

Bases de données & POWER AMC Conception de Base de Données

Travaux Pratiques N° :1

Objectifs du TP

- Comprendre et appliquer la méthode Merise pour concevoir un Modèle Conceptuel de Données (MCD).
- Savoir identifier les entités, associations, attributs et cardinalités à partir d'un cahier des charges.
- Utiliser PowerAMC Designer pour formaliser la conception d'un système d'information médical, et d'un système de gestion de location de voiture
- Transformer le MCD en **Modèle Logique de Données (MLD)**.
- Générer le **Modèle Physique de Données (MPD)** sous **PowerAMC Designer**.
- Comprendre les concepts de **clés primaires, étrangères et normalisation**.

Exercice 1:

Problématique

Une clinique privée souhaite mettre en place un système d'information pour gérer les patients, les médecins et les consultations médicales.

Exigences du système

- Chaque **patient** est identifié par un numéro unique. Il a un nom, un prénom, une date de naissance, un numéro de téléphone et une adresse.
- Chaque **médecin** est identifié par un matricule et possède un nom, une spécialité et un numéro de téléphone.
- Un **patient** peut effectuer plusieurs **consultations** au sein de la clinique. Une consultation est associée à un **médecin**, une date, un diagnostic et éventuellement une ordonnance.
- Une **ordonnance** contient plusieurs médicaments, identifiés par un code, un nom et une posologie.
- Un médecin peut voir plusieurs patients, mais une consultation est toujours effectuée par un seul médecin.

A faire :

1. Créer un nouveau projet sous PowerAMC Designer.
2. Ajouter les entités suivantes avec leurs attributs :
 - **PATIENT** (NumPatient, Nom, Prénom, DateNaissance, Téléphone, Adresse).
 - **MEDECIN** (Matricule, Nom, Spécialité, Téléphone).
 - **CONSULTATION** (NumConsultation, Date, Diagnostic).
 - **ORDONNANCE** (NumOrdonnance, Date).
 - **MEDICAMENT** (CodeMedicament, NomMedicament, Posologie).
3. Créer les associations entre ces entités en respectant les cardinalités :
 - Un patient peut avoir **plusieurs consultations**, mais une consultation concerne **un seul patient**.
 - Un médecin peut effectuer **plusieurs consultations**, mais une consultation est effectuée **par un seul médecin**.
 - Une consultation peut être associée à **une ordonnance**, mais ce n'est pas obligatoire.
 - Une ordonnance peut contenir **plusieurs médicaments**, et un médicament peut apparaître dans **plusieurs ordonnances**.

Exercice 2:

Problématique

Une **agence de location de voitures** souhaite informatiser la gestion de ses locations. Le système doit permettre :

- La gestion des **clients** (nom, prénom, email, téléphone, permis de conduire).
- La gestion des **véhicules** (marque, modèle, immatriculation, état, kilométrage, disponibilité).
- La gestion des **contrats de location** (client, véhicule, date début, date fin, prix total).
- La gestion des **paiements** (montant, mode de paiement, date, contrat associé).

Exigences du système

- Un client peut louer plusieurs voitures, mais un contrat concerne un seul client et un seul véhicule.
- Un véhicule ne peut être loué que s'il est disponible.
- Un contrat peut être associé à un ou plusieurs paiements.

A faire :

A. Conception du MCD

- 1) **Créer un projet PowerAMC** et ouvrir un **Modèle Conceptuel de Données (MCD)**.
- 2) **Créer les entités** avec leurs attributs :
CLIENT (IdClient, Nom, Prénom, Email, Téléphone, Permis)
VEHICULE (IdVehicule, Marque, Modèle, Immatriculation, État, Kilométrage, Disponibilité)
CONTRAT (IdContrat, DateDébut, DateFin, PrixTotal)
PAIEMENT (IdPaiement, Montant, ModePaiement, DatePaiement)
- 3) **Créer les relations** en respectant les cardinalités :
 - Un client peut avoir plusieurs contrats, mais un contrat est lié à un seul client.
 - Un véhicule peut être loué plusieurs fois, mais un contrat concerne un seul véhicule.
 - Un contrat peut être payé en plusieurs fois (paiements multiples).

B. Génération du MLD

- 1) Convertir le MCD en MLD sous PowerAMC :
 - Ouvrir le menu "Génération" → "Modèle Logique de Données".
 - Vérifier que toutes les relations sont bien transformées en clés étrangères.
 - Vérifier la normalisation (pas de redondance inutile).
- 2) Vérification des clés et types de données :
 - IdClient devient une clé étrangère dans CONTRAT.
 - IdContrat devient une clé étrangère dans PAIEMENT.

