

Projet de DSB
RAPPORT : Galerie d'Art
JULLION – LECORDIER
Promotion 2018-2019



Sommaire

Introduction / Objectif.....	4
I-Jeu de données.....	4
II-Modele Conceptuel/Physique.....	5
III-Relationel/SQL.....	6
A-Selection avec Projection.....	7
B-Jointure.....	7
C-Moyenne sur intégralité d'un Attribut.....	7
D-Regroupement avec calcul.....	8
E-Différence.....	8
F-Division.....	8
IV-PHP.....	9
V-DTD/XML.....	9
VI-Remarque.....	9

Introduction / Objectif

L'objectif du Projet en DSB est de modéliser et construire une base de données sur un thème au choix. Pour ensuite, manipuler la base et ces données grâce aux différents langages et méthodes de représentation des données vues en cours. Cela passera donc par plusieurs étapes importantes : dans un premier temps, quel est notre jeu de données et que voulons nous faire avec. Ensuite il faudra réaliser des schémas Conceptuel et Physique, représentant cette bdd sous différentes formes. Il y aura aussi l'implémentation de nos données et la création de table contenant celles-ci à l'aide de MySQL, puis l'interaction avec les données grâce à des requetes. Enfin, nous utiliseront le format DTD et XML pour varier la représentation des données.

Durant tout ce projet, nous illustrerons nos taches effectuées avec un site XHTML(contenant du CSS et du PHP) qui contiendra tout nos résultats, schémas et données.

I-Jeu de données

Nous avons choisis comme thème de ce projet : la peinture. La modélisation la plus propice pour intégrer ce thème fut donc une galerie d'art. Malheureusement, nous n'avons pas trouvés de CSV contenant déjà une modélisation de galerie, ce qui expliquera donc pourquoi le nombres de n-uplets sera moyen. Des recherches personnels ont été effectuées pour récupérer des informations autour d'un tableau et ses éléments clés.

Voici un exemple de donnée extraite lors de nos recherches :

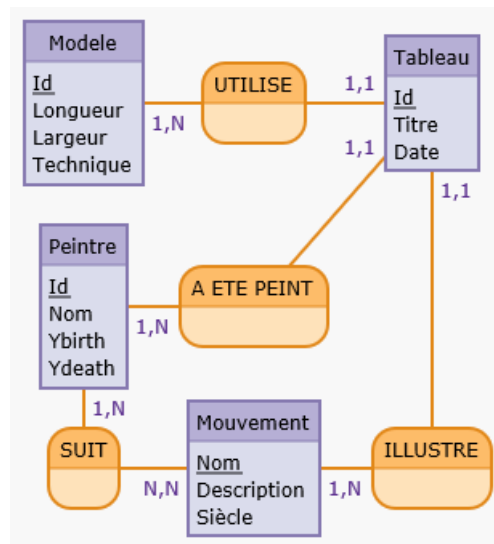
Sandro Botticelli 1444-1510	Le printemps 1478/1482	Tempera sur panneau de bois 203x314cm	Première renaissance italienne
--------------------------------	---------------------------	------------------------------------------	-----------------------------------

Nous avons donc réunis un tableau d'environ 15 lignes comme celle-ci. Il nous reste donc à organiser toutes ces informations sous forme d'ensemble de table bien construite et cohérente

II-Modele Conceptuel/Physique

Pour le modèle conceptuel, nous avons donc divisé nos données dans 4 catégories, puis les reliées entre elles avec le système de clé primaire et étrangère. Nous avons donc une table Modèle, une table Peintre, une table mouvement et une table tableau, qui possédera l'identifiant du modèle du peintre et du modèle.

