

IUT de Montpellier - Base de Données - TD7

SQL Manipulation des données

Rémi COLETTA

14 octobre 2021

Le langage SQL comporte 3 sous-ensembles contenant des commandes différentes :

- langage de requêtes : commande SELECT
- langage de description de données (LDD) : commandes CREATE, ALTER, DROP
- langage de manipulation de données (LMD) : commandes INSERT, DELETE, UPDATE

Ce TD porte sur l'utilisation des parties LDD et LMD toujours sur le cas Produit-Achat-Client

1 Langage de description de données

1.1 Creation des tables

La création des tables précise :

- les colonnes avec leur type et leur contrainte (par exemple NOT NULL)
- les contraintes (clé primaire et clé étrangère)

Colonnes et contraintes sont séparées par des virgules.

Exemple 1 Ici on crée une table avec 2 attributs, et une clé primaire composée d'un seul attribut.

```
CREATE TABLE client
(ncli INTEGER ,
 nom VARCHAR(10) NOT NULL,
 CONSTRAINT pk_client PRIMARY KEY(ncli));
```

En cas de violation d'une contrainte d'intégrité, c'est son nom qui vous sera affiché. Il faut donc choisir des noms intelligibles.

Par convention, "pk" (pour Primary Key) suivi du nom de la table.

Un second exemple avec des contraintes de clés étrangères :

Exemple 2 Ici la table a une clé primaire composite (qui porte sur deux attributs).

La définition d'une clé étrangère décrit quel est l'attribut de la table qui est une clé étrangère ainsi que l'attribut pointé/référencé dans l'autre table.

```
CREATE TABLE achat
(ncli INTEGER ,
 np INTEGER ,
 qa INTEGER,
 CONSTRAINT stock check (qa>0),
 CONSTRAINT pk_achat PRIMARY KEY (ncli,np),
 CONSTRAINT fk_achat_ncli FOREIGN KEY (ncli) REFERENCES client(ncli),
 CONSTRAINT fk_achat_np FOREIGN KEY (np) REFERENCES produit(np));
```

Par convention on nomme les clés étrangère, "fk" (pour Foreign Key) suivi du nom de la table, suivi du nom de l'attribut, car il peut y avoir plusieurs clés étrangères par table.

1.2 Suppression des tables

Du fait des contraintes d'intégrité référentielles (clés étrangères), il faut supprimer les tables dans l'ordre inverse de l'ordre de création.

Exemple 3

```
DROP TABLE achat;  
DROP TABLE client;  
DROP TABLE produit;
```

1.3 Modification des tables

Exemple 4 Ajouter une colonne date_inscription de type DATE à une la table Client :

```
ALTER TABLE client ADD date_inscription DATE DEFAULT CURRENT_DATE;
```

Ici DEFAULT spécifie une valeur par défaut pour l'attribut date_inscription, qui sera CURRENT_DATE (la date du jour).

On peut ajouter une contrainte avec la même syntaxe.

Exemple 5 Modifier la colonne nom de la table Client pour passer de 10 à 20 caractères :

```
ALTER TABLE client MODIFY nom VARCHAR(20);
```

Exemple 6 Supprimer la colonne date_inscription de la table Client :

```
ALTER TABLE client DROP COLUMN date\_inscription;
```

On peut aussi supprimer une contrainte :

```
ALTER TABLE marchand DROP CONSTRAINT pk_marchand;
```

2 Langage de modification de données

2.1 Insertion de tuple

Exemple 7 Pour insérer un tuple en renseignant tous les attributs

```
INSERT INTO produit VALUES (1,'Agrafeuse','Rouge',180);
```

Si on ne veut renseigner que les valeurs de certains attributs, on peut en préciser la liste :

Exemple 8 Pour insérer un tuple en ne renseignant que certains attributs

```
INSERT INTO produit(np,lib,qs) VALUES (21,'Agrafeuse',180);
```

Les attributs non renseignés prendront la valeur NULL ou la valeur par DEFAULT si elle a été spécifiée dans la définition de la colonne (voir l'exemple 4).

2.2 Suppression de tuple

Attention on supprime toute une ligne d'un coup, pas valeur par valeur.

Exemple 9 Pour supprimer un tuple unique, il convient de l'identifier par sa clé primaire

```
DELETE FROM produit WHERE np= 21 ;
```

On peut aussi faire des suppressions de masse :

Exemple 10 Supprimer tous les produits orange (j'aime pas le orange, sauf pour les Kangoo tunning).

```
DELETE FROM produit WHERE coul = 'Orange' ;
```

Le cas des jointures :

Exemple 11 Je veux les lignes de la table achat qui correspondent à des achat de 'Lampe' :

```
SELECT a.np,ncli FROM achat a JOIN produit p ON a.np = p.np WHERE lib = 'Lampe' ;
```

Cette requête peut aussi s'écrire avec une sous-requête imbriquée :

```
SELECT a.np,ncli FROM achat a
WHERE np IN (SELECT np FROM produit p WHERE lib = 'Lampe');
```

Si je veux supprimer de la table achat, tous les achats de lampe, je suis obligé d'utiliser la forme imbriquée :

```
DELETE FROM achat a
WHERE np IN (SELECT np FROM produit p WHERE lib = 'Lampe');
```

En effet le JOIN n'est pas autorisé dans un DELETE, car SQL ne saurait dans laquelle des deux tables (achat ou produit) vous souhaitez supprimer des tuples.