- 4

Bases de données & SGBD MYSQL

Langage de définition des données (LDD)

Travaux Pratiques N° :2

Objectifs

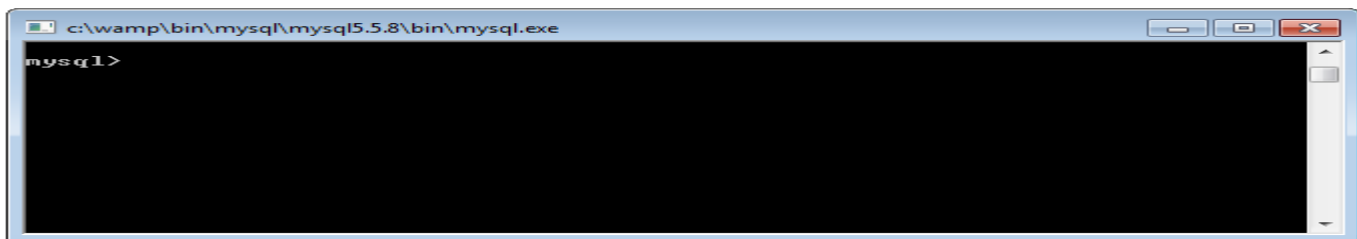
Le but de ces séances de ce TP est de se familiariser avec le concept des bases de données relationnelles, notamment savoir définir un schéma relationnel par une démarche conceptuelle et maîtriser la norme SQL, en particulier le *langage de description des données (LDD)*.

PARTIE 1 :

- ✓ Installez le pack **XampServer** avec les configurations par défaut.

1- Informations sur les bases de données existantes :

- A l'aide de **MYSQL CONSOLE** :



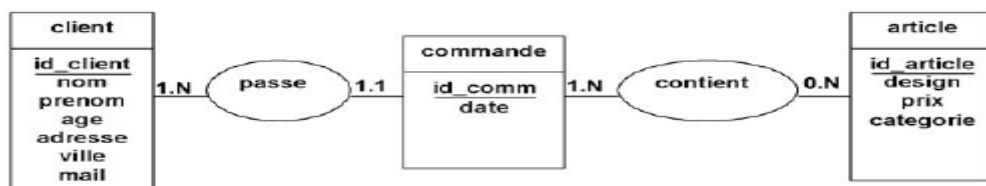
- 1.1) Déterminer le répertoire d'installation de MySQL (basedir)
- 1.2) Déterminer le répertoire qui contient les bases de données MySQL (datadir)
- 1.3) Déterminer le nombre de connexions sur le serveur MySQL en cours
- 1.4) Déterminer la durée de fonctionnement du serveur MySQL.
- 1.5) Affichez les bases de données existantes
- 1.6) Sélectionner une base de données de votre choix
- 1.7) Affichez les tables de la base de données sélectionnée
- 1.8) Affichez la structure (noms des colonnes, types et autres attributs)
- 1.9) Déterminez la base de données active

- A l'aide de gestionnaire graphique *phpMyAdmin* :

Refaites les questions 1.5) à 1.9).

2-Création et gestion des tables d'une base de données :

Soit le modèle conceptuel des données (MCD) de la BD magasin suivant :



2. Proposer le schéma logique (MLD) :

- 2-1) Créer la base de données magasin
- 2-2) Supprimer la base de données magasin
- 2-3) Recréer la base de données magasin
- 2-4) Sélectionner la base de données magasin
- 2-5) Créer les tables client, commande, ligne et article du schéma logique précédent
- 2-6) Afficher les informations globales de la table client (taille max de données, Date de création, type d'engine utilisé...)

La table client

Le code suivant définit les colonnes de la table client.

```
client(id_client, nom, prenom, age, adresse, ville, mail)
```

La colonne id_client est :

- ✓ de type entier positif moyen (**MEDIUMINT**),
- ✓ Clé primaire de la table (**PRIMARY KEY**),
- ✓ champ id_client doit avoir obligatoirement une valeur(**NOT NULL**),
- ✓ Ce champ étant incrémenté automatiquement d'une unité à chaque nouvelle insertion de donnée (**AUTO_INCREMENT**),

Les colonnes nom, prenom, adresse et **ville** sont de type

- ✓ VARCHAR,
- ✓ avec l'option NOT NULL sélectionnée.

