : elle permet de suivre les « liens » entre les tables (contraintes d'intégrités référentielles) et donc de regrouper des données cohérentes

- A noter : $|R \bowtie_K S| = |S|$ alors que $|R \bowtie_Q S| \leq |R| * |S|$
donc des requêtes plus faciles à calculer en cas de $R \rightarrow T$

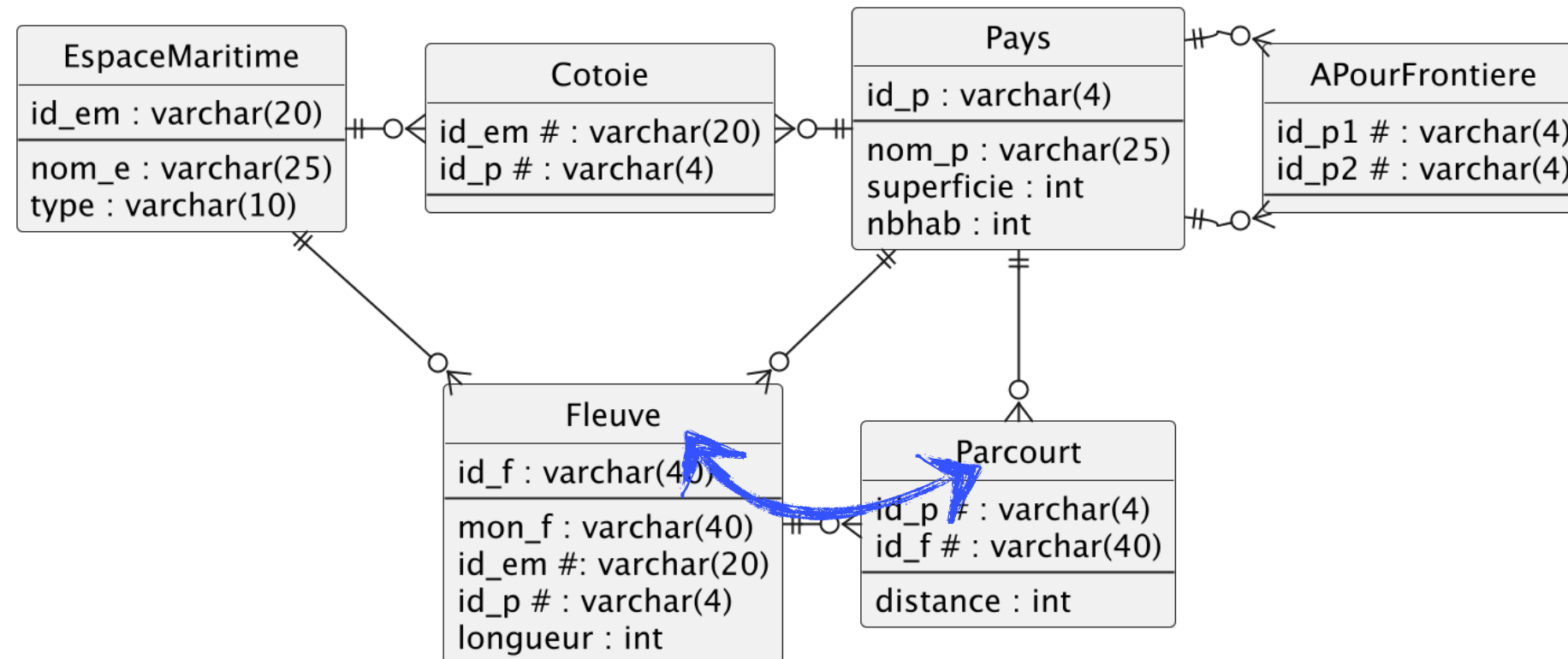
Cas de la jointure de clés



```
Select nom_f, Parcours.id_p  
From Fleuve Join Parcours on Fleuve.id_f = Parcours.id_f;
```

```
Select nom_p, nom_e  
From Pays Join Cotoie Using(id_p) Join EspaceMaritime Using(id_em) ;
```

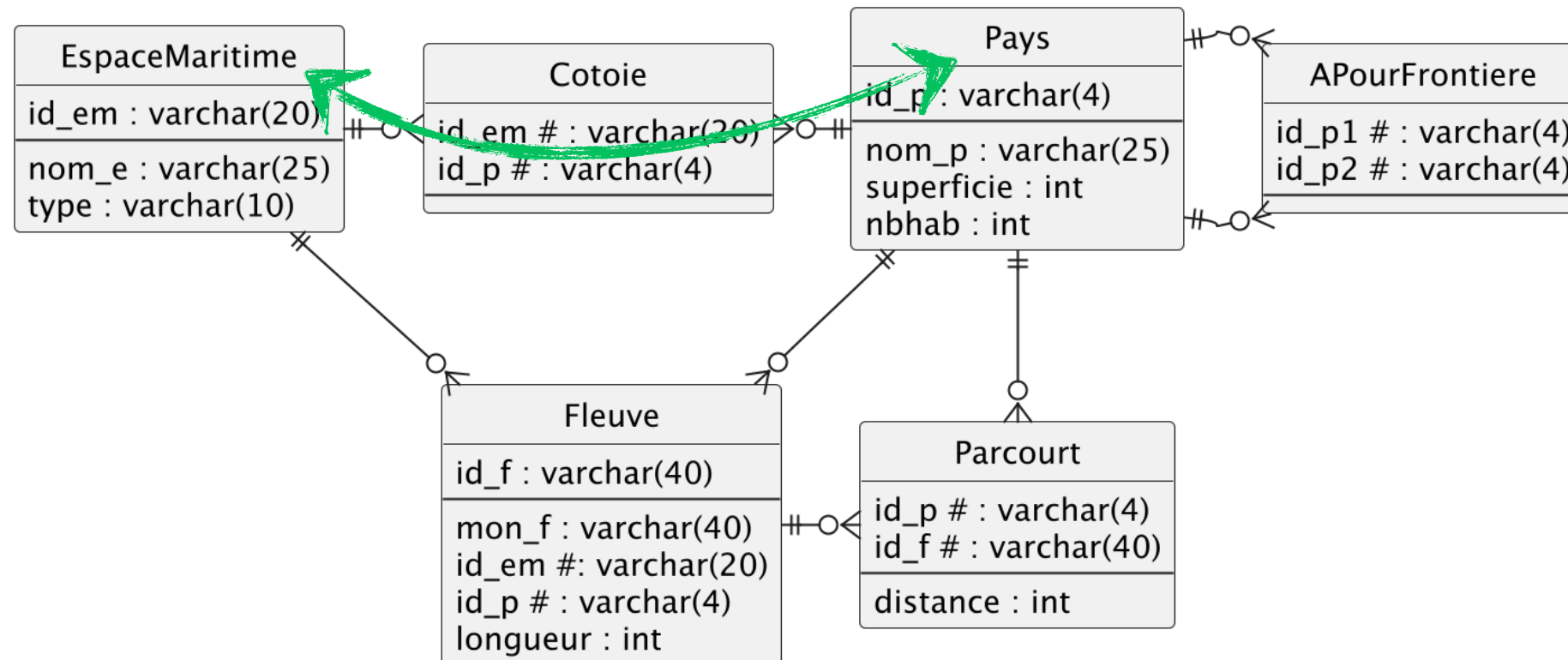
Cas de la jointure de clés



```
Select nom_f, Parcourt.id_p  
From Fleuve Join Parcourt on Fleuve.id_f = Parcourt.id_f;
```

```
Select nom_p, nom_e  
From Pays Join Cotoie Using(id_p) Join EspaceMaritime Using(id_em) ;
```

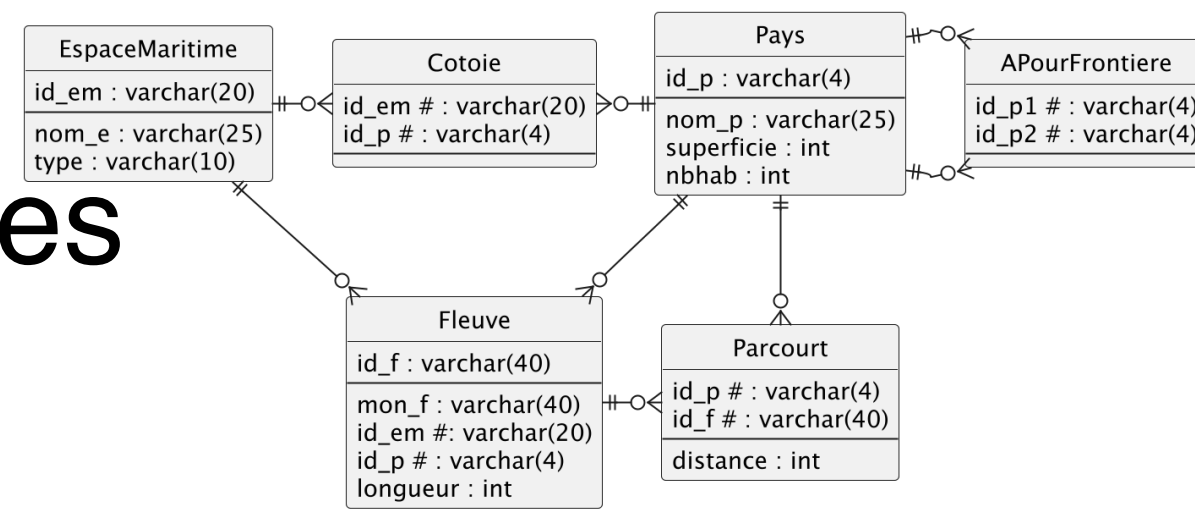
Cas de la jointure de clés



```
Select nom_f, Parcours.id_p  
From Fleuve Join Parcours on Fleuve.id_f = Parcours.id_f;
```

```
Select nom_p, nom_e  
From Pays Join Cotoie Using(id_p) Join EspaceMaritime Using(id_em) ;
```

