420-KB6-LG

Le modèle relationnel

Saliha Yacoub

Le modèle relationnel

- Un modèle de données est un ensemble de concepts utilisés pour décrire la structure d'une base de données.
- Par structure de base de données, nous entendons: les tables, les types de données, les relations, et les contraintes qui définissent le gabarit de la base de données.
- Tous les SGBDs offrent des outils qui permettent de modéliser la base de données.
- Certains outils permettent même de générer le code SQL issu du modèle de données. Comme l'outil Oracle Data Modeler.

Le modèle relationnel

- **Rappels:**

On appelle modèle relationnel, le modèle qui représente les tables et les liens entre les tables.

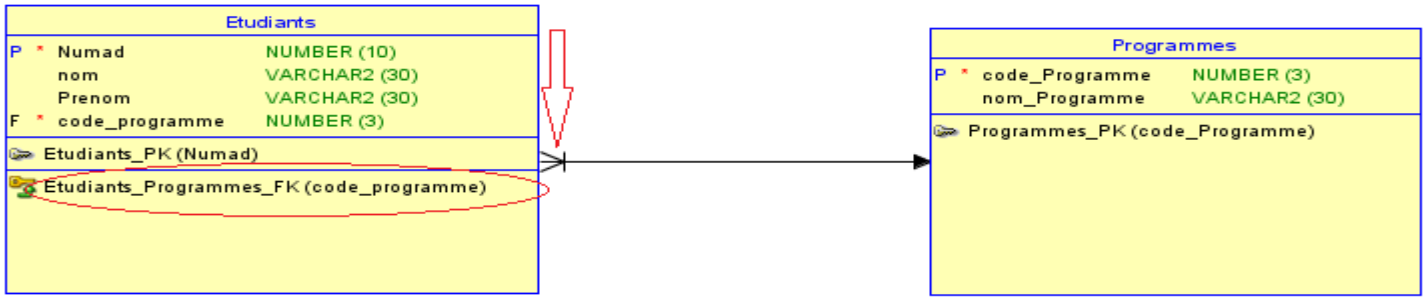
Lors de la création des tables, le lien entre deux tables se matérialise par le concept de Foreign Key (ou clé étrangère).

1. Lorsqu'il y a une relation entre deux tables, alors la table qui reçoit la clé étrangère est appelé table ENFANT. La table d'où vient la clé étrangère est appelé table PARENT.
2. Lorsqu'il y a une relation entre deux tables, alors la table qui reçoit la clé étrangère est appelé table CIBLE. La table d'où vient la clé étrangère est appelé table SOURCE

Les deux définitions sont équivalentes.

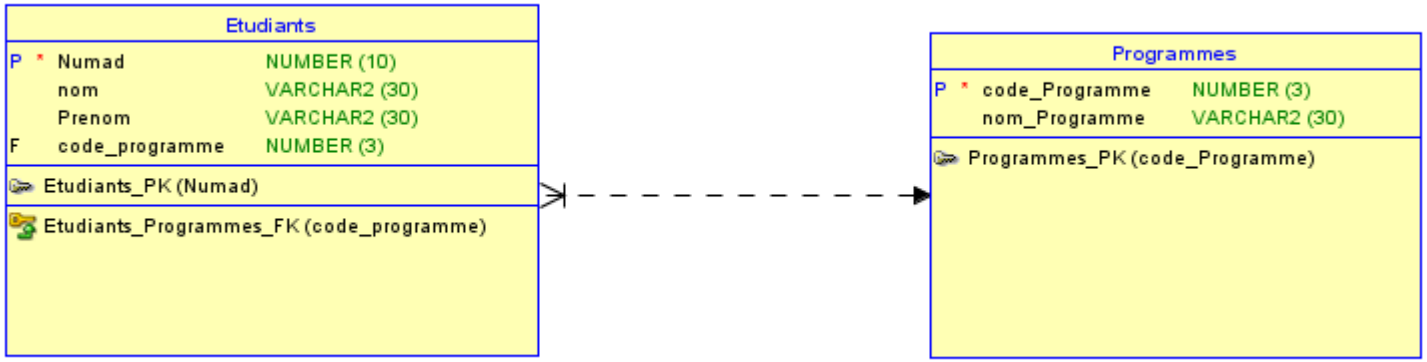
Le modèle relationnel

- Exemple 1: (**le modèle M1)**, dans cet exemple, on voit:
  - Deux tables avec chacune sa clé primaire.
  - Dans la table Etudiants, on voit bien la contrainte de FK.
  - Il y a un lien entre les deux tables. Ce lien a ces caractéristiques:
    - Un trait plein
    - Une fourche



