|  |  |
| --- | --- |
|  | **Conception & Développement Informatique**  **Développer les composants d’accès aux données**  Apprentissage | Travaux pratiques |

**Création d’un site e-commerce**

Merise, HTML, CSS, Javascript, PHP, MySQL

Contenu

[Compétences mises en œuvre 1](#_Toc535777317)

[Gestion de projet : La démarche Merise 1](#_Toc535777318)

[Contexte du projet 1](#_Toc535777319)

[Gestion du catalogue de produits 2](#_Toc535777320)

[Analyse du texte 2](#_Toc535777321)

[Dictionnaire de données 2](#_Toc535777322)

[Modèle Conceptuel de Données 2](#_Toc535777323)

[Modèle Logique de Données 3](#_Toc535777324)

[Modèle Physique de Données 3](#_Toc535777325)

[Crédits 4](#_Toc535777326)

# Compétences mises en œuvre

* Concevoir une base de données en utilisant la démarche Merise.
* Mettre en place une base de données MySQL.
* Développer et mettre en forme des pages web avec HTML et CSS.
* Développer les composants d’accès aux données avec PHP.
* Mettre en lien les pages web avec les données MySQL.

# Gestion de projet : La démarche Merise

La méthode Merise est une méthode d'analyse, de conception et de réalisation de systèmes d'informations.

Nous nous intéresserons particulièrement à la gestion des données de la méthode Merise.

**La méthode Merise est découpée en plusieurs étapes :**

1. Le niveau **Conceptuel** s’intéresse aux variables métiers (quelles sont les données à manipuler ?) décrit par le dictionnaire de données et le Modèle Conceptuel de Données.
2. Le niveau **Logique**,décrit par le Modèle Logique de Données, définit la structure et l’organisation des données définies au niveau conceptuel telles qu’elles pourront être implémentées.
3. Le niveau **Physique** détermine le Système de Gestion de Bases de Données (SGBD) à utiliser et implémente le modèle logique généralement avec le langage SQL.

# Contexte du projet

Une entreprise locale fait appel à nos services dans le but d’informatiser son catalogue de produits afin de les vendre sur le web.

L’équipe commerciale nous indique que le projet se déroulera en 5 étapes :

1. Mettre à disposition un site web simple et rapide permettant aux employés de gérer un catalogue de produits classés par catégories. Un modèle de l’interface web à réaliser est fourni.
2. Permettre aux employés de gérer une base de données de clients.
3. Permetrre aux employés d’inscrire de nouvelles commandes au nom d’un client.
4. Créer un site web public où les clients pourront directement passer commande.
5. Ajouter une gestion fine des caractéristiques des produits.

# Gestion du catalogue de produits

Soit la description métier suivante :

Tous les produits du catalogue sont classés par catégories. Une catégorie est caractérisée par un nom et une description. Un produit est décrit par un nom, une description, un prix d’achat, un prix de vente. Un produit doit être lié à une et une seule catégorie.

## Analyse du texte

Pour bien définir les données avec lesquelles il sera nécessaire d’interagir, nous allons utiliser le texte précedent pour définir les différentes entités ainsi que leur caractéristiques. Une entité est un objet contenant un certain nombre d’attributs un peu à la manière d’une classe dans un diagramme UML. En règle générale, le nom d’une entité est au pluriel. Il pourra, par la suite, exister une ou plusieurs occurrences de chaque entité

Dans le texte précédent, deux entités (catégories, produits) ont été identifiées et surlignées en jaune. Pour chacune de ces entités, nous avons identifié leurs attributs (surlignés en vert pour l’entité categories et en bleu pour l’entité produits).

## Dictionnaire de données

Les informations recueillies sont reportées dans un tableau appelé le dictionnaire de données.

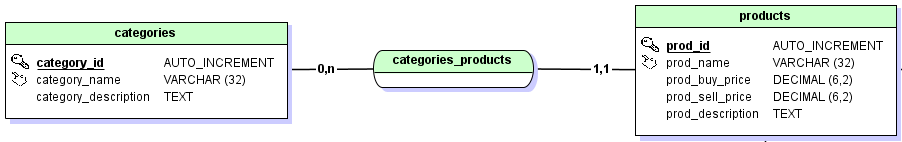
Ce dictionnaire permet de spécifier différentes caractéristiques à nos attributs tels que leur nom, type, longeur, taille et éventuelles contraintes.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Entité | Mnémonique | Description | Type | Taille | Remarques |
| categories | category\_id |  | N | 11 | identifiant, auto\_increment |
| categories | category\_name |  | AN | 32 | unique |
| categories | category\_description |  | AN | 65535 | nullable |
| products | prod\_id |  | N | 11 | identifiant, auto\_increment |
| products | prod\_name |  | A | 32 |  |
| products | prod\_buy\_price |  | N | 6,2 | >0 |
| products | prod\_sell\_price |  | N | 6,2 | >prod\_buy\_price |
| products | prod\_description |  | AN | 65535 | nullable |

Remarquez la présence de deux attributs rajoutés (category\_id, prod\_id). Ils nous serviront d’identifiant pour leur entité respective.

En effet, chaque entité doit posséder un identifiant

## Modèle Conceptuel de Données



## Modèle Logique de Données

**categories(category\_id, category\_name,** category\_description**)**

**products(prod\_id, prod\_name, prod\_buy\_price, prod\_sell\_price,** prod\_description**, #category\_id)**

### Modèle Physique de Données

Le modèle physique de données correspond aux scripts ecrits avec le langage SQL.

DDL : Data Definition Language

|  |
| --- |
| CREATE TABLE categories(  category\_id Int Auto\_increment NOT NULL ,  category\_name Varchar (32) NOT NULL,  category\_description Text NULL ,  CONSTRAINT categories\_Idx INDEX (category\_name),  CONSTRAINT categories\_PK PRIMARY KEY (category\_id)  )ENGINE=InnoDB;  CREATE TABLE products(  prod\_id Int Auto\_increment NOT NULL ,  prod\_name Varchar (32) NOT NULL ,  prod\_buy\_price Decimal (6,2) NOT NULL ,  prod\_sell\_price Decimal (6,2) NOT NULL ,  prod\_description Text NOT NULL ,  category\_id Int NOT NULL,  CONSTRAINT products\_Idx INDEX (prod\_name),  CONSTRAINT products\_PK PRIMARY KEY (prod\_id),  CONSTRAINT products\_categories\_FK FOREIGN KEY (category\_id) REFERENCES categories(category\_id)  )ENGINE=InnoDB; |

# Crédits

<http://www.arfp.asso.fr>

--- FIN DU DOCUMENT ---

|  |  |
| --- | --- |
| Légende des icônes | |
|  | Information complémentaire |
|  | Point d’attention particulier |
|  | Intervention du formateur possible |
|  | Lien vers une ressource externe |