Inéqui-jointures et auto-jointures



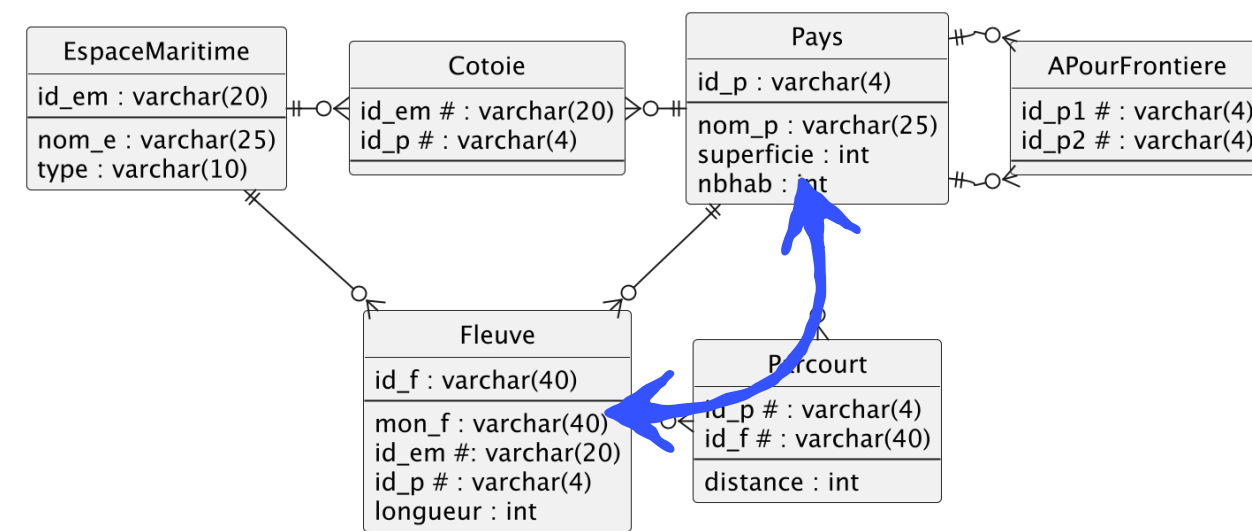
- Inéqui-jointure : souvent sur des attributs non-clés et numériques (pour avoir les opérateurs de comparaison)
 - Souvent sur des dates, des « mesures », des coûts...
- Auto-jointure : jointure d'une table avec elle-même
 - Alias obligatoire
 - Complexe à appréhender.
 - Reconnue comme étant une des opérations les plus complexes en SQL
 - Souvent demandée dans les sujets de CAPES...
 - Souvent dans le cas de « boucle » dans le modèle relationnel, exemple :
Employé(id, nom, prénom, #idChef) avec Employé.idChef → Employé.id
- Souvent ensembles
- Requêtes à laisser en fin d'exercice...

```
Select f1.nom-f
From Fleuve f1 Join Fleuve f2
On f1.longueur > f2.longueur
Where f2.nom-f = "Seine"
and f1.mon-p = 'France' ;
```

Cas de la jointure naturelle

- Jointure $R \bowtie S$ avec condition implicite/calculée
 - Tout attribut de même nom doit avoir la même valeur
- Soit $R(a,b,d,d)$ et $S(a,b,e,f,g)$ alors : $R \bowtie S = R \bowtie_{R.a=S.a \wedge R.b=S.b} S$
- Pas indispensable en NSI

Cas de la jointure naturelle



- Attention, parfois elle va « trop loin »
 - Exemple : « Le nom des pays parcourus par le Rhone ».

```
Select nom_p
From Fleuve as f Inner Join Parcours as pa On f.id_f=pa.id_f
Inner Join Pays as p On p.id_p=pa.id_p
Where nom_f = 'Rhone' ;
```

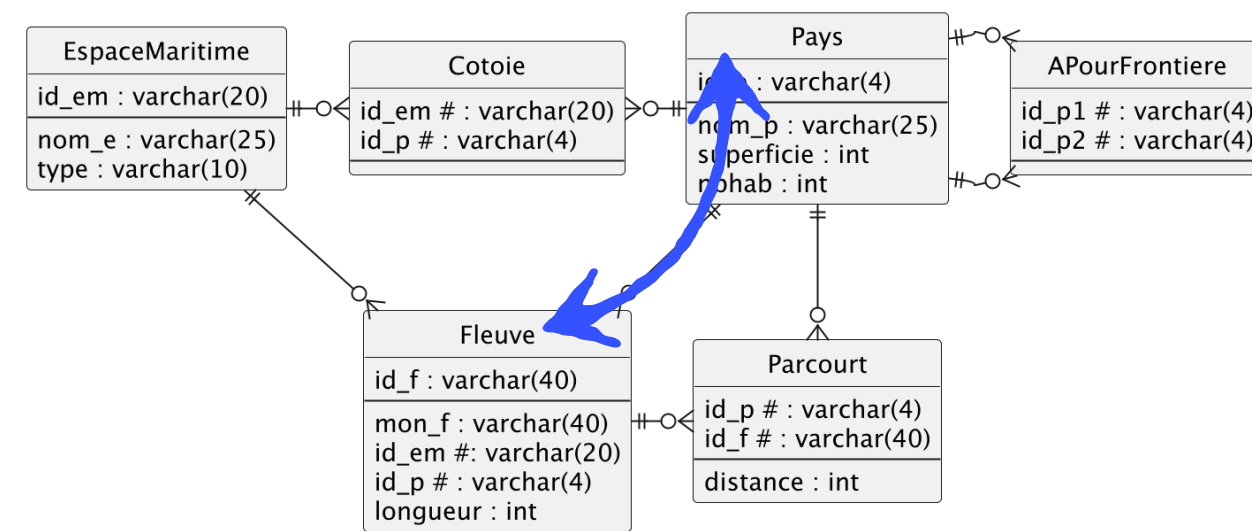
Blue arrow with a green checkmark points to the first line of the query. A red 'X' is placed over the second line of the query.

```
Select nom_p
From Fleuve Natural Join Parcours Natural Join Pays
Where nom_f = 'Rhone' ;
```

```
Select nom_p
From Fleuve as f Inner Join Parcours as pa On f.id_f=pa.id_f and f.id_p=pa.id_p
Inner Join Pays as p On p.id_p=pa.id_p
Where nom_f = 'Rhone' ;
```

Blue arrow with a green checkmark points to the first line of the query. The condition `and f.id_p=pa.id_p` in the second line is highlighted in pink.

Cas de jointures externes



- Permet de conserver les tuples de la table de gauche ($R \cdot \bowtie_Q S$), de droite ($R \bowtie_Q \cdot S$) ou des deux tables ($R \cdot \bowtie_Q \cdot S$).

Structure plus complexe, 3 variantes, présence du Null (logique tri-valuée, calculs un peu différents)
→ A éviter en NSI !

- Utilisée souvent pour :
 - Des calculs sur des regroupements (permet un Count à 0 par exemple)
 - Une des solutions pour traiter une négation.
Exemple : « les pays qui n'ont pas de source »

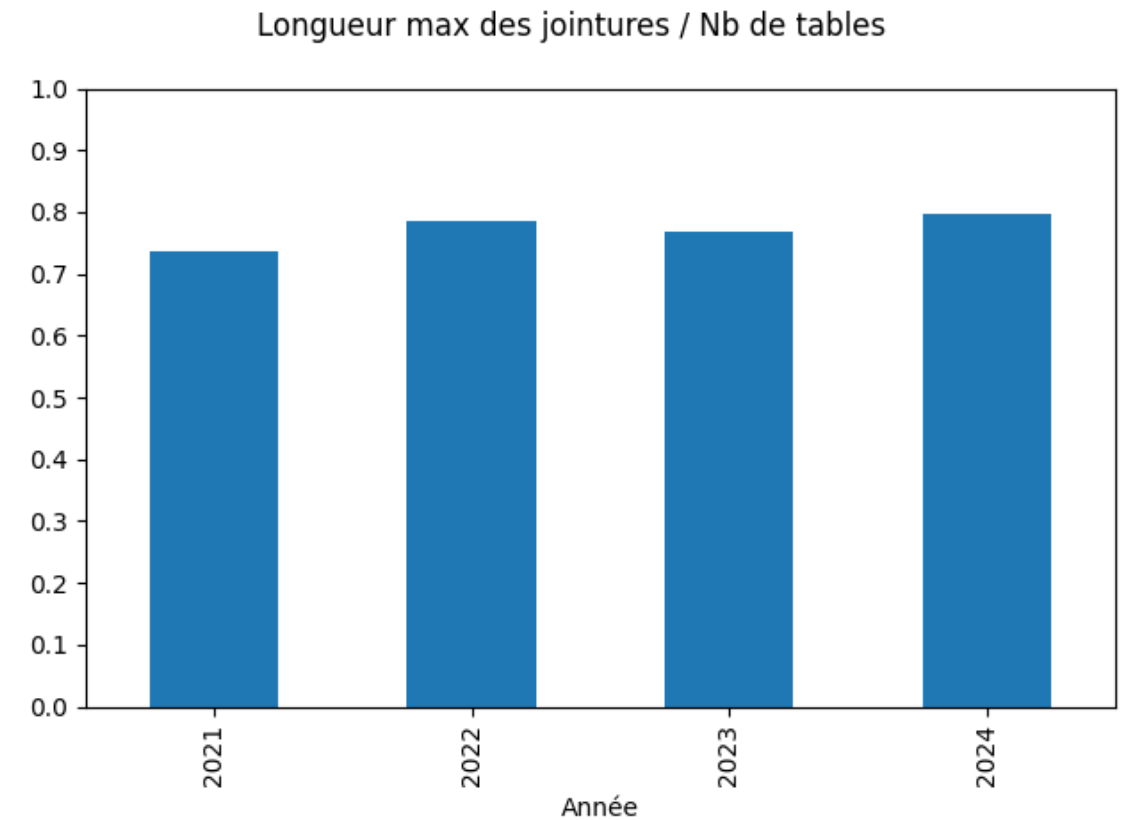
$\pi_{id_p, nom_p}(\sigma_{id_f=NULL}(Fleuve \bowtie_{F.id_p=P.id_p} \cdot Pays))$

```
Select id_p, nom_p
From Fleuve Right Outer Join Pays Using(id_p)
Where id_f is Null ;
```

