



TP SGBD

BOUSSAOUI Elias

GAUTIER Titouan

Info 3, Groupe 2

Chapitre 2 : PL SQL, Vues, triggers, référence, procédures stockées, parcours hiérarchique, Python

1 - Contraintes d'intégrités

Question 1.1 : Créer la table CompositionIngrédient et implémenter ces contraintes dans la base de données.

```
CREATE TABLE COMPOSITIONINGREDIENT (
    nomIngredient      VARCHAR2(50),
    nomSousIngredient  VARCHAR2(100),
    quantiteUtilise    NUMBER,
    PRIMARY KEY (nomIngredient, nomSousIngredient),
    FOREIGN KEY (nomIngredient) REFERENCES INGREDIENT (nomIng
    CONSTRAINT maxQte CHECK ( quantiteUtilise <= 5 ),
    CONSTRAINT ingredientDifferent check ( nomSousIngredient
```

	NOMINGREDIENT	NOMSOUSINGREDIENT	QUANTITEUTILISE
1	Feuille de Laurier	Herbe	2
2	Racine de Mandragore	Arbre	1
3	Serpentardium	Ecaille	3
4	Baie de Sureau	Fruit	2
5	Serpentardium	Venin	4
6	Œil de Trithemius	Pupille	1
7	Feuille de Laurier	Branche	1
8	Baie de Sureau	Baguette	1

2 - Clé primaire

Question 2.1 : Réaliser l'incrémentation de l'identifiant.

```
-- Création d'une séquence pour l'auto-incrémentation --
-- une séquence permet de prendre des valeurs (ici à partir de 1)
CREATE SEQUENCE seq_noPotion START WITH 1 INCREMENT BY 1;

-- Modification de la table Potion pour ajouter l'identifiant
-- Par défaut noPotion aura une valeur de la sequence --
ALTER TABLE POTION
ADD noPotion INT DEFAULT seq_noPotion.NEXTVAL;
```

3 - Attribut calculé

Question 3.1 : Rajouter à la table de votre choix une colonne nbIngrédients qui compte ("automatiquement") le nombre d'ingrédients composant une potion.

```
alter table POTION add nbIngrédients int;
```

Question 3.2 : Ecrire une procédure stockée permettant de renseigner la colonne nbIngrédients pour les tuples déjà présents dans la base.

```
CREATE OR REPLACE PROCEDURE UpdateNbIngredients AS
BEGIN
    FOR potion_rec IN (SELECT DISTINCT * FROM POTION) LOOP
        update POTION
```

```
        set nbIngrédients = (select count(*) from COMPOSITION
                               where nomPotion = potion_rec.nomPotion;
    END LOOP;
END;
```

Question 3.3 : Ecrire un trigger permettant de mettre à jour automatiquement cette colonne lors des nouvelles mises à jour de la base.

```
CREATE OR REPLACE TRIGGER nbIngrédientPotion
BEFORE INSERT OR UPDATE
ON POTION
FOR EACH ROW
    DECLARE
        v_nbIngrédient NUMBER;
    BEGIN
        SELECT COUNT(*) INTO v_nbIngrédient
        FROM COMPOSITION c
        WHERE c.nomPotion = :NEW.nomPotion;

        :NEW.nbIngrédient := v_nbIngrédient;
    END;
```

4 - Vues modifiables

Question 4.1 : Définir une vue permettant de consulter tous les renseignements sur les potions et leurs ingrédients.

```
Create or replace view VuePotionsIngredients as
select * from COMPOSITION C JOIN POTION P ON C.nomPotion = P.
```

Question 4.2 : Supprimer à partir de cette vue les potions comptant plus de trois ingrédients.

```
DELETE FROM VuePotionsIngredients
WHERE nomPotion IN (
    SELECT nomPotion
    FROM VuePotionsIngredients
    GROUP BY nomPotion
    HAVING COUNT(DISTINCT nomIngredient) > 3
);
```

5 - Requêtes hiérarchique

Question 5.1 : Lister pour un ingrédient donné tous les ingrédients directs et indirects qui le composent.

```
SELECT distinct
    CI.nomIngredient AS ingredient,
    CI.nomSousIngredient AS sousIngredient,
    LEVEL AS hierarchy_level
FROM
    CompositionIngredient CI, INGREDIENT i
START WITH
    CI.nomIngredient = 'Baie de Sureau'
CONNECT BY
    PRIOR CI.nomSousIngredient = CI.nomIngredient
ORDER BY
    hierarchy_level, sousIngredient;
```

