Corrigé du R2.06-Exploitation BD (Séance n° 6)
Tables système, Vues et Privilèges d'accès et Rappels SQL

Première étape : Consultation des tables systèmes

Q1 Quel est le nom de tous les attributs de la relation PROF ?

```
SELECT COLUMN_NAME
FROM USER_TAB_COLUMNS
WHERE TABLE_NAME = 'PROF' ;
```

Q2 Donnez la liste des tables de la base IUT ?

```
SELECT OBJECT_NAME
FROM USER_OBJETS
WHERE OBJECT_TYPE = 'TABLE';
```

Q3 Quelles sont toutes les contraintes d'intégrité définies sur les relations de données ?

```
SELECT CONSTRAINT_NAME
FROM USER_CONSTRAINTS ;
```

Q4 Retrouvez le nom des contraintes définies sur toutes les relations de la base IUT ayant un attribut de type NUMBER.

```
SELECT CONSTRAINT_NAME
FROM USER_CONSTRAINTS
WHERE TABLE_NAME IN (SELECT TABLE_NAME
                     FROM USER_TAB_COLUMNS
                     WHERE DATA_TYPE = 'NUMBER') ;
```

Q5 Retrouvez le nom des contraintes et le nom de l'attribut de type NUMBER sur lequel elles portent pour toutes les relations de la base IUT.

```
SELECT DISTINCT CC. CONSTRAINT_NAME, CC. COLUMN_NAME, DATA_TYPE
FROM USER_CONS_COLUMNS CC, USER_TAB_COLUMNS ATT
WHERE CC. COLUMN_NAME = ATT.COLUMN_NAME AND
       CC.TABLE_NAME = ATT.TABLE_NAME AND DATA_TYPE = 'NUMBER' ;
```

Q6 Retrouvez le nom des contraintes, le nom de l'attribut sur lequel elles portent et le type de cet attribut, pour toutes les relations de la base IUT ayant un attribut de type NUMBER.

```
SELECT CI. CONSTRAINT_NAME, CI. COLUMN_NAME, DATA_TYPE
FROM USER_CONS_COLUMNS CI, USER_TAB_COLUMNS ATT
WHERE CI. COLUMN_NAME = ATT.COLUMN_NAME
AND CI. TABLE_NAME = ATT.TABLE_NAME AND CI. TABLE_NAME IN (
            SELECT TABLE_NAME
            FROM USER_TAB_COLUMNS
            WHERE DATA_TYPE = 'NUMBER') ;
```

Q7 Trouvez le nom des contraintes de clef primaire dans la base IUT.

```
SELECT CONSTRAINT_NAME
FROM USER_CONSTRAINTS
WHERE CONSTRAINT_TYPE = 'P' ;
```

Q8 Quels sont les attributs portant le même nom dans des relations différentes de la base ? Affichez nom attribut 1, nom relation 1, nom attribut 2, nom relation 2.

```
SELECT AT1.COLUMN_NAME, AT1.TABLE_NAME, AT2.COLUMN_NAME, AT2.TABLE_NAME
FROM USER_TAB_COLUMNS AT1, USER_TAB_COLUMNS AT2
WHERE AT1.COLUMN_NAME = AT2.COLUMN_NAME AND
      AT1.TABLE_NAME <> AT2.TABLE_NAME ;
```

Q9 Comment formuler la requête précédente en évitant que les couples soient affichés deux fois ((A1, A2) et (A2, A1)) ?

```
SELECT AT1.COLUMN_NAME, AT1.TABLE_NAME, AT2.COLUMN_NAME, AT2.TABLE_NAME
FROM USER_TAB_COLUMNS AT1, USER_TAB_COLUMNS AT2
WHERE AT1.COLUMN_NAME = AT2.COLUMN_NAME AND
      AT1.TABLE_NAME < AT2.TABLE_NAME ;
```

Deuxième étape : Création et manipulation de vues

Vues de consultation

Q10 Créez une vue appelée PROF_INFO2 donnant le numéro, le nom et le prénom des professeurs du département Informatique enseignant en deuxième année.
Puis consultez la vue par une requête d'interrogation.