	id_p	nom_p	id_f
59	GB	United Kingdom	river Trent
60	GB	United Kingdom	river Clyde
61	GB	United Kingdom	river Ness
62	EW	Estonia	<null>
63	SK	Slovakia	<null>

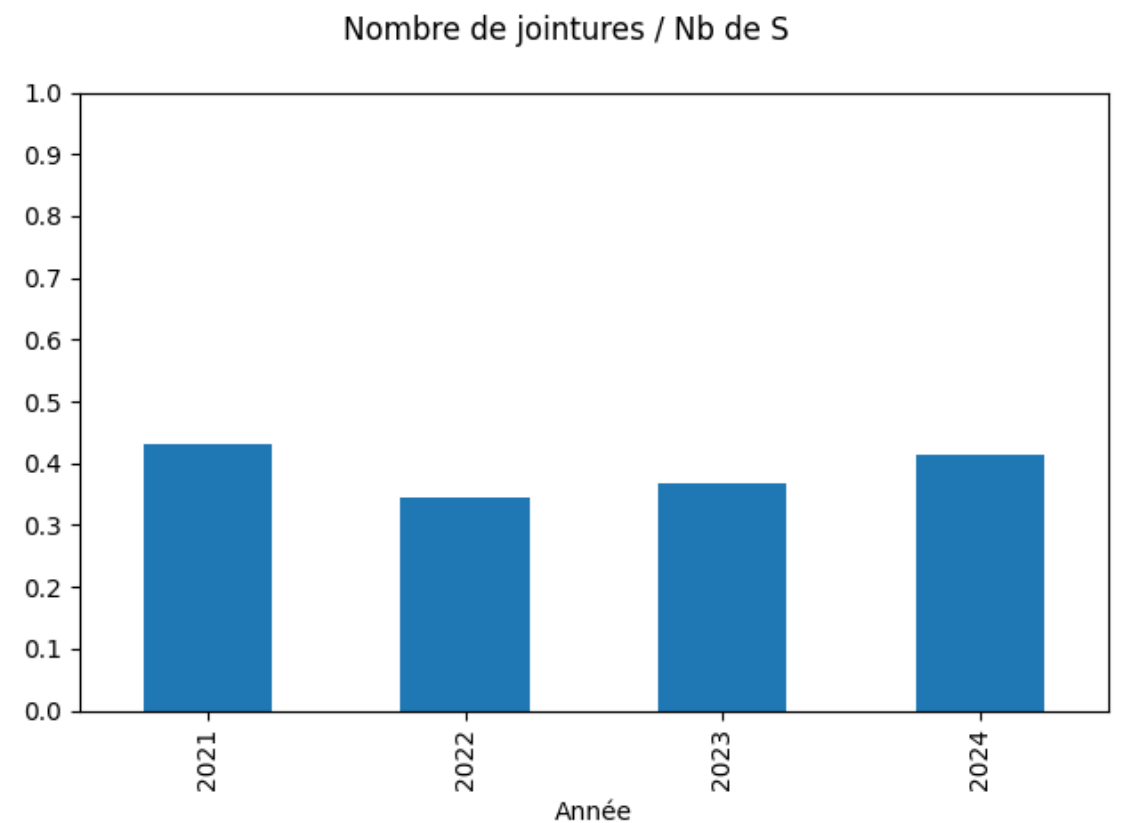
Et au bac ?

- 63 modèles pour 194 tables (plus ou moins biens présentées) : entre 1 et 6 tables par exercice (3 en moyenne)
- 89 requêtes de jointure !
Soit 25% des requêtes
- En moyenne, 1.41 jointures par exercice (entre 0 et 4 requêtes)
- Longueur : entre 2 et 4 tables (2.5 en moyenne)



Et au bac ?

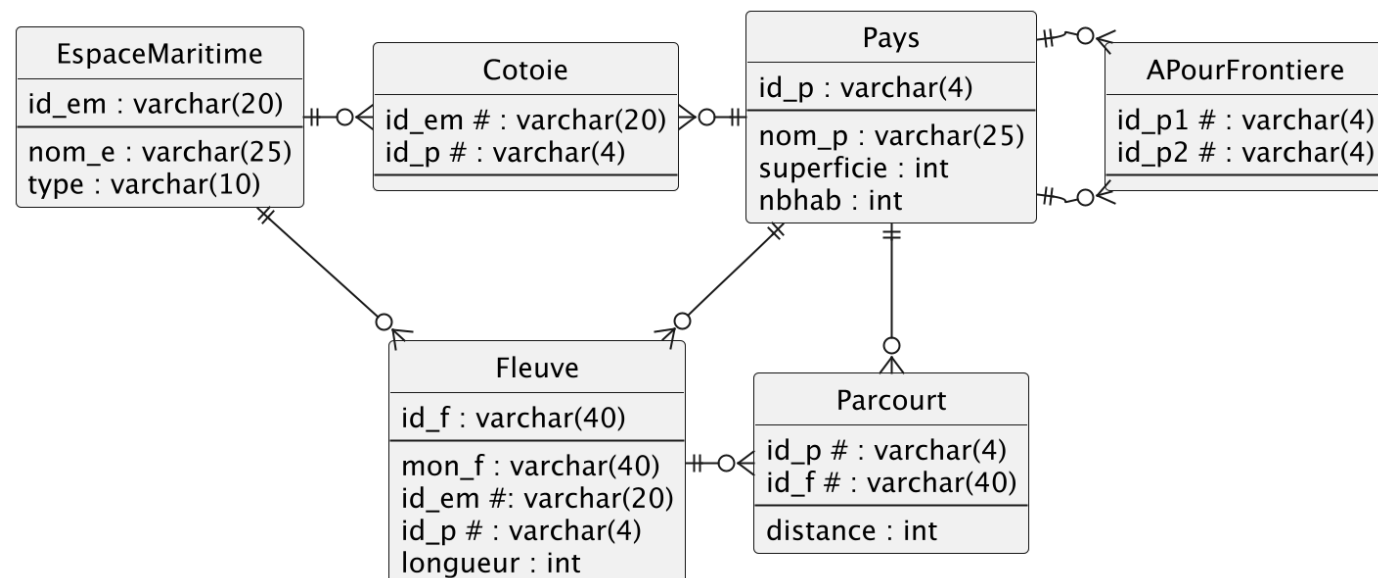
- 38% des requêtes S (2/5)
- Uniquement « ... [Inner] Join .. On ... » dans les aides ou les requêtes $l \rightarrow R$.
- Uniquement des jointures de clés !



Vers une méthode de construction de requête ?

- Méthode pour construire une requête (niveau Bac)
 1. Identifier les informations dont on a besoin (projections, filtres...),
 2. Proposer une chaîne de jointures de clés pour connecter ces informations,
 3. Mettre en place l'expression logique de sélection,
 4. Définir les attributs à présenter (projection) et les fonctions d'agrégation,
 5. Se poser la question du Distinct,
 6. Se poser la question du tri et du limit.

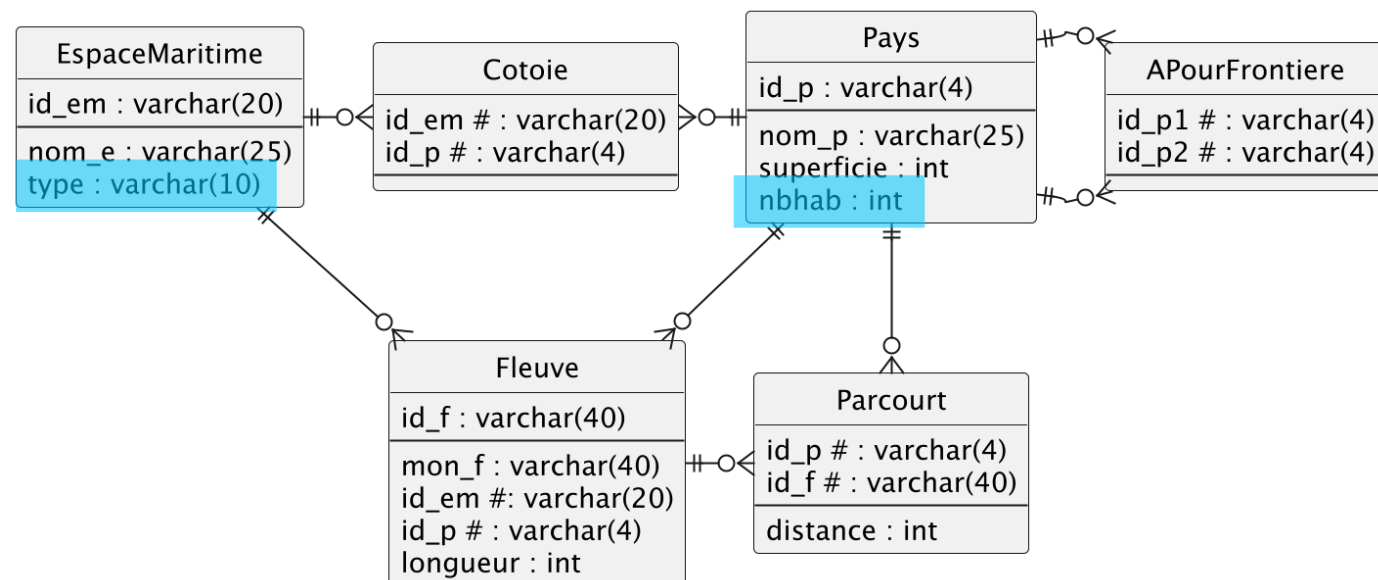
Donner les types d'espace maritime dans lequel se jettent les fleuves qui prennent leur source dans un pays de plus de 60 millions d'habitants.