	INGREDIENT	SOUSINGREDIENT	HIERARCHY_LEVEL
1	Baie de Sureau	Baguette	1
2	Baie de Sureau	Feuille de Laurier	1
3	Baie de Sureau	Fruit	1
4	Feuille de Laurier	Branche	2
5	Feuille de Laurier	Herbe	2

Question 5.2 : Lister pour un ingrédient donné, les ingrédients dans lesquelles il intervient directement ou indirectement.

```
SELECT
    ci.nomSousIngrédient as sousIngrédient,
    ci.nomIngrédient as ingrédient,
    LEVEL as hierarchy_level
FROM
    COMPOSITIONINGREDIENT ci
START WITH
    ci.nomSousIngrédient = 'Feuille de Laurier'
CONNECT BY
    PRIOR ci.nomIngrédient = ci.nomSousIngrédient
ORDER BY
    hierarchy_level, ingrédient;
```

	<input type="checkbox"/> SOUSINGREDIENT	<input type="checkbox"/> INGRÉDIENT	<input type="checkbox"/> HIERARCHY_LEVEL
1	Feuille de Laurier	Baie de Sureau	1
2	Baie de Sureau	Serpentarium	2

Chapitre 3 : Administration de la BD

1 - Consulter le dictionnaire des données

Question 1.1 : Quels sont les utilisateurs existants dans la base ?

```
SELECT username FROM all_users;
```

Question 1.2 : Quels sont les privilèges systèmes ?

```
SELECT grantee, privilege FROM dba_sys_privs
WHERE grantee IN (SELECT username FROM all_users);
```

Question 1.3 : Quels sont les rôles prédéfinis ?

```
select distinct granted_role from dba_role_privs where default
```

Question 1.4 : Quels sont les utilisateurs qui ont ces rôles et qui les leur a octroyé ?

```
select dba_role.grantee, dba_role.granted_role, dba_tabs.gran
from dba_role_privs dba_role join dba_tab_privs dba_tabs
on dba_role.grantee = dba_tabs.grantee where dba_role.def
```

Question 1.5 : Quels sont les quotas de chaque utilisateur ?

```
SELECT username, tablespace_name, bytes, max_bytes, max_block
FROM dba_ts_quotas;
```

2 - Gestion des utilisateurs

Question 2.1 : créer un utilisateur E0XXXXPANORAMIX

```
create user E219118XPANORAMIX identified by E219118XPANORAMIX
```

Question 2.2 : Se connecter sous le nom de E0XXXXPANORAMIX

```
grant create session to E219118XPANORAMIX;
grant restricted session to E219118XPANORAMIX;
```

Question 2.3 : En tant qu'administrateur, donner le droit de créer des tables à PANORAMIX.

```
grant create table to E219118XPANORAMIX;

/* Changement de user à E219118XPANORAMIX */
create table test(id int primary key );
```

Question 2.4 : Se connecter en tant que PANORAMIX et tenter de créer une table. En tant qu'administrateur, affecter à PANORAMIX votre espace de travail comme espace de travail par défaut en limitant à 20K la taille de l'espace utilisable, vérifier les résultats.

```
alter user E219118XPANORAMIX default tablespace USERS quota 2
```

Question 2.5 : Se connecter en tant que PANORAMIX, créer une table Test et donner les droits d'interrogation sur cette table à tout utilisateur. Si la création échoue, expliquer pourquoi et corriger puis vérifier.

```
/* Changement de user à E219118XPANORAMIX */
create table test(
    id number,
    nom varchar(15)
);

grant select on test to public;

-- verification--
select table_name from all_tables where owner='PANORAMIX';
```

Question 2.6 : Permettre à PANORAMIX de ne consulter que les potions dont il est l'émetteur ou le récepteur.

```
create or replace view PotionPanoramix as
    select * from POTION po, PRISE_POTION pp
        where po.nomPotion = pp.nomPotion
            and (po.druidCreateur = 'Panoramix'
                or pp.nomPatient='Panoramix');

grant select on PotionPanoramix to E219118XPANORAMIX;
```

3 - Notion de rôle

Question 3.1 : Créer un rôle E219118XRoleIndex ayant comme privilèges create index, alter any index, drop any index. Allouer ce rôle à E219118XPANORAMIX.

```
create role E219118XRoleIndex;
grant create any index to E219118XRoleIndex;
grant alter any index to E219118XRoleIndex;
```

```
grant drop any index to E219118XRoleIndex;  
grant E219118XRoleIndex to E219118XPANORAMIX;
```

