

# Modélisation d'une base de données relationnelle

## Projet « Showroom »

### Objectif

Etre capable de modéliser une base de données et d'en déduire le code SQL-DDL.

### Présentation du projet

Dans les jours qui viennent, il vous est demandé de remplir le WIKI d'organisation sur Moodle en indiquant votre nom, votre prénom, votre thème pour l'application et le gabarit de page Web pré-existant choisi.

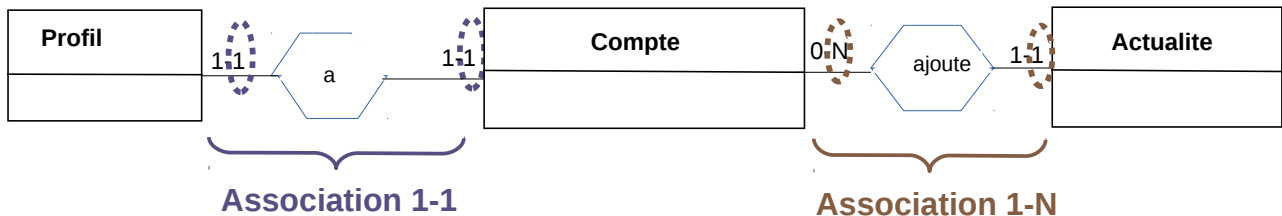
### Compréhension du cahier des charges

1. Qui est le client ?
2. Qui est le chef de projet ?
3. Quels seront les différents utilisateurs de l'application de showroom en ligne ?
4. A quelles données les différents utilisateurs de l'application pourront-ils avoir accès ?
5. Donnez au moins 3 exemples différents d'utilisation possible de l'application de showroom en ligne dans le cadre de notre Université. Quel sera le thème de votre showroom ?
6. Puis donnez un exemple de sélection et de ses éléments associés.
7. Qu'est-ce qu'un « lien » ? Donnez un exemple !
8. Quelle sera la particularité de la table de présentation de la structure ?
9. Comment se passe la gestion des actualités (ou « news ») ?  
→ Pour répondre à cette question, proposez des **exemples concrets (instances de classes)**.
10. Déduisez de votre réponse précédente, uniquement pour la gestion des actualités
  1. le diagramme UML de classes modélisant votre petite base de données,
  2. le schéma relationnel découlant de votre diagramme UML de classes précédent,
  3. le code SQL-DDL de création des différentes tables impliquées dans la gestion des actualités.
11. Puis préparez le diagramme UML de classes, le schéma relationnel puis le code SQL-DDL de la base de données complète à créer pour le projet « Showroom ».
12. Préparez aussi les requêtes SQL-DML d'insertion de lignes (enregistrements) dans les tables.

## 9. et 10. Comment se passe la gestion des actualités (ou « news ») ?

Le MCD (Modèle Conceptuel de Données) est le point de départ de la modélisation.

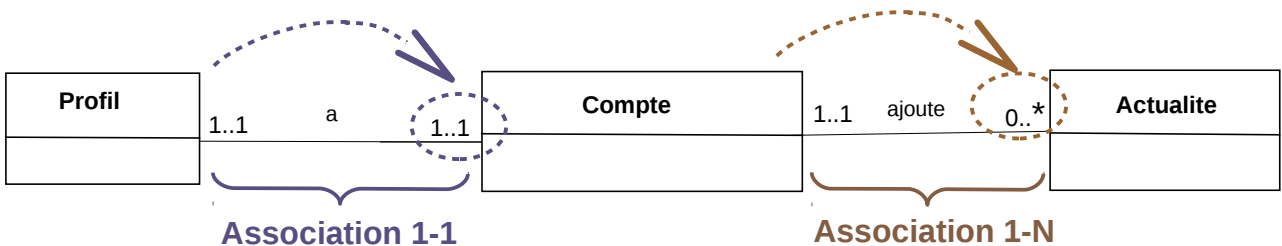
### 1. ((Schéma Entité-Association concernant la gestion des actualités (« news »)))



- Il y a 3 entités sur ce schéma E-A : *Profil*, *Compte*, *Actualite*.

**IMPORTANT** ⇒ A l'heure actuelle, on préfère utiliser le diagramme UML de classes pour modéliser une base de données relationnelle.

### 2. Diagramme UML de classes concernant la gestion des actualités (« news »)



**Rappel concernant le diagramme de classes UML :** les multiplicités concernant une classe sont inscrites à l'extrémité opposée de l'association (contrairement aux cardinalités sur un schéma E-A).

- Il y a 3 classes sur ce diagramme UML de classes : *Profil*, *Compte*, *Actualite*.
- Associations et multiplicités :

« 1 profil a au minimum 1 compte et au maximum 1 seul compte **et** 1 compte est détenu par 1 et 1 seul profil »

« 1 compte permet d'ajouter au minimum 0 et au maximum N actualités **et** 1 actualité est mise en ligne par 1 et 1 seul compte »

### 3. Détermination des multiplicités

RAPPEL : il s'agit de préciser

- le nombre **minimum et maximum** d'instances de la classe **B** pour chaque instance de la classe **A**  
et
- le nombre **minimum et maximum** d'instances de la classe **A** pour chaque instance de la classe **B**.