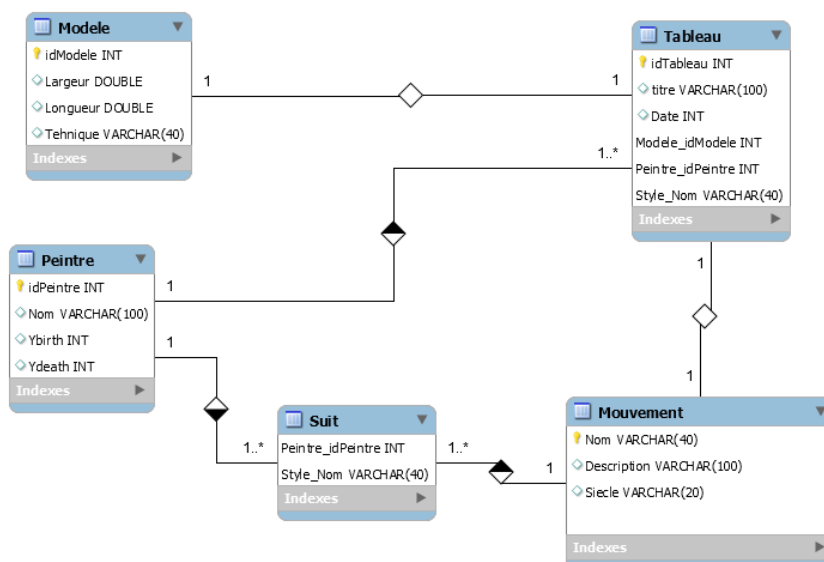
Voici le modèle relationnel/Merise, généré par l'outil MOCODO Online :



Voici le modèle MPD, construit avec les règles vues en cours sur la conversion de MCD à MPD:

Modele(ID, Largeur, Longueur, Technique)
Peintre(ID, Nom, Ybirth, Ydeath)
Mouvement(Nom, Description, Siècle)
Tableau(ID, Titre, Date, ID_Modele, ID_Peintre, Nom_Mouvement)
Suit(ID_Peintre, Nom_Mouvement)

Avec le schéma correspondant, fait sur MySQL Workbench :



III-Relationnel/SQL

Puisque nous avons implémentés ces tables sur PhpMyAdmin en SQL, voici les commandes exécutées pour chacune des tables.

Nom de la table	Requêtes CREATE TABLE
Modele	CREATE TABLE Modele (ID INT PRIMARY KEY NOT NULL, Longueur double, Largeur double, Technique varchar(40))
Peintre	CREATE TABLE Peintre (ID INT PRIMARY KEY NOT NULL, nom varchar(100), ybirth INT, ydeath INT)
Mouvement	CREATE TABLE Mouvement(Nom varchar(40) PRIMARY KEY NOT NULL, Description varchar(100), Siecle varchar(25))
Tableau	CREATE TABLE Tableau (ID INT PRIMARY KEY NOT NULL, Titre varchar(100), Date_ Date, Modele_ID INT, FOREIGN KEY (Modele_ID) REFERENCES Modele(ID), Peintre_ID INT, FOREIGN KEY (Peintre_ID) REFERENCES Peintre(ID), Nom_Mouvement varchar(40), FOREIGN KEY (Nom_Mouvement)REFERENCES Mouvement(Nom))
Suit	CREATE TABLE suit(ID_Peintre INT, Nom_Mouvement varchar(25), FOREIGN KEY (ID_Peintre) REFERENCES Peintre(ID_Peintre), FOREIGN KEY (Nom_Mouvement) REFERENCES Mouvement(Nom), PRIMARY KEY(ID_Peintre, Nom_Mouvement))

Voici aussi des exemples de commandes INSERT INTO sur chacune des tables

Nom de la table	Exemple de commande INSERT INTO
Peintre	INSERT INTO `peintre`(`ID`, `nom`, `ybirth`, `ydeath`) VALUES (1,"Sandro Botticelli",1445,1510)
Modele	INSERT INTO `modele`(`ID`, `Longueur`, `Largeur`, `Technique`) VALUES (1,203,314,"Tempera sur panneau de bois")
Mouvement	INSERT INTO `mouvement`(`Nom`, `Description`, `Siecle`) VALUES ("Première renaissance Italienne","Evolutions techniques lumineuses","XV siecle")
Tableau	INSERT INTO `tableau`(`ID`, `Titre`, `Date_`, `Modele_ID`, `Peintre_ID`, `Nom_Mouvement`) VALUES (1,"Persistance de la mémoire",1931,13,15,"Surréalisme")
Suit	INSERT INTO `suit`(`ID_Peintre`, `Nom_Mouvement`) VALUES (15,"Surréalisme")

Dans le cas ou nous avons oubliés de mettre l'attribut Technique dans la table modèle, voici la requête SQL que nous exécuterions afin de modifier la table et de rajouter un attribut :

ALTER TABLE `modele` ADD `Technique` VARCHAR(50) NOT NULL AFTER `Largeur`;

Pour chacune des 6 requêtes demandées, nous allons la scénariser et l'exprimer en SQL et algèbre relationnel.

