ETUDE DE CAS

Objectif :

L'objectif de cette etude de cas est d'apprendre sur un cas pratique comment utiliser les commandes suivantes :

Le service commercial de l'entreprise PRODX dispose d'une base de données permettant la gestion de production de tubes.

La commande d'un client concerne une ou plusieurs références de tube avec une quantité à produire pour chaque référence.

Une commande déclenche une ou plusieurs productions.

Une production peut concerner plusieurs références de tubes : pour chaque référence de tube, il est alors nécessaire de connaître la quantité fabriquée par jour.

schéma relationnel utilisé :

production : nom de la base de donnee

USINE (num, ville)
num : clé primaire

ATELIER (num, numUsine)
num : clé primaire
numUsine : clé étrangère en référence à num de USINE

TUBE (ref, libelle, diametre, poids, numAtelier)
ref : clé primaire
numAtelier : clé étrangère en référence à num de ATELIER

CLIENT (ref, nom, adr, pays)
ref : clé primaire

COMMANDE (num, refClient, date)
num : clé primaire
refClient : clé étrangère en référence à ref de CLIENT
PRODUCTION (num, numCommande)
num : clé primaire
numCommande : clé étrangère en référence à num de COMMANDE

COMPOSER_PRODUCTION (numProduction, refTube, dateJour, quantite)
numProduction, refTube, dateJour : clé primaire
numProduction : clé étrangère en référence à num de PRODUCTION
refTube : clé étrangère en référence à ref de TUBE

COMPOSER_COMMANDE (numCommande, refTube, quantite)

numCommande, refTube : clé primaire
numCommande : clé étrangère en référence à num de COMMANDE
refTube : clé étrangère en référence à ref de TUBE

I) IMPLANTATION DE LA BASE DE DONNE SOUS MYSQL

1.1) Lancez le serveur mysql

1.2) Connexion au serveur mysql
mysql –h –uroot –p

1.3) Creation de la base de donnee
create database production;

1.4) Ouverture de la base de donnee
use production ;

1.5) Creation de la table usine

```
create table usine (
num int unsigned not null auto_increment,
ville varchar(30) not null,
PRIMARY KEY(num)
);
```

1.6) Creation de la table atelier

```
create table atelier (
num int unsigned not null auto_increment,
numusine int unsigned not null,
PRIMARY KEY(num)
);

alter table atelier add Constraint  FK_Atelier Foreign key (numusine) references
usine(num);
```

1.7) Creation de la table tube
```
create table tube (
ref varchar(30) not null ,
libelle varchar(30) not null,
diametre double (7,2),
poids double (5,2),
numatelier int unsigned not null,
```

```
PRIMARY KEY(ref)
);

alter table tube add constraint fk_tube foreign key(numatelier) references atelier(num);
```

1.8) Creation de la table client

```
create table client(
ref varchar(30) not null,
nom varchar(30) not null,
adr varchar(30) not null,
pays varchar(30) not null,
primary key(ref)
);
```

1.10) Creation de la table commande

```
create table commande (
num int unsigned not null auto_increment,
refClient varchar(30) not null,
primary key(num),
constraint commande_fk foreign key(refclient) references client(ref)
);
```

1.11) Creation de la table production

```
create table production(
num int unsigned  not null auto_increment,
numCommande int unsigned  not null,
primary key(num),
constraint fk_production foreign key(numCommande) references commande(num)

);
```

1.12) Creation de la table composer_production

```
create table composer_production(
numproduction int unsigned not null auto_increment,
reftube varchar(30) not null,
datejour date,
quantite double(7,2),
primary key(numproduction, reftube,datejour),
constraint fk_composer1 foreign key(numProduction) references production(num),
constraint fk_composer2 foreign key(reftube) references tube(ref)
);
```

1.13) Creation de la table composer_commande

```
create table composer_commande(
numcommande int unsigned not null auto_increment,
refTube varchar(30) not null,
primary key(numcommande, reftube),
constraint k_composer_commande foreign key(numcommande) references commande(num),
constraint k1_composer_commande foreign key(reftube) references tube(ref)
);
```

1.12) Ajouter la table suivante

OPERATEUR (num, nom, prenom, numAtelier)
num : clé primaire
numAtelier : clé étrangère en référence à num de ATELIER

```
CREATE TABLE OPERATEUR (
  num integer not null,
  nomOperateur varchar(30),
  prenomOperateur varchar(30),
  numAtelier integer,
  Constraint FK_p primary key(num)
  );
```

```
alter table atelier add Constraint  FK_Atelier Foreign key (numusine) references usine(num);
```

1.13) Ajout d'un enregistrement dans la table commande

```
INSERT INTO commande(num, refClient) VALUES (1,1);
```

Que constate t'on ?
Pourquoi ?

