

BASES DE DONNÉES :

ALGÈBRE RELATIONNELLE - REQUÊTES SQL

RADOME :							
numer_sta	id_omm	departement	nom_usuel	latitude	longitude	altitude	
(entier)	(entier)	(entier)	(texte)	(réel)	(réel)	(réel)	
89295001	7266	89	AUXERRE-PERRIGNY	47.824667	3.549667	152	
58062001	7270	58	CHATEAU CHINON	47.070333	3.934333	598	
21154001	7276	21	CHATILLON/SEINE	47.849333	4.581333	262	
21473001	7280	21	DIJON-LONGVIC	47.267833	5.088333	219	
52269001	7283	52	LANGRES	47.843667	5.3375	466	
25056001	7288	25	BESANCON	47.249	5.988833	307	

Annotations :
 - clé primaire : sur numer_sta
 - attributs : sur les colonnes de la tête
 - domaines : sur les types de données de la tête
 - tuple : sur une ligne de données

Objectifs

A la fin de la séquence d'enseignement l'élève doit pouvoir :

- utiliser une application offrant une interface graphique pour créer une base de données et l'alimenter
- utiliser une application offrant une interface graphique pour lancer des requêtes sur une base de données
- distinguer les rôles respectifs des machines client, serveur, et éventuellement serveur de données
- traduire dans le langage de l'algèbre relationnelle des requêtes écrites en langage courant
- concevoir une base constituée de plusieurs tables, et utiliser les jointures symétriques pour effectuer des requêtes croisées

Table des matières

1	Introduction	2
1.1	Nécessité de structures de données	2
1.1.1	Météo France	2
1.1.2	Professeurs et élèves d'un lycée	2
1.2	Principe des bases de données	2
2	La structure des données - Le modèle relationnel	3
2.1	Niveaux de modèles	3
2.2	Définitions	3
2.2.1	Les clés	4
2.2.2	Base de données relationnelle	4
3	L'algèbre relationnelle	5
3.1	Problématique	5
3.2	Notations ensemblistes - formalisme	6
3.2.1	L'algèbre relationnelle	6
3.2.2	Le langage SQL	6
3.3	Opérateurs relationnels	6
3.3.1	La projection	6
3.3.2	La sélection	7
3.3.3	Le renommage	8
3.3.4	La jointure	8
3.3.5	L'agrégation	9
3.4	Opérateurs ensemblistes	10
3.4.1	L'union	10
3.4.2	L'intersection	10
3.4.3	La différence	11
3.4.4	Le produit cartésien	11
3.4.5	La division cartésienne	12
4	Les tables	13

1 Introduction

1.1 Nécessité de structures de données

1.1.1 Météo France

Météo France publie et stocke chaque jour des informations météorologiques issues de ses stations (réseau RADOME).



Ces données d'observations peuvent par exemple être destinées à des messages internationaux d'observation en surface (SYNOP) qui circulent sur le système mondial de télécommunication (SMT) de l'Organisation Météorologique Mondiale (OMM). Elles contiennent des paramètres atmosphériques mesurés (température, humidité, direction et force du vent, pression atmosphérique, hauteur de précipitations) ou observés (temps sensible, description des nuages, visibilité) depuis la surface terrestre. Selon l'instrumentation et des spécificités locales, d'autres paramètres peuvent être disponibles (hauteur de neige, état du sol, etc.).

PROBLÈME: Comment stocker et accéder à ces quantités gigantesques de données ?

1.1.2 Professeurs et élèves d'un lycée

Le lycée souhaite créer un fichier de renseignement sur les élèves. Sur chaque fiche, on inscrit des renseignements : nom, prénom, adresse, lycée d'origine, nom des professeurs, ...

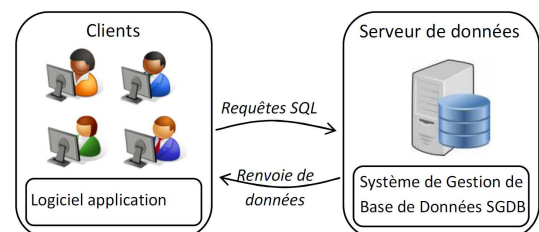
PROBLÈMES:

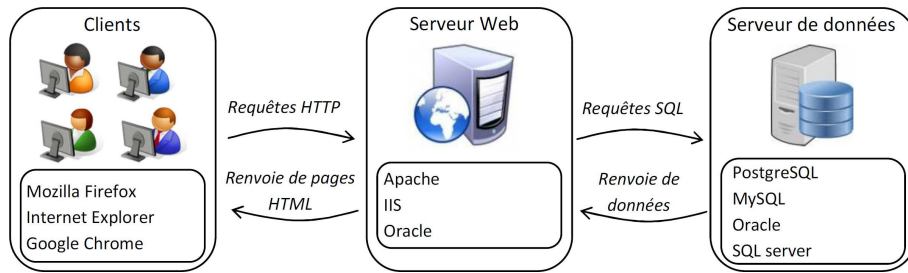
- si plusieurs élèves ont même nom, prénom ?
- comment trouver tous les élèves ayant un même enseignant ? d'une même classe ? Si on doit mettre à jour la classe ou le nom d'un professeur ? Faut-il tout refaire ?

