Initiation SQL

Workbench permet de saisir des requêtes

Plusieurs types: structure, mise à jour, consultations, gestion des droits.

Xampp start les 2 premiers

Puis créer une connection dans workbench +

Tu peux ajouter un nom de connexion

Hostname= nom post local

Schema= structure

Va dans edit, preference: sql editor puis descend et décoche safe updates.

Ferme l’appli et rouvre. Sinon pas de sauvegarde des requêtes.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dans le navigateur:

Partie principale de saisie

Action output:

Voir les resultats, codes erreurs

1ere requête: create créer quelque chose

On peut soit executer la totalité de la saisie soit une partie avec une sélection +clic sur éclair avec un I et dans out put le résultat apparaît.

Rafraichir la liste de gauche pour voir apparaître le résultat.

Ouvrir un nouveau fichier à côté : file new query table

Actions :

drop suppression d’un élement ou du schéma complet (seulement si pas de données)

add = dois-je vraiment expliquer ?

modify= la même…

ressources : SQL.sh

Sur visual code : extension : SQL Tools et SQL Tools my SQL/Maria DB

**Etape 1** : créer une base de données :

**create database** nom choisi**;**

**use** nom choisi**;**

**Etape 2 :** créer des tables

**create table** nom de l’entité **(**

**id\_**nom de l’entité **int primary key auto\_increment not null,**

**attribut\_**nom de l’entité **type(varchar/int/text) not null**

**)Engine = InnoDB ;**

**Etape 3**: les tables d’associations

**Create table** nom du machin **(**

Id\_de la table a relier 1 **int not null,**

**Id**\_de la table à relier 2 **int not null,**

Attribut\_nom du machin type not null,

**Primary key(id\_1,id\_2)**

**)Engine=InnoDB ;**

Etape 4: relier le bazar entre ses différents morceaux

**alter table** nom de la table qui se modifie

**add constraint** fk\_action\_cible

**foreign key (id\_cible)**

**references cible(id\_cible);**

**NB: ne jamais oublier les ; ça serait con que ça ne fonctionne pas.**

Prochaine partie : Inclure des données dans la structure

**La requête INSERT :** on ne note pas la clé primaire elle est déjà là par système

**Insert into** nom de la table **(‘varchar ‘, ‘date’, chiffre)**

**Value** (‘contenu1’, ‘contenu date’, chiffre) ;

**Si value est écrit au singulier on entre les données 1 à 1, si values alors c’est un paquet.**

(c1,c2,c3),

( c1,c2,c3),

( c1,c2,c3) ;

NB : attention aux espaces dans les guillemets ils annulent le contenu.

La **requête UPDATE :** (mise à jour des valeurs)

Attention ! **apparition des conditions. (where)**

**update** nom\_table **set** nom\_colonne **=** ‘nom/donnée’ **where** nom\_colonne = ‘nom/donnée’ ;

les conditions peuvent s’ajouter ou être un choix (comme if/else if/else).

Elles apparaissent sous les termes :

**And** : **ajout de conditions, les 2 doivent être ok.**

**Or** : **l’une ou l’autre doit être ok. Attention si la 1 est ok la 2 n’est pas lue.**

Ex : **update** nom\_table **set** nom\_colonne **=** ‘nom/donnée’ **where** nom\_colonne = ‘nom/donnée’ **and** nom\_colonne = ‘nom/donnée’ ;

**Il en existe d’autres comme :**

Between, like, is null / is not null…

Les %: point particulier les % dans la condition like :

Like %ur : tous les enregistrements finissant en ur.

Like mise% : tous les enregistrements commençant par mise.

Like %mise% : tous les enregistrements commençant, possédant au milieu ou finissant par mise.

La requête **SQL IN** permet d’ajouter en une seule requête plusieurs demandes.

**Les effets ajoutés :**

On peut ajouter des effets aux données dans les parenthèses comme majuscule, envers etc…

Forme :

Majuscule : **upper(‘donnée’)**

Envers : **reverse(‘donnée’)**

**Md5(‘**password**’)-> criptage du password** attention elle est réversible.

La **requête DELETE** : la suppression de donnée

**Attention conditions impératives**

Forme de la requête :

**Delete from** nom\_table **where** nom\_colonne <2;

**Attention difference avec truncate: delete ne réinitialise pas à 0 les id.**

Un élément relié à d’autres table doit d’abord voir ses connexions coupées pour pouvoir être supprimé, très fastidieux donc on préfère faire passer la donnée sous silence que la supprimer.