1.15) Ajout d'enregistrement dans la base

1.15.1) Ajout d'enregistrement dans la table usine

```
INSERT INTO usine(num,ville) VALUES (1,"paris");
INSERT INTO usine(num,ville) VALUES (2,"lyon");
```

1.15.2) Ajout d'enregistrements dans la table atelier

```
INSERT INTO atelier(num, numusine) VALUES (1,1);
INSERT INTO atelier(num, numusine) VALUES (2,1);
INSERT INTO atelier(num, numusine) VALUES (3,2);
INSERT INTO atelier(num, numusine) VALUES (4,2);
```

1.15.3) Ajout d'enregistrements dans la table client

```
INSERT INTO client(ref, nom, adr, pays) VALUES ("client1","Entreprise client1","30 rue ville client1","pays du client1") ;
INSERT INTO client(ref, nom, adr, pays) VALUES ("client2","Entreprise client2","30 rue ville client2","pays du client2") ;
INSERT INTO client(ref, nom, adr, pays) VALUES ("client3","Entreprise client3","30 rue ville client3","pays du client3")   ;
```

1.15.4) Ajout d'enregistrements dans la table commande

```
INSERT INTO commande(num, refClient) VALUES (1,"client1");
INSERT INTO commande(num, refClient) VALUES (2,"client1");
INSERT INTO commande(num, refClient) VALUES (3,"client1");
INSERT INTO commande(num, refClient) VALUES (4,"client2");
INSERT INTO commande(num, refClient) VALUES (5,"client2");
INSERT INTO commande(num, refClient) VALUES (6,"client2");
```

1.15.5) Ajout d'enregistrements dans la table tube

```
INSERT INTO tube(ref, libelle, diametre, poids, numatelier) VALUES ("tube1","tube 1",1.5,5.00,1);
INSERT INTO tube(ref, libelle, diametre, poids, numatelier) VALUES ("tube2","tube 2",2.5,6.00,1);
INSERT INTO tube(ref, libelle, diametre, poids, numatelier) VALUES ("tube3","tube 3",6.5,7.15,1);
INSERT INTO tube(ref, libelle, diametre, poids, numatelier) VALUES ("tube4","tube 4",3.5,5.65,2);
INSERT INTO tube(ref, libelle, diametre, poids, numatelier) VALUES ("tube5","tube 5",2.5,3.00,2);
INSERT INTO tube(ref, libelle, diametre, poids, numatelier) VALUES ("tube6","tube 6",4.5,2.50,3);
INSERT INTO tube(ref, libelle, diametre, poids, numatelier) VALUES ("tube7","tube 7",3.5,1.15,3);
INSERT INTO tube(ref, libelle, diametre, poids, numatelier) VALUES ("tube8","tube 8",0.5,2.67,4);
INSERT INTO tube(ref, libelle, diametre, poids, numatelier) VALUES ("tube9","tube 5",6.5,1.45,4);
```

1.15.6) Ajout d'enregistrements dans la table composer_commande

```
INSERT INTO composer_commande(numcommande, refTube) VALUES (1,"tube1");
INSERT INTO composer_commande(numcommande, refTube) VALUES (1,"tube2");
INSERT INTO composer_commande(numcommande, refTube) VALUES (1,"tube3");
INSERT INTO composer_commande(numcommande, refTube) VALUES (1,"tube5");
INSERT INTO composer_commande(numcommande, refTube) VALUES (2,"tube6");
INSERT INTO composer_commande(numcommande, refTube) VALUES (2,"tube7");
INSERT INTO composer_commande(numcommande, refTube) VALUES (2,"tube5");
INSERT INTO composer_commande(numcommande, refTube) VALUES (3,"tube5");
INSERT INTO composer_commande(numcommande, refTube) VALUES (3,"tube6");
INSERT INTO composer_commande(numcommande, refTube) VALUES (3,"tube8");
INSERT INTO composer_commande(numcommande, refTube) VALUES (4,"tube1");
INSERT INTO composer_commande(numcommande, refTube) VALUES (5,"tube7");
INSERT INTO composer_commande(numcommande, refTube) VALUES (5,"tube8");
INSERT INTO composer_commande(numcommande, refTube) VALUES (6,"tube1");
INSERT INTO composer_commande(numcommande, refTube) VALUES (6,"tube3");
INSERT INTO composer_commande(numcommande, refTube) VALUES (6,"tube5");
```

