

INSA DE ROUEN
DÉPARTEMENT GÉNIE MATHÉMATIQUE

23 avril 2014
Projet Base de Données

**Modélisation d'un forum associé à un site internet,
création d'une base utilisateurs.**

Alexandre VIEIRA

Encadrant : Nathalie CHAIGNAUD

Table des matières

Introduction	2
1 Modèle Entités/Associations	3
1.1 Schéma du modèle	3
1.2 Commentaire et justification	5
2 Modèle relationnel	6
2.1 Avant dénormalisation	6
2.2 Dénormalisation	7
2.3 Modèle relationnel final	7
3 Codage SQL de la base de données	8
3.1 Création et remplissable de la base	8
3.2 Requêtes sur la base	9

Introduction

Le but de ce projet est de modéliser un site internet de nouvelles en ligne. On retrouve bien souvent des sites internet qui réunissent en leur sein deux services :

- Une ou plusieurs pages présentant plusieurs nouvelles (news), qui peuvent être commentées
- Un forum où les utilisateurs peuvent échangés publiquement (posts) ou de façon plus privée (messages privés, ou MP)

Au cœur de tout cela se trouve donc une base d'utilisateurs, qui forment des groupes :

- des utilisateurs communs qui postent des commentaires sur les news, sur le forum ou qui s'envoient des messages privés
- des modérateurs pour le forum ou les commentaires
- des "newseurs" qui postent les nouvelles sur le site, et qui utilisent parfois certaines sources qu'il faut pouvoir citer et retrouver

Le forum aura également sa propre organisation interne, que la base de données devra également présenter :

- les posts publics seront inclus dans des sous-sections, eux-mêmes inclus dans des sections
- les messages privés s'organisent par sujet qui connaîtront un certain nombre de destinataires qui échangeront toute une conversation

La base de données devra donc pouvoir répondre aux problèmes suivants :

- Quelles news ont été postées à une date donnée et qui les a posté ?
- Où telle source a été utilisée ? Quand ? Et par qui ?
- Qui a écrit un commentaire donné ? Sur quelle news ?
- Quelles sont les sous-section de telle section ? Quels sont les sujets compris dans telle sous-section ?
- Qui a écrit tel post ? Dans quel sujet ? Quand ?
- Quels sont les destinataires d'un sujet de message privé ? Qui a écrit tel message privé ? En rapport avec quel sujet ?
- Quels sont les conversations par messages privés ou sections sur le forum auquel un utilisateur a accès ?
- A quel groupe appartient un utilisateur donné ? Qui sont les modérateurs du forum ?

Dans un premier temps, le modèle Entités/Associations sera présenté, puis on en déduira le modèle relationnel. Chacun de ces modèles sera commenté et justifié.

1 Modèle Entités/Associations

1.1 Schéma du modèle

Analysons tout d'abord les besoins pour pouvoir construire notre schéma Entités/Associations.

- Nous aurons une entité, Personne, qui se trouvera au cœur de notre schéma. En effet, l'inscription des membres sera autant utile aux entités faites pour le site que pour le forum.
- Le forum se décomposera en deux parties indépendantes, mais proches dans la forme
 - Les posts seront plutôt des messages échangés publiquement. Ils devront bien évidemment être rangés, organisés. Ainsi, des sections contenant elles-même des sous-sections seront créées. Les sous-sections contiendront des sujets, qui contiendront des posts créés par les membres.
 - Les membres doivent également pouvoir échanger de façon plus privée. C'est ce à quoi servent les messages privés (MP). Pour classer les MP, on les range par sujet. On pensera également que plusieurs personnes pourront participer à ces conversations.
- Cependant, toutes les personnes ne doivent pas toujours voir tous les messages postés sur le forum (comme une partie réservée uniquement aux modérateurs et administrateurs). Il peut être également pratique de voir qui sont les modérateurs par exemple. Une entité Groupe doit donc être créée pour pouvoir identifier plus facilement ces personnes.
- De la même façon, le site s'organise en deux parties :
 - Il devra présenter plusieurs nouvelles journalières, appelées news. Elle sera créée par une personne, qui devra également utiliser une source, qui sera toujours enregistrée.
 - Chaque news comprendra aussi potentiellement des commentaires, postés par des membres.