Question 3.2 : Créer un rôle Role1 ayant Grant any privilege comme privilèges.

```
Create role role1;  
grant grant any privilege to role1;  
grant role1 to PANORAMIX;
```

Question 3.3 : En tant qu'utilisateur PANORAMIX, vérifiez ses privilèges (Connect to)

```
-- Alternative pour verifier, car j'arrive pas à me connecter  
select grantee, granted_role from dba_role_privs where grantee  
  
-- Possibilité de se connecter à l'utilisateur Panoramix  
-- et tester de créer une table par exemple --
```

Question 3.4 : Faire le ménage : supprimer tous les objets créés (utilisateurs, tables, rôles etc). Voir entre autres, drop user ... cascade.

```
revoke all privileges from E219118XPANORAMIX;  
drop user E219118XPANORAMIX cascade;  
drop role E219118XRoleIndex;  
drop role role1;
```

Chapitre 4 - Stockage, Index

1 - Stocker les données dans la base

Question 4.1 : Lister les identifiants physiques des patients, leur no d'ordre (rownum) et leur identifiant logique.

```
SELECT ROWNUM as numero_physique,  
       ROWID as numero_logique,
```



```
nom_Patient
FROM PATIENT;
```

	NUMERO_PHYSIQUE	NUMERO_LOGIQUE	NOM_PATIENT
1	1	AAAXvxAAPAAAFaTAAC	Garcia
2	2	AAAXvxAAPAAAFaTAAE	Kim
3	3	AAAXvxAAPAAAFaTAAB	Martin
4	4	AAAXvxAAPAAAFaTAAD	Mueller
5	5	AAAXvxAAPAAAFaTAAA	Smith

Question 4.2 : Dans quel tablespace de la base se trouve la table patients, dans quel schéma et dans quel fichier physique est-elle stockée ?

```
-- Dans quel table space et quel schema est elle stocké
SELECT table_name, tablespace_name, owner
FROM all_tables
WHERE table_name = 'PATIENT';

-- Dans quel fichier physique
SELECT tablespace_name, file_name
FROM dba_data_files
WHERE tablespace_name = 'USERS';
```

La table PATIENT se trouve dans le tablespace USERS, dans le schéma E217657J.

	TABLE_NAME	TABLESPACE_NAME	OWNER
1	PATIENT	USERS	E217657J

La table PATIENT se trouve dans le fichier physique :

	TABLESPACE_NAME	FILE_NAME
1	USERS	/opt/oracle/oradata/ORCLCDB/ORCLPDB1/users01.dbf
2	USERS	/opt/oracle/oradata/ORCLCDB/ORCLPDB1/users02.dbf
3	USERS	/opt/oracle/oradata/ORCLCDB/ORCLPDB1/users03.dbf
4	USERS	/opt/oracle/oradata/ORCLCDB/ORCLPDB1/users04.dbf

Question 4.3 :

```
SELECT segment_name, sum(bytes) as nb_octet, SUM(blocks) as n
FROM dba_segments
WHERE segment_name = 'PATIENT'
      or segment_name = 'COMPOSITION'
      or segment_name = 'COMPOSITIONINGREDIENT'
      or segment_name = 'POTION'
      or segment_name = 'INGREDIENT'
group by segment_name;
```

	SEGMENT_NAME	NB_OCTET	NB_BLOCK
1	PATIENT	1966080	240
2	POTION	2031616	248
3	INGREDIENT	2097152	256
4	COMPOSITION	1966080	240
5	COMPOSITIONINGREDIENT	1245184	152

Question 4.4 :

```
SELECT TABLESPACE_NAME AS "TABLESPACE",
       SEGMENT_TYPE AS "TYPE OBJET",
       Sum(BYTES) / 1024 / 1024 AS "TAILLE (Mb)"
FROM DBA_EXTENTS
WHERE OWNER = USER
GROUP BY OWNER, TABLESPACE_NAME, SEGMENT_TYPE
ORDER BY OWNER, TABLESPACE_NAME
```

	TABLESPACE	"TYPE OBJET"	"TAILLE (Mb)"
1	USERS	INDEX	0.375
2	USERS	TABLE	0.375

La requête fournit une vue agrégée de l'utilisation de l'espace disque par les objets (tables, indexes, etc.) pour l'utilisateur courant, en présentant la taille totale en mégaoctets pour chaque type d'objet dans chaque tablespace.