

Exercice 1: (08 pts)

1. Dans quelle clause SQL est utilisée le mot clé LIKE ?

WHERE

2. Est-il possible d'interroger plusieurs tables à la fois, avec une seule requête SQL ?

Oui

3. Donner trois exemples de privilèges système ?

CREATE USER, CREATE TABLE, DROP USER

4. Donner trois exemples de privilèges objet ?

ALTER, DELETE, INSERT

5. Comment appelle-t-on un regroupement de privilèges ?

ROLE

6. Donner une brève description des commandes SQL suivantes :

ALTER : Modification d'un objet de base de données existant, comme une table

DROP: Suppression de toute une table, une vue ou un autre objet dans la base de données

REVOKE: Enlever les privilèges accordés aux utilisateurs

7. Que va-t-il se passer si on n'ajoute pas la clause WHERE dans une requête de suppression ?

Toutes les lignes de la table seront supprimées

8. Cocher les mots-clés compatibles avec le langage SQL

☒ FROM ☐ WHEN ☒ IN ☒ WHILE ☒ IF THEN .. ELSE ☐ CONTINUE ☒ WHERE

9. Parmi ces différentes fonctionnalités, une seule n'est pas du domaine des SGBD

☐ l'ajout de données ☐ **la prévision de données** ☐ la recherche des données ☐ la mise à jour des données

10. Un administrateur souhaite afficher que les villes dont le résultat du count(*) est supérieur à 10.000 , alors il doit écrire ?

☐ SELECT ville, count(*) FROM table_ville GROUP BY ville
WHERE count(id_ville)>10000

☒ SELECT ville, count(*) FROM table_ville GROUP BY ville
HAVING count(*)>10.000

☐ SELECT ville, count(*) FROM table_ville GROUP BY ville
WHERE count(*)>10.000

☐ SELECT ville, count(*) FROM table_ville GROUP BY ville
count(*)>10.000

11. Le mot clé généralement attendu après la clause GROUP BY (6 caractères majuscules) permettant une restriction sur les données à grouper est le mot **HAVING**

12. Quel est l'intérêt de l'utilisation des transactions ?

Assurer le passage d'un état cohérent et correct vers un autre état cohérent et correct de la BDD

13. En SQL, quelles commandes ne peuvent pas être annulées dans une transaction ?

☐ INSERT ☐ UPDATE ☒ TRUNCATE ☒ DROP ☒ ALTER ☐ SELECT

14. Peut-on renommer une base de données ?

NON

Exercice 2: (12 pts)

Soit le schéma relationnel suivant:

Enseignant (numEns, nomEns, prenomEns, Ville)

Article (numArt, titre, genre, dateSortie, numEns)

Avec: numEns, numArt sont de type **entier**. nomEns, prenomEns, ville, titre et genre de type **chaîne de caractère** et dateSortie de type **Date**.

1. Ecrire une requête SQL pour la création de la base de données EnsDB

CREATE DATABASE EnsDB

2. Ecrire deux requêtes SQL pour la création des deux tables Enseignant et Article

Enseignant	Article
CREATE TABLE Enseignant (numEns INT NOT NULL, , nomEns VARCHAR(30), prenomEns VARCHAR(30), ville VARCHAR(30));	CREATE TABLE Article (numArt INT, titre VARCHAR(30), genre VARCHAR(30), dateSortie DATE, NumEns INT NOT NULL, CONSTRAINT pk_Article PRIMARY KEY (numArt));

3. Ajouter la contrainte d'intégrité de clé primaire numEns sur la table Enseignant

ALTER TABLE Artiste ADD CONSTRAINT pk_Enseignant PRIMARY KEY(numEns);

4. Ajouter la contrainte d'intégrité de clé étrangère numEns sur la table Article qui fait référence au champ numEns de la table Enseignant

ALTER TABLE Article ADD CONSTRAINT fk FOREIGN KEY(numEns) REFERENCES Enseignant(numEns);

5. Ajouter une colonne Type (de type chaîne de caractères), dans la table Article avec 'Notype' comme valeur par défaut.

ALTER TABLE Article ADD Type VARCHAR(20) DEFAULT 'NoType';

6. Sélectionner le nombre d'Article par Enseignant

SELECT COUNT(numArt) AS NbrArticle FROM Article GROUP BY (numEns);

7. Créer un utilisateur **EnsUser** et un utilisateur **ArtUser**

CREATE USER EnsUser;

CREATE USER ArtUser;

8. Créer une vue **ArtView** permettant de consulter les champs: nom, ville, titre et dateSortie

CREATE VIEW ArtView(nom, ville, titre, dateSortie) AS SELECT nomEns, ville, titre, dateSortie FROM Enseignant, Article Where Enseignant.numEns = Article.numArt;

9. Attribuer le privilège **SELECT** sur la vue **ArtView** à l'utilisateur **EnsUser**

GRANT SELECT ON ArtView TO EnsUser;

10. Attribuer les privilèges, insertion et mise à jour, sur les deux tables **Enseignant** et **Article** pour l'utilisateur **ArtUser**

GRANT INSERT,UPDATE ON Enseignant TO ArtUser;

GRANT INSERT,UPDATE ON Article TO ArtUser;

11. Créer un rôle **AdminEns**

CREATE ROLE AdminEns

12. Attribuer tous les privilèges sur la table **Enseignant** pour le rôle **AdminEns**

GRANT ALL PRIVILEGES ON Enseignant TO AdminEns

13. Supposant qu'il existe initialement 10 tuples dans la table Enseignant (dernier tuple ayant numEns=10). Soit la transaction suivante avec autocommit = 0:

**INSERT INTO Enseignant (numEns, nomEns) VALUES (11, 'NomEns11');
INSERT INTO Enseignant (numEns, nomEns) VALUES (12, 'NomEns12');
COMMIT;**

INSERT INTO Enseignant (numEns, nomEns) VALUES (13, 'NomEns13');

UPDATE Enseignant SET ville = 'Mascara' WHERE numEns = 1;

DELETE FROM Enseignant WHERE numEns = 11;

SELECT count(numEns) FROM Enseignant;

SAVEPOINT P1;

UPDATE Enseignant SET ville = 'Oran' WHERE numEns = 3;

INSERT INTO Enseignant (numEns, nomEns) VALUES (14, 'NomEns14');

SELECT count(numEns) FROM Enseignant;

ROLLBACK TO SAVEPOINT P1;

SELECT count(numEns) FROM Enseignant;

- Donner le nombre de tuples dans la table Enseignant à chaque sélection?

sélection1: **12** Sélection 2: **13** Sélection 3: **12**

- Donner le nombre de mises à jour validées si autocommit = 1

2 mises à jour