1.2 Principe des bases de données

Différentes architectures sont possibles :

- **pair à pair (Peer to peer P2P)**
Chaque utilisateur est aussi serveur de données. Un logiciel remplissant les fonctions de **client** et de **serveur** est nécessaire pour chaque unité connectée au réseau. C'est un système de partage. Il n'y a pas de forcément de centralisation des données.
- **client - Serveur**
Les utilisateurs (clients) envoient des requêtes au serveur qui les traite et retourne des résultats. Le serveur de données intègre un Système de Gestion de Base de Données **SGBD** (logiciel) ainsi qu'un espace de stockage des données (support physique). **EXEMPLE :** bibliothèque universitaire.
- **trois tiers (3-tiers)**
Les clients sont équipés d'une interface utilisateur chargée de la présentation. Un serveur d'application (**middleware**) fournit la ressource, mais en faisant appel à un autre serveur. Un serveur de données fournit au serveur d'application les données requises pour répondre au client. **EXEMPLE :** base de données sur le Web.





Un Système de Gestion de Base de Données (**SGBD**) est un logiciel conçu pour remplir deux buts :

- définir des tables et insérer des données.
- ajouter, effacer et mettre à jour les données dans la base : plusieurs utilisateurs peuvent utiliser simultanément et à tout moment la base.
- fournir différentes manières de visualiser tout ou partie des données de la base : fonctions de recherche multicritères, interface graphiques, ...

Nous utiliserons SQLite (le moteur de base de données le plus distribué au monde) pour les deux derniers points.

2 La structure des données - Le modèle relationnel

2.1 Niveaux de modèles

Un **schéma** est une description des données selon un modèle.

On distingue trois niveaux de modèles :

- **le niveau externe** : ce qui est vu par l'utilisateur (l'applicatif),
- **le niveau logique** : le modèle relationnel,
- **le niveau physique** : l'organisation du stockage des données.

Ces trois niveaux sont dans l'idéal, indépendants. On s'intéresse par la suite au modèle relationnel.

2.2 Définitions

Une **relation** (ou **table**) permet de représenter des objets et leurs associations. Une relation forme un tableaux à deux dimensions. Il traduit un lien entre plusieurs **attributs**.

- **relation** : tableau à deux dimensions. **EXEMPLE** : RADOME
- **attribut** : colonne qui compose la table. **EXEMPLE** : altitude
- **domaine d'un attribut** : ensemble de valeurs que peut prendre un attribut (entier, réel, chaîne de caractères, ...). **EXEMPLE** : le domaine de altitude est les réels (\mathbb{R})
- **schéma** : liste des attributs d'une relation (et de façon optionnelle de leur domaine). **EXEMPLE** :
(num_sta, id_omm, nom_usuel, département, commune, altitude, latitude, longitude)
- **tuple (ou n-uplet)**: ligne d'une relation. **EXEMPLE** :
(21473001, 07280, DIJON-LONGVIC, 47.268, 5.088, 219, 19210101)
- **population** : ensemble des tuples d'une relation
- **base de données** : ensemble de plusieurs relations

RADOME :	numer_sta	id_omm	departement	nom_usuel	latitude	longitude	altitude
	(entier)	(entier)	(entier)	(texte)	(réel)	(réel)	(réel)
	89295001	7266	89	AUXERRE-PERRIGNY	47.824667	3.549667	152
	58062001	7270	58	CHATEAU CHINON	47.070333	3.934333	598
	21154001	7276	21	CHATILLON/SEINE	47.849333	4.581333	262
	21473001	7280	21	DIJON-LONGVIC	47.267833	5.088333	219
	52269001	7283	52	LANGRES	47.843667	5.3375	466
	25056001	7288	25	BESANCON	47.249	5.988833	307

TABLE 1 – Relation RADOME

2.2.1 Les clés

L'ordre des lignes n'est pas important. Par contre, chaque tuple doit être **unique**. Pour les identifier, on définit une **clé** qui peut être constituée d'un ou plusieurs attributs.

DÉFINITION: Clé

|| Ensemble d'attributs qui permet d'identifier un tuple unique de la relation

EXEMPLE : Relation gens, de schéma (nom, prénom, age) :

gens :	nom	prénom	age
clé →	Terrieur	Alain	36
	Terrieur	Alex	36
	Proviste	Alain	22

Dans la relation gens, nom ne peut être une clé puisque deux personnes (tuples...) ont la même valeur nom.

De même, dans la relation gens, prénom ne peut être une clé puisque deux personnes (tuples...) ont la même valeur prénom.

