Département Informatique et Statistique, Faculté de SEG, Université Lumière Lyon 2
M1 Informatique – Année 2010-2011

Langages de requêtes – TD n° 5

J. Darmont (*http://eric.univ-lyon2.fr/~jdarmont/*), 05/01/11

Exercice 1 : Vues

Soit le schéma relationnel suivant, disponible depuis le compte de l'utilisateur DARMONT.

CLIENT (NumCli, Nom, Prenom, DateNaiss, CP, Rue, Ville)
PRODUIT (NumProd, Desi, PrixUni) Clés primaires Clés étrangères#
COMMANDE (NumCli#, NumProd#)

1. Formuler à l'aide du langage SQL les requêtes suivantes (sans recopier les tables – rappel : l'accès aux tables d'un autre utilisateur se fait en préfixant le nom de la table par le nom de l'utilisateur, ex. `darmont.client`).

- Liste des clients (nom + prénom) qui ont commandé le produit n° 102.
- Nom des clients qui ont commandé au moins un produit de prix supérieur ou égal à 500 €.
- Nom des clients qui n'ont commandé aucun produit.
- Nom des clients qui n'ont pas commandé le produit n° 101.
- Nom des clients qui ont commandé tous les produits.

2. Créer une vue nommée `clicopro` permettant de visualiser les caractéristiques des produits commandés par chaque client (attributs à sélectionner : NumCli, Nom, Prenom, NumProd, Desi, PrixUni).

3. Lister le contenu de la vue `clicopro`.

4. Reformuler les deux premières requêtes de la question 1 en utilisant la vue `clicopro`. Commentaire ?

5. Formuler les requêtes suivantes en utilisant la vue `clicopro`.

- Pour chaque client, prix du produit le plus cher qui a été commandé.
- Pour chaque client dont le prénom se termine par la lettre 'e', prix moyen des produits commandés.
- Maximum des totaux des prix pour tous les produits commandés par les différents clients.
- Numéros des produits commandés plus de deux fois.

6. Créer une vue nommée `clipro` basée sur `clicopro` et permettant d'afficher seulement les attributs Nom, Prenom et Desi. Lister le contenu de la vue `clipro`.

7. Détruire la vue `clicopro`. Lister le contenu de la vue `clipro`. Conclusion ?

Exercice 2 : Catalogue du système

1. Lister toutes les tables qui vous sont accessibles (par leur nom), ainsi que leurs propriétaires, en interrogeant la vue système `ALL_TABLES`.

2. Lister les tables et les vues de votre compte, ainsi que leurs types (table ou vue), à l'aide de la vue système `USER_CATALOG`.

3. Lister toutes les contraintes d'intégrité définies sur vos tables à l'aide de la vue système `USER_CONSTRAINTS`. Afficher pour chaque contrainte son nom, la table à laquelle elle s'applique, son type, et sa « condition de recherche ».

4. Créer la table `DEMO_CONTRAINTES` en définissant les attributs et les contraintes suivants, **tels quels**.

- `NUM NUMBER(1) PRIMARY KEY`
- `PASNUL NUMBER(1) NOT NULL`
- `PASNUL2 NUMBER(1)`
- `CONSTRAINT PASNEG CHECK (PASNUL>=0)`
- `CONSTRAINT PASNUL2C CHECK (PASNUL2 IS NOT NULL)`
- `CONSTRAINT REFNUM FOREIGN KEY(PASNUL2) REFERENCES DEMO_CONTRAINTES(NUM)`

5. Même question que la question 3, mais uniquement pour la table `DEMO_CONTRAINTES`. Comment et par qui sont nommées les contraintes ?

6. Lister les contraintes d'intégrité `NOT NULL` de la table `DEMO_CONTRAINTES` à l'aide de la commande `DESC`. Retrouve-t-on toutes les contraintes définies à la question 4 ? D'où vient celle qui apparaît en plus ?

7. À l'aide de la vue système `ALL_CONS_COLUMNS`, lister les contraintes qui sont associées à la table `DEMO_CONTRAINTES` ainsi que les attributs auxquels elles s'appliquent. Penser à filtrer le résultat selon le propriétaire de la table (votre login).