→ Pour faciliter cette opération, il est possible de s'aider d'instances des classes identifiées.

**Instances de la classe Profil**

| Pseudo / ID du Compte | Nom    | Prenom   | E-mail                | ... |
|-----------------------|--------|----------|-----------------------|-----|
| <b>vmarc</b>          | Marc   | Valérie  | vmarc@univ-brest.fr   | ... |
| <b>mdurand</b>        | Durand | Marcel   | mdurand@univ-brest.fr | ... |
| <b>lduval</b>         | Duval  | Laurence | lduval@univ-brest.fr  | ... |
| <b>dmasse</b>         | Masse  | Damien   | dmasse@univ-brest.fr  | ... |
| <b>gdupond</b>        | Dupond | Gérard   | gdupond@univ-brest.fr | ... |
| ...                   | ...    | ...      | ...                   | ... |

Associée à l'instance de l'entité Profil ayant le prénom « Valérie » et le nom « MARC »

**Instances de la classe Compte**

| Pseudo / ID    | Mot de passe |
|----------------|--------------|
| <b>vmarc</b>   | *****        |
| <b>mdurand</b> | *****        |
| <b>lduval</b>  | *****        |
| <b>dmasse</b>  | ...          |
| <b>gdupond</b> |              |
| ...            |              |

**Instances de la classe Actualite**

| Pseudo / ID du Compte | Numero   | Texte  | Date de publication |
|-----------------------|----------|--|---------------------|
| <b>vmarc</b>          | <b>1</b> | C'est la rentrée !   | 01/09/18 à 9h       |
| <b>vmarc</b>          | <b>2</b> | Notre showroom a fait peau neuve !                                   | 02/09/18 à 14h      |
| <b>lduval</b>         | <b>3</b> | Nouveau : les projets des étudiants de bio-informatique !            | 09/09/18 à 9h       |
| <b>lduval</b>         | <b>4</b> | Votez pour votre projet préféré par e-mail : dept-info@univ-brest.fr | 10/09/18 à 14h      |
| <b>vmarc</b>          | <b>5</b> | Bientôt de nouveaux projets !  | 12/09/18 à 11h      |
| ...                   | ...      | ...  | ...                 |

#### 4. Schéma relationnel concernant la gestion des actualités

Il faut ensuite traduire le schéma conceptuel en schéma relationnel, en respectant les règles de traduction suivantes :

| Classe et attributs  | ----->  | Commentaires   |
|--|---|--|
| <div> <div>CLIENT</div> <div>NumClient</div> <div>id: NumClient</div> </div> | <div> <div>t_client_cli</div> <div>P cli_id</div> <div>t_client_cli_PK(cli_id)</div> </div> | Chaque classe est traduite par une table et chaque attribut est traduit en une colonne de la table |

| Associations  | ----->  | Commentaires  |
|---|---|---|
| <b>un-à-plusieurs (1:N)</b><br><div> <div>A</div> <div>1..1</div> <div>est associé à</div> <div>0..*</div> <div>B</div> </div> <div> <div>IdA</div> <div>id:IdA</div> </div>        | <div> <div>t_A_a</div> <div>P a_id</div> <div>t_A_a_PK(a_id)</div> </div> <div> <div>t_B_b</div> <div>F a_id</div> </div>   | ⇒ Traduction d'une association <b>1:N</b> par une <b>clé étrangère</b> vers la clé primaire de l'autre table  |
| <b>un-à-un (1:1)</b><br><div> <div>A</div> <div>1..1</div> <div>est associé à</div> <div>0..1</div> <div>B</div> </div> <div> <div>IdA</div> <div>id:IdA</div> </div>               | <div>1)</div> <div> <div>t_A_a</div> <div>P a_id</div> <div>t_A_a_PK(a_id)</div> </div> <div> <div>t_B_b</div> <div>PF a_id</div> <div>t_B_b_PK(a_id)</div> </div> <div>2)</div> <div> <div>t_A_a</div> <div>P a_id</div> <div>t_A_a_PK(a_id)</div> </div> <div> <div>t_B_b</div> <div>P b_id</div> <div>FU a_id</div> <div>t_B_b_PK(b_id)</div> </div> | ⇒ Traduction d'une <b>1..1 ---- 0..1</b> par<br><b>1) une clé étrangère identifiante</b><br>ou<br><b>2) une clé étrangère UNIQUE</b>                              |
| <div> <div>A</div> <div>1..1</div> <div>est associé à</div> <div>1..1</div> <div>B</div> </div> <div> <div>IdA</div> <div>id:IdA</div> </div>                                       | <div>t_AB_ab</div> <div>P ab_id</div> <div>t_A_a_PK(ab_id)</div>  | ⇒ Traduction d'une <b>1..1 ---- 1..1</b> par une seule table  |
| <b>plusieurs-à-plusieurs (N:N)</b><br><div> <div>A</div> <div>0..*</div> <div>est associé à</div> <div>0..*</div> <div>B</div> </div> <div> <div>IdA</div> <div>id:IdA</div> </div> | <div> <div>t_A_a</div> <div>P a_id</div> <div>t_A_a_PK(a_id)</div> </div> <div> <div>t_B_b</div> <div>P b_id</div> <div>t_B_b_PK(b_id)</div> </div> <div> <div>t_R_r</div> <div>F a_id</div> <div>F b_id</div> <div>t_R_r_PK(a_id,b_id)</div> </div>  | ⇒ Traduction d'une association <b>N:N</b> par la création d'une <b>table associative</b> munie de <b>2 clés étrangères</b> et <b>d'une clé primaire COMPOSITE</b> |