En revanche, le couple (nom, prénom) peut être un clé de la relation nom.

DÉFINITION: Clé primaire

|| Attribut qui permet d'identifier un tuple unique de la relation

On parle aussi d'identifiant. On la note en gras ou on lui associe le symbole d'une clef **⬤**.

EXEMPLE : num_sta est la clé primaire du schéma de relation RADOME. Il peut y avoir plusieurs stations météorologiques dans une grande ville. Par contre, chacune a une clé primaire distincte.


DÉFINITION: Clé étrangère

|| Ensemble d'attributs égal à la clé primaire d'une autre relation : un tuple de la relation ne peut exister que si sa clé étrangère est égale à la clé primaire d'un tuple dans l'autre relation.

EXEMPLE : id_omm est une clé étrangère pour SYNOP .

2.2.2 Base de données relationnelle

clé étrangère



SYNOP :	id_omm	date	pmer	tend	cod_tend	dd	ff	t
	7255	20160503210000	102820	130	1	40	1.7	283.45
	7280	20160503210000	102810	190	1	330	3.2	279.95
	7299	20160503210000	102770	150	1	170	0.8	280.75

TABLE 2 – Relation SYNOP

Le schéma d'une **base de données relationnelle** est constitué d'un ensemble de schémas relationnels.

EXEMPLE : Schéma de la base de donnée relationnelle BDD Météo France.

Cette base de données comporte de nombreux schémas de relation. On ne s'intéresse qu'aux deux suivants. En plus du schéma de relation RADOME, il y a le schéma SYNOP. On peut écrire :

```
BDD Météo France = {RADOME, SYNOP}, avec
RADOME = (num_sta, id_omm, nom_usuel, département, commune, altitude, latitude, longitude)
SYNOP = (id_omm, date, pmer, tend, cod_tend, dd, ff, t)
```

REMARQUE: ces deux schémas de relation ont un attribut en commun : `id_omm`. Cela permet d'effectuer des intersections de schémas de relation lors de l'exécution de requêtes.

3 L'algèbre relationnelle

3.1 Problématique

PROBLÈME: on souhaite effectuer des requêtes sur la base de données BDD Météo France.

- Quelles sont les stations situées en Bourgogne ?
- Quelles sont les stations où la pression au niveau de la mer est passée en dessous de 100000 Pa le 3 mai 2016 ?
- Quelles sont les villes où la vitesse du vent moyen a dépassé les 20 m/s en 1999 ?

On entend par **algèbre relationnelle**, une collection d'**opérations formelles** (Op) qui agissent sur des relations et **produisent une relation** en résultat : $R_3 \leftarrow R_1 Op R_2$.

Dans la plupart des SGBD relationnels, la réponse à une requête s'obtient par l'utilisation d'un ou plusieurs opérateurs relationnels.

Parfois, il est possible d'exprimer de plusieurs manières une même requête. Même si le résultat est identique pour chacune d'entre elles, il faut privilégier celle qui est la plus efficace en terme de rapidité de réponse. Certaines requêtes s'optimisent.

On précisera dans ce qui suit les opérateurs de l'algèbre relationnelle, ainsi que les requêtes **SQL (Structured Query Language)** : langage de requête structurée) correspondantes, puisqu'elles sont largement utilisées dans les SGBD (et bien sûr, au programme d'info !).

Si le langage SQL permet d'interroger la base de donnée avec une syntaxe comprise par les SGBD, l'algèbre relationnelle permet, quant à elle, de formaliser les requêtes d'un point de vue ensembliste.

3.2 Notations ensemblistes - formalisme

3.2.1 L'algèbre relationnelle

Quelques notations d'algèbre relationnelle :

- les attributs A_i sont les éléments d'un ensemble fini \mathcal{A} .
- si $A \in \mathcal{A}$, on note $dom(A)$ le domaine de A , i.e. l'ensemble des valeurs que peut prendre un élément de l'attribut A
- une relation R d'attributs A_1, \dots, A_n est notée $R(A_1, \dots, A_n)$ avec les A_i distincts deux à deux
- un schéma relationnel S est un n -uplet de la forme $S = (A_1, \dots, A_n)$ avec les A_i distincts deux à deux
- on note $B \in S$ pour dire que B est un des attributs de S
- par extension, si $X = B_1, \dots, B_p$ où les B_j sont des attributs distincts, on note $X \subset S$ pour dire que $X \subset A_1, \dots, A_n$
- si t est un tuple de R alors on note $t \in R$
- la valeur de l'attribut A_i dans t se note $t.A_i$ ou $t[A_i]$
- le p -uplets des attributs B_1, \dots, B_p dans t se note $t[B_1, \dots, B_p]$
- le nombre d'éléments d'une relation R est appelé cardinal, noté $\#R$

EXEMPLE : Dans la relation *personne*, (num_p, nom, prenom, tel) est le schéma relationnel.

