Cours sur le SQL

I. Présentation du langage SQL

1. DEF:

SQL: Structure Query Language.

-C'est un langage informatique normalise servant a exploiter les bases de données.

-Conçu en 1990 par Edgar Frank CODD

-Le SQL est utilisé par des SGBDs

-SQL est compose de 4 sous-ensembles.

2. SOUS ENSEMBLE DU LANGAGE SQL

- LDD : Langage de Définition de Données

-Création de la base de données

-Création des tables

-Modification de tables

-Suppression de tables

- LMD : Langage de Manipulation de Données

-Insertion de données

-Modification de données

-Suppression de données

- LID : Langage d'Interrogation de données

-Consultation de données

- LCD : Langage de Contrôle de Données
- LCT : langage de contrôle des transactions.

II. LES OUTILS

- SGBD : Système de Gestion de Base de Données
- SGBDR : Système de Gestion de Base de Données Relationnelles

Les SGBDR les plus utilises :

-MySQL (logiciel libre et gratuit)

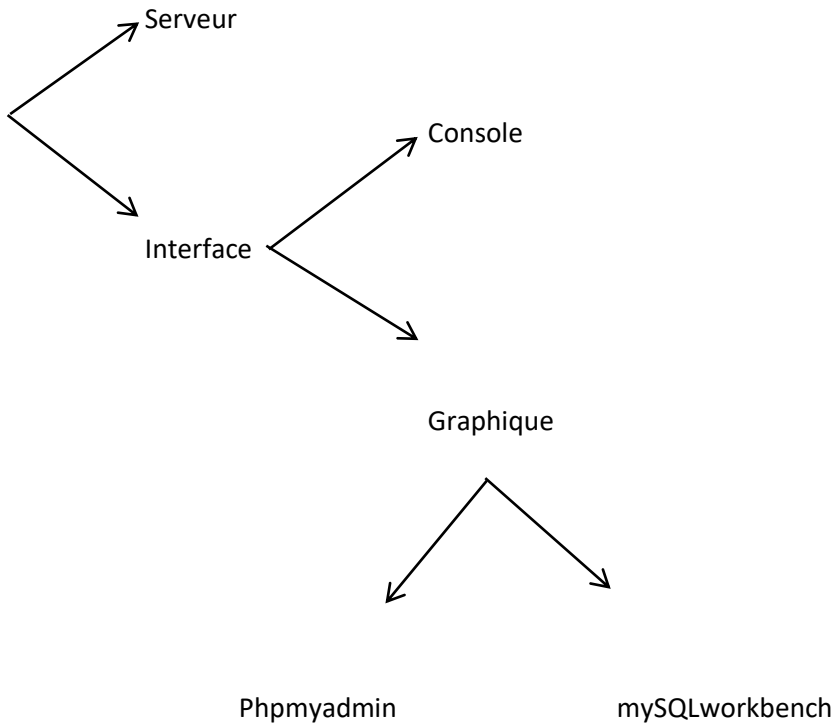
-Oracle (complet et non gratuit)

-SQL server (complet et non gratuit)

-Access

-PostgreSQL (concurrence Oracle et SQL server mais est gratuit)

- MySQL



- Wamp :Windows Apache MySQL PhP
- Lamp:Linux Apache MySQL PhP
- XAMPP:XOs Apache MySQL PhP
- Mamp:MacOs Apache MySQL PhP

III. COMMANDES SQL

1. Quelques commandes usuelles

- Demarrer mySQL

Mysql -h nom_hôte –u nom _utilisateur _ pmot_de_passe ;

Exple :mysql -u root -p ;

- Afficher les bases de données

Show databases ;

- Commande de création de la base de données

CREATE DATABASE nom_base ;

ou

CREATE DATABASE CHARACTERSET utf8;

- Afficher les utilisateurs

SELECT user FROM mysql.user;

2. Types de donnees

-Type numerique

- INT/DOUBLE

-Type alphanumerique

- CHAR/VARCHAR

-Type date pour les dates et heures

- DATE,TIME,DATETIME

3. CREATION DE TABLE

Une table est creee dans une base de donnee.