Exercice 3 : Vues et catalogue du système

1. Créer la vue PNOM (PLNUM, PLNOM) à partir de la table PILOTE du TD n° 1. Vérifier son contenu.
2. À travers la vue PNOM, modifier le nom du pilote n° 5 en « DARMONT ». Consulter le contenu de la vue PNOM et de la table PILOTE.
3. Créer la vue VOLS (VOLNUM, PLNOM, AVNOM), associant à chaque numéro de vol le nom du pilote et le nom de l'avion, à partir des tables PILOTE, AVION et VOL du TD n° 1. Vérifier son contenu. Quel est l'intérêt de définir cette vue ?
4. À travers la vue VOLS, modifier le nom du pilote du vol n° 4 en « Sinbad ». Que se passe-t-il ?
5. Insérer un n-uplet quelconque dans la vue VOLS. Que se passe-t-il ?
6. À partir de la vue système USER_TAB_COLUMNS, afficher les attributs de la table PILOTE.
7. À partir de la vue système USER_TAB_COLUMNS, afficher le nom des tables et des vues qui ont pour attribut PLNUM.

8. À partir de la vue système ALL_TABLES, afficher le nom des tables dont vous êtes propriétaire.
9. Créer la vue MES_TABLES à partir du résultat de la requête 8. Vérifier son contenu.

Exercice 4 complémentaire : Vues, catalogue du système et droits d'accès

1. Une base de données ancienne, gérée par M. Scott, aujourd'hui à la retraite, doit être réorganisée et mise en troisième forme normale (3FN). Pour cela, il faut déterminer les dépendances fonctionnelles entre les attributs de cette base.

- Vous avez accès en lecture (privilège SELECT) à toutes les données de l'utilisateur SCOTT. Créer à l'aide de SQL une vue ATTRIBUTS permettant de lister tous les attributs de toutes les tables de l'utilisateur SCOTT ainsi que leur type (sans doublon). Utiliser pour cela la vue ALL_TAB_COLUMNS du catalogue du système. Appliquer la commande `DESC` à ALL_TAB_COLUMNS pour visualiser ses attributs.
- Vérifier le résultat obtenu. Ca ne vous rappelle rien ?
- La mise en 3FN devant être effectuée par quelqu'un d'autre, octroyer à tous les utilisateurs le droit d'accéder en lecture à la vue ATTRIBUTS.

2. Créer, à partir du catalogue du système, une vue nommée MES_CONTRAINTES permettant d'afficher les contraintes définies sur chacun des attributs de vos tables et contenant les attributs suivants :

- nom de la table,
- type de la table,
- nom de l'attribut,
- type de l'attribut,
- nom de la contrainte,
- type de la contrainte,
- condition de recherche (pour les contraintes de domaine).

Correction Exercice 1

```
-- 1.1
select nom, prenom
from darmont.client c1, darmont.commande c2
where c1.numcli=c2.numcli
and numprod=102;

-- 1.2
select distinct nom
from darmont.client c1, darmont.commande c2, darmont.produit p
where c1.numcli=c2.numcli
and c2.numprod=p.numprod
and prixuni>=500;

-- 1.3
select nom from darmont.client C1 where not exists(
select * from darmont.commande C2 where C2.numcli = C1.numcli);

-- 1.4
select nom from darmont.client where numcli not in (
select numcli from darmont.commande where numprod = 101);

-- 1.5
select nom from darmont.client C1 where not exists (
select * from darmont.produit P where not exists (
select * from darmont.commande C2
where C2.numcli = C1.numcli and C2.numprod = P.numprod));
ou
select nom from darmont.client c1, darmont.commande c2
where c1.numcli=c2.numcli
group by nom
having count(distinct numprod)=(select count(numprod) from produit);

-- 2
create view clicopro as
select c1.numcli, nom, prenom, p.numprod, desi, prixuni
from darmont.client c1, darmont.commande c2, darmont.produit p
where c1.numcli=c2.numcli
and c2.numprod=p.numprod;

-- 3
select * from clicopro;

-- 4.1
select nom, prenom from clicopro where numprod=102;

-- 4.2
select distinct nom from clicopro where prixuni>=500;

-- 5.1
select nom, max(prixuni) from clicopro group by nom;

-- 5.2
select nom, avg(prixuni) from clicopro where prenom like '%e' group by nom;

-- 5.3
select max(sum(prixuni)) from clicopro group by numcli;

-- 5.4
select numprod from clicopro group by numprod having count(*)>2;
```

```
-- 6.1
create view clipro as select nom, prenom, desi from clicopro;

-- 6.2
select * from clipro;

-- 7.1
drop view clicopro;

-- 7.2
select * from clipro;
```

Correction Exercice 2

```
-- 1
select owner, table_name from all_tables;

-- 2
select * from user_catalog;

-- 3
select table_name, constraint_name, constraint_type, search_condition
from user_constraints;

-- 4
create table demo_contraintes (
     num number(1) primary key,
     pasnul number(1) not null,
     pasnul2 number(1),
     constraint pasneg check (pasnul>=0),
     constraint pasnul2c check(pasnul2 is not null),
     constraint refnum foreign key(pasnul2) references demo_contraintes(num));

-- 5
select constraint_name, constraint_type, search_condition from user_constraints
where table_name='DEMO_CONTRAINTES';
```

Les contraintes nommées `SYS*` sont définies par le système.

```
-- 6
desc demo_contraintes
```

La contrainte `NOT NULL` sur `PASNUL2` n'apparaît pas car elle est définie comme contrainte de domaine (`CHECK`).
La clé primaire (qui doit être UNIQUE et NON NULLE) apparaît en plus.

```
-- 7
select constraint_name, column_name from all_cons_columns
where owner=USER and table_name='DEMO_CONTRAINTES';
```

Correction Exercice 3

```
-- 1
create view pnom as select plnum, plnom from pilote;

-- 2
update pnom set plnom='DARMONT' where plnum=5;
```



```
-- 3
create view vols as select volnum, plnom, avnom from pilote p, avion a, vol v
where p.plnum=v.plnum and a.avnum=v.avnum;

-- 4
update vols set plnom='Sinbad' where volnum=4;
--ORA-01779: impossible de modifier une colonne correspondant à une table non
--protégée par clé

-- 5
insert into vols values (17,'Toto','Bibi');
--ORA-01776: impossible de modifier plus d'une table de base via une vue jointe

-- 6
select column_name from user_tab_columns where table_name='PILOTE';

-- 7
select table_name from user_tab_columns where column_name='PLNUM';

-- 8
select table_name from all_tables where owner=USER;

-- 9
create view mes_tables as select table_name from all_tables
where owner=USER;
```

Correction Exercice 4

```
-- 1.1
DESC ALL_TAB_COLUMNS

-- 1.2
CREATE VIEW ATTRIBUTS AS
SELECT DISTINCT COLUMN_NAME, DATA_TYPE
FROM ALL_TAB_COLUMNS
WHERE OWNER='SCOTT';

-- 1.3
GRANT SELECT ON ATTRIBUTS TO PUBLIC;

-- 2
CREATE VIEW MES_CONTRAINTES AS
SELECT t.TABLE_NAME, TABLE_TYPE, a.COLUMN_NAME, DATA_TYPE, c2.CONSTRAINT_NAME,
     CONSTRAINT_TYPE, SEARCH_CONDITION
FROM USER_CONS_COLUMNS c1, USER_CATALOG t, USER_TAB_COLUMNS a,
     USER_CONSTRAINTS c2
WHERE c1.TABLE_NAME = t.TABLE_NAME
AND c1.TABLE_NAME = a.TABLE_NAME AND c1.COLUMN_NAME = a.COLUMN_NAME
AND c1.CONSTRAINT_NAME = c2.CONSTRAINT_NAME
ORDER BY t.TABLE_NAME, a.COLUMN_NAME, c2.CONSTRAINT_NAME;
```