Pour le tuple $t=(4027,Fraichi,Sarah,078989xxxx)$:

$$t \in \text{personne} \quad ; \quad t[\text{nom}] = \text{Fraichi} \quad \text{et} \quad t[\text{num_p}, \text{tel}] = 4027,078989xxxx$$

3.2.2 Le langage SQL

Pour obtenir des informations à partir d'une base de données, il faut *interroger* cette base à partir de requêtes. Les **requêtes** seront formulées en **SQL** et se présentent sous la forme :

```
1 SELECT ...
2 FROM ....
3 ;
```

```
1 SELECT ... FROM ... JOIN ... ON ... WHERE ... ; -- commentaire
```

Le point virgule ; termine une requête. On peut donc écrire cette requête sur plusieurs lignes.

3.3 Opérateurs relationnels

3.3.1 La projection

OBJECTIF : Sélectionner certaines colonnes ; éliminer des attributs.

DÉFINITION: Projection

Soit $R(S)$ une relation de schéma $S = (A_1, \dots, A_n)$ et $X \subset S$.

On appelle **projection** de R selon X la relation notée $\pi_X(R)$ obtenue en ne gardant que les attributs X pour

chacun des tuples de R : $\pi_X(R) = \{t[X] | t \in R\}$

```
SELECT A1, A3, ... FROM R;
```

EXEMPLE : Quels sont les numéros de station et les villes correspondantes ?

```
SELECT num_sta, commune FROM RADOME ;
```

REMARQUE: en cas de doublon dans le résultat SQL, on utilise `DISTINCT`. En algèbre relationnelle, les doublons n'existent pas.

```
SELECT DISTINCT nom, age FROM gens ;
```

$\pi_{\text{num_sta, commune}}(\text{RADOME})$: num_sta	nom_usuel
89295001	AUXERRE-PERRIGNY
58062001	CHATEAU CHINON
21154001	CHATILLON/SEINE
21473001	DIJON-LONGVIC
52269001	LANGRES
25056001	BESANCON

3.3.2 La sélection

OBJECTIF : Sélectionner certaines lignes (sous-ensemble de tuples) qui vérifient une condition.

DÉFINITION: Sélection

Soient $R(S)$ une relation de schéma S , $A \in S$ un attribut du schéma S et $a \in \text{dom}(A)$ un élément du domaine de A . On appelle **sélection** de R selon $A = a$, que l'on note $\sigma_{A=a}(R)$, la relation obtenue en sélectionnant dans R uniquement les tuples t tels que $t[A] = a$ (ou $t.A = a$). On obtient

$$\sigma_{A=a}(R) = \{t \in R \mid t[A] = a\}$$

REMARQUE: on peut utiliser des conditions autres que l'égalité.

Si E est un ensemble et P une propriété pouvant être décidée sur les éléments de E , on note $\{x \in E \mid P(x)\}$ l'ensemble des éléments de E qui vérifient P . Ainsi, la sélection de R sur une propriété p (ou prédicat p) est la relation constituée des tuples de R vérifiant p , que l'on note alors

$$\sigma_p(R) = \{t \in R \mid p(t)\}$$

- **en algèbre relationnelle :** on peut utiliser les opérateurs logiques suivants :

$=, \neq, <, >, \leq, \geq, \wedge$ (et logique), \vee (ou logique), \neg (complément logique)

- **dans les requêtes SQL :** on peut utiliser les éléments du tableau suivant :

=	>	<=	AND	IS NOT NULL	NOT
!= ou <>	<	>=	OR	IS NULL	BETWEEN ... AND ...

```
SELECT * FROM R WHERE A = a;
```

REMARQUE: `*` permet de sélectionner tous les attributs.

EXEMPLE : Quels sont les numéros de station météo et les noms des stations situées en Côte d'Or ?

$$R_{sel} \leftarrow \pi_{\text{num_sta, nom_usuel}}(\sigma_{\text{département}=21}(\text{RADOME}))$$

```
SELECT num_sta, nom_usuel FROM RADOME WHERE département=21
```

R_{sel} : num_sta	nom_usuel
21154001	CHATILLON/SEINE
21473001	DIJON-LONGVIC

REMARQUE: on peut aussi utiliser le mot clé `LIKE` `'...'` associé au tiret `-` pour remplacer un caractère et le pourcentage `%` pour remplacer un nombre quelconque de caractères :

- `LIKE 'A%'` pour un mot commençant par A
- `LIKE '%a%'` pour un mot contenant a

- LIKE 'po%ur' pour un mot commençant par *po* et finit par *ur* (pour, porteur, pollueur, ...).
- LIKE 'a_c' pour remplacer un et un seul caractère dans une chaîne, ici, entre a et c.

3.3.3 Le renommage

OBJECTIF : changer le nom d'un attribut ou d'un ensemble d'attributs pour rendre des schémas compatibles.