Le modèle relationnel

- Exemple 2: (**le modèle M2)**, dans cet exemple, on voit:
  - Deux tables avec chacune sa clé primaire.
  - Dans la table Etudiants, on voit bien la contrainte de FK.
  - Il y a un lien entre les deux tables. Ce lien a ces caractéristiques:
    - Un trait en pointillé
    - Une fourche



Le modèle relationnel

- Fourche ? Trait plein, trait en pointillé ? Y a-t-il d'autre choses ? Et ça veut dire quoi ?
- Oui, il y a d'autres choses qu'on va voir plus loin. Et ça veut dire cardinalités.
- Cardinalité: c'est quoi ? Une cardinalité est le nombre de fois minimum et maximum qu'une occurrence d'une table participe à une relation.

Je vous comprends!!!! cette définition de cardinalité n'est pas claire du tout.

Au fait, lorsque vous créez une table, par exemple la table Programmes, vous connaissez tout sur la table (attributs, types, PK…etc).

Vous savez, entre autres, :

- Que l'attribut clé primaire est UNIQUE et NOT NULL. Donc, l'attribut code_Programme dans la table Programmes est UNIQUE et NOT NULL (car c'est la clé primaire).

Le modèle relationnel

- Mais qu'en est-il de l'attribut code_programme dans la table ETUDIANTS ? Est-il unique, est-il not null ?
- La cardinalité sert à ça. À déterminer comment va se comporter l'attribut de la FK.
- Si vous avez une FOURCHE cela veut dire que l'attribut de la FK est MULTIPLE (dans notre cas, cela veut dire que le code_programme dans la table Etudiants est MULTIPLE).
- Multiple veut dire NON UNIQUE.
- MULTIPLE veut dire que les valeurs de l'attribut de la FK peut se retrouver plusieurs fois.
- Dans notre cas, cela veut dire que dans la table Etudiants, une valeur de l'attribut code_programme peut se retrouver plusieurs fois.
- Ce qui veut dire, par exemple, qu'on peut trouver plusieurs fois le code 420 dans la table ETUDIANTS. Ce qui est normal, car il y a beaucoup d'étudiants en informatique.

Le modèle relationnel

- Donc la fourche, détermine la multiplicité.
- Mais comment savoir si l'attribut de la FK est NOT NULL ? Comment savoir si l'attribut code_programme dans la table ETUDIANTS est NOT NULL ?
- Ça… c'est le rôle du trait.
- Lorsqu'il est PLEIN, cela veut dire l'OBLIGATION.
- Lorsqu'il est pointillé, cela veut dire optionnel.
- Pour nous, lorsque le trait est plein, cela veut dire qu'un étudiant est OBLIGÉ d'être dans un programme.
- Pour nous, lorsque le trait est plein, cela veut dire: lors de la création de la table ETUDIANTS, le code_Programme est NOT NULL.

Le modèle relationnel

- Voici le code SQL issu du modèle M1(Trait plein)

```
CREATE TABLE etudiants (
    numad           NUMBER(10) NOT NULL,
    nom             VARCHAR2(30),
    prenom          VARCHAR2(30),
    code_programme   NUMBER(3) NOT NULL
);

ALTER TABLE etudiants ADD CONSTRAINT etudiants_pk PRIMARY KEY ( numad );
```

Le modèle relationnel

Suite du code:

```
CREATE TABLE programmes (
    code_programme   NUMBER(3) NOT NULL,
    nom_programme    VARCHAR2(30)
);

ALTER TABLE programmes ADD CONSTRAINT programmes_pk PRIMARY KEY ( code_programme );

ALTER TABLE etudiants
    ADD CONSTRAINT etudiants_programmes_fk FOREIGN KEY ( code_programme )
        REFERENCES programmes ( code_programme );
```

Le modèle relationnel

- Voici le code SQL issu du modèle M2(Trait pointillé, il n'y a pas le NOT NULL)

```
CREATE TABLE etudiants (
    numad           NUMBER(10) NOT NULL,
    nom             VARCHAR2(30),
    prenom          VARCHAR2(30),
    code_programme   NUMBER(3)
);
 ALTER TABLE etudiants ADD CONSTRAINT etudiants_pk PRIMARY KEY ( numad );
```