Avec toutes ces considérations, on en arrive finalement au schéma figure 1.

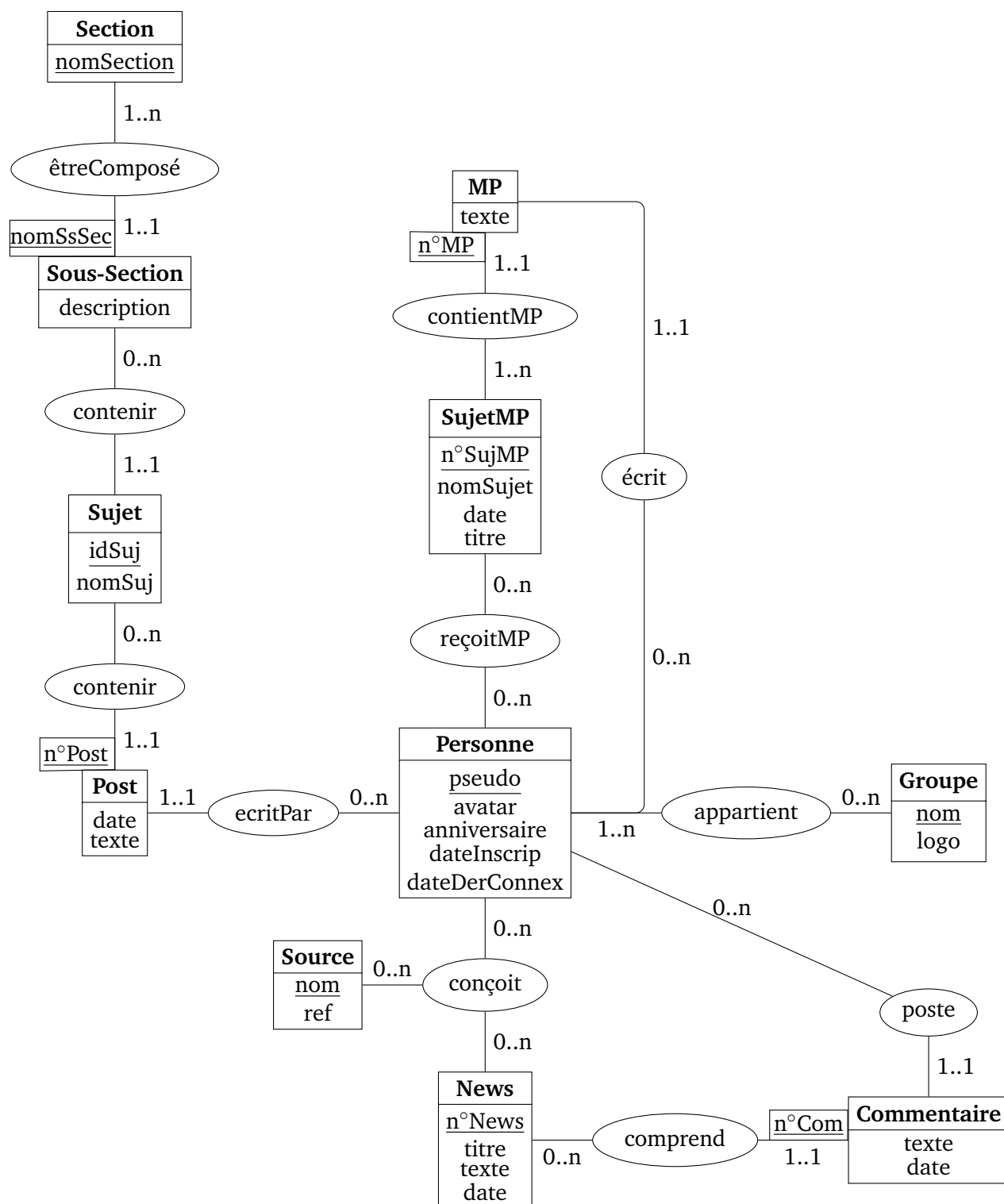


FIGURE 1 – Schéma Entités-Associations

1.2 Commentaire et justification

êtreComposé Une section peut contenir entre 1 et n sous-section. Cependant, une sous-section doit obligatoirement être contenue dans une et une seule section. Comme plusieurs sous-sections peuvent avoir le même nom mais dans des sections différentes, on introduit la sous-section comme une entité faible d'une section avec un nom de sous-section servant de clé primaire. Cela obligera donc chaque sous-section d'une section à avoir des noms différents, ce qui apporte une certaine clarté.