DÉFINITION: Renommage

Soit $R(X, Y)$ avec X, Y des attributs ou ensembles d'attributs. Si Z et X ont même taille et si Z n'apparaît pas dans le schéma de R , on note $\rho_{X \rightarrow Z}(R)$ la relation de schéma (Z, Y) obtenue en renommant X en Z , telle que $\rho_{X \rightarrow Z}(R) = \{t ; \exists t' \in R | t[Z] = t'[X] \text{ et } t[Y] = t'[Y]\}$

```
SELECT X AS Z FROM R;
```

$\rho_{\text{nom_usuel} \rightarrow \text{Nom de la station}} (\sigma_{(\text{id_omm} \neq \text{NULL}) \wedge (\text{departement}=21)} (\text{RADOME}))$

ou encore :

$\rho_{\text{nom_usuel} \rightarrow \text{Nom de la station}} (\sigma_{(\text{id_omm} \neq \text{NULL})} (\sigma_{\text{departement}=21} (\text{RADOME})))$

```
SELECT nom_usuel
AS 'Nom_de_la_station'
FROM RADOME
WHERE id_omm IS NOT NULL
AND departement=21;
```

```
SELECT nom_usuel
AS 'Nom_de_la_station'
FROM (SELECT
      nom_usuel , departement
      FROM RADOME
      WHERE id_omm
      IS NOT NULL)
WHERE departement=21 ;
```

3.3.4 La jointure

OBJECTIF : Combiner une paire de tuples de deux relations en un seul tuple.

DÉFINITION: Jointure

Soient $R_1(X, Y)$ et $R_2(U, V)$ deux relations avec X, Y, U, V des attributs ou des ensembles d'attributs tels que $X \neq \emptyset$ et $Y \neq \emptyset$. On appelle **jointure** de R_1 et R_2 selon la condition $X = U$ (ou $R_1.X = R_2.U$) la relation notée $R_1 \bowtie_{R_1.X=R_2.U} R_2$ (ou $R_1[R_1.X = R_2.U]R_2$) dont les tuples sont de la forme (x, y, u, v) tels que $x = u$, $(x, y) \in R_1$ et

$(u, v) \in R_2$: $R_1[R_1.X = R_2.U]R_2 = R_1 \bowtie_{R_1.X=R_2.U} R_2 = \{t | t[X, Y] \in R_1; t[U, V] \in R_2; t[X] = t[U]\}$

REMARQUE: On ne s'intéresse ici qu'à la jointure symétrique simple qui permet de recoller deux relations ayant un attribut en commun.

```
SELECT * FROM R1 JOIN R2 ON R1.attribut = R2.attribut
```

EXEMPLE : Jointure de personne et enseigne sur `personne.num_p` :

$R_{\text{join}} \leftarrow \text{personne} \bowtie_{\text{personne.num}_p=\text{enseigne.num}_\text{prof}} \text{enseigne}$

```
SELECT * FROM personne pe JOIN enseigne en ON pe.num_p=en.num_prof;
```


R_{join} : num_p	nom	prénom	tel	classe	num_prof	matière
9509	McGonagall	Minerva	061889xxxx	1	9509	Math
9826	Ombrage	Dolores	074136xxxx	1	9826	Physique
9095	Dumbledore	Albus	068779xxxx	1	9095	SI
9756	Quirrell	Quirinus	071426xxxx	5	9756	Math
9407	Maugrey	Alastor	062643xxxx	5	9407	Physique
9095	Dumbledore	Albus	068779xxxx	5	9095	SI
9095	Dumbledore	Albus	068779xxxx	5	9095	Info
9995	Snape	Severus	066230xxxx	5	9995	Chimie
9509	McGonagall	Minerva	061889xxxx	1	9509	Info

```
SELECT nom, prénom
FROM personne pe
JOIN enseigne en
ON pe.num_p=en.num_prof
WHERE matière='Info';
```

R_{info}	nom	prénom	matière
	Dumbledore	Albus	Info
	McGonagall	Minerva	Info

EXEMPLE : Quelles sont les villes où la vitesse du vent moyen a dépassé les 20 m/s ?

$$\pi_{\text{commune}} \left(\sigma_{ff > 20} \left(\text{RADOME} \bowtie_{\text{id_omm}} \text{SYNOP} \right) \right)$$

```
SELECT RADOME.commune
FROM RADOME JOIN SYNOP ON RADOME.id_omm=SYNOP.id_omm WHERE SYNOP.ff > 20
```

EXEMPLE : Donner le nom et la classe des élèves des classes ayant moins de 35 élèves :

```
SELECT p.nom, c.nom
FROM classe c JOIN eleve e ON c.no=e.num_classe
JOIN personne p ON p.num_p = e.num_p WHERE effectif <= 35;
```

qui est plus rapide que :

```
SELECT p.nom, tel FROM classe c, eleve e, personne p
WHERE num_classe = no AND p.num_p = e.num_p AND effectif <= 35;
```

nom	classe
Attan	MPSI 1
Hindailaibile	MPSI 1
Assain	MPSI 1
Fraichi	MPSI 1

3.3.5 L'agrégation

OBJECTIF : permettre d'appliquer des fonctions aux attributs d'une table et retourner une valeur unique.