A-Selection avec Projection

Victor doit réaliser une soutenance sur les peintres du XXIème siècle, il exécute donc cette requête pour récupérer les peintres ayant vécu entre 1900 et 2000.

SELECT `nom` FROM `peintre` WHERE `ybirth`>1900 AND `ydeath`<2000
Π nom (σ ybirth>1900 AND ydeath<2000 (peintre))

B-Jointure

Le directeur d'une galerie souhaite savoir quelle sont les dimensions des tableau qu'il va recevoir pour sa futur exposition, cette requête permet de récupérer Longueur, Largeur et Technique de chaque tableau de la base de données

SELECT `Titre`, `Longueur`, `Largeur`, `Technique` FROM `tableau` NATURAL JOIN `modele`
Π Titre,Longueur,Largeur,Technique (tableau \bowtie modele)

C-Moyenne sur intégralité d'un Attribut

Le directeur souhaite maintenant savoir quelle est la longueur moyenne des tableaux pour organiser la place disponible. Il fera donc cette requête.

```
SELECT AVG(`Longueur`)
FROM `modele`
```

D-Regroupement avec calcul

Victor souhaite savoir le nombre de tableau par mouvement dans la galerie, il exécute donc cette requête.

```
SELECT DISTINCT `suit`.`Nom_Mouvement`, COUNT(*)
FROM `suit` NATURAL JOIN `tableau`
GROUP BY(`Peintre_ID`) ORDER BY (`Nom_Mouvement`) ASC
```

E-Différence

Victor souhaite comparer 2 tables contenant des tableaux, une classique et une contenant des tableaux dont la date est paire. Il exécute donc cette requête pour récupérer la différence entre les deux, c'est-à-dire, les tableaux apparaissant dans la première table et pas la deuxième.

```
SELECT `Titre`,`Date_`
FROM `tableau` t1
WHERE t1.`Date_` NOT IN (SELECT `Date_` FROM `tableau` t2 WHERE `Date_`%2=0)
\sigma Date_%2=0 (tableau)
 $\pi$  Titre,Date_ (tableau - tableau2)
```

F-Division

Jean souhaite obtenir comme résultat, tout les tableaux utilisant la technique 'Huile sur Toile'. On exécute donc cette division pour obtenir le résultat voulu grâce à une jointure et une sélection.

```
SELECT `Titre`
FROM `tableau`
WHERE(SELECT `ID`
      FROM `modele`
      WHERE `Technique`='Huile sur Toile'
      AND `tableau`.`Modele_ID`=`modele`.`ID`)
 $\pi$  titre (tableau  $\bowtie$   $\pi$  'Technique'='Huile sur Toile' (modele))
```


IV-PHP

Nous avons créé un page PHP pour chaque requête demandée dans le sujet. Le format du fichier PHP est le même et fut donné lors du TD PHP. La seule différence entre les 6 fichiers qui représentent les 6 requêtes, ce sont les query, qui seront différents selon le type de requête.

```
<?php
define("SERVERNAME","localhost");//mysql.istic.univ-rennes1.fr
define("USERNAME","root");//18002286
define("PASSWORD","");
define("DATABASE","dsb");//base_18002886
$query = "SELECT * from `tableau`";
```

1-Définition des paramètres de connections (define) et de la requête (query)

```
try {
    $bd = new PDO("mysql:host=".SERVERNAME.";dbname=".DATABASE.";charset=utf8", USERNAME , PASSWORD);

    $bd->setAttribute(PDO::ATTR_ERRMODE,PDO::ERRMODE_EXCEPTION);

    $table=$bd->query($query);

    $table->setFetchMode(PDO::FETCH_ASSOC);
```

2-Début du try /catch, on se connecte en mettant nos paramètre définit précédemment, puis on définit le mode d'erreur, on crée une table dans laquelle on va ranger nos resultats de la requete et on exécute la requête pour récupérer la premier ligne. Le setfetchMode définit sous quelle forme va être notre ligne, dans notre cas, c'est sous forme d'association (clé,valeur)

```
echo "<p align='center'>$query</p>";
echo"<table border='1' align='center'>";

$nuplet=$table->fetch();

echo"<tr>";

    foreach ($nuplet as $attribut => $valeur) {
        echo "<th>".$attribut."</th>";
    }

    do {
        echo"<tr>";
        foreach ($nuplet as $attribut => $valeur) {
            echo "<td>".$valeur."</td>";
        }
        echo "</tr>";
    } while ($nuplet=$table->fetch());
echo "</table>";
}
```