contenir1 De même que précédemment, une sous-section contient plusieurs sujets de conversation. Un sujet doit forcément être dans une sous-section. On introduit également le sujet comme une entité faible de la sous-section, avec le nom du sujet servant de clé primaire.

contenir2 On retrouve encore le même genre de cas, avec cette fois-ci les posts des utilisateurs et les sujets

ecritPar Un post ne peut être écrit que par une et une seule personne, et une personne peut écrire autant de posts qu'elle veut (ou même aucun). Par ailleurs, le pseudo est ici utilisé comme clé primaire, ce qui obligera chaque personne à avoir un pseudo différent (ce qui est plus commode).

contientMP Un SujetMP contient plusieurs messages privés (MP). Cependant, un message privé se trouve nécessairement dans un seul sujet. On utilise encore une entité faible dans ce cas là.

reçoitMP Les conversations pouvant se faire en groupe même par messages privés, le choix a été fait de faire passer tout cela par le sujet de la conversation. Ainsi, une personne peut participer à plusieurs conversations, et une conversation peut avoir plusieurs personnes pouvant y accéder. Le choix a été fait de mettre un numéro pour reconnaître le sujet des messages, car un même nom de sujet pourrait être utilisé plusieurs fois facilement.

écrit Le mécanisme est ici le même que pour écritPar pour les posts.

appartient Ici, on choisit de pouvoir mettre une personne dans plusieurs groupes (et bien sûr qu'un groupe puisse être vide ou avoir autant de personnes que nécessaire). Par exemple, on pourrait avoir une personne qui serait à la fois rédacteur de news et modérateur en même temps.

conçoit Une personne peut utiliser plusieurs sources et écrire plusieurs news. Une source peut être utilisée plusieurs fois pour plusieurs news différentes. Enfin, une news ne peut être écrite que par une personne (une des limites des schémas E/A) mais peut aussi bien n'utiliser aucune source ou en utiliser plusieurs. Enfin, pour une news et une personne donnée, il ne peut y avoir qu'une seule date possible.

comprend On retrouve une nouvelle fois le même schéma qu'avec post et sujet.

poste Il s'agit ici d'un utilisateur qui poste un commentaire sur une news. Un commentaire ne peut être posté que par une seule personne, mais une personne peut poster autant de commentaires qu'il veut.

2 Modèle relationnel

2.1 Avant dénormalisation

Tout d'abord, nous allons déduire directement du modèle entités/associations le modèle relationnel en utilisant les règles suivantes :

- Chaque entité devient une relation.
- Une association qui donne une entité faible transforme la relation induite par cette dernière. Dans l'entité faible, on utilise un mécanisme de clé étrangère pour référencer l'entité forte.
- On crée des relations à partir des autres associations, en reprenant les clés des deux entités liées comme clé de la nouvelle relation, et les propriétés comme attributs.

On en arrive à la liste de relations suivante :

Section (nomSection)

Sous-Section (nomSection, nomSsSection, description)

Sujet (nomSsSec, nomSujet)

Post (nomSujet, n°Post, date, texte, pseudo)

Personne (pseudo, avatar, anniversaire, dateInscript, dateDerConnex)

Groupe (nomGroupe, logo)

SujetMP (n°SujetMP, nomSujet, date, titre)

MP (n°SujetMP, n°MP, texte, pseudo)

Source (nomSource, ref)

News (n°News, texte, titre)