Vers une méthode de construction de requête ?

- Méthode pour construire une requête (niveau Bac)
 1. Identifier les informations dont on a besoin (projections, filtres...),
 2. Proposer une chaîne de jointures de clés pour connecter ces informations,
 3. Mettre en place l'expression logique de sélection,
 4. Définir les attributs à présenter (projection) et les fonctions d'agrégation,
 5. Se poser la question du Distinct,
 6. Se poser la question du tri et du limit.

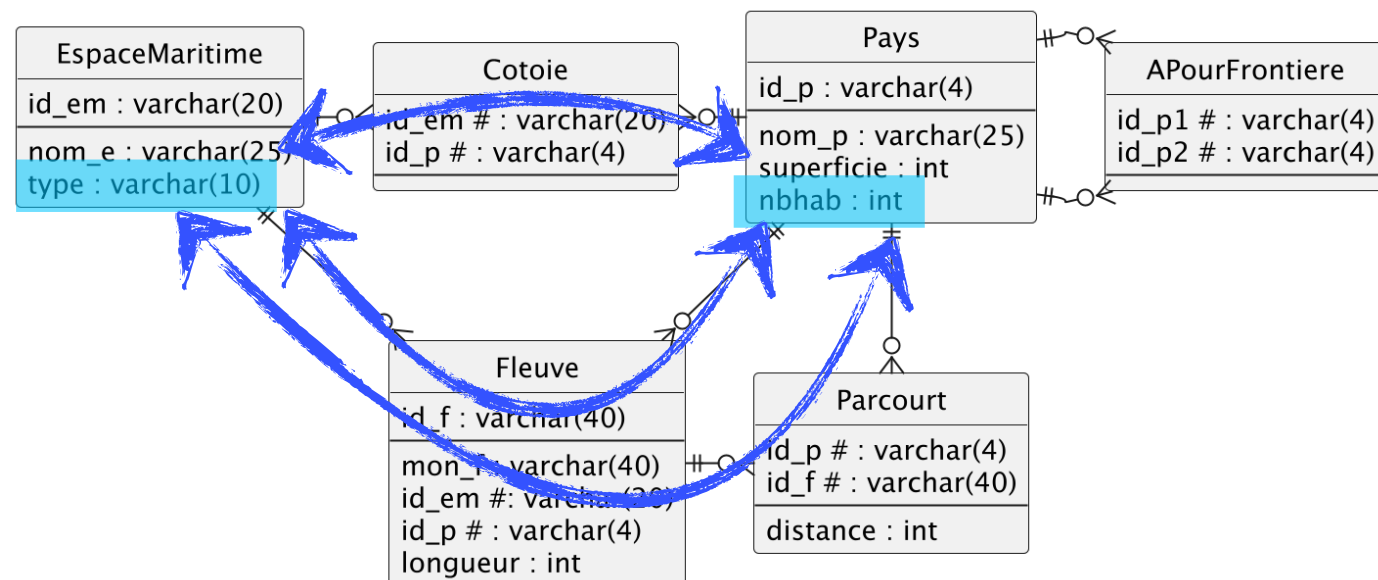
Donner les types d'espace maritime dans lequel se jettent les fleuves qui prennent leur source dans un pays de plus de 60 millions d'habitants.



Vers une méthode de construction de requête ?

- Méthode pour construire une requête (niveau Bac)
 1. Identifier les informations dont on a besoin (projections, filtres...),
 2. Proposer une chaîne de jointures de clés pour connecter ces informations,
 3. Mettre en place l'expression logique de sélection,
 4. Définir les attributs à présenter (projection) et les fonctions d'agrégation,
 5. Se poser la question du Distinct,
 6. Se poser la question du tri et du limit.

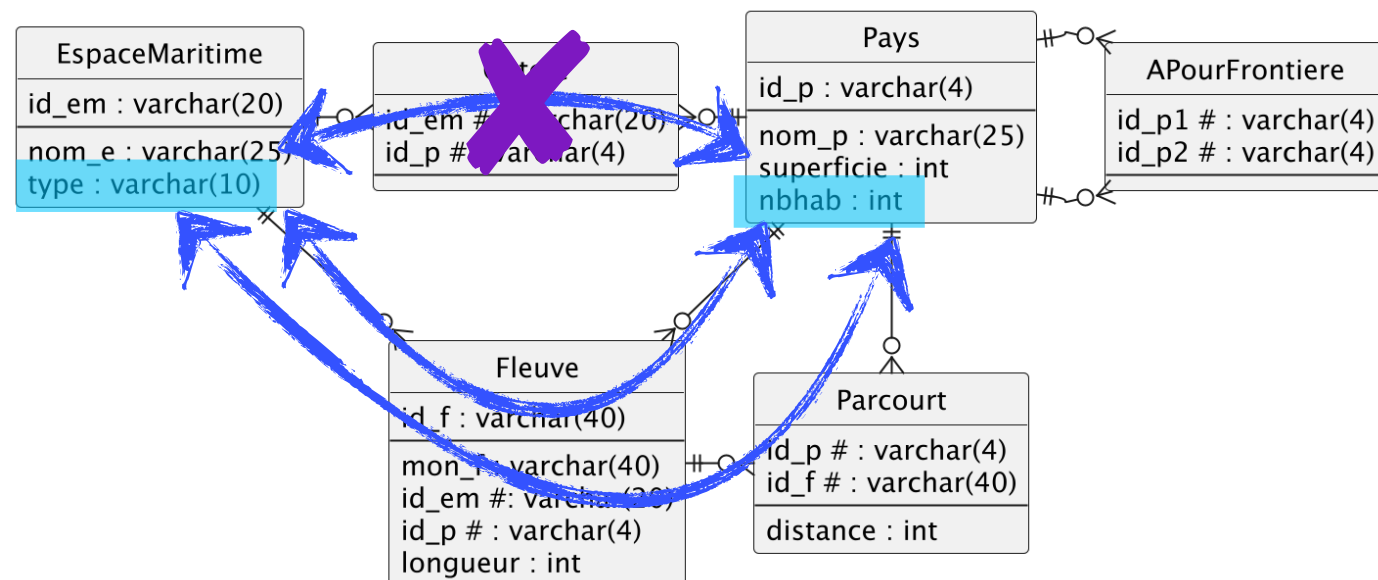
Donner les types d'espace maritime dans lequel se jettent les fleuves qui prennent leur source dans un pays de plus de 60 millions d'habitants.



Vers une méthode de construction de requête ?

- Méthode pour construire une requête (niveau Bac)
 1. Identifier les informations dont on a besoin (projections, filtres...),
 2. Proposer une chaîne de jointures de clés pour connecter ces informations,
 3. Mettre en place l'expression logique de sélection,
 4. Définir les attributs à présenter (projection) et les fonctions d'agrégation,
 5. Se poser la question du Distinct,
 6. Se poser la question du tri et du limit.

