

# Conception Orientée Objet

## Objectifs du Cours

prof : Kim Mens

## Thèmes

- » Introduction aux bases de données et modélisation de données
- » Conception de programmes orientés objet
- » Méthodologie d'aide au développement de programmes
- » Réalisation ( analyse et implémentation ) de programmes Java ( android ) de complexité moyenne

## Durée

**9 semaines** à raison de 2H/semaine

## Projet - EZMeal

1. Ecran de Login
2. Gestion du profil utilisateur
3. Menu d'accueil
4. Recettes recommandées
5. Catalogue de recettes
6. Détail d'une recette
7. Recherche de recettes
8. différentes extensions possibles

## Contenu du cours

1. Gestion de données
2. Conception orientée Objet
3. Programmation Android

## Evaluation

50 % projet et 50 % Travail  
Il faut un minimum de 10/20 à chacun d'entre eux pour réussir

## Examen

Modélisation et conception d'une application similaire à l'étude de cas développée dans les séances pratiques

5 questions sur des sujets vus au cours

ORM, SQL, diagrammes de classes, diagrammes de séquences, ...

- » Question type: à partir d'un modèle incomplet
- » corrigez / critiquez / discutez ce modèle
- » complétez ou raffinez le modèle pour une extension donnée

## Introduction

**Une Base de Donnée** : est un système informatique de stockage d'informations. Les plus utilisés s'appuient sur le **modèle relationnel** et utilisent le **language SQL**

```
select Item1, Item2, Item3
from DataBase
where condition
```

ceci retourne un tableau de 3 colonnes et d'autant de lignes qu'il y a d'éléments dans la BD qui satisfont la condition.

```
select D1.ITEM2, D2.ITEM3, sum(D.QUANTITE*P.PRIX)
from CLIENT D1 , DETAIL D , PRODUIT P
where C.CLI + D.NCLI
and D.NPRO = P.NPRO
group by C.LOCALITE, P.NPRO
```

Une **base de données** est constituée d'un ensemble de **tables**.

Une **table** contient les données relatives à une **collection d'entités** de même nature.

Chaque **ligne** d'une table reprend les données relatives à une **entité**.

Chaque **colonne** d'une table décrit une propriété commune des entités.

Un **identifiant** de la table est le jeu de colonnes dont les valeurs sont uniques.

Une **clé étrangère** vers cette autre table sont les lignes d'une table peuvent faire référence chacune à une ligne d'une autre table.

On évite d'enregistrer des données qu'il est possible de calculer à partir d'autres données enregistrées.

## Les SGBD ( Systèmes de Gestion de Bases de Données )

Un système de gestion de bases de données est un système informatique permettant de gérer une BD.(maintenir la DB & répondre à des requêtes)

- » Organisation des données
- » Gestion des données
- » Accès aux données
- » Protection contre les accidents
- » Gestion des accès concurrents
- » Contrôle des accès

## SQLite

Le moteur de **base de données relationnelles** est accessible par le langage SQL.

- » il doit être intégré aux programmes et n'a pas besoin de serveur séparé.
- » Sa particularité est de stocker toutes ses données dans un seul fichier .sqlite sur le disque
- » SQLite est utilisé pour les données des applications de type Android et iPhone

## Univers de discours (UoD)

L'UoD décrit le domaine de l'application. Cet univers décrit typiquement une partie du monde "réel".

## ORM

**ORM** = Object-Role Modeling

**Base de données** = ensemble de relations n-aires

**Schéma de base de données** = constitué de l'ensemble des signatures des relations de la base.

Le modèle relationnel ne se base pas sur la notation positionnelle (1,2,3... comme un tableau) mais sur les identificateur pour identifier les entités.

L'ORM est *fact-oriented* = modélise l'info comme des **faits** suivant des **règles**.

## Exemples

(12,Antoine,Lambert,SINF12BA) : **Tuple**

(ID,NOM,PRENOM,UCL\_INFO) : **Shéma de la relation**

STUDENT(ID,NOM,PRENOM,UCL\_INFO) : **Signature de la relation n-aire**

## Modélisation ORM

Suivant le processus *Conceptual Schema Design Procedure*

1. Trouver les faits élémentaires à partir des exemples de données

- » prendre des exemples concrets
- » les transformer en faits élémentaires

**Les entités** : Type('Client') - (Mode de référence('nom'))- Valeur('Bob')

**Les faits élémentaires** expriment des relations entre les objets et ne peuvent être divisés.

Ils peuvent être unaire (propriété de l'objet[...est...]), binaire (relation entre 2 objets[...a...]) ou n-aire

1. Dessiner les types de faits et ajouter une population au diagramme >  
**entités** : cercles plein (entité) > **valeurs** : cercle pointillé ou (valeur) en dessous des entités : (...valeur...) > référencement des éléments : -[][- > rôle : []
2. Combiner des types d'entités et noter des dérivations arithmétiques
3. Ajouter des contraintes d'unicité et vérifier les arités des types de faits  
**Les contraintes d'unicité** : définissent les éléments qui peuvent apparaître 2 fois dans une base de donnée et ceux qui sont uniques  
>**Attention !!** : l'emplacement des contraintes est l'inverse du diagramme UML

L'arité des contraintes doit toujours être de n-1. Sinon, soit il y a une erreur soit on peut subdiviser

-°(Truc) : cet élément est obligatoire (en uml l'autre à une arité de 1 et non 0...1)

cercle bleu signifie 'doit avoir'

(Etudiant (Noma)) = (Etudiant) -[<->, <->] - (Noma)

En DB, on utilise souvent les ID's pour les données. mais ceux-ci ne doivent pas être indiqués dans l'ORM

En schéma : les contraintes d'unicité sont représentées par un soulignement de l'élément

**Clé candidate** : colonne/ combinaison de colonnes qui permet d'identifier de manière unique les éléments dans la DB.

*“ Dans le cas ou plusieurs clé candidates existent, on doit en déterminer qui sera la clé primaire. cette clé sera souligné deux fois.*

*”*

si deux éléments d'une clé candidate ne sont pas côte à côte, on peut utiliser les fleches pour indiquer que la colonne en question n'est pas une clé candidate à elle seule.

Une clé candidate ne permet pas de valeur nulle

## Règles d'intégrité du modèle relationnel

### 1. Règle d'intégrité des entités

Clé primaire != null

## 2. Règle d'intégrité référentielle

chaque valeur non-nulle d'une clé étrangère doit correspondre à la valeur d'une clé primaire.

## Rmap

\< Skip >

## Cours S4 - notes

Le sous langage DDL permet de créer des structures de données et les modifier.

Data Description Language

## cours S6 - notes

UML = Unified Modelling language - langage architecture (ex: package) - UML

class diagrams - UML sequence diagrams -... User stories = récit utilisateurs