Commentaire (n°News, n°Com, pseudo, texte, date)

reçoitMP (n°SujetMP, pseudo)

appartient (pseudo, nomGroupe)

conçoit (n°News, pseudo, nomSource, date)

Les clés sont pour la plupart issues directement du modèle E/A. On doit tout de même garder en tête que les clés d'une relation peut être un sous-ensemble de la concaténation des clés. Quelques clés des relations ci-dessus méritent donc une justification :

- Dans **MP**, on aurait aussi bien pu mettre le pseudo en clé, mais les deux numéros (de sujet et du MP) suffisent. En effet il suffit de pouvoir reconnaître la conversation puis le message en lui-même pour pouvoir retrouver un message en particulier. Il en est de même avec **Commentaire**.
- Dans **reçoitMP** et **appartient**, pour reconnaître de manière unique un tuple, il nous faut forcément garder les deux clés des deux associations. Par exemple, pour le premier, si nous prenions que le numéro de sujet, il y aurait plusieurs pseudos associés (au moins 2, pour une conversation). Pour le second, si nous prenions que le pseudo, on aurait la plupart du temps un seul groupe associé. Mais comme précisé plus haut, une personne peut appartenir à plusieurs groupes. On doit donc garder les deux clés.
- Dans **conçoit**, le numéro de la news suffit à accéder de manière unique à toutes les autres informations. Aucun besoin donc de rajouter d'autres clés.

2.2 Dénormalisation

1. On supprime la relation **Section**, car on la retrouve dans **Sous-Section**
2. On peut supprimer la relation **Groupe**, car elle contient en fin de compte relativement peu d'information. On duplique ces informations dans **appartient**, qui devient donc :

appartient (pseudo, nomGroupe, logo)

3. On pourrait possiblement faire de même avec **News** et l'intégrer à chaque fois dans **conçoit**, mais l'accès à ce dernier sera probablement fréquent et pourrait être l'objet d'une recherche particulière. Avoir une table directement prête à disposition permettrait d'avoir un accès plus rapide à l'information recherchée.
4. Cependant, le choix a été fait de supprimer la relation **Source**, car une recherche particulière sur ce sujet reste tout de même plus rare. On intègre les informations dans la relation **conçoit**, qui devient :

conçoit (n°News, pseudo, nomSource, date, ref)

2.3 Modèle relationnel final

Après dénormalisation, notre modèle relationnel devient le suivant :

Sous-Section (nomSection, nomSsSection, description)

Sujet (idSuj, nomSection, nomSsSection, nomSujet)

Post (idSuj, n°Post, date, texte, pseudo)

Personne (pseudo, avatar, anniversaire, dateInscript, dateDerConnex)

SujetMP (n°SujetMP, nomSujet, date, titre)

MP (n°SujetMP, n°MP, texte, pseudo)

News (n°News, texte, titre)

Commentaire (n°News, n°Com, pseudo, texte, date)

reçoitMP (n°SujetMP, pseudo)

appartient (pseudo, nomGroupe, logo)

conçoit (n°News, pseudo, nomSource, date, ref)

3 Codage SQL de la base de données

Maintenant que la base de données est bien définie, nous allons enregistrer des données sur le disque dur. Pour cela, nous utiliserons le langage SQL, et plus spécifiquement le système de gestion de base de données MySQL.

Après avoir créé la base de données et l'avoir remplie avec certaines informations, nous exécuterons certaines requêtes sur celle-ci, qui seront traduites (dans la mesure du possible) en algèbre relationnelle, puis en requête SQL.

