Travaux pratiques 2: BD01

(select … from… where)

Objectif : - exécuter un script SQL
- Apprendre progressivement le SQL et la construction des requêtes en SQL
(cet apprentissage se fera par plusieurs travaux pratiques)

Qu'est ce qu'un fichier script ?

Un **script** est un fichier texte qui contient une **séquence d'instructions SQL**.
Ce fichier doit être ASCII pur. (Il ne peut pas provenir d'un traitement de texte qui injecte des caractères de mise en page)
Chaque instruction est séparée d'un point virgule.
Si on souhaite indiquer un commentaire, on l'entoure ainsi : /* …commentaire…….*/
Exécuter un fichier script produit le même effet qu'écrire et exécuter successivement chacune de ses instructions en mode commande.
Les erreurs présentes dans un script sont signalées à son exécution.
Seules les instructions qui ont généré ces erreurs ne sont pas exécutées.
Si vous relancer dès lors le script après avoir corrigé ces erreurs, vous engendrez de nouvelles erreurs puisque vous tentez d'exécuter à nouveau les instructions qui avaient correctement été exécutées la première fois.

Voici le Script de création et d'insertion de données d'une base de données :

```
CREATE TABLE  `J` (
  `ID_J` char(4) NOT NULL,
  `JNAME` char(20) DEFAULT NULL,
  `CITY` char(10) DEFAULT NULL,
  PRIMARY KEY (`ID_J`)
);


INSERT INTO `J` (`ID_J`, `JNAME`, `CITY`) VALUES
('J1', 'Sorter', 'Paris'),
('J2', 'Display', 'Rome'),
('J3', 'OCR', 'Athens'),
('J4', 'Console', 'Athens'),
('J5', 'RAID', 'London'),
('J6', 'EDS', 'Oslo'),
('J7', 'Tape', 'London');


CREATE TABLE  `P` (
  `ID_P` char(4) NOT NULL,
  `PNAME` char(20) DEFAULT NULL,
  `COLOR` char(8) DEFAULT NULL,
  `WEIGHT` int(11) DEFAULT NULL,
  `CITY` char(10) DEFAULT NULL,
  PRIMARY KEY (`ID_P`)
) ;
```

```
INSERT INTO `P` (`ID_P`, `PNAME`, `COLOR`, `WEIGHT`, `CITY`) VALUES
('P1', 'Nut', 'Red', 12, 'London'),
('P2', 'Bolt', 'Green', 17, 'Paris'),
('P3', 'Screw', 'Blue', 17, 'Rome'),
('P4', 'Screw', 'Red', 14, 'London'),
('P5', 'Cam', 'Blue', 12, 'Paris'),
('P6', 'Cog', 'Red', 19, 'London');


CREATE TABLE `S` (
  `ID_S` char(4) NOT NULL,
  `SNAME` char(20) DEFAULT NULL,
  `STATUS` char(2) DEFAULT NULL,
  `CITY` char(10) DEFAULT NULL,
  PRIMARY KEY (`ID_S`)
);


INSERT INTO `S` (`ID_S`, `SNAME`, `STATUS`, `CITY`) VALUES
('S1', 'Smith', '20', 'London'),
('S2', 'Jones', '10', 'Paris'),
('S3', 'Blake', '30', 'Paris'),
('S4', 'Clark', '20', 'London'),
('S5', 'Adams', '30', 'Athens');

CREATE TABLE `SPJ` (
  `ID_S` char(4) NOT NULL,
  `ID_P` char(4) NOT NULL,
  `ID_J` char(4) NOT NULL,
  `QTY` int(11) DEFAULT NULL,
  `DATE_DERNIERE_LIVRAISON` date DEFAULT NULL,
  PRIMARY KEY (`ID_S`,`ID_P`,`ID_J`),
  KEY `pk_SPJ_P` (`ID_P`),
  KEY `pk_SPJ_J` (`ID_J`)
);


INSERT INTO `SPJ` (`ID_S`, `ID_P`, `ID_J`, `QTY`, `DATE_DERNIERE_LIVRAISON`) VALUES
('S1', 'P1', 'J1', 200, '2001-10-05'),
('S1', 'P1', 'J4', 700, '2001-05-10'),
('S2', 'P3', 'J1', 400, '2001-05-20'),
('S2', 'P3', 'J2', 200, '2000-07-30'),
('S2', 'P3', 'J3', 200, '2001-05-10'),
('S2', 'P3', 'J4', 500, '2001-10-03'),
('S2', 'P3', 'J5', 600, '2001-09-20'),
('S2', 'P3', 'J6', 400, '2000-05-12'),
('S2', 'P3', 'J7', 800, '2001-08-23'),
('S2', 'P5', 'J2', 100, '2000-06-23'),
('S3', 'P3', 'J1', 200, '2001-07-07'),
('S3', 'P4', 'J2', 500, '2001-05-18'),
('S4', 'P6', 'J3', 300, '2001-05-10'),
('S4', 'P6', 'J7', 300, '2001-09-16'),
('S5', 'P1', 'J4', 100, '2000-03-18'),
('S5', 'P2', 'J2', 200, '2001-11-10'),
('S5', 'P2', 'J4', 100, '2001-04-17'),
('S5', 'P3', 'J4', 200, '2001-05-19'),
('S5', 'P4', 'J4', 800, '2001-05-10'),
('S5', 'P5', 'J4', 400, '2001-12-16'),
('S5', 'P5', 'J5', 500, '2001-02-08'),
('S5', 'P5', 'J7', 100, '2001-06-25'),
('S5', 'P6', 'J2', 200, '2001-02-09'),
('S5', 'P6', 'J4', 500, '2001-10-10');

ALTER TABLE `SPJ`
  ADD CONSTRAINT `pk_SPJ_S` FOREIGN KEY (`ID_S`) REFERENCES `S` (`ID_S`),
  ADD CONSTRAINT `pk_SPJ_P` FOREIGN KEY (`ID_P`) REFERENCES `P` (`ID_P`),
  ADD CONSTRAINT `pk_SPJ_J` FOREIGN KEY (`ID_J`) REFERENCES `J` (`ID_J`);
```