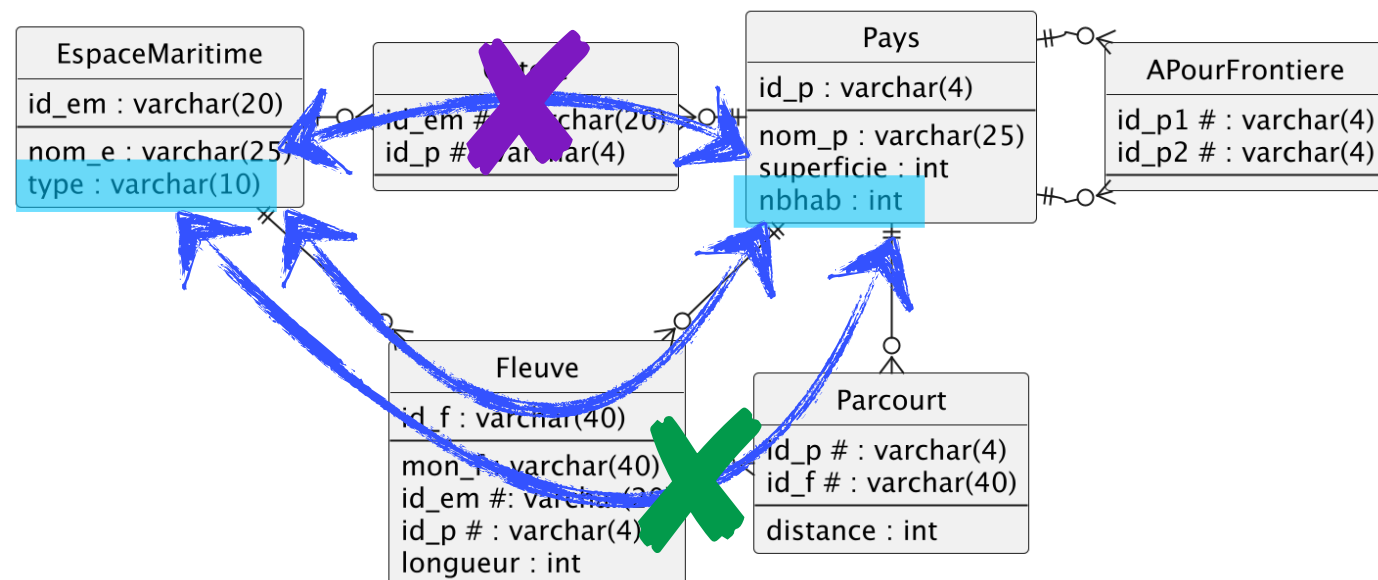
Donner les types d'espace maritime dans lequel se jettent les fleuves qui prennent leur source dans un pays de plus de 60 millions d'habitants.



Vers une méthode de construction de requête ?

- Méthode pour construire une requête (niveau Bac)
 1. Identifier les informations dont on a besoin (projections, filtres...),
 2. Proposer une chaîne de jointures de clés pour connecter ces informations,
 3. Mettre en place l'expression logique de sélection,
 4. Définir les attributs à présenter (projection) et les fonctions d'agrégation,
 5. Se poser la question du Distinct,
 6. Se poser la question du tri et du limit.

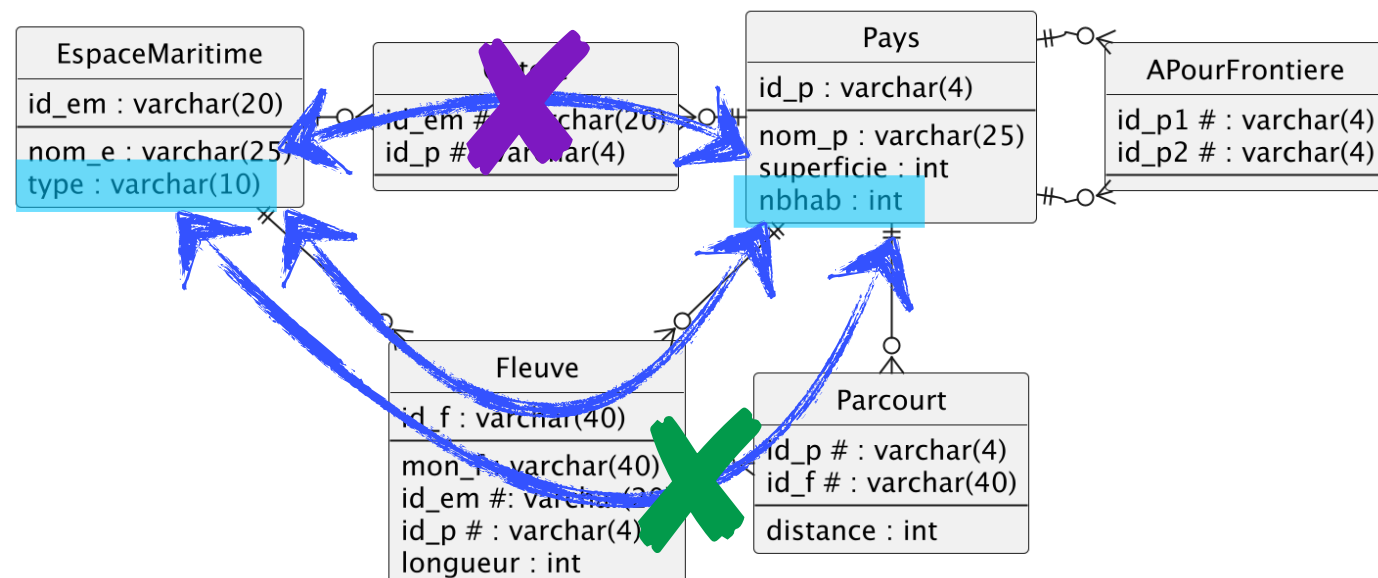
Donner les types d'espace maritime dans lequel se jettent les fleuves qui prennent leur source dans un pays de plus de 60 millions d'habitants.



Vers une méthode de construction de requête ?

- Méthode pour construire une requête (niveau Bac)
 1. Identifier les informations dont on a besoin (projections, filtres...),
 2. Proposer une chaîne de jointures de clés pour connecter ces informations,
 3. Mettre en place l'expression logique de sélection,
 4. Définir les attributs à présenter (projection) et les fonctions d'agrégation,
 5. Se poser la question du Distinct,
 6. Se poser la question du tri et du limit.

Donner les types d'espace maritime dans lequel se jettent les fleuves qui prennent leur source dans un pays de plus de 60 millions d'habitants.

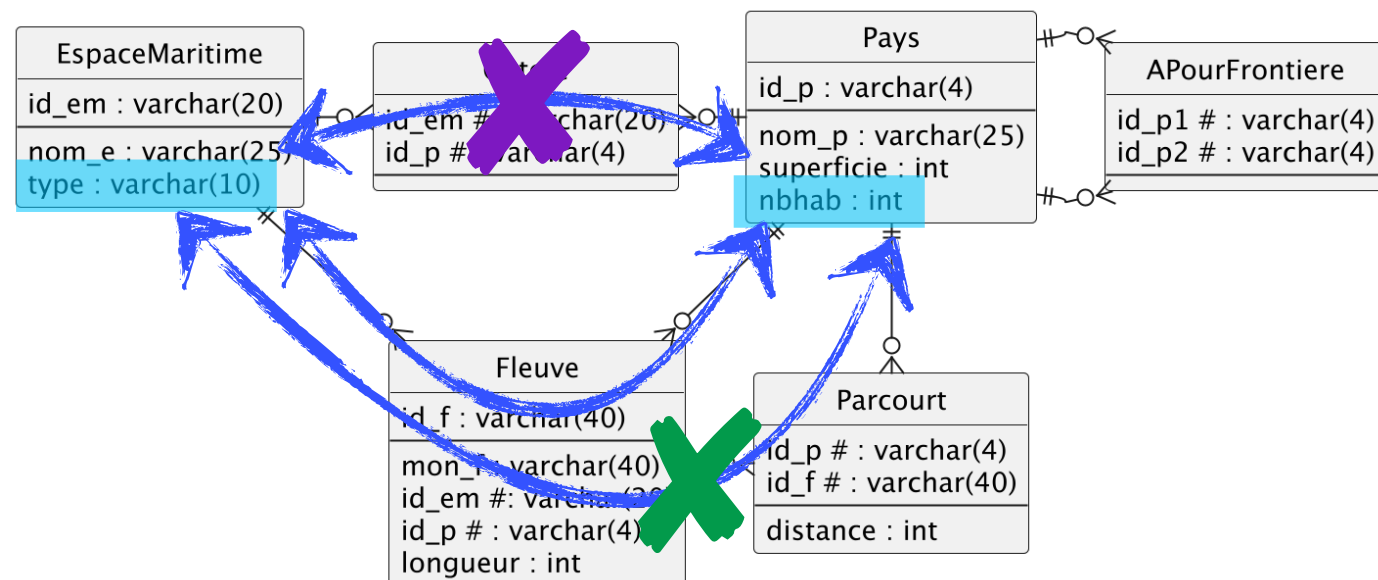


```
Select * -- Projection ?
From Pays Inner Join Fleuve Using(id_p)
      Inner Join EspaceMaritime Using(id_em)
-- Sélection ?
```

Vers une méthode de construction de requête ?

- Méthode pour construire une requête (niveau Bac)
 1. Identifier les informations dont on a besoin (projections, filtres...),
 2. Proposer une chaîne de jointures de clés pour connecter ces informations,
 3. Mettre en place l'expression logique de sélection,
 4. Définir les attributs à présenter (projection) et les fonctions d'agrégation,
 5. Se poser la question du Distinct,
 6. Se poser la question du tri et du limit.

Donner les types d'espace maritime dans lequel se jettent les fleuves qui prennent leur source dans un pays de plus de 60 millions d'habitants.



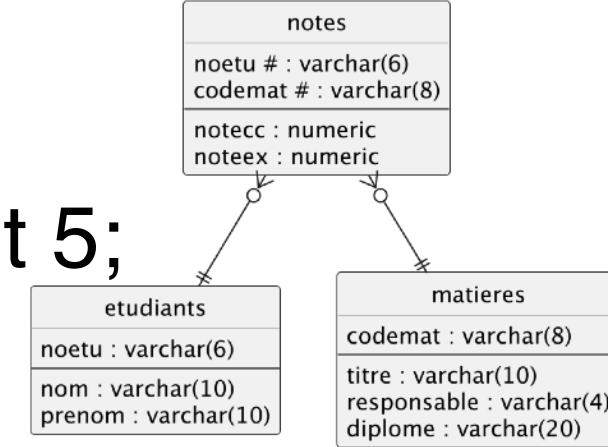
```
Select Distinct type
From Pays Inner Join Fleuve Using(id_p)
      Inner Join EspaceMaritime Using(id_em)
Where nbHab > 600000 ;
```

Vers du « pas à pas » ?