3.1 Création et remplissage de la base

Nous commençons par créer la base de données, et créer les différentes tables qui ont été définies dans le modèle relationnel. Le codage SQL est montré en partie ici, le reste est présenté en annexe ??

```
CREATE DATABASE IF NOT EXISTS forumSite;

CREATE TABLE Sous-section(
nomSection VARCHAR(50),
nomSsSection VARCHAR(100),
description TEXT,
PRIMARY KEY nomSection, nomSsSection);

CREATE TABLE Sujet(
idSuj UNSIGNED SMALLINT AUTO_INCREMENT,
nomSection VARCHAR(50),
nomSsSection VARCHAR(100),
nomSujet VARCHAR(200),
PRIMARY KEY idSuj);

CREATE TABLE Post(
idSuj UNSIGNED SMALLINT,
noPost UNSIGNED SMALLINT AUTO_INCREMENT,
datePost DATETIME,
textePost LONGTEXT,
pseudo VARCHAR(200),
PRIMARY KEY idSuj, noPost,
FOREIGN KEY idSuj REFERENCES Sujet);

CREATE TABLE Personne(
pseudo VARCHAR(50),
avatar MEDIUMBLOB,
anniversaire DATE,
dateInscrip DATE,
dateDerConnex DATE,
PRIMARY KEY pseudo);

CREATE TABLE SujetMP(
noSujetMP UNSIGNED SMALLINT AUTO_INCREMENT,
```

```
nomSujet VARCHAR(150),
dateSujMP DATE,
PRIMARY KEY noSujetMP);
```

De même, on crée des entrées dans la base de données (pour la plupart quelconques). Une partie du code est présenté ici, le reste est accessible en annexe ??

3.2 Requêtes sur la base

Nous nous intéressons à présent aux requêtes sur la base de données. Il y en a en tout une quinzaine, ne recouvrant qu'une partie de ce que peut offrir un langage comme SQL. A chaque fois, la question sera traduite (si possible) en algèbre relationnelle puis sous forme de requête SQL. Nous présenterons ensuite les résultats sur la base de données.

- Quel est le nombre de posts dans chaque sujet de la base de données ?

$$\prod_{\text{idSuj, count(noPost)}} (\text{Post})$$

```
SELECT idSuj, COUNT(noPost) FROM Post
GROUP BY idSuj;
```

- Création d'une vue donnant les personnes ayant déjà écrit une news

```
CREATE VIEW redacteurEffectifNews (pseudo)
AS SELECT pseudo FROM Concoit;
```

- Une liste des newseurs n'ayant jamais posté de News.

$$\prod_{\text{pseudo}} (\sigma_{\text{nomGroupe}=\text{"Newseur"}}(\text{Appartient})) - \prod_{\text{pseudo}} (\text{Concoit})$$

```
SELECT pseudo FROM Appartient
WHERE nomGroupe="Newseur"
AND pseudo NOT IN redacteurEffectifNews;
```

- Liste des noms des sous-sections dont le nom est utilisé deux fois mais dans 2 sections différentes ?

$$\prod_{\text{nomSsSection}} \left(\begin{array}{ccc} \text{Sous-section} & & \text{Sous-section} \\ s1.\text{nomSousSection} & = & s2.\text{nomSousSection} \\ s1.\text{nomSection} & \neq & s2.\text{nomSection} \end{array} \right)$$

```
SELECT nomSsSection FROM Sous-section Ss1
WHERE EXISTS(SELECT *
              FROM Sous-section Ss2
              WHERE Ss1.nomSsSection=Ss2.nomSsSection
              AND Ss1.nomSection<>Ss2.nomSection);
```

- Liste des sujets de MP où 2 personnes (pseudo1 et pseudo2) apparaissent obligatoirement ?

$$\sigma_{\text{pseudo}=\text{pseudo1}}(\text{RecoitMP}) \underset{\text{noSujetMP}}{\infty} \sigma_{\text{pseudo}=\text{pseudo2}}(\text{RecoitMP})$$

```
SELECT noSujetMP FROM RecoitMP r1
WHERE EXISTS(SELECT pseudo
              FROM recoitMP r2
              WHERE r1.pseudo="Pseudo1"
              AND r2.pseudo="Pseudo2");
```

- Création d'une vue donnant les news conçues par deux personnes différentes mais utilisant une même source.

```
CREATE VIEW news2Personnes1Source (noNews, pseudo, nomSource, ref)
AS      SELECT * FROM Concoit c1
        WHERE EXISTS (SELECT * FROM Concoit c2
                      WHERE c1.nomSource=c2.nomSource
                      AND c1.pseudo<>c2.pseudo);
```

- Liste des news où une même source a été utilisée par deux personnes différentes.