Exécution d'un fichier script

avant propos.

Ce premier travail consiste à créer la base de données qui servira pour tous vos travaux pratiques.
Importer le script **spj_mysql.sql** et exécutez le. Vous créerez ainsi la base de données 'fournisseur'
Si l'exécution du script ne se déroule pas correctement, un message comportant les erreurs s'affiche.
Ces messages sont souvent peu explicites mais guident quand même un peu la correction de l'erreur.

Travail 1:

Exécutez le script spj_mysql.sql et vérifiez que tout c'est bien passé en exécutant
select * from S.
Vous devrez voir apparaître les contenus suivants :

ID_S	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

La suite du travail se présente sous forme de tutoriel.

Vous êtes invité à lire les informations données et à taper les instructions mises en ***italique*.**

N'oubliez pas d'éxécuter chaque instruction une par une.

SELECT … FROM

Select *liste des champs à afficher* **from** *nom de la table*

Projection

Exemples :

1) affiche la colonne sname de la table s
construction logique: S[sname]
implementation sql : ***Select sname from s***

2) affiche les colonnes sname et city de la table S
S[sname,city]
Select sname,city from s

3) affiche la colonne sname en l'appelant nom
S [sname]
Select sname nom from s

Dans cet exemple on renomme le champ sname pour des raisons d'affichage.
Cela n'a donc aucun sens de le faire dans la construction logique de la requête. On ne renomme un champ ou une table dans une construction logique que si on souhaite exprimer une opération logique entre deux champs de même nom.
La construction logique est un outil de raisonnement, pas de présentation.

4) Affichez les villes des fournisseurs

S[city]

Select city from S

Ou

select distinct city from S /* distinct évite les affichages de doublons */

Travail 2:

Imaginez mentalement ce que l'instruction affichera et exécutez la **ensuite** pour vérifier ?

1. ***Select * from J***
2. ***Select sname , city from S***
3. ***Select sname nom,city ville from S***

SELECT … FROM….. WHERE ….

Select *liste des champs à afficher* **from** *nom de la table* **WHERE** *critère de sélection.*

Projection

Selection

La clause where exprime la condition de sélection (restriction) à effectuer sur le résultat obtenu de select… from.

Principe :

1. On construit la table correspondante à select … from
2. On ne conserve que les lignes pour lesquelles l'évaluation de la condition est vrai.

En réalité, l'instruction SQL sera optimisée par le système. Rien ne dit que le travail s'effectuera vraiment de cette manière. La seule certitude qu'on ait est que le résultat sera semblable si on effectuait le travail de cette manière.

La condition peut être une condition composée s'exprimant entre autre à l'aide des opérateurs =,>,<, and, or , not, between, in, any, exists (in, any,exists seront vu plus tard)

Exemples

1) (S where status=20)[sname]
   ☛ ***select sname from S where status=20***

2) S where status>20 and city='Paris'
   ☛ ***select * from s where status>20 and city='Paris'***

3) P where (color='Red' or color='Green') and weight >15
   - ☛ ***select * from P where (color='Red' or color='Green') and weight >15***

4) (spj where qty >= 500 and qty <=700)[id_p,qty]
   - ☛ ***select id_p,qty from spj where qty >=500 and qty <=700***
   ou
   - ☛ ***select id_p,qty from spj where qty between 500 and 700***

5) (spj where qty < 500 or qty >700)[id_p,qty]
   - ☛ ***select id_p,qty from spj where qty < 500 or qty >700***
   ou
   - ☛ ***select id_p,qty from spj where qty not between 500 and 700***

6) (P where Pname='Bolt' or Pname='Cam')[pname,color]
   - ☛ ***select pname,color from P where pname='Bolt' or pname='Cam'***
   ou
   - ☛ ***select pname,color from P where pname in ('Bolt' ,'Cam')***

EXERCICES

1) On veut afficher le contenu de la table des fournisseurs.

ID_S	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

2) On veut le nom et la ville de tous les fournisseurs avec comme intitulé NOM et VILLE

NOM	VILLE
Smith	London
Jones	Paris
Blake	Paris
Clark	London
Adams	Athens

3) On veut le nom des fournisseurs dont la ville est London ou Paris

SNAME
Smith
Jones
Blake
Clark

4) On veut le nom des fournisseurs dont le status vaut strictement moins de 25 et qui sont de Paris.

SNAME
Jones

5) On veut obtenir le nom des fournisseurs dont le status **n'est pas** dans l'intervalle fermé [15,25]

SNAME
Jones
Blake
Adams