

Examen PL/SQL L3

Voici les réponses avec les questions intégrées :

Exercice 1 : Questions à Choix Multiples (QCM)

- 1. Une base de données est définie comme :
Réponse : B - Un ensemble structuré et organisé de données interreliées.
- 2. Le rôle principal d'un SGBD est de :
Réponse : B - Gérer les bases de données et minimiser la redondance.
- 3. L'architecture ANSI/SPARC comporte combien de niveaux ?
Réponse : B - 3 niveaux.
- 4. Le niveau externe dans l'architecture ANSI/SPARC correspond à :
Réponse : B - Les vues des utilisateurs.
- 5. Une entité faible est caractérisée par :
Réponse : B - L'impossibilité d'être identifiée sans une entité forte.
- 6. Quelle est la contrainte imposée par l'intégrité de clé ?
Réponse : A - Les clés primaires doivent être uniques et non nulles.
- 7. Le processus de normalisation a pour objectif :
Réponse : C - Éviter les anomalies et réduire les redondances.
- 8. Une relation est en 1ère forme normale (1FN) si :
Réponse : B - Tous ses attributs sont atomiques.
- 9. Une relation en 2ème forme normale (2FN) :
Réponse : B - Est en 1FN et tous ses attributs dépendent de la clé entière.
- 10. Une relation est en 3ème forme normale (3FN) si :
Réponse : A - Elle est en 2FN et élimine les dépendances transitives.
- 11. Une clé étrangère dans une relation :
Réponse : A - Fait référence à une clé primaire dans une autre relation.
- 12. Le modèle relationnel a été proposé par :
Réponse : C - E. F. Codd.

1/5

```
(4, 'Lemoine', 'Claire', 'claire.lemoine@example.com', '0745123698', 'Bordeaux'),
(5, 'Moreau', 'Luc', 'luc.moreau@example.com', '0712345698', 'Nice');
```

3. Affichez toutes les commandes avec le nom et prénom du client, la date de commande et le statut.

```
sql

SELECT Commande.id_commande, Client.nom, Client.prenom, Commande.date_commande,
Commande.statut
FROM Commande
JOIN Client ON Commande.id_client = Client.id_client;
```

4. Affichez l'ID de chaque commande et le montant total de la commande.

```
sql

SELECT Commande.id_commande, SUM(Détail_Commande.quantite * Produit.prix) AS
montant_total
FROM Détail_Commande
JOIN Produit ON Détail_Commande.id_produit = Produit.id_produit
JOIN Commande ON Détail_Commande.id_commande = Commande.id_commande
GROUP BY Commande.id_commande;
```

5. Affichez tous les produits d'une commande donnée.

```
sql

SELECT Produit.nom, Détail_Commande.quantite, Produit.prix,
(Détail_Commande.quantite * Produit.prix) AS total
FROM Détail_Commande
```

4/5

- 13. Une projection dans l'algèbre relationnelle :
Réponse : B - Supprime les attributs non mentionnés dans le résultat.
- 14. La jointure naturelle est :
Réponse : B - Une jointure basée sur l'égalité des attributs communs.
- 15. Une transaction dans un SGBD doit être :
Réponse : B - Atomique, cohérente, isolée, et durable (ACID).
- 16. La 3ème forme normale de Boyce-Codd (BCNF) :
Réponse : A - Élimine les dépendances fonctionnelles non triviales.
- 17. L'intégrité référentielle assure que :
Réponse : B - Les clés étrangères pointent vers des valeurs existantes.
- 18. Une union dans l'algèbre relationnelle nécessite :
Réponse : B - Des relations avec des schémas identiques.
- 19. La décomposition des relations dans le processus de normalisation vise à :
Réponse : C - Minimiser les anomalies d'insertion et de suppression.
- 20. L'algèbre relationnelle est :
Réponse : B - Une collection d'opérations pour manipuler les relations.

Exercice 2 : Gestion des commandes d'un magasin

1. Écrivez les requêtes SQL pour créer les tables avec les contraintes appropriées.

```
sql

CREATE TABLE Produit (
  id_produit INT PRIMARY KEY,
  nom