$$\text{News} \underset{\text{noNews}}{\infty} \sigma_{\text{noNews}} \left(\begin{array}{ccc} \text{Concoit} & & \text{Concoit} \\ & \text{c1.nomSource} = \text{c2.nomSource} & \\ & \text{c1.pseudo} \neq \text{c2.pseudo} & \end{array} \right)$$

```
SELECT * FROM News
WHERE noNews IN (SELECT noNews FROM news2Personnes1Source);
```

- Quelle news ont été postées à une date donnée ? Et qui les a posté ?

$$\prod_{\substack{\text{noNews} \\ \text{titre} \\ \text{date}}} (\text{News}) \underset{\text{noNews}}{\infty} \prod_{\substack{\text{noNews} \\ \text{pseudo}}} (\sigma_{\text{date}=20-03-2014}(\text{Concoit}))$$

```
SELECT noNews, titre, jour, pseudo FROM News n1, Concoit c1
WHERE jour=20-03-2014
AND n1.noNews=c1.noNews;
```

- Quand telle source a-t-elle été utilisée ?

$$\prod_{\text{jour}} (\sigma_{\text{nomSource}=\text{"Le lorem Ipsum"}}(\text{Concoit}))$$

```
SELECT jour FROM Concoit
WHERE nomSource="Le lorem ipsum";
```

- Combien de commentaires a écrit une personne, toute news confondues ?

$$\prod_{\text{COUNT}(\text{noCom})} (\sigma_{\text{pseudo}=\text{"Pseudo3"}}(\text{Commentaire}))$$

```
SELECT COUNT(noCom) FROM Commentaire
WHERE pseudo="Pseudo3";
```

- Combien de commentaires ont écrit toutes les personnes inscrites, toute news confondues ?

$$\prod_{\text{pseudo}, \text{COUNT}(\text{noCom})} (\text{Commentaire})$$

```
SELECT pseudo, COUNT(noCom) FROM Commentaire
GROUP BY pseudo;
```

- Création d'une vue donnant le nombre de posts par personne.

```
CREATE VIEW nbPostParPersonne (pseudo, nb)
AS SELECT pseudo, COUNT(noPost) FROM Post
GROUP BY pseudo;
```

- Quelle est la personne ayant le plus posté sur le forum ? Et combien de posts a-t-il créé ?

```
SELECT pseudo, MAX(nb) FROM nbPostParPersonne;
```

- A quel groupe appartiennent les différentes personnes participant à une conversation par message privé ?

$$\prod_{\text{pseudo}} (\sigma_{\text{noSujetMP}=1}(\text{RecoitMP})) \underset{\text{pseudo}}{\infty} \prod_{\substack{\text{pseudo} \\ \text{nomGroupe}}} (\text{Appartient})$$

```
SELECT pseudo, nomGroupe FROM Appartient
WHERE pseudo IN (SELECT pseudo FROM RecoitMP
WHERE noSujetMP=1)
GROUP BY pseudo;
```

- Parmi les personnes ayant écrit un message dans ce sujet, lesquels sont modérateurs ?

$$\prod_{\text{pseudo}} (\sigma_{\text{idSuj}=2}(\text{Post})) \underset{\text{pseudo}}{\infty} \prod_{\text{pseudo}} (\sigma_{\text{nomGroupe}=\text{"Moderateur"}}(\text{Appartient}))$$

```
SELECT pseudo, nomGroupe FROM Appartient
WHERE nomGroupe="Moderateur"
AND pseudo IN (SELECT pseudo FROM Post
WHERE idSuj=2)
GROUP BY pseudo;
```

- Liste des posts inclus dans une sous-section donnée ?

$$\text{Post} \underset{\substack{\text{idSuj} \\ \text{idSuj}}}{\infty} \prod_{\text{idSuj}} (\sigma_{\text{nomSsSection}=\text{"Petits jeux"}}(\text{Sujet}))$$

```
SELECT * FROM Post
WHERE idSuj IN (SELECT idSuj FROM Sujet
                WHERE nomSsSection="Petits jeux");
```