On distingue deux types de fonction sur les attributs d'une table :

- les fonction s'appliquant à un attribut d'une table en modifiant dans chaque tuple l'attribut correspondant.

EXEMPLE : convertir la pression de la mer en hPa à partir de la valeur en Pa :

```
SELECT numer_sta, pmer/100 FROM SYNOP;
```

- les **fonctions d'agrégation** utilisant toutes les valeurs des tuples d'une table extraite suivant un attribut.

Soient R une relation de schéma S , A un attribut de S et f une fonction définie sur $dom(A)$. On note $f(R.A)$ ou $\gamma_{f(A)}(R)$ le résultat de l'application de la fonction f au tuple des valeurs de R pour l'attribut A .

Voici la liste des fonctions d'agrégation que nous utiliserons :

MIN, MAX, SUM (somme), AVG (moyenne),
COUNT (comptage du nombre de lignes).

$$\pi_{\gamma_{MAX(pmer)}(SYNOP)}$$

```
SELECT MAX(pmer) FROM SYNOP
```

EXEMPLE : Quelle est la plus haute valeur de pression mesurée ?

3.4 Opérateurs ensemblistes

Puisqu'une relation est un ensemble fini de n -uplets de $D_1 \times \dots \times D_n$ où les D_i sont les domaines de ses attributs, si deux relations ont le même schéma, alors il est possible de leur appliquer des opérateurs ensemblistes.

On utilisera principalement l'union (\cup), l'intersection (\cap) et la différence ($-$).

Soient alors deux relations :

<i>mpsi1</i> : num_prof matière	
9509	Math
9826	Physique
9509	Info
9095	SI

<i>pcsi2</i> : num_prof matière	
9756	Math
9407	Physique
9095	SI
9095	Info
9995	Chimie

3.4.1 L'union

DÉFINITION: Union

|| L'*union* de deux relations $R_1(S)$ et $R_2(S)$ est l'ensemble des valeurs comprises dans R_1 *ou* R_2 .

On la note $R_1 \cup R_2$ de schéma S .

```
SELECT * FROM R_1 UNION SELECT * FROM R_2;
```

ATTENTION ! si en algèbre relationnelle les doublons sont supprimés, il n'en va de même en SQL. Il faut le demander explicitement (car cela demande un peu plus de travail !).

```
SELECT DISTINCT * FROM R_1 UNION SELECT * FROM R_2;
```

EXEMPLE : Quels sont les numéros de stations situées en Côte d'Or et celles situées en Haute-Marne ?

$\pi_{\text{num_sta}}(\sigma_{\text{département}=21}(\text{RADOME}) \cup \sigma_{\text{département}=52}(\text{RADOME}))$

```
SELECT num_sta FROM
SELECT RADOME WHERE departement = 21
UNION
SELECT num_sta FROM RADOME WHERE departement = 52;
```

<i>mpsi1</i> \cup <i>pcsi2</i> : num_prof matière	
9509	Math
9826	Physique
9509	Info
9095	SI
9756	Math
9407	Physique
9095	SI
9095	Info
9995	Chimie

La répétition n'existe pas en algèbre relationnelle.

```
SELECT num_sta
FROM RADOME
WHERE departement = 21
OR departement = 52;
```

3.4.2 L'intersection

DÉFINITION: Intersection

|| L'*intersection* de deux relations $R_1(S)$ et $R_2(S)$ est l'ensemble des valeurs comprises dans R_1 *et* dans R_2 .

On la note $R_1 \cap R_2$ de schéma S .

```
SELECT FROM R_1
INTERSECT
SELECT * FROM R_2;
```

<i>mpsi1</i> \cap <i>pcsi2</i> : num_prof matière	
9095	SI

EXEMPLE : Quels sont les numéros de station situés en Côte d'Or et à une altitude supérieure à 70 m ?

```
SELECT num_sta FROM RADOME WHERE departement = 21
INTERSECT SELECT num_sta FROM RADOME WHERE altitude > 70
```

$\pi_{\text{num_sta}}(\sigma_{\text{département}=21}(\text{RADOME}) \cap \sigma_{\text{altitude}>70}(\text{RADOME}))$

3.4.3 La différence

DÉFINITION: Différence

La *différence* entre deux relations $R_1(S)$ et $R_2(S)$ est l'ensemble des valeurs comprises dans $R_1(S)$ *mais pas* dans $R_2(S)$.

On la note $R_1 - R_2$ de schéma S .

```
SELECT * FROM R_1 EXCEPT SELECT * FROM R_2;
```

ATTENTION ! relation non commutative et non associative.

On a viré le prof de SI !!