Pas à pas ?

- Constat : absence d'outil pour détailler l'exécution d'une requête SQL
- Défi : faire ça... avec des requêtes SQL !
- 2 difficultés identifiées :
 - Le « From » et les jointures : nécessité d'un arbre syntaxique
 - Le « Group By ... Having ... » : comment faire un regroupement sans « Group by » (ou presque) ?
- 1 difficulté « surprise » : la gestion des SGBD (identification des tables, des attributs, des clés...)
 - PostgreSQL : 😊
 - MySQL : 😊
 - SQLite : 😞
- Python 3
 - Plusieurs bibliothèques
 - GitLab : <https://gitlab.univ-nantes.fr/ls2n-didactique/querycraft>

select noetu, codemat, noteex
 from etudiants join notes using(noetu) order by noteex limit 5;

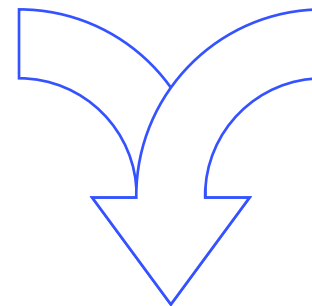


```
select noetu, codemat, noteex FROM #etudiants# JOIN
notes USING (noetu) ORDER BY noteex LIMIT 5 ;
```

etudiants.noetu	etudiants.nom	etudiants.prenom
28936E	Dupont	Franck
46283B	Dupont	Isabelle
86719E	Martin	Adrien
99628C	Robert	Adrien
99321C	Denou	Michelle
99322C	Dupont	Isabelle

```
select noetu, codemat, noteex FROM etudiants JOIN
#notes# USING (noetu) ORDER BY noteex LIMIT 5 ;
```

notes.noetu	notes.codemat	notes.noteex	notes.notecc
99628C	MIAS2I5	12.0	15.5
46283B	MIAS2I5	9.5	2.0
86719E	IUP2MA	12.0	5.5
99321C	LIL6	18.0	16.5
28936E	MIAS2I5	13.5	13.5
86719E	IUP2IS	8.5	10.0
99321C	LIL5	15.0	14.5
99322C	MIAS2I5	12.0	15.5
46283B	MIAS2I6	8.0	12.0
99628C	MIAS2I6	3.0	7.0
28936E	MIAS2I6	12.0	null



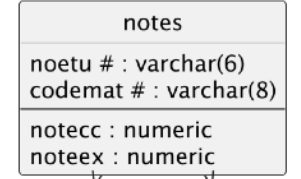
```
select noetu, codemat, noteex FROM #etudiants JOIN notes USING (noetu)#
ORDER BY noteex LIMIT 5 ;
```

etudiants.noetu	etudiants.nom	etudiants.prenom	notes.codemat	notes.noteex	notes.notecc
99628C	Robert	Adrien	MIAS2I5	12.0	15.5
46283B	Dupont	Isabelle	MIAS2I5	9.5	2.0
86719E	Martin	Adrien	IUP2MA	12.0	5.5
99321C	Denou	Michelle	LIL6	18.0	16.5
28936E	Dupont	Franck	MIAS2I5	13.5	13.5
86719E	Martin	Adrien	IUP2IS	8.5	10.0
99321C	Denou	Michelle	LIL5	15.0	14.5
99322C	Dupont	Isabelle	MIAS2I5	12.0	15.5
46283B	Dupont	Isabelle	MIAS2I6	8.0	12.0
99628C	Robert	Adrien	MIAS2I6	3.0	7.0
28936E	Dupont	Franck	MIAS2I6	12.0	null



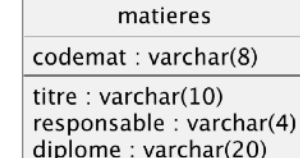
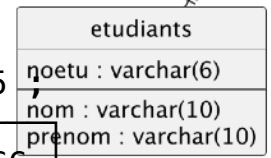
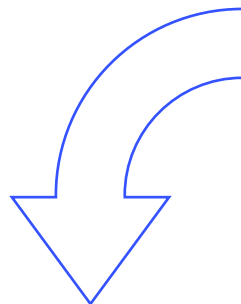
<https://codimd.math.cnrs.fr/s/OJX0u8dFC>

select noetu, codemat, noteex from etudiants join notes using(noetu) order by noteex limit 5;



```
select noetu, codemat, noteex FROM #etudiants JOIN notes USING (noetu)# ORDER BY noteex LIMIT 5 ;
```

etudiants.noetu	etudiants.nom	etudiants.prenom	notes.codemat	notes.noteex	notes.notecc
99628C	Robert	Adrien	MIAS2I5	12.0	15.5
46283B	Dupont	Isabelle	MIAS2I5	9.5	2.0
86719E	Martin	Adrien	IUP2MA	12.0	5.5
99321C	Denou	Michelle	LIL6	18.0	16.5
28936E	Dupont	Franck	MIAS2I5	13.5	13.5
86719E	Martin	Adrien	IUP2IS	8.5	10.0
99321C	Denou	Michelle	LIL5	15.0	14.5
99322C	Dupont	Isabelle	MIAS2I5	12.0	15.5
46283B	Dupont	Isabelle	MIAS2I6	8.0	12.0
99628C	Robert	Adrien	MIAS2I6	3.0	7.0
28936E	Dupont	Franck	MIAS2I6	12.0	null

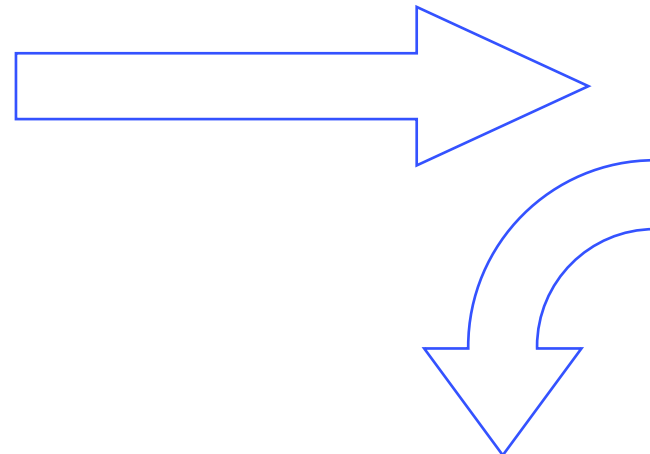


```
select #noetu, codemat, noteex FROM etudiants JOIN notes USING (noetu)# ORDER BY noteex LIMIT 5 ;
```

etudiants.noetu	notes.codemat	notes.noteex
99628C	MIAS2I5	12.0
46283B	MIAS2I5	9.5
86719E	IUP2MA	12.0
99321C	LIL6	18.0
28936E	MIAS2I5	13.5
86719E	IUP2IS	8.5
99321C	LIL5	15.0
99322C	MIAS2I5	12.0
46283B	MIAS2I6	8.0
99628C	MIAS2I6	3.0
28936E	MIAS2I6	12.0

```
select #noetu, codemat, noteex FROM etudiants JOIN notes USING (noetu) ORDER BY noteex# LIMIT 5 ;
```

etudiants.noetu	notes.codemat	notes.noteex
99628C	MIAS2I6	3.0
46283B	MIAS2I6	8.0
86719E	IUP2IS	8.5
46283B	MIAS2I5	9.5
28936E	MIAS2I6	12.0
86719E	IUP2MA	12.0
99322C	MIAS2I5	12.0
99628C	MIAS2I5	12.0
28936E	MIAS2I5	13.5
99321C	LIL5	15.0
99321C	LIL6	18.0



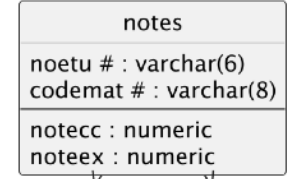
```
select #noetu, codemat, noteex FROM etudiants JOIN notes USING (noetu) ORDER BY noteex LIMIT 5# ;
```

etudiants.noetu	notes.codemat	notes.noteex
99628C	MIAS2I6	3.0
46283B	MIAS2I6	8.0
86719E	IUP2IS	8.5
46283B	MIAS2I5	9.5
86719E	IUP2MA	12.0