La **requête SELECT** : sélectionner des données pour divers usages

Pas **\***mauvaise pratique : trop général et expose des données en plus de ce qui est nécessaire nuisible à la sécurité des données.

La forme basique :

**Select** nom\_colonne **from** table ;

Apparition des **alias (as)** et des spécifications de recherche

L’alias est pris en compte uniquement dans la requête utilisée.

**Select** id\_classe **as** id, nom\_classe **as** classe **from** classe **where** nom\_classe = ‘6G5’;

**Le tri prend la forme order by :**

**Order by** nom\_classe **asc/desc**;

Mail: si non spécifié il prend un tri asc.

Les colonnes triées peuvent se suivre séparées par une virgule.

Si pas d’order by spécifié il trie par id.

**Group by :** regroupe de l’information ensemble (compter combien de type dans un groupe)

**Les jointures de tables en SQL**

Les jointures permettent de mettre côte à côte des informations situées dans des tables différentes.

**INNER JOIN :** met en commun les infos qui matche entre 2 colonnes.

**LEFT JOIN** : prend toute la table (nécessité condition) et renvoie le côté gauche avec ce qui matche.

**RIGHT JOIN :** prend toute la table (nécessité condition) et renvoie le côté droit avec ce qui matche. Permet aussi de voir les éléments manquants dans une table

**JOIN :** comme inner join, moins utilisé

**UNION :** peut joindre des requêtes ensembles

**Limit :** peut ben limiter à tant de valeur **utile pour une sélection ciblée dans un tri**

**Limit 3,1, J’en veux 1 à partir du 3eme.**

NB : la précision des données est importante. Pour cibler avec précision : nom\_table.attribut (penser que cela doit cibler un point commun)

Ordre d’apparition dans la requête :

**Select, Inner join, on, 2eme inner join, on, where puis enfin tri avec order by/ group by etc…**

**Les fonctions en SQL**

**Les fonctions d’agrégation**

Ensemble d’outils permettant d’effectuer diverses opérations de manière rapide.

Exemple :

**LTRIM/ RTRIM() :** supprime espace avant/après chaine de caractères

**AVG() :** calcule la moyenne

**ROUND() :** arrondi, a tant après la virgule

**SUM() :** somme

**COUNT() :** compteur de ce que l’on veut ça dépend avec quoi on l’associe

**MAX ():** renvoie le plus grand

**MIN () :** renvoie le plus petit

**DAY ()** : retourne date numero

**MONTH () :** retourne numero du mois

**DAYOFYEAR () :** retourne numero du jour de l’année

**DAYOFWEEK () :** retourne numero du jour de la semaine

**RAND() :** renvoie un numero à virgule au hasard situé entre 0 et 1.

**CONCAT** (,”€”): **concaténation** :permet de mettre côte a côte 2 infos comme texte et chiffre ou symbole

(en js dans console.log c’est le + ‘’)

Ex : select **round(** **sum(**prix\_article)/count(id\_article**)**,**0)** from article;

Là le résultat n’a pas d’information après la virgule

Round (avg(prix\_article),1) le resultat apparaît avec 1 chiffre après la virgule.

**Les requêtes imbriquées :**

**Select** \* from ticket where id\_vendeur =

(**select** id\_vendeur from vendeur where prenom\_vendeur = ‘julie’) ;

Le résultat (julie) est d’abord calculé

SQL fait d’abord l’intérieur de la parenthèse et ensuite le reste.

Peut pas mal rallonger le truc suivant le nombre de données à analyser. A utiliser si nécessaire mais sinon préférer des usages moins gourmands.

Quitte à faire 2 requêtes.

**Permet de réaliser des calculs complexes sans utiliser 20 requêtes.**

**Créer une vue :** ce n’est pas une table mais le résultat d’une requête fonctionne comme une table quand même.

**Create view** as vw\_complet **as select** inserer les différentes colonnes et tables jointes en requête

Données accessibles en lecture seule

Se met à jour seule si on change les données, peut donc solutionner des calculs.

**Gestion des droits**

**Create user ‘nom\_user’ ‘@’%’ identify by ‘personne’** commande spéciale qui crée donc un compte utilisateur auquel on va associer des droits

Fondamental pour la gestion sécuritaire de vos sites.

On ne donne pas tous les droits à tout le monde.