<i>mpsi1 - pcsi2:</i>	num_prof	matière
	9509	Math
	9826	Physique
	9509	Info

<i>pcsi2 - mpsi1 :</i>	num_prof	matière
	9756	Math
	9407	Physique
	9095	Info
	9995	Chimie

EXEMPLE : Quels sont les numéros de station situés en Côte d'Or et qui ne sont pas à une altitude supérieure à 250 m ?

$\pi_{\text{num_sta}}(\sigma_{\text{département}=21}(\text{RADOME}) - \sigma_{\text{altitude}>70}(\text{RADOME}))$

```
SELECT num_sta FROM RADOME WHERE departement = 21 EXCEPT SELECT num_sta FROM RADOME WHERE altitude >250
```

3.4.4 Le produit cartésien

OBJECTIF : assembler deux tables sans avoir recours l'opérateur relationnel *jointure*.

DÉFINITION: Produit cartésien

Soient deux relations R_1 de schéma $S_1 = (A_1, \dots, A_m)$ et R_2 de schéma $S_2 = (B_1, \dots, B_n)$ n'ayant aucun attribut en commun. On note $R_1 \times R_2$ la relation obtenue en concaténant tout tuple de R_1 avec tout tuple de R_2 .

$$R_1 \times R_2 = \{t[t[A_1, \dots, A_m] \in R_1 \text{ et } t[B_1, \dots, B_n] \in R_2]\}$$

On note $R_1 \times R_2$ de schéma $S = S_1 \uplus S_2 = (S_1, S_2)$, le produit cartésien des relations R_1 de schéma S_1 et R_2 de schéma S_2 .

```
SELECT * FROM R_1 , R_2;
```

ATTENTION ! Si R_1 (resp. R_2) a m (resp. n) tuples et j (resp. k) attributs alors $R_1 \times R_2$ a $m.n$ tuples et $j + k$ attributs, ce qui fait $m.n.(j + k)$ "cases" dans la table.

occupation :	salle	classe
	167	MPSI_1
	208	PCSI_2

capacité :	salle	chaises
	167	90
	261	32
	208	48

capacité × occupation :	salle	chaises	salle	classe
	167	90	167	MPSI_1
	167	90	208	PCSI_2
	208	48	167	MPSI_1
	208	48	208	PCSI_2
	261	32	167	MPSI_1
	261	32	208	PCSI_2

REMARQUE: Il n'est pas nécessaire en SQL que les attributs de R_1 et R_2 soient tous différents (ceux de R_1 sont préfixés par $R_1.$ et ceux de R_2 par $R_2.$).

```
SELECT ca.salle , ca.chaises , oc.classe FROM capacite ca , occupation oc ;
```

R_{cart} :	salle	chaises	salle	classe
	167	90	MPSI_1	
	167	90	PCSI_2	
	208	48	MPSI_1	
	208	48	PCSI_2	
	261	32	MPSI_1	
	261	32	PCSI_2	

$$R_{cart} \leftarrow \sigma_{ca.salle, ca.chaises, oc.classe}(\rho_{capacité \rightarrow ca}(capacité) \times \rho_{occupation \rightarrow oc}(occupation))$$

$$R_{join} \leftarrow \sigma_{ca.salle, ca.chaises, oc.classe} \left(\rho_{capacite \rightarrow ca} (capacite) \bowtie_{salle} \rho_{occupation \rightarrow oc} (occupation) \right)$$

```
SELECT ca.salle, ca.chaises, oc.classe
FROM capacite ca JOIN occupation oc ON ca.salle=oc.salle;
```

R _{join} : salle chaises classe		
167	90	MPSI_1
208	48	PCSI_2

REMARQUE:

Afin d'éviter des calculs inutiles, on préfère la requête 2 à la requête 1 :

Requête 1 :

```
SELECT ... FROM R1, R2 WHERE R1.X=R2.A;
```

Requête 2 :

```
SELECT ... FROM R1 JOIN R2 ON R1.X=R2.A;
```

3.4.5 La division cartésienne

OBJECTIF : obtenir la plus grande relation vis-à-vis de l'inclusion.

DÉFINITION: Division cartésienne

Soient deux relations R_1 de schéma $S_1 = (A_1, \dots, A_n)$ et R_2 de schéma $S_2 = (A_1, \dots, A_m)$ telles que $m < n$. On note $R_1 \div R_2$ de schéma $S = (A_{m+1}, \dots, A_n)$ la division cartésienne de R_1 par R_2 telle que :

$$R_1 \div R_2 = \{t | \forall t' \in R_2, (t, t') \in R_1\}$$

La combinaison de chaque tuple de $R_1 \div R_2$ avec les tuples de R_2 est contenue dans R_1 .

REMARQUE: en pratique, on peut l'exprimer à partir des autres opérateurs. Trop coûteuse, elle n'est pas présente directement dans les langages de requêtes de type SQL.