La colonne **age** est un

- ✓ entier positif de petite taille(TINYINT UNSIGNED).
- ✓ L'option NULL permet à un client de ne pas saisir son âge.

La colonne mail est également de type :

- ✓ VARCHAR,
- ✓ l'option NULL sélectionnée, car un client peut ne pas donner son adresse e-mail.

La table commande

La table commande a la structure suivante :

```
commande(id_comm, id_client, date)
```

La colonne `id_comm` est de type :

- ✓ MEDIUMINT, avec les options UNSIGNED, NOT NULL et AUTO_INCREMENT sélectionnées, de façon que chaque commande ait un numéro différent.
- ✓ Clé primaire de la table (PRIMARY KEY),

La colonne `id_client` est une clé étrangère issue de la table `client`. Elle a les mêmes caractéristiques que `id_comm`.

La colonne date est de type

- ✓ DATE, avec l'option NOT NULL sélectionnée, de façon que les valeurs soient insérées automatiquement sans intervention du client et qu'il n'y ait pas à craindre un oubli.

La table article

La table article a la structure suivante :

- A l'aide de gestionnaire graphique *phpMyAdmin* :

- 3-8) Ajoutez le champ ville après l'adresse
- 3-9) Ajouter le champ âge après le champ prénom
- 3-10) Modifier le nom du champ nom en nom_client et son type en CHAR(50)
- 3-11) Supprimer le champ ville
- 3-12) changer la clé primaire de la table client en (id_client, nom).
- 3-13) Renommer la table client en clientbis
- 3-14) Ajouter les différentes clés étrangères de la table ligne
- 3-15) Afficher le modèle physique de base de données
- 3-16) Exporter les tables de la base magasin dans des fichiers SQL.
- 3-17) Supprimer toutes les tables.
- 3-18) Recréer les tables à l'aide d'un fichier SQL importé.

Bases de données relationnelles & SGBD MYSQL

Langage de manipulation des données (LMD)

Travaux Pratiques N° :3

Objectifs

Le but de ces séances de ce TP est de se familiariser et maîtriser la norme SQL spécifique au SGBD MySQL, en particulier le *langage de manipulation des données (LMD)*.

1-Insertion de données

Syntaxe :

```
INSERT INTO nom de table ( champ1 , champ2 , champ3 , champ4... )
VALUES (
    Valeur1, Valeur2, Valeur3, Valeur4
);
```

1-1) Insérer ligne par ligne à l'aide de l'interface console les données de la table client

id_client	nom	prenom	age	adresse	ville	mail
1	Marti	Jean	36	5 av. Einstein	Orléans	mart@marti.com
2	Rapp	Paul	44	32 av. Foch	Paris	rapp@libert.com
3	Devos	Marie	18	75 bd Hochimin	Lille	grav@waladoo.fr
4	Hochon	Paul	22	33 rue Tsétsé	Chartres	hoch@fiscali.fr
5	Grave	Nuyen	18	75 bd Hochimin	Lille	grav@waladoo.fr
6	Hachette	Jeanne	45	60 rue d'Amiens	Versailles	NULL
7	Marti	Pierre	25	4 av. Henri	Paris	martin7@fiscali.fr
8	Mac Neal	John	52	89 rue Diana	Lyon	mac@freez.fr
9	Basile	Did	37	26 rue Gallas	Nantes	bas@walabi.com
10	Darc	Jeanne	19	9 av. d'Orléans	Paris	NULL
11	Gaté	Bill	45	9 bd des Bugs	Lyon	bill@microhard.be

1-2) Insérer ligne par ligne à l'aide de l'interface graphique *phpMyadmin* les données de la table commande :