Création de la vue :

```
CREATE VIEW PROF_INFO2 (NUM, NOM, PRENOM) AS
SELECT PROF.NUM_PROF, NOM_PROF, PRENOM_PROF
FROM PROF, ENSEIGNT, ETUDIANT
WHERE ANNEE = 2 AND
           PROF.NUM_PROF = ENSEIGNT.NUM_PROF AND
           ENSEIGNT.NUM_ET = ETUDIANT.NUM_ET ;
```

Consultation de la vue :

```
SELECT * FROM PROF_INFO2 ;
```

Créer des vues permettant de simplifier les requêtes suivantes (en remplaçant les requêtes imbriquées dans le FROM).

Q11 Donnez l'effectif moyen des groupes de deuxième année.

Création de la vue :

```
CREATE VIEW Q11(EFFECTIF) AS
SELECT COUNT(*)FROM ETUDIANT WHERE ANNEE = 2
GROUP BY GROUPE ;
```

Consultation de la vue :

```
SELECT AVG(EFFECTIF)FROM Q11 ;
```

Q12 Quel est le plus grand nombre d'étudiants qu'un professeur a en cours ?

Création de la vue :

```
CREATE VIEW Q12(EFFECTIF_ET) AS
SELECT COUNT(DISTINCT NUM_ET)
FROM ENSEIGNT
GROUP BY NUM_PROF ;
```

Consultation de la vue :

```
SELECT MAX(EFFECTIF_ET)FROM Q12 ;
```

Q13 Pour les diverses matières (code et libellé) donnez le pourcentage que représente le volume horaire de cours par rapport au total des heures de cours dans la discipline correspondante.

Création de la vue :

```
CREATE VIEW HORAIRE(TOTAL, DISCIPLINE) AS
SELECT SUM(H_COURS_PREV), DISCIPLINE
FROM MODULE
GROUP BY DISCIPLINE ;
```

Consultation de la vue :

```
SELECT CODE, H_COURS_PREV / TOTAL "% discipline"
FROM  HORAIRE, MODULE
WHERE MODULE.DISCIPLINE = HORAIRE.DISCIPLIN
```

Q14 Pour chaque matière, donnez le nombre de professeurs qui l'enseignent ainsi que la note maximale obtenue en test.

Création de deux vues :

```
CREATE VIEW Q141(NB_PRO, CODE) AS
SELECT COUNT(DISTINCT NUM_PROF), CODE
FROM ENSEIGNT
GROUP BY CODE ;

CREATE VIEW Q142(MOY_MAX, CODE) AS
SELECT MAX(MOY_TEST), CODE
FROM NOTATION
GROUP BY CODE ;
```

Consultation de la vue :

```
SELECT Q141.CODE, NB_PROF, MOY_MAX
FROM  Q141, Q142
WHERE Q141.CODE= Q142.CODE ;
```

Q15 Pour la matière de code ACSI et pour les étudiants (dont on projettera le numéro), donnez l'écart entre d'une part la note de test obtenue par les étudiants et d'autre part la meilleure et la moins bonne note de test en ACSI.

Création de la vue :

```
CREATE VIEW Q15(MAX_MOY, MIN_MOY) AS
SELECT MAX(MOY_TEST), MIN(MOY_TEST)
      FROM NOTATION
      WHERE CODE ='ACSI';
```

Consultation de la vue :

```
SELECT NUM_ET, MOY_TEST - MAX_MOY, MOY_TEST - MIN_MOY
FROM NOTATION, Q15
WHERE CODE =’ACSI’ ;
```

Vues de Mise à jour

Q16 Il s'agit de prendre en compte l'intégrité de domaine de l'attribut DISCIPLINE dans la relation MODULE. En effet, ces valeurs admissibles ne peuvent appartenir qu'à l'ensemble suivant : {Informatique, Gestion, Maths}. Créez une vue DIS, permettant d'assurer que toute insertion à travers cette vue vérifie la contrainte de domaine énoncée. Vérifiez l'opération réalisée en tentant une insertion invalide.