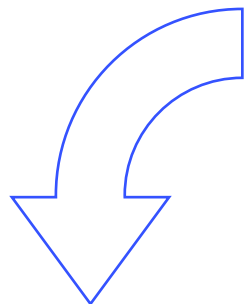
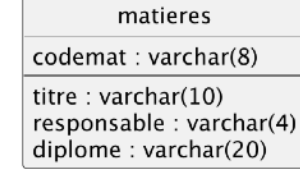
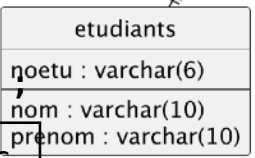
<https://codimd.math.cnrs.fr/s/OJX0u8dFC>

select noetu, codemat, noteex from etudiants join notes using(noetu) order by noteex limit 5;



select noetu, codemat, noteex FROM #etudiants JOIN notes USING (noetu)# ORDER BY noteex LIMIT 5

etudiants.noetu	etudiants.nom	etudiants.prenom	notes.codemat	notes.noteex	notes.notecc
99628C	Robert	Adrien	MIAS2I5	12.0	15.5
46283B	Dupont	Isabelle	MIAS2I5	9.5	2.0
86719E	Martin	Adrien	IUP2MA	12.0	5.5
99321C	Denou	Michelle	LIL6	18.0	16.5
28936E	Dupont	Franck	MIAS2I5	13.5	13.5
86719E	Martin	Adrien	IUP2IS	8.5	10.0
99321C	Denou	Michelle	LIL5	15.0	14.5
99322C	Dupont	Isabelle	MIAS2I5	12.0	15.5
46283B	Dupont	Isabelle	MIAS2I6	8.0	12.0
99628C	Robert	Adrien	MIAS2I6	3.0	7.0
28936E	Dupont	Franck	MIAS2I6	12.0	null



select #noetu, codemat, noteex FROM etudiants JOIN notes USING (noetu)# ORDER BY noteex LIMIT 5 ;

etudiants.noetu	notes.codemat	notes.noteex
99628C	MIAS2I5	12.0
46283B	MIAS2I5	9.5
86719E	IUP2MA	12.0
99321C	LIL6	18.0
28936E	MIAS2I5	13.5
86719E	IUP2IS	8.5
99321C	LIL5	15.0
99322C	MIAS2I5	12.0
46283B	MIAS2I6	8.0
99628C	MIAS2I6	3.0
28936E	MIAS2I6	12.0

select #noetu, codemat, noteex FROM etudiants JOIN notes USING (noetu) ORDER BY noteex# LIMIT 5 ;

etudiants.noetu	notes.codemat	notes.noteex
99628C	MIAS2I6	3.0
46283B	MIAS2I6	8.0



```

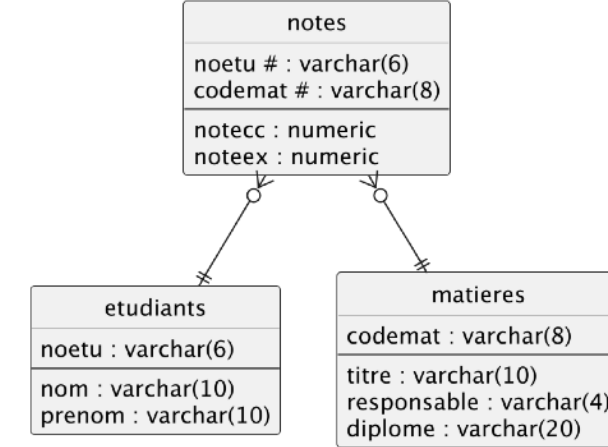
Select * From etudiants ;
Select * From notes ;
Select * FROM etudiants JOIN notes USING (noetu) ;
select noetu, codemat, noteex FROM etudiants JOIN notes USING (noetu) ;
select noetu, codemat, noteex FROM etudiants JOIN notes USING (noetu) ORDER BY noteex ;
select noetu, codemat, noteex FROM etudiants JOIN notes USING (noetu) ORDER BY noteex LIMIT 5 ;
  
```

46283B	MIAS2I6	8.0
86719E	IUP2IS	8.5
46283B	MIAS2I5	9.5
86719E	IUP2MA	12.0



<https://codimd.math.cnrs.fr/s/0JX0u8dFC>

SELECT noetu, AVG(noteex) FROM etudiants JOIN notes USING (noetu)
WHERE codemat LIKE 'MIAS2I%' GROUP BY noetu HAVING AVG(noteex) > 8

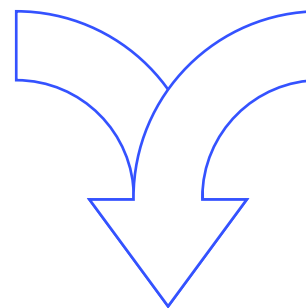


select noetu, AVG(noteex) FROM #etudiants# JOIN notes
USING (noetu) WHERE codemat LIKE 'MIAS2I%'
Group By etudiants.noetu, notes.noetu HAVING AVG(noteex) > 8 ;

etudiants.noetu	etudiants.nom	etudiants.prenom
28936E	Dupont	Franck
46283B	Dupont	Isabelle
86719E	Martin	Adrien
99628C	Robert	Adrien
99321C	Denou	Michelle
99322C	Dupont	Isabelle

select noetu, AVG(noteex) FROM etudiants JOIN #notes#
USING (noetu) WHERE codemat LIKE 'MIAS2I%'
Group By etudiants.noetu, notes.noetu HAVING AVG(noteex) > 8 ;

notes.noetu	notes.codemat	notes.noteex	notes.notecc
99628C	MIAS2I5	12.0	15.5
46283B	MIAS2I5	9.5	2.0
86719E	IUP2MA	12.0	5.5
99321C	LIL6	18.0	16.5
28936E	MIAS2I5	13.5	13.5
86719E	IUP2IS	8.5	10.0
99321C	LIL5	15.0	14.5
99322C	MIAS2I5	12.0	15.5
46283B	MIAS2I6	8.0	12.0
99628C	MIAS2I6	3.0	7.0
28936E	MIAS2I6	12.0	null



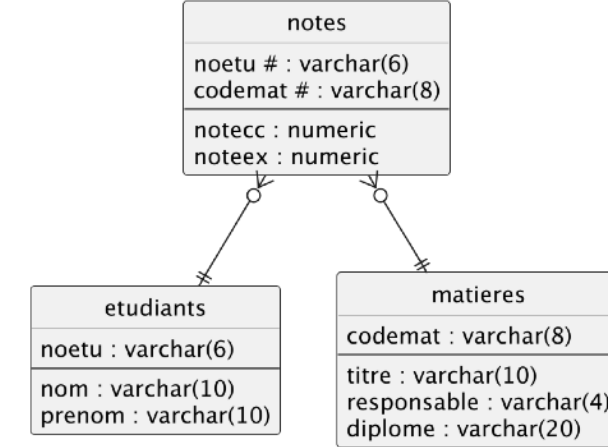
select noetu, AVG(noteex) FROM #etudiants JOIN notes USING (noetu)#
WHERE codemat LIKE 'MIAS2I%' Group By etudiants.noetu, notes.noetu HAVING AVG(noteex) > 8 ;

etudiants.noetu	etudiants.nom	etudiants.prenom	notes.codemat	notes.noteex	notes.notecc
99628C	Robert	Adrien	MIAS2I5	12.0	15.5
46283B	Dupont	Isabelle	MIAS2I5	9.5	2.0
86719E	Martin	Adrien	IUP2MA	12.0	5.5
99321C	Denou	Michelle	LIL6	18.0	16.5
28936E	Dupont	Franck	MIAS2I5	13.5	13.5
86719E	Martin	Adrien	IUP2IS	8.5	10.0
99321C	Denou	Michelle	LIL5	15.0	14.5
99322C	Dupont	Isabelle	MIAS2I5	12.0	15.5
46283B	Dupont	Isabelle	MIAS2I6	8.0	12.0
99628C	Robert	Adrien	MIAS2I6	3.0	7.0
28936E	Dupont	Franck	MIAS2I6	12.0	null



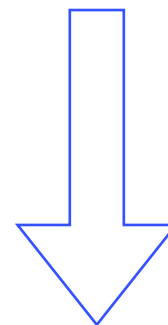
<https://codimd.math.cnrs.fr/s/0JX0u8dFC>

SELECT noetu, AVG(noteex) FROM etudiants JOIN notes USING (noetu)
 WHERE codemat LIKE 'MIAS2I%' GROUP BY noetu HAVING AVG(noteex) > 8



select noetu, AVG(noteex) FROM #etudiants JOIN notes USING (noetu)#
 WHERE codemat LIKE 'MIAS2I%' Group By etudiants.noetu, notes.noetu HAVING AVG(noteex) > 8 ;

etudiants.noetu	etudiants.nom	etudiants.prenom	notes.codemat	notes.noteex	notes.notecc
99628C	Robert	Adrien	MIAS2I5	12.0	15.5
46283B	Dupont	Isabelle	MIAS2I5	9.5	2.0
86719E	Martin	Adrien	IUP2MA	12.0	5.5
99321C	Denou	Michelle	LIL6	18.0	16.5
28936E	Dupont	Franck	MIAS2I5	13.5	13.5
86719E	Martin	Adrien	IUP2IS	8.5	10.0
99321C	Denou	Michelle	LIL5	15.0	14.5
99322C	Dupont	Isabelle	MIAS2I5	12.0	15.5
46283B	Dupont	Isabelle	MIAS2I6	8.0	12.0
99628C	Robert	Adrien	MIAS2I6	3.0	7.0
28936E	Dupont	Franck	MIAS2I6	12.0	null



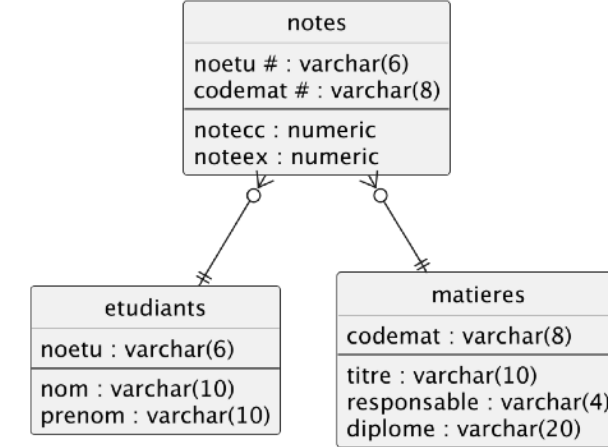
select noetu, AVG(noteex) # FROM etudiants JOIN notes USING (noetu) WHERE codemat LIKE 'MIAS2I%'#
 Group By etudiants.noetu, notes.noetu HAVING AVG(noteex) > 8 ;

etudiants.noetu	etudiants.nom	etudiants.prenom	notes.codemat	notes.noteex	notes.notecc
99628C	Robert	Adrien	MIAS2I5	12.0	15.5
46283B	Dupont	Isabelle	MIAS2I5	9.5	2.0
28936E	Dupont	Franck	MIAS2I5	13.5	13.5
99322C	Dupont	Isabelle	MIAS2I5	12.0	15.5
46283B	Dupont	Isabelle	MIAS2I6	8.0	12.0
99628C	Robert	Adrien	MIAS2I6	3.0	7.0
28936E	Dupont	Franck	MIAS2I6	12.0	null