Avant tout, il faut la selection de la base de donnee.

- Selection de base de donnee

USE nom_Bd ;

- Afficher la liste des tables d une Bd

SHOW TABLE ;

❖ Syntaxe de création d une table

CREATE TABLE nom_table(attr1 type [contrainte d attribut]

attr 2 type [contrainte d attribut]

. .

. .

attr n type [contrainte d attribut]

exple :

ETUDIANT
Matricule
Nom
Prénom
Date_naissance
Poids

ETUDIANT (matricule,nom,prenom,date_naissance,poids).

```
CREATE TABLE ETUDIANT(
Matricule INT(6) PRIMARY KEY,
Nom VARCHAR(20),
Prenom VARCHAR(20),
Date_naissance DATE,
Poids DOUBLE,
) ;
```

❖ Syntaxe de modification d'une table

-Ajouter un attribut a la table : ALTER TABLE nom_table

ADD attr Type ;

Exple : Ajouter l'attribut 'adresse' a la table 'étudiant'.

ALTER TABLE étudiant ADD adresse VARCHAR(30) ;

-Modifier le type de l'attribut matricule de int vers VARCHAR.

ALTER TABLE étudiant MODIFY matricule VARCHAR(6) ;

-Supprimer l'attribut adresse de la table 'étudiant '

ALTER TABLE étudiant DROP adresse ;

-Ajouter une contrainte de clé primaire sur l'attribut matricule

ALTER TABLE étudiant ADD constraint pk_etudiant PRIMARY KEY(matricule);

-Contrainte de clé étrangère

CONSTRAINT nom_contrainte FOREIGN KEY(attr) REFERENCE nom_table(attr_ref);

LMD:

- ✓ Contraintes d'integrite:

Une contrainte d'integrite est une regle qui permet de controler la validite et la coherence des valeurs entree dans les differentes tables de la base

-Dans les commandes de creation des tables.

-Au moment de la modification de la structure de la table.

La contrainte porte sur un seul attribut. Ces contraintes sont

- NOT NULL : adresse VARCHAR(15) NOT NULL ;
- UNIQUE : VARCHAR(15) UNIQUE ;
- PRIMARY KEY : MATRICULE INT(8) PRIMARY KEY ;
- FOREINGN KEY :

7) INSERTION DE DONNER DANS LA TABLE

SYNTAXE :

-INSERT INTO **NOM_table(**attr1,attr2,…attrN)

VALUES(val1 ,…..valN) ;

Exemple :

INSERT INTO etudient(matricule,nom,prenom)
VALUE('20100','AFATOLO','steeve') ;

Ou bien esseré pour toute la table

INSERT INTO etudient

VALUE('20100','AFATOLO','steeve','2000/01/10', 60) ;

8) MODIFICATION DE DONNEES

SYNTAXE :

-UPDATE Nom_table set

Exemple : Modifier le poids de l'etudiant AFATOLO

UPDATE etudiant set poid=70 where matricule= '20100' ;

9) SUPPRESSION DE DONNE

Syntaxe :

-DELETE from nom_table Where condition

Ex :Suppression de l'etudiant AFATOLO

Delete from etudiant Where matricule= ‘20100’ ;