EXEMPLE : Quels sont les numéros de station qui ont rencontré tous les types de tendance barométrique ?

$$\pi_{num_sta} (SYNOP \div \pi_{cod_tend} (SYNOP))$$

EXEMPLE ::

R ₂ : B C
1 1
2 0

R ₃ : B C
1 1

R ₄ : B C
3 5

R ₁ : A B C
1 1 1
1 2 0
1 2 1
1 3 0
2 1 1
2 3 3
3 1 1
3 2 0
3 2 1

R ₁ ÷ R ₂ : A
1
3

R ₁ ÷ R ₃ : A
1
2
3

R ₁ ÷ R ₄ : A
/

Bibliographie

Ce cours a été réalisé à l'aide :

- du livre *Informatique pour tous en classe préparatoire aux grandes écoles*, chez EYROLLES (<http://www.eyrolles.com/Sciences/Livre/informatique-pour-tous-en-classes-preparatoires-aux-grandes-ecoles-9782212137002>)
- du cours de J. Larochette (<http://sup3carnot.free.fr/spip3/>)
- du cours de P. Beynet
- du site SQL.sh
- du site Documentation PostgreSQL 7.2 (<http://www.linux-france.org/article/serveur/psql/Postgres-7.2/book18.html>)

4 Les tables

Descriptif	Attribut	Type	Unité
Indicatif INSEE station	num_sta	car	
Indicatif OMM station	id_omm	int	
Date (UTC)	date	car	AAAAMMJJHHMISS
Pression au niveau mer	pmer	int	Pa
Variation de pression en 3 heures	tend	int	Pa
Type de tendance barométrique	cod_tend	int	code
Direction du vent moyen 10 mn	dd	int	degré
Vitesse du vent moyen 10 mn	ff	réel	m/s

TABLE 3 – Attributs inclus dans les données SYNOP

eleve :	num_p	num_etudiant	lycée_origine	num_classe
	8227	379	Beaux Bâtons	2
	2899	609	Hogwarts	1
	1409	440	Hogwarts	1
	8862	881	Carnot	3
	2159	918	Beaux Bâtons	3
	7160	921	Hogwarts	5
	4027	105	Hogwarts	1
	1605	438	Beaux Bâtons	2
	2394	839	Beaux Bâtons	1
	6356	246	Durmstrang	3
	8684	580	Durmstrang	4
	3955	211	Carnot	3
	4777	361	Carnot	5
	8871	777	Durmstrang	2
	1457	351	Durmstrang	3

TABLE 4 – Relation **eleve**

enseigne :	num_classe	num_prof	matière
	1	9509	Math
	1	9826	Physique
	1	9095	SI
	1	9509	Info
	5	9756	Math
	5	9407	Physique
	5	9095	SI
	5	9095	Info
	5	9995	Chimie

TABLE 5 – Relation **enseigne**

personne:	num_p	nom	prénom	tel
	8227	Aitinnegueux	Marc	070890xxxx
	2899	Assain	Marc	069803xxxx
	1409	Attan	Charles	070184xxxx
	9870	Binns	Cuthbert	067362xxxx
	8862	Blechose	Lorie	063264xxxx
	2159	Caperdunaufage	Thérèse	066508xxxx
	7160	Culaire	Lorie	070049xxxx
	9095	Dumbledore	Albus	068779xxxx
	9632	Flitwick	Filius	076523xxxx
	4027	Fraichi	Sarah	078989xxxx
	1605	Gnorancedeumapart	Sophie	066595xxxx
	2394	Hindailaibile	Marc	069308xxxx
	6356	Insonsec	Thérèse	063364xxxx
	9305	Longbottom	Neville	069154xxxx
	9407	Maugrey	Alastor	062643xxxx
	9509	McGonagall	Minerva	061889xxxx
	9826	Ombrage	Dolores	074136xxxx
	9756	Quirrell	Quirinus	071426xxxx
	8684	Skicoulafleau	Louis	069497xxxx
	9700	Slughorn	Horace	064326xxxx
	9995	Snape	Severus	066230xxxx
	3955	Stickey	Sophie	079525xxxx
	4777	Terrieur	Alain	062858xxxx
	8871	Terrieur	Alex	068036xxxx
	9535	Trelawney	Sibylle	064815xxxx
	1457	Vigote	Sarah	076870xxxx

TABLE 6 – Relation personne

prof :	num_p	anciennete	grade
	9870	200	Pr
	9095	130	Pr
	9305	1	Cer
	9407	40	Agr CS
	9509	60	Pr
	9826	30	Agr HC
	9756	25	Agr
	9700	70	Pr
	9995	25	Agr CS
	9535	40	Cer
	9632	65	Pr

TABLE 7 – Relation prof

classe :	no	nom	effectif	prof_princ
	1	MPSI 1	35	9509
	2	MPSI 2	41	9632
	3	MPSI 3	42	9870
	4	PCSI 1	42	9995
	5	PCSI 2	40	9407

TABLE 8 – Relation classe