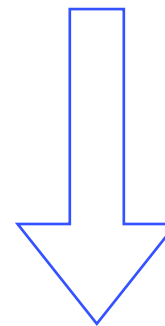
<https://codimd.math.cnrs.fr/s/OJX0u8dFC>

SELECT noetu, AVG(noteex) FROM etudiants JOIN notes USING (noetu) WHERE codemat LIKE 'MIAS2I%' GROUP BY noetu HAVING AVG(noteex) > 8



`select noetu, AVG(noteex) # FROM etudiants JOIN notes USING (noetu) WHERE codemat LIKE 'MIAS2I%'# Group By etudiants.noetu, notes.noetu HAVING AVG(noteex) > 8 ;`

etudiants.noetu	etudiants.nom	etudiants.prenom	notes.codemat	notes.noteex	notes.notecc
99628C	Robert	Adrien	MIAS2I5	12.0	15.5
46283B	Dupont	Isabelle	MIAS2I5	9.5	2.0
28936E	Dupont	Franck	MIAS2I5	13.5	13.5
99322C	Dupont	Isabelle	MIAS2I5	12.0	15.5
46283B	Dupont	Isabelle	MIAS2I6	8.0	12.0
99628C	Robert	Adrien	MIAS2I6	3.0	7.0
28936E	Dupont	Franck	MIAS2I6	12.0	null



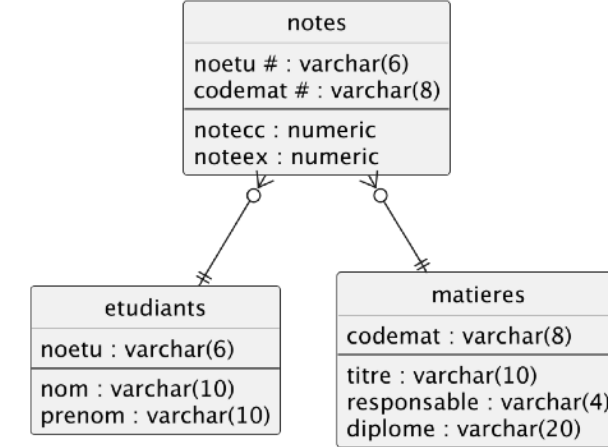
`select noetu, AVG(noteex) # FROM etudiants JOIN notes USING (noetu) WHERE codemat LIKE 'MIAS2I%' Group By etudiants.noetu, notes.noetu# HAVING AVG(noteex) > 8 ;`

etudiants.noetu	notes.noetu	etudiants.nom	etudiants.prenom	notes.codemat	notes.notecc	notes.noteex
28936E	28936E	["Dupont", "Dupont"]	["Franck", "Franck"]	["MIAS2I5", "MIAS2I6"]	[13.5, null]	[13.5, 12.0]
46283B	46283B	["Dupont", "Dupont"]	["Isabelle", "Isabelle"]	["MIAS2I6", "MIAS2I5"]	[12.0, 2.0]	[8.0, 9.5]
99322C	99322C	["Dupont"]	["Isabelle"]	["MIAS2I5"]	[15.5]	[12.0]
99628C	99628C	["Robert", "Robert"]	["Adrien", "Adrien"]	["MIAS2I6", "MIAS2I5"]	[7.0, 15.5]	[3.0, 12.0]



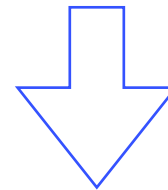
<https://codimd.math.cnrs.fr/s/0JX0u8dFC>

SELECT noetu, AVG(noteex) FROM etudiants JOIN notes USING (noetu) WHERE codemat LIKE 'MIAS2I%' GROUP BY noetu HAVING AVG(noteex) > 8



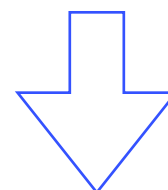
`select noetu, AVG(noteex) # FROM etudiants JOIN notes USING (noetu) WHERE codemat LIKE 'MIAS2I%' Group By etudiants.noetu, notes.noetu# HAVING AVG(noteex) > 8 ;`

etudiants.noetu	notes.noetu	etudiants.nom	etudiants.prenom	notes.codemat	notes.notecc	notes.noteex
28936E	28936E	["Dupont", "Dupont"]	["Franck", "Franck"]	["MIAS2I5", "MIAS2I6"]	[13.5, null]	[13.5, 12.0]
46283B	46283B	["Dupont", "Dupont"]	["Isabelle", "Isabelle"]	["MIAS2I6", "MIAS2I5"]	[12.0, 2.0]	[8.0, 9.5]
99322C	99322C	["Dupont"]	["Isabelle"]	["MIAS2I5"]	[15.5]	[12.0]
99628C	99628C	["Robert", "Robert"]	["Adrien", "Adrien"]	["MIAS2I6", "MIAS2I5"]	[7.0, 15.5]	[3.0, 12.0]



`select noetu, AVG(noteex) # FROM etudiants JOIN notes USING (noetu) WHERE codemat LIKE 'MIAS2I%' Group By etudiants.noetu, notes.noetu HAVING AVG(noteex) > 8# ;`

etudiants.noetu	notes.noetu	etudiants.nom	etudiants.prenom	notes.codemat	notes.notecc	notes.noteex
28936E	28936E	["Dupont", "Dupont"]	["Franck", "Franck"]	["MIAS2I5", "MIAS2I6"]	[13.5, null]	[13.5, 12.0]
46283B	46283B	["Dupont", "Dupont"]	["Isabelle", "Isabelle"]	["MIAS2I6", "MIAS2I5"]	[12.0, 2.0]	[8.0, 9.5]
99322C	99322C	["Dupont"]	["Isabelle"]	["MIAS2I5"]	[15.5]	[12.0]



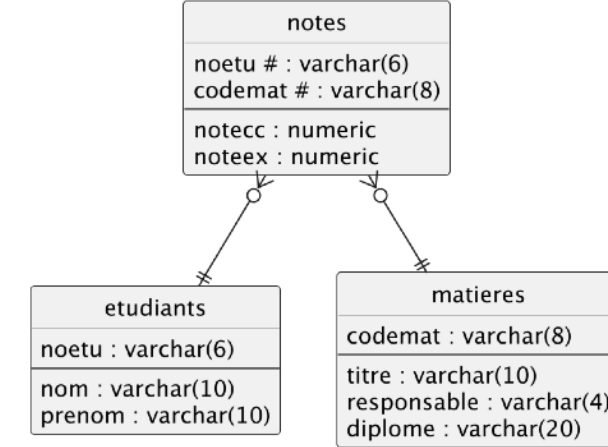
`select #noetu, AVG(noteex) FROM etudiants JOIN notes USING (noetu) WHERE codemat LIKE 'MIAS2I%' Group By etudiants.noetu, notes.noetu HAVING AVG(noteex) > 8# ;`

etudiants.noetu	avg
28936E	12.7500000000000000
46283B	8.7500000000000000
99322C	12.0000000000000000



<https://codimd.math.cnrs.fr/s/OJX0u8dFC>

SELECT noetu, AVG(noteex) FROM etudiants JOIN notes USING (noetu)
WHERE codemat LIKE 'MIAS2I%' GROUP BY noetu HAVING AVG(noteex) > 8



select noetu, AVG(noteex) # FROM etudiants JOIN notes USING (noetu) WHERE codemat LIKE 'MIAS2I%' Group By etudiants.noetu, notes.noetu#
HAVING AVG(noteex) > 8 ;

```

Select * From etudiants ;

Select * From notes ;

Select * FROM etudiants JOIN notes USING (noetu) ;

select noetu, codemat, noteex FROM etudiants JOIN notes USING (noetu) ;

select * FROM etudiants JOIN notes USING (noetu) WHERE codemat LIKE 'MIAS2I%' ;

select etudiants.noetu,etudiants.nom,etudiants.prenom,notes.codemat,notes.noetu,notes.notecc,notes.noteex
FROM etudiants JOIN notes USING (noetu)
WHERE codemat LIKE 'MIAS2I%' Order By etudiants.noetu, notes.noetu ;

select etudiants.noetu,etudiants.nom,etudiants.prenom,notes.codemat,notes.noetu,notes.notecc,notes.noteex
FROM etudiants JOIN notes USING (noetu)
WHERE codemat LIKE 'MIAS2I%' and (etudiants.noetu, notes.noetu) in (
    select etudiants.noetu, notes.noetu FROM etudiants JOIN notes USING (noetu)
    WHERE codemat LIKE 'MIAS2I%'
    Group By etudiants.noetu, notes.noetu HAVING AVG(noteex) > 8
) Order By etudiants.noetu, notes.noetu ;

select noetu, AVG(noteex) FROM etudiants JOIN notes USING (noetu)
WHERE codemat LIKE 'MIAS2I%' Group By etudiants.noetu, notes.noetu HAVING AVG(noteex) > 8 ;
    
```

28936E	12.7500000000000000
46283B	8.7500000000000000
99322C	12.0000000000000000



<https://codimd.math.cnrs.fr/s/OJX0u8dFC>

12

U