Le modèle relationnel

Suite du code:

```
CREATE TABLE programmes (
    code_programme   NUMBER(3) NOT NULL,
    nom_programme    VARCHAR2(30)
);

ALTER TABLE programmes ADD CONSTRAINT programmes_pk PRIMARY KEY ( code_programme );

ALTER TABLE etudiants
    ADD CONSTRAINT etudiants_programmes_fk FOREIGN KEY ( code_programme )
        REFERENCES programmes ( code_programme );
```

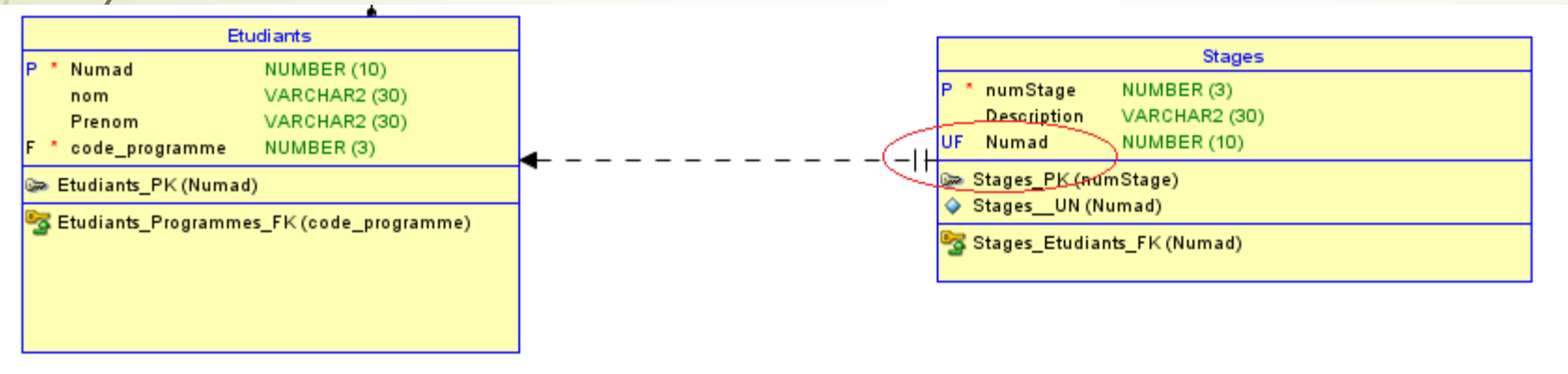
Le modèle relationnel

Autre exemple:

Dans le modèle suivant, nous n'avons pas la fourche. Nous avons deux barres | |

Ces deux barres veulent dire UNIQUE.

Ce qui veut dire que le NUMAD qui est dans STAGES est UNIQUE. Le même stage ne peut être attribué à deux étudiants.



Le modèle relationnel

```
CREATE TABLE stages (
    numstage      NUMBER(3) NOT NULL,
    description   VARCHAR2(30),
    numad         NUMBER(10)
);
ALTER TABLE stages ADD CONSTRAINT stages_pk PRIMARY KEY ( numstage );
ALTER TABLE stages ADD CONSTRAINT stages__un UNIQUE ( numad );

ALTER TABLE stages
    ADD CONSTRAINT stages_etudiants_fk FOREIGN KEY ( numad )
        REFERENCES etudiants ( numad );
```

Le modèle relationnel

On résume:

- **Règle R1:** Lorsqu'il y a un lien entre deux tables, du côté de la table ENFANT (la table de la FK) soit:
  - Nous avons une FOURCHE, ce qui veut dire que l'attribut de la FK est MULTIPLE.
  - Soit nous avons deux barres | |, ce qui veut dire que l'attribut de la FK est UNIQUE.
- **Règle R2:** Lorsque nous avons un lien entre deux tables, ce lien est soit:
  - Un trait plein: ce qui veut dire que l'attribut de la FK est OBLIGATOIRE
  - Un trait en pointillé: ce qui veut dire que l'attribut de la FK est optionnel.
- Le sens de la flèche indique la référence (on l' a vu en classe).
- Dans le modèle relationnel d'ORACLE, la fourche est du côté d'UNE SEULE TABLE.

Le modèle relationnel

Comment lire un modèle relationnel ?

Par la lecture de chacune des relations. Il faudra tenir compte des cardinalités.