2.3 Modification de tuple

Exemple 12 Changer la couleur des Calculatrice

```
UPDATE produit SET coul='Blanc' WHERE lib='Calculatrice';
```

On peut modifier 2 valeurs en même temps :

Exemple 13 Changer la couleur du produit 2 et augmenter le stock de 100

```
UPDATE produit SET coul='Blanc', qs = qs +100 WHERE np = 2 ;
```

3 A vous de jouer :

Se connecter à Oracle avec isqlplus

Dans un navigateur, ouvrez l'url <http://orainfo.iutmontp.univ-montp2.fr:5560/isqlplus/> (accessible aussi de chez vous) et identifiez vous avec votre login, votre mot de passe, et mettez IUT comme identificateur de connexion.

Pas besoin de ré-exécuter le script le script PAC.sql à l'url <http://www.lirmm.fr/~coletta/Cours/BD/PAC.sql> . Les tables et les tuples doivent toujours exister.

A tout moment,

- connaître la liste de vos tables : `SELECT table_name FROM user_tables;`
- pour connaître la structure d'une table : `DESCRIBE le_nom_de_la_table;`
- Lister les tuples d'une table : `SELECT * FROM le_nom_de_la_table;`

3.1 INSERT

1. insérer dans la relation produit un nouveau produit de numéro 20 correspondant à un cahier blanc dont on dispose de 100 exemplaires.
2. insérer un nouveau produit dans la relation produit de numéro 21 correspondant à une gomme dont on dispose de 500 exemplaires (on ne connaît pas sa couleur). (Utiliser la syntaxe présentée à l'exemple 8).
3. essayer d'insérer dans la relation produit un nouveau produit de libellé "trombone", de couleur "gris", disponible en 1000 exemplaires mais sans préciser de numéro.
4. (★) tenter d'insérer à nouveau le produit de la question 1.
 - Pourquoi ça ne marche pas ?
 - Aider vous du nom de la contrainte violée pour comprendre.
5. insérer dans la relation achat un nouvel achat du client 101 pour le produit 20 de quantité 12.

3.2 DELETE

1. supprimez de la relation produit le tuple concernant le produit 21.
2. tentez de faire la même chose pour le produit 20.
 - Pourquoi ça ne marche pas ? (Le nom de la contrainte violée doit vous aider)
 - Que devez vous faire préalablement
3. supprimez de la relation achat tous les achats dont la quantité achetée est inférieure à 5.
4. supprimez de la relation achat tous les achats du client Ullman (sans utiliser directement son ncli). Il faudra utiliser une requête imbriquée. Les étapes ci-dessous sont facultatives, juste là pour vous aidez.
 - écrivez la requête qui récupère le numéro de client de Ullman
 - écrivez la requête "comme si" vous connaissiez par coeur le numéro de client de Ullman
 - (★) remplacez ce numéro de client par une la sous requête ci-dessus.

3.3 UPDATE

1. augmenter de 10 la quantité d'agrafeuses achetées par Defrere.
2. augmenter simultanément la quantité en stock de 10 et modifier en jaune la couleur des crayons de luxe rouges.
3. (★) augmenter de 2 toutes les quantité achetées lors des achats effectués par Defrere.
4. essayer de remplacer le numéro de produit 10 par le numéro 30. Que se passe-t-il ?

4 Définition des relations

4.1 ALTER

1. ajouter à la relation client un attribut adr (adresse) de type chaîne de caractères de longueur 40 max.
2. modifier le type de l'attribut adr de la relation client pour accepter des chaînes de 80 caractères max.
3. ajouter 'Montpellier' comme adresse à tous les clients.
4. (★) faire en sorte que tous les futurs clients aient 'Mende' comme adresse par défaut.
5. tester en créant un nouveau client sans lui spécifier d'adresse.