1.15.7) Ajout d'enregistrements dans la table production

```
INSERT INTO production(num, numCommande) VALUES (1,1);
INSERT INTO production(num, numCommande) VALUES (2,2);
INSERT INTO production(num, numCommande) VALUES (3,3);
INSERT INTO production(num, numCommande) VALUES (4,4);
INSERT INTO production(num, numCommande) VALUES (5,5);
INSERT INTO production(num, numCommande) VALUES (6,6);
```

1.15.7) Ajout d'enregistrements dans la table composer_production

```
INSERT INTO composer_production(numproduction, reftube, datejour, quantite) VALUES (1,"tube1","2015-01-01",10);
INSERT INTO composer_production(numproduction, reftube, datejour, quantite) VALUES (1,"tube2","2015-01-01",4);
INSERT INTO composer_production(numproduction, reftube, datejour, quantite) VALUES (1,"tube3","2015-01-02",3);
INSERT INTO composer_production(numproduction, reftube, datejour, quantite) VALUES (1,"tube5","2015-01-02",18);
INSERT INTO composer_production(numproduction, reftube, datejour, quantite) VALUES (2,"tube6","2015-01-02",17);
INSERT INTO composer_production(numproduction, reftube, datejour, quantite) VALUES (2,"tube7","2015-01-03",15);
INSERT INTO composer_production(numproduction, reftube, datejour, quantite) VALUES (2,"tube5","2015-01-03",12);
INSERT INTO composer_production(numproduction, reftube, datejour, quantite) VALUES (3,"tube5","2015-01-04",11);
INSERT INTO composer_production(numproduction, reftube, datejour, quantite) VALUES (3,"tube6","2015-01-05",13);
INSERT INTO composer_production(numproduction, reftube, datejour, quantite) VALUES (3,"tube8","2015-01-05",6);
INSERT INTO composer_production(numproduction, reftube, datejour, quantite) VALUES (4,"tube1","2015-01-06",7);
```

1.16) Modification d'enregistrements dans la base

UPDATE composer_production SET datejour="2015-01-07",quantite=18 WHERE numproduction=4 and reftube="tube1";

1.18) Controler que le changement a eu lieu

select * from composer_production wHERE numproduction=4 and reftube="tube1";

1.19) Supprimer l'enregistrement cree ci-dessus

delete from composer_production wHERE numproduction=4 and reftube="tube1";

1.20) controler que l'enregistrement a bien été supprimé

select * from composer_production wHERE numproduction=4 and reftube="tube1";

1.21) Creez l'utilisateur albert avec le mot de passe 123

mysql -h localhost -uroot -p
CREATE USER 'albert'@'localhost' IDENTIFIED BY '123';

1.22) Se deconnecter et se connecter avec les informations du nouvel utilisateur

exit
mysql -h localhost -ualbert –p123

1.23) tenter d'ouvrir la base de donné production

Use production ;

Que se passe t'il ?

1.24) Accorder tous les privileges à l'utilisateur albert sur toutes les bases de donnée

exit
mysql -h localhost -uroot -p
grant all privileges on *.* to 'albert'@'localhost';

1.22) Se deconnecter et se connecter avec les informations de l'utilisateur albert et faire les operations suivantes :

exit
mysql -h localhost -ualbert –p123
show databases;
use production;
select * from production;
select * from client;

Que se passe t'il ?

1.23) Accorder tous les privileges à l'utilisateur albert sur toutes les tables de la base de donnee production

exit
mysql -h localhost -uroot -p
grant all privileges on production.* to 'albert'@'localhost';*

1.24 Se connecter en tant que albert et verifier les privileges

mysql -h localhost -ualbert -p123
show databases;
use production;

show tables;
select * from commande;
insert into commande (num, refClient) values (7,"client1");
create user 'patrick'@'localhost' identified by '123';
Que se passe t'il ?

1.26) Supprimer les privileges de l'utilisateur albert

mysql -h localhost -uroot -p
REVOKE ALL PRIVILEGES ON *.* FROM 'albert'@'localhost';
Exit
mysql -h localhost -ualbert –p123
show databases;
use production;
que se passe t'il?