Création de la vue :

```
CREATE VIEW DIS AS
SELECT * FROM MODULE
WHERE DISCIPLINE IN ('INFORMATIQUE', 'GESTION', 'MATHS')
WITH CHECK OPTION ;
```

Tentative d'insertion dans la vue d'un tuple erroné :

```
INSERT INTO DIS(CODE, DISCIPLINE) VALUES (99, 'SPORT') ;
```

Q17 Dans la base actuellement définie, la contrainte suivante n'est pas vérifiée : "Le responsable d'une matière doit forcément enseigner cette matière". Comment prendre en compte cette contrainte d'intégrité dynamique par une vue ?

Création de la vue :

```
CREATE VIEW MAT AS
SELECT * FROM MODULE M
WHERE RESP IN (
            SELECT NUM_PROF
            FROM ENSEIGNT
            WHERE M. CODE = ENSEIGNT.CODE)
 WITH CHECK OPTION ;
```

Vérification par insertion d'un tuple erroné :

```
INSERT INTO MAT (CODE, RESP) VALUES ('XXX', 99) ;
```

Remarque : l'obligation dans une vue d'exprimer les jointures sous forme imbriquée impose l'utilisation d'un alias pour MODULE dans la définition de la vue MAT : le responsable de la matière traitée au niveau du premier bloc doit enseigner cette matière là.

Autre possibilité pour la création de la vue :

```
CREATE VIEW MAT AS
SELECT * FROM MODULE M
WHERE (CODE, RESP) IN (
            SELECT CODE, NUM_PROF
            FROM ENSEIGNT)
 WITH CHECK OPTION ;
```

Q18 Définissez la vue nécessaire pour prendre en compte toutes les contraintes d'intégrité de référence mises en jeu lors d'une insertion dans la relation ENSEIGNT ?

Création de la vue :

```
CREATE VIEW ENS AS
SELECT *
FROM ENSEIGNT
WHERE  CODE IN       (SELECT CODE FROM MODULE) AND
       NUM_ET IN     (SELECT NUM_ET FROM ETUDIANT) AND
       NUM_PROF IN   (SELECT NUM_PROF FROM PROF)
 WITH CHECK OPTION ;
```

Vérification par insertion d'un tuple erroné :

```
INSERT INTO ENS (CODE, NUM_ET, NUM_PROF) VALUES ('SPO', 2112, 12) ;
```

Q19 Définissez la vue nécessaire pour prendre en compte toutes les contraintes d'intégrité de référence mises en jeu lors d'une suppression de matière dans la relation MODULE ?

La relation MODULE est, via sa clef primaire CODE, associée aux relations :
- PROF par la clef étrangère MAT_SPEC ; - ENSEIGNT par la clef étrangère CODE ;
- NOTATION par la clef étrangère CODE ; - MODULE par la clef étrangère CODEPERE

La prise en compte de l'intégrité de référence en suppression signifie qu'une matière ne peut être supprimée de la relation que s'il n'existe :
- aucun professeur dont cette matière est la spécialité ;
- aucun enseignement réalisé dans cette matière ;
- aucune note attribuée dans cette matière ;
- aucune matière dépendant de cette matière.

Création de la vue :

```
CREATE VIEW MAT AS
SELECT * FROM MODULE
WHERE CODE NOT IN
              (SELECT DISTINCT CODE FROM ENSEIGNT)
      AND CODE NOT IN
              (SELECT DISTINCT MAT_SPEC FROM PROF)
      AND CODE NOT IN
              (SELECT DISTINCT CODE FROM NOTATION)
      AND CODE NOT IN
              (SELECT DISTINCT CODEPERE FROM MODULE)
WITH CHECK OPTION ;
```

Vérification par tentative de suppression d'une matière enseignée :

```
DELETE FROM MAT WHERE CODE = 'ACSI' ;
```

Troisième étape : Gestion des privilèges d'accès aux données

Q20 Donnez l'autorisation à tous de consulter la relation ETUDIANT.

```
GRANT SELECT ON ETUDIANT TO PUBLIC ;
```

Q21 Effectuez la consultation de tous les étudiants résidant à Marseille, dans l'espace de travail d'un autre utilisateur de login GALAXE. Vérifiez que vous n'avez que le droit de consultation, en tentant une opération de mise à jour sur cette relation.

```
SELECT NOM_ET, PRENOM_ET FROM GALAXE.ETUDIANT
WHERE VILLE_ET = 'MARSEILLE' ;
```

Vérification de l'accord des privilèges :

```
 SELECT * FROM GALAXE.ETUDIANT
/
UPDATE GALAXE.ETUDIANT SET VILLE_ET = 'PARIS' WHERE NUM_ET = 2401
/
```

L'ordre de modification est rejeté, faute d'avoir le droit nécessaire.