A noter :

**P** : Primary Key (clé primaire)

**F** : Foreign Key (clé étrangère)

**PF** : Primary/Foreign Key (clé étrangère identifiante)

Il est aussi demandé de respecter les conventions de nommage des tables et des colonnes suivantes :

**Table « Vehicule » :**

**t\_vehicule\_veh**

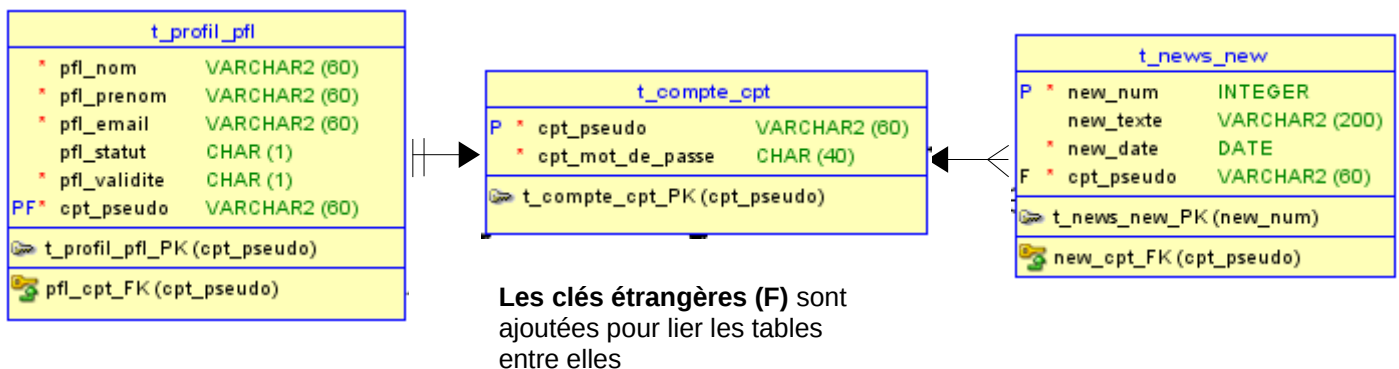
t\_NomTable\_Code3Caractères

**Colonne « Identifiant » de la table « Vehicule » :**

**veh\_id**

Code3CaractèresTable\_NomColonne

→ On en déduit le schéma relationnel suivant :



## 5. Code SQL-DDL concernant la gestion des actualités

On en déduit enfin le code SQL-DDL de création des tables et d'ajout des contraintes de table (clés primaires et étrangères).

```
CREATE TABLE t_compte_cpt
(
    cpt_pseudo          VARCHAR (60) NOT NULL ,
    cpt_mot_de_passe    VARCHAR (40) NOT NULL
);
ALTER TABLE t_compte_cpt ADD CONSTRAINT t_compte_cpt_PK PRIMARY KEY ( cpt_pseudo
);

CREATE TABLE t_profil_pfl
(
    pfl_nom             VARCHAR (60) NOT NULL ,
    pfl_prenom          VARCHAR (60) NOT NULL ,
    pfl_email           VARCHAR (60) NOT NULL ,
    pfl_statut          CHAR (1) NOT NULL,
    pfl_validite        CHAR(1) NOT NULL ,
    cpt_pseudo          VARCHAR (60) NOT NULL
);
ALTER TABLE t_profil_pfl ADD CONSTRAINT t_profil_pfl_PK PRIMARY KEY ( cpt_pseudo
);

CREATE TABLE t_news_new
(
    new_num             INTEGER NOT NULL ,
    new_texte           VARCHAR (200) ,
    new_date            DATE NOT NULL ,
    cpt_pseudo          VARCHAR (60) NOT NULL
);
```

## *Bases de données*

```
ALTER TABLE t_news_new ADD CONSTRAINT t_news_new_PK PRIMARY KEY ( new_num ) ;
```

```
ALTER TABLE t_profil_pfl ADD CONSTRAINT t_pfl_t_cpt_FK FOREIGN KEY  
( cpt_pseudo ) REFERENCES t_compte_cpt ( cpt_pseudo ) ;
```

```
ALTER TABLE t_news_new ADD CONSTRAINT t_new_t_cpt_FK FOREIGN KEY ( cpt_pseudo )  
REFERENCES t_compte_cpt ( cpt_pseudo ) ;
```