3-On affiche la requête puis on commence en créant une ligne qui va contenir tout les attributs du tableau. Puis on fait un do while, qui va mettre chaque valeur de la ligne dans une colonne, sur la même ligne (utilisation des tr et td). La condition du while signifie que la boucle continue tant qu'il y a une ligne disponible dans la requête sinon il ferme le tableau.

```
catch(PDOException $e){
    echo "Erreur : " . $e->getMessage();
}
```

4-Le catch permet d'afficher les erreurs si jamais il y en a.

V-DTD/XML/XPath

Pour la DTD, on a un premier élément galerie qui représente un ensemble de tableau. Le tableau possède lui-même 5 éléments : le titre, la date, le peintre, le mouvement et le modèle. Les deux premiers sont des éléments représentés par des chaînes de caractères et les 3 autres possèdent des sous-éléments (qui sont aussi des PCDATA). On remarquera que notre DTD est cohérente avec nos modèles précédents.

Pour le XML, nous avons donc utilisé un script PHP et l'objet DOMImplementation afin d'importer notre dtd avec un CreateDocumentType. Puis on crée le xml avec CreateDocument, avec comme paramètre la valeur représentant la DTD.

Voici 4 requêtes utilisés pour récupérer certaines données.

```
SELECT Titre,Date_
FROM tableau
INNER JOIN mouvement ON tableau.Nom_Mouvement=mouvement.Nom
INNER JOIN modele ON tableau.Modele_ID=modele.ID
INNER JOIN peintre ON tableau.Peintre_ID=peintre.ID
```

```
SELECT Nom_Mouvement,Description,Siecle
FROM tableau
INNER JOIN mouvement ON tableau.Nom_Mouvement=mouvement.Nom
INNER JOIN modele ON tableau.Modele_ID=modele.ID
INNER JOIN peintre ON tableau.Peintre_ID=peintre.ID
```

```
SELECT Longueur,Largeur,Technique
FROM tableau
INNER JOIN mouvement ON tableau.Nom_Mouvement = mouvement.Nom
INNER JOIN modele ON tableau.Modele_ID = modele.ID
INNER JOIN peintre ON tableau.Peintre_ID = peintre.ID
```

```
SELECT peintre.nom,ybirth,ydeath
FROM tableau
INNER JOIN mouvement ON tableau.Nom_Mouvement = mouvement.Nom
INNER JOIN modele ON tableau.Modele_ID = modele.ID
INNER JOIN peintre ON tableau.Peintre_ID = peintre.ID
```

On crée un élément xml, puis un élément galerie qui sera fils de xml

Pour chaque tour de boucle, voici le déroulement : On crée 4 éléments tableau,peintre, modèle, mouvement avec les valeurs mises dans ces éléments, qui sont récupérés à l'aide d'un fetch de toutes les requêtes. Ensuite on définira peintre, modèle et mouvement comme des fils de tableau puis tableau comme fils de galerie.

Cela nous donnera bien une galerie avec un tableau construit d'un titre, d'une date et de ses 3 sous-éléments. Ensuite on vérifie si l'objet xml est valide : si oui, on sauvegarde l'objet dans un fichier xml et on l'affiche sinon on affiche un message d'erreur.

Pour le Xpath, on utilise DOMDocument pour créer un document et importer le fichier xml précédent dans celui-ci. En suite, on utilise l'objet DOMXPath pour exécuter une requête sur l'objet xml.

Notre requête sera donc : /galerie/tableau/peintre[ybirth>1900 and ydeath<2000]/nom

Qui correspond a notre première requête vu précédemment.

VI-Remarque

Pour le HTML et le CSS, nous avons optes pour un affichage simple, sans template, mais facile à mettre en place afin de passer plus de temps sur les questions du projet. Le code reste disponible dans le dossier du projet.

Pour DBPedia, nous avons eu l'idée d'utiliser une requête pour récupérer toutes les pages utilisant le Mouvement artistique. Ce qui est en rapport avec notre projet puisque chaque tableau et peintre utilise au minimum un mouvement artistique dans le domaine de la peinture.

```
SELECT ?s WHERE {  
  ?s rdfs:label "Mouvement artistique"@fr  
}
```