TP

/CRATION DE LA BASE /

```
create database gestion_vente;

use gestion_vente;
create table categorie(
        code_cat varchar(10) primary key,
    libelle_cat varchar(30)
    );

    CREATE TABLE Client (
        num_cli INT(4)    PRIMARY KEY,
    nom varchar (30),
    telephone VARCHAR(12),
    adresse VARCHAR(50),
    email varchar(30)
    );

        CREATE TABLE produit(
    ref_prod varchar(6) primary key ,
    libelle_prod varchar(30),
    prix_unit int(7),
```

```
    refcat varchar(10),
    CONSTRAINT fk_categorie FOREIGN KEY (refcat) REFERENCES
categorie (code_cat)

    );




    CREATE TABLE commande(
    num_cmd int(4) primary key auto_increment,
    date_cmd date,
    num_cli int(4),
    CONSTRAINT fk_client FOREIGN KEY (num_cli) REFERENCES Client
(num_cli)
);




CREATE TABLE concerner(
ref_prod varchar(6),
num_cmd int(4),
qte int(6),
constraint pk_concerner primary key(ref_prod,num_cmd),
constraint fk_produit foreign key (ref_prod) references produit(ref_prod),
constraint  fk_commande foreign key(num_cmd) references
commande(num_cmd)
);
```

***/ INSERTION DE 15 CLIENTS DANS LA TABLE CLIENT */**

INSERT INTO client
VALUES(1001,"BINESSI",90121325,"Tokoin","Binessi@gmail.com"),
(1002,"GBADO",97232526,"Kegue","Gbado@gmail.com"),

(1003,"TCHAMDJA",92959893,"Agbalepedo","Tchamdja@gmail.com"),

(1004,"FOLLY",98969392,"Hedrzanawoe","Folly@gmail.com"),

(1005,"HEMAZRO",92979423,"Hedzranawoe","Hemazro@gmail.com"),
(1006,"AFATOLO",93252427,"Adjidogome","Afatolo@gmail.com"),

(1007,"TALLE",70252426,"Djidjole","Talle@gmail.com"),

(1008,"PALABI",70525421,"Tokoin","Palabi@gmail.com"),

(1009,"NYAKOU",93363534,"Tokoin","Nyakou@gmail.com"),
(1010,"BOATENG",92989441,"Adjidogome","Boateng@gmail.com"),

(1011,"AFIVI",91121315,"Kpogan","Afivi@gmail.com"),
(1012,"GABA",90141718,"Tokoin","Gaba@gmail.com"),

(1013,"ODADJE",90181613,"Adjidogome","Odadje@gmail.com"),

(1014,"LAWSON",98932328,"Hedrzanawoe","Lawson@gmail.com"),
(1015,"DZAGO",92476525,"WUITI","Dzag@gmail.com");

***/ INSERTION DANS LA TABLE CATEGORIE */**

INSERT INTO categorie VALUES("ORD","ORDINATEUR"),

("TEL","TELEPHONE"),

("IMP","IMPRIMANTE"),

("TV","TELEVISION"),

("MO","MOTO"),

("AV","AVION"),

("VOI","VOITURE"),

("VEN","VENTILATEUR");

***/ INSERTION DANS LA TABLE COMMANDE */**

```
INSERT INTO commande(date_cmd, num_cli)
VALUES("2019/04/21",1009),
      ("2019/05/25",1001),
      ("2019/04/01",1010),
      ("2019/07/14",1010),
      ("2019/04/28",1015),
      ("2019/04/29",1009),
      ("2019/01/07",1002),
      ("2019/02/14",1003),
      ("2019/03/23",1004),
      ("2019/10/10",1011),
      ("2019/11/21",1006),
      ("2019/06/01",1007),
      ("2019/09/09",1008),
      ("2019/12/25",1012);
```

***/ INSERTION DANS LA TABLE PRODUIT */**

```
INSERT INTO produit VALUES("HP2","HP 2000",230000,"ORD"),
                          ("HT1","HTC One",90000,"TEL"),
                          ("S7","Samsung galaxy",130000,"TEL"),
                          ("NOTE","Infinix note",120000,"TEL"),
                          ("HOT","Infnix hot6",90000,"TEL"),
                          ("HJ","Haojue",500000,"MO"),
                          ("AV","Avensis",330000,"VOI"),
                          ("AS","Asus",240000,"ORD"),
                          ("AC","Accer",310000,"ORD"),
                          ("IMPI","Imprimante",200000,"IMP"),
```