1.27) Accorder le priviliege de faire des requêtes sur la table production de la base production à albert ainsi que le droit d'accorder ce privilege à d'autres utilisateurs

mysql -h localhost -uroot -p
grant SELECT ON production . * TO 'albert'@'localhost' WITH GRANT OPTION ;
create user 'pierre'@'localhost' identified by '123';
Exit
mysql -h localhost -ualbert -p123
grant select on production.* to 'pierre'@'localhost';

1.28) Afficher la liste de toutes les informations de la table composer_production

select * from composer_production;
select numproduction, reftube,datejour,quantite from composer_production;

1.28) Afficher la quantite total, la moyenne, le maximum, le minimum des productions

select sum(quantite),avg(quantite), max(quantite),min(quantite) from composer_production;

1.29) Afficher le poids total des productions

select sum(quantite*poids) from composer_production,tube where reftube=ref;

1.30) Afficher le poids total des productions en utilisant les alias

mysql> select sum(c.quantite*t.poids) from composer_production c,tube t where reftube=ref;

1.31) Afficher le poids total des productions par tube

select reftube,sum(quantite*poids) from composer_production,tube where reftube=ref group by reftube;

1.32) Afficher les productions tries par quantite

select * from composer_production order by quantite;

1.33) Afficher les productions tries par quantite par decroissant

select * from composer_production order by quantite desc;

1.34) Afficher les commandes qui ne sont pas encore en production

select num ,refClient from commande where num not in (select numcommande from production);

1.35Écrire la commande SQL qui permet d'afficher en début de journée le poids total de tubes à produire pour chaque commande

Raisonnement:

1) Il faut trouver les informations à afficher dans la requête
   numCommande ,Poids total des tubes=quantite*poids
2) Il faut trouver à quelle table elles appartiennent

3) il faut determiner les conditions
4) Il faut etablir les jointures entre les tables

SELECT numCommande, SUM(quantite * poids) FROM PRODUCTION,COMPOSER_PRODUCTION, TUBE WHERE PRODUCTION.num = numProduction AND refTube = TUBE.ref AND dateJour = "2010-10-14" GROUP BY numCommande

1.36) Realiser la requête pour determinez le nombre de commande par tube

select reftube, count(*) from composer_commande group by reftube;

1.37) **Réalisez la requête permettant de determinez les tubes pour lesquels le nombre de commande dépasse 3**

select reftube, count(*) from composer_commande group by reftube having c
ount(*)>3;

1.38) On decide de changer les codes des tubes avec le tableau ci dessous

ANCIEN CODE	NOUVEAU CODE
tube1	tub1
tube2	tub2
tube3	tub3
tube4	tub4
tube5	tub5
tube6	tub6
tube7	tub7
tube8	tub8

Que faut il faire d'après vous sur la base de donnée ? Proposer une solution ?

1.39) calculer le nombre de production par atelier pour le mois de janvier 2015
select numatelier, count(*) from composer_production, tube where reftube=ref and datejour between "2015-01-01" and "2015-01-31" group by numatelier;

140) Autoriser l'utilisateur LATE à modifier et à supprimer des lignes dans la table client.

grant update, delete on production.client to albert;

1.50) LiSTER La quantite produite PAR TUBE ET PAR ATELIER
SELECT reftube, numatelier, sum(quantite) from composer_production,tube where reftube=ref group by reftube, numatelier

1.51) Lister la production pour laquelle la quantite produite est maximum

select numproduction from composer_production where quantite=(select Max(quantite) from composer_production);

1.52) Lister les quantites maximales produites par tube;
select max(quantite) from composer_production group by reftube;

1.53) Lister le nombre de commande par client
select refClient, count(*) from cOMMANDE group by REFclient;

1.54) Lister la quantite produite par client
mysql> select refclient, sum(quantite) from composer_production cp , production p, commande c where p.num=cp.numproduction and p.numcommande=c.num ;

1.55) Realiser des transactions sur la base de donnee

use production;
start transaction;
insert into client (ref,nom,adr,pays) values ("client4","entreprise du client4","ville du client4","pays du client4");
insert into client (ref,nom,adr,pays) values ("client5","entreprise du client5","ville du client5","pays du client5");
commit;
select * from client;

use production;
start transaction;
insert into client (ref,nom,adr,pays) values ("client6","entreprise du client6","ville du client6","pays du client6");
insert into client (refclient,nom,adr,pays) values ("client7","entreprise du client7","ville du client7","pays du client7");
rollback;
select * from client;