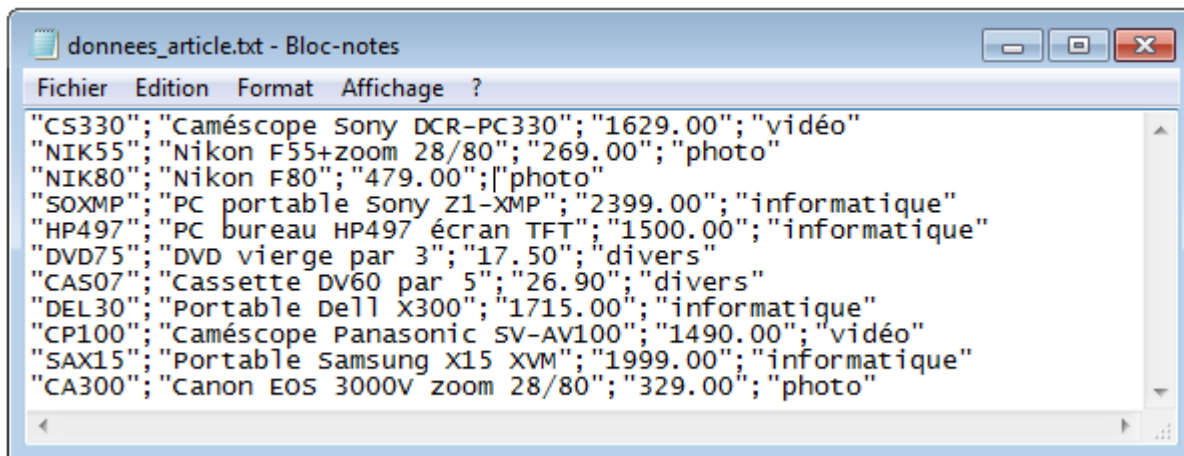
id_comm	id_client	date
1	5	2012-06-11
2	9	2012-06-25
3	1	2012-07-12
4	3	2012-07-14
5	9	2012-07-31
6	10	2012-08-08
7	2	2012-08-25
8	7	2012-09-04
9	11	2012-10-15
10	4	2012-11-23
11	8	2013-01-21

1-3) L'insertion ligne par ligne devient couteuse lorsqu'il s'agit de saisir un grand nombre de Données, soient les données ci-dessous de la table article :

id_article	designation	prix	categorie
CS330	Caméscope Sony DCR-PC330	1629.00	vidéo
NIK55	Nikon F55+zoom 28/80	269.00	photo
NIK80	Nikon F80	479.00	photo
SOXMP	PC portable Sony Z1-XMP	2399.00	informatique
HP497	PC bureau HP497 écran TFT	1500.00	informatique
DVD75	DVD vierge par 3	17.50	divers
CAS07	Cassette DV60 par 5	26.90	divers
DEL30	Portable Dell X300	1715.00	informatique
CP100	Caméscope Panasonic SV-AV100	1490.00	vidéo
SAX15	Portable Samsung X15 XVM	1999.00	informatique
CA300	Canon EOS 3000V zoom 28/80	329.00	photo

L'intérêt de pouvoir insérer un nombre important des données en un seul coup, en un utilisant un fichier texte.

Pour cela, saisir les données précédentes dans un fichier texte de la forme :

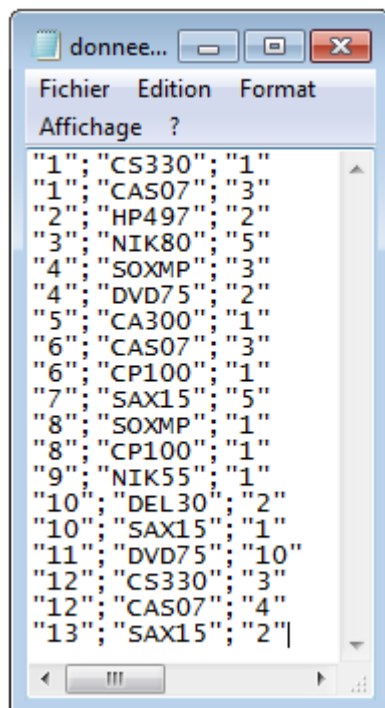


-Utiliser le code SQL suivant pour insérer les données dans la table article à partir de ce fichier:

```
LOAD DATA INFILE 'fichier.texte'
INTO TABLE article
FIELDS TERMINATED BY ';'
ENCLOSED BY '"'
ESCAPED BY '\\'
LINES TERMINATED BY '\r\n'
```

phpMyadmin :

id_comm	id_article	quantite
1	CS330	1
1	CAS07	3
2	HP497	2
3	NIK80	5
4	SOXMP	3
4	DVD75	2
5	CA300	1
6	CAS07	3
6	CP100	1
7	SAX15	5
8	SOXMP	1
8	CP100	1
9	NIK55	1
10	DEL30	2
10	SAX15	1
11	DVD75	10
12	CS330	3
12	CAS07	4
13	SAX15	2