("SEN","Senli",430000,"MO"),
("LJ","Luojia",250000,"MO"),
("MEC","Mercedes",730000,"VOI"),
("LB","Lamborgini",900000,"VOI"),
("SAM","Samsung",400000,"TV");

***/ INSERTION DANS LA TABLE CONCERNER */**

INSERT INTO concerner VALUES("HP2",5,10),
("HP2",2,20),
("HT1",5,1),
("AV",1,5),
("HJ",2,3),
("AC",9,15),
("IMPI",8,4),
("MEC",14,30),
("NOTE",5,40),
("LB",5,20),
("SAM",3,50),
("SEN",6,3),
("HT1",7,1),
("AS",4,14);

***/ augmenter le prix de 20% le prix_unit de tous les produit*/**

UPDATE produit set prix_unit= 1.2 * prix_unit;

/ Donez la liste des commande lancé avant le 17/05/2019 /

SELECT * FROM Commande WHERE date_cmd < '2019/05/17';

/ Liste des produit dont le prix est compris entre 200000 et 250000/

SELECT *FROM Produit WHERE prix_unit BETWEEN 200000 AND 250000 ;

/Lise des client (nom,telephone) commencent par 't' /

SELECT Nom,Telephone FROM Client WHERE Nom LIKE 't%' ;

/*Liste des etudiant dont la date se trouve parmi les date suivante :

- **21/04/2019**
- **14/07/2019**
- **25/052019**

SELECT * FROM Commande WHERE Date_unit in ('21/04/2019','14/07/2019','25/05/2019') ;

/*liste des produit qui appartiennent à une categorie */

SELECT *FROM Produit WHERE code_cat is not NULL ;

C) COMMANDE SELECT -> TRI DES DONNEES

Pour trier les données affichées par la commande de SELECT on utilise la clause ORDER BY

SYNTAXE :

SELECT attre1, attre2..

FROM Noxm_table

WHERE Condition

ORDER BY attr1[asc/desc], … atr[asc/desc]

EXEMPLE :

SELECT Nom FROM Client ORDER BY Nom desc ;//tri descendant

EXEMPLE 2 : trié client par ordre alphabetique et par date de naissance

SELECT *FROM Client ORDER BY date_naissance DESC, nom ASC ;

13- LES FONCTIONS

NOM DE LA FONTION	ROLE DE LA FONTION
AVG	Moyenne
SUM	Somme
MAX	Maximum
MIN	Minimum
COUNT(*)	Nombre de ligne
COUNT(ATTR)	Nombre de valeur non nul de attr
CONT([DISTINCT)ATTR]	Nombre

REQUETE 12 : Nombre total de commande du client 1010

SELECT COUNT(*) FROM commande WHERE num_cli=1010 ;

REQUETE 13 : Quelle est la moyenne des prix unitaire des produit

SELECT AVG(prix_unit) from produit ;

REQUETE 14 : Donnez le montant total des telephones

SELECT SUM(prix_unit) FROM produit WHERE code_cat='tel' ;

REQUETE 15 : Quelle est e prix unitaire le plus élevé

SELECT Max(prix_unit) FROM produit ;

REQUETE 16 : Quelle est la date de la cmmnde la plus ancienne

SELECT MIN(date_cmd) FROM commande ;

14- LES ALIAS

Dans le langage sql . Cette astuce est particulièrement utile pour faciliter la lecture des requête.

APPLICATION :SELECT lib_prod , 2*prix_unit FROM produit ;

***ALIAS SUR UNE COLONNE**

SYNTAXE : SELECT Colonne1 AS C1, Colonne 2

Cette syntaxe permet de s rennoé la collone collone 1 en c1.