4.2 CREATE

1. créer une relation correspondant au schéma de relation suivant : FOURNISSEUR(NF, NOM, ADR) avec NF : numéro de fournisseur (entier) clé primaire NOM et ADR : nom et adresse du fournisseur de type chaîne de caractères. Ces attributs sont obligatoires.
2. Insérer les tuples suivants :

nf	nom	adr
1001	Bureau Valle	Les Sables d Olonne
1002	BIC	Montpellier
1003	NADA	Mende

3. Essayer d'insérer les tuples suivants. Si vos contraintes d'intégrité sont correctes cela ne doit pas fonctionner.

nf	nom	adr	Erreur ?
1004	Clairefontaine		
1002	Epson	Paris	

4. (★) créer une relation correspondant au schéma de relation suivant : REAP(NF,NP,DATE_REAP, QR) qui spécifie pour un fournisseur donné, un produit donné à une date donnée une quantité de réapprovisionnement. Par défaut [default] (ie. si la date n'est pas précisée lors de l'insertion d'un tuple), la date est la date courante [current_date].
5. Insérer les tuples suivants :

nf	np	date_reap	qr
1001	1	date du jour	51
1001	4	date du jour	100
1001	5	date du jour	100
1001	6	date du jour	100
1001	7	date du jour	100
1001	7	'01/01/2020'	100
1002	7	'01/01/2020'	500
1003	1	'01/01/2020'	49
1003	9	'01/01/2020'	100
1003	9	'01/02/2020'	50
1003	10	'01/01/2020'	200

6. Essayer d'insérer les tuples suivants. Si vos contraintes d'intégrité sont correctes cela ne doit pas fonctionner.

nf	np	date_reap	qr	Erreur ?
1002	1	date du jour		
1001	7	'01/01/2020'	101	
1004	4	date du jour	3	

4.3 Quelques requêtes avec les opérateurs relationnels et ensemblistes

Q1 Quels sont les numéros fournisseurs qui ont déjà réapprovisionné des 'Lampe' ou des 'Agriculteur' ?

Correction

```
SELECT DISTINCT nf FROM reap r JOIN produit p ON p.np=r.np WHERE lib ='Lampe' OR Lib= 'Agriculteur'
```

NF

1003
1001
1002

Q2 Quels sont les noms de fournisseurs qui ont déjà approvisionné des 'Lampe' et des 'Agriculteur' ?

Correction

```
SELECT nom FROM fournisseur f JOIN reape r ON r.nf = f.nf JOIN produit p ON p.np=r.np WHERE  
INTERSECT  
SELECT nom FROM fournisseur f JOIN reape r ON r.nf = f.nf JOIN produit p ON p.np=r.np WHERE  
NOM  
-----  
Bureau Valle
```

Q3 Quels sont les fournisseurs qui n'ont jamais réapprovisionné de 'Lampes' ?

Correction

```
SELECT nom FROM fournisseur  
MINUS  
SELECT nom FROM fournisseur f JOIN reape r ON r.nf = f.nf JOIN produit p ON p.np=r.np WHERE  
NOM  
-----  
NADA
```

Q4 Quels sont les fournisseurs qui n'ont plus réapprovisionné depuis le '02/02/2020' ?

Correction

```
SELECT nom FROM fournisseur  
MINUS  
SELECT nom FROM fournisseur f JOIN reape r ON r.nf = f.nf WHERE date_reape > '02/02/2020'  
NOM  
-----  
BIC  
NADA
```

Q5 (★) Quels sont les numéros de fournisseurs qui ont déjà approvisionné (au moins) deux produits ?

Correction

```
SELECT DISTINCT r.nf FROM reape r JOIN reape r2 ON r.nf = r2.nf WHERE r.np != r2.np  
NF  
---  
1003  
1001
```

Q6 Quels sont les numéros des produits qui ont déjà réapprovisionné (au moins) deux fois ?
Attention : deux réapprovisionnements peuvent différer par la date ou par le fournisseur.

Correction

```
SELECT DISTINCT r.np FROM reape r JOIN reape r2 ON r.np = r2.np WHERE (r.nf != r2.nf OR r  
NP  
----  
1  
7  
9
```

Q7 (★) Quels sont les produits qui ont déjà approvisionné (au moins) deux fois par des fournisseurs différents et en quantité supérieure à 50 ?

Correction

```
SELECT DISTINCT r.np FROM reap r JOIN reap r2 ON r.np = r2.np WHERE r.nf != r2.nf AND r.np > 50
```

NP

7

Q8 Quels sont les numéros de fournisseurs qui ont déjà approvisionné tous les produits ?

Correction

```
SELECT nf FROM Fournisseur
MINUS
SELECT nf FROM (
    SELECT nf, np FROM Fournisseur, Produit
MINUS
    SELECT nf, np FROM reap);
```

aucune ligne sélectionnée

Q9 (★) Quels sont les fournisseurs qui ont déjà approvisionné toutes les Lampe(s) ?

Correction

```
SELECT nf FROM Fournisseur
MINUS
SELECT nf FROM (
    SELECT nf, np FROM Fournisseur, Produit WHERE lib='Lampe'
MINUS
    SELECT nf, np FROM reap);
```

NF

1001