- 13

- 2.4 Sélectionner les clients dont les noms commencent par une lettre entre A et H, la lettre H n'étant pas prise en compte
- 2.5 Sélectionner les articles dont le prix est compris entre 1 500 et 2 000 euros
- 2.6 Sélectionner les clients dont l'âge figure dans la liste (18, 19, 20)
- 2.7 Sélectionner les clients qui habitent une avenue, et dont l'adresse contient donc la chaîne av
- 2.8 Sélectionner les articles de la marque Sony, dont la désignation contient donc la chaîne Sony
- 2.9 Sélectionner les clients dont les noms contiennent le caractère 'a' dans la deuxième position
- 2.10 Sélectionner les clients qui ont moins de 30 ans et qui habitent Paris
- 2.11 Sélectionner les clients qui habitent Lyon ou dont le nom commence par la lettre H
- 2.12 Sélectionner les clients qui n'habitent ni à Paris ni à Lyon

Les fonctions :

- ✓ Une fonction monoligne agit sur une ligne à la fois et ramène un résultat par ligne.
 (Caractères, numériques, dates et conversions de types de données).

Exemple

POW(n,2) retourne la colonne à la puissance 2

MAX(COS(ABS(n))) désigne le maximum des cosinus de la valeur absolue de la colonne n.

- ✓ Une fonction multiligne (fonction d'agrégat) agit sur un ensemble de lignes pour ramener un résultat

Fonction	Description
AVG(colonne)	Retourne la moyenne des valeurs de la colonne précisée.
COUNT(colonne)	Retourne le nombre de lignes dont la valeur n'est pas NULL dans la colonne précisée. COUNT (*) retourne le nombre total de lignes, même si certaines ont la valeur NULL.
COUNT(DISTINCT colonne)	Retourne le nombre de lignes ayant une valeur non NULL et distincte (en éliminant les doublons).
MAX(colonne)	Retourne la valeur maximale de la colonne précisée.
MIN(colonne)	Retourne la valeur minimale de la colonne précisée.
SUM(colonne)	Retourne la somme des valeurs de la colonne précisée.

- 2.13 Sectionner l'id_com, la quantité et la quantité à la puissance 2 sous l'alias Produit_QT à partir de la table ligne
- 2.14 Calculer l'âge moyen des clients
- 2.15 Calculer le prix moyen des articles
- 2.16 Calculer le nombre de client dans la table client
- 2.17 Calculer le nombre des prénoms différent dans la table client
- 2.18 Calculer le nombre de clients ayant indiqué leur adresse e-mail (donc le nombre de lignes pour lesquelles la colonne mail n'est pas NULL)
- 2.19 Calculer le nombre de villes différentes de la table client
- 2.20 Déterminer le prix maximal de la table article
- 2.21 Déterminer l'âge du client le plus jeune
- 2.22 Calculer le nombre total d'articles commandés

Regroupement :

La clause :

```
GROUP BY {col1 | expr1 | position1} [{col2... }]
[HAVING condition]
```

permet de regrouper des lignes ayant une caractéristique commune.

- ✓ La clause GROUP BY liste les colonnes du groupement.
- ✓ La clause HAVING permet de poser des conditions sur chaque groupement.

- 2.23 Calculer l'âge moyen des clients par ville
- 2.24 Calculer l'âge minimal des clients par ville
- 2.25 Calculer l'âge maximal des clients par ville
- 2.26 Regrouper les lignes par ville et afficher dans un même tableau l'âge minimal, l'âge moyen et l'âge maximal des clients
- 2.27 Afficher le nombre d'articles de chaque commande
- 2.28 Calculer l'âge moyen des clients qui habitent des villes dont l'initiale est L
- 2.29 Calculer le montant total de chaque catégorie
- 2.30 Calculer le montant total des catégories dont ce montant dépasse 1500

3-Les jointures

La syntaxe la plus courante d'une jointure est la suivante :

```
SELECT col1,col2,...
FROM table1,table2,...
WHERE condition_de_jointure
```

La condition de jointure est de la forme :

table1.colX = table2.colY

Dans laquelle **colX** et **colY** contiennent des données représentant la même information, comme l'identifiant de client.

- 3.1 Retrouver toutes les commandes faites par un client(id_com,nom,prenom,ville),classez les par nom
- 3.2 Sélectionner les commandes du client dont l'identifiant est 5
- 3.3 Etablir la liste des articles les plus vendus et à afficher leur code, leur désignation, leur prix et le nombre total d'articles (Vous appliquez ensuite à la colonne id_article la clause GROUP BY pour obtenir la somme des ventes de chaque article et la clause ORDER BY pour afficher les résultats en ordre décroissant).
- 3.4 Sélectionner les numéros des commandes, les nom, prénom et ville du client associé à la commande et le montant total de la commande obtenu et de regrouper les résultats par numéro de commande à l'aide de la clause GROUP BY