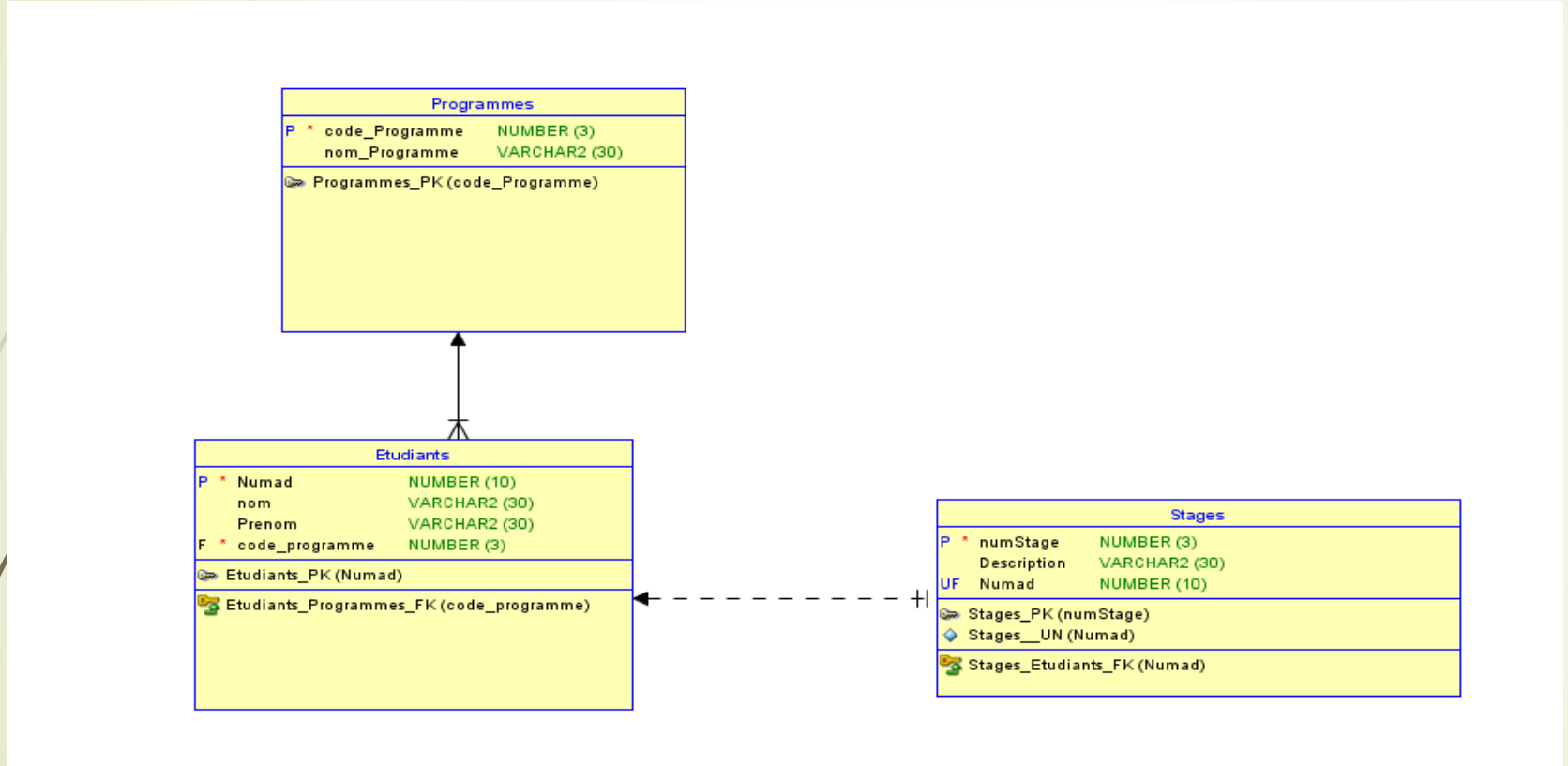
Par exemple, pour le modèle suivant (page suivante), on lira:

1. Un étudiant est OBLIGÉ d'être dans un programme, car c'est un trait plein.
2. Dans un programme, nous avons plusieurs étudiants (fourche du côté de l'étudiant).

Pour les stages:

1. Un étudiant a un seul STAGE (pas de fourche du côté de stage on a | |).
2. Un stage peut ne pas être affecté (peut ne pas avoir d'étudiant, trait pointillé).

Le modèle relationnel



Le modèle relationnel

Comment faire un modèle relationnel ?

1. D'abord télécharger Oracle Data Modeler .

2. Puis lire le cours : http://www.salihayacoub.com/420kb6/Semaine%2010/MR.pdf

3. Et il faut apprendre à normaliser ses données, car sinon le modèle ne vaut absolument rien. Pour cela, suivre le cours : Normalisation.