APPLICATION : SELECT Libelle_prod AS "produit" , 2*prix_unitaire AS "prix_unitaire" AS "prix double"

La SYNTAXE prcédenete peux egalement s'ecrire de la facon suivante :

REQUETE 17 : afficher le minimum et le maximum des prix unitaire des produit ainsi que le nombre total de produit.

SELECT MIN (Pric_unit) AS "prix_unit", MAX (prix_unit) AS "prix_max", COUNT(*) AS "nombre total de produit FROM Produit ;

LA SYNTAXE pour renomer une table dans dans une requete est la suivate:

SYNTAXE SELECT *FROM nom_table AS t ;

Ou SELECT *FROM nom_table t ;

APPLICATION : afficher les produit dont le prix est sup a 100000.

SELECT *FROM Produit AS p where p.prix unit> 100000 ;

15- LES JOINTURES

Jointure 1

Il s'agit ici de selectionner les donnees provenant de plusieurs tables ayant 1 ou plusieurs attrs communs

Il existe plusieurs types de jointures :

--Les Jointures internes : Elles ne selectionnent que les donnees ayant une correspondance ds les 2 tables

--Les Jointures externes : Elles selectionnent ttes les donnees meme si certaines n'ont pas de correspondances dans d'autres tables

```
SELECT  *
FROM tab1, tab2
WHERE attr1 = attr2;
```

--Produit Cartesien

```
SELECT  *
FROM tab1, tab2;
```

Jointure 2

```
SELECT libelle_prod, prix_unit, libelle_cat FROM produit, categorie
WHERE produit.ref_cat = categorie.code_cat AND categorie.libelle_cat =
'ORDINATEUR';
```

Jointure 3

```
SELECT libelle_prod, prix_unit, libelle_cat
FROM produit
JOIN categorie
ON produit.ref_cat = categorie.code_cat
WHERE categorie.libelle_cat = 'ORDINATEUR';
```

***EXERCICE :**

1) **Donner les produits command »s au cours de l'année 2019 et qui sont vendus aux client de Hedzranawoe**.

```
SELECT Produit.ref_prod, Produit.lebelle_prod FROM produit AS p
JOIN concerner AS c ON p.ref_prod=c.ref_prod
JOIN commande AS cm ON c.num_cmd=cm.num_cmd
JOIN client AS cl ON cm.num_cli=cl.num_cli
WHERE date_cmd BETWEEN '2019/01/01' AND '2019/12/31' AND
adresse='hedrzanawoe' ;
```

2) **Donner les clients(nom) qui ont une fois commandé un produit de quanlité >10 par ordre décroissant.**

```
SELECT DISTINCT n.nom, n.telephone
FROM client AS n
JOIN commande AS cm
ON cm.num_cli = n.num_cli
JOIN concerner AS c
ON c.num_cmd = cm.num_cmd
WHERE c.qte > 10
ORDER BY nom desc;
```

Joincture externe à gauche

```
SELECT *FROM compte LEFT [outer] JOIN client
```

Joincture externe à droite

SELECT *FROM compte RIGTH [outer] JOIN client

16- LES OPERATEUR ENSEMBLISTE

A)l'union

Donner l'ensemble des clients de tokoin et de djidjolé qui on des commandes.

SELECT cl.nom, cl.adresse FROM client cl, commande cm WHERE cl.num_cli=cm.num_cli AND cl.adresse='tokoin'

UNION

SELECT cl.nom, cl.adresse FROM client cl, commande cm WHERE cl.num_cli=cm.num_cli AND cl.adresse='djidjolé'

B)INTERSECTION

Donner l'ensemble des produit commun au commande 2 et 5.

SELECT *FROM Produit JOIN concerner cn ON p.ref_prod=cn.ref_prod JOIN commande cm ON cn.num_cmd=cm.num_cmd WHERE cm.num_cmd=2

INTERSECT

SELECT *FROM Produit JOIN concerner cn ON p.ref_prod=cn.ref_prod JOIN commande cm ON cn.num_cmd=cm.num_cmd WHERE cm.num_cmd=2