Q22 Accordez le droit de consultation et de mise à jour des données de la relation PROF à un autre utilisateur, qui vous accordera les mêmes privilèges.
Consultez puis effectuez une modification sur la relation PROF de l'autre utilisateur.

```
GRANT SELECT, INSERT, UPDATE, DELETE ON PROF TO GALAXE
WITH GRANT OPTION ;
```

Vérification de l'accord des privilèges :

```
SELECT * FROM GALAXE.PROF
/
INSERT INTO GALAXE.PROF(NUM_PROF, NOM_PROF, PRENOM_PROF)
VALUES (21,'EINSTEIN', 'ALBERT')
/
```

Remarque : la visualisation du tuple inséré par l'utilisateur GALAXE n'est possible qu'après Validation de votre transaction par **COMMIT**.

Quatrième étape : Requêtes complexes

Q23 Quels sont les étudiants (numéro, nom, prénom) ayant été noté dans une matière rattachée au module intitulé Principes des BD ?

```
SELECT DISTINCT ETUDIANT.NUM_ET, NOM_ET, PRENOM_ET
FROM ETUDIANT, NOTATION
WHERE ETUDIANT.NUM_ET = NOTATION.NUM_ET
AND CODE IN (
            SELECT CODE
            FROM MODULE
            CONNECT BY CODEPERE = PRIOR CODE
            START WITH    LIBELLE = 'PRINCIPES DES BD') ;
```

Q24 Pour chaque module, donnez son libellé et le libellé de son module père direct dans la hiérarchie.

```
SELECT M.LIBELLE, MPERE.LIBELLE
FROM MODULE M, MODULE MPERE
WHERE M.CODEPERE = MPERE.CODE ;
```

Q25 Quels sont les groupes de deuxième année, dans lesquels un étudiant a obtenu, pour la matière de libellé Conception de SI, une meilleure note de test que le meilleur étudiant du groupe 3 dans la même matière ?

```
SELECT GROUPE
FROM ETUDIANT
WHERE ANNEE = 2 AND GROUPE IN
            (SELECT GROUPE
            FROM ETUDIANT, NOTATION, MODULE
            WHERE ETUDIANT.NUM_ET = NOTATION.NUM_ET AND
                      MODULE.CODE = NOTATION.CODE AND
                      LIBELLE = 'CONCEPTION DE SI'
            GROUP BY  GROUPE
            HAVING MAX(MOY_TEST) > (
                      SELECT   MAX(MOY_TEST)
                      FROM ETUDIANT, NOTATION, MODULE
                      WHERE ETUDIANT.NUM_ET = NOTATION.NUM_ET   AND
                      MODULE.CODE = NOTATION.CODE AND GROUPE = 3 AND
                      LIBELLE = 'CONCEPTION DE SI' AND ANNEE = 2))
GROUP BY GROUPE ;
```

```
SELECT GROUPE
FROM ETUDIANT
WHERE ANNEE = 2 AND GROUPE IN (
            SELECT GROUPE
            FROM ETUDIANT, NOTATION, MODULE
            WHERE ETUDIANT.NUM_ET = NOTATION.NUM_ET AND
                      MODULE.CODE = NOTATION.CODE AND
                      LIBELLE = 'CONCEPTION DE SI' AND MOY_TEST > (
                          SELECT MAX(MOY_TEST)
                          FROM ETUDIANT, NOTATION, MODULE
                          WHERE ETUDIANT.NUM_ET = NOTATION.NUM_ET   AND
                          MODULE.CODE = NOTATION.CODE AND GROUPE = 3
                          AND LIBELLE = 'CONCEPTION DE SI' AND ANNEE = 2))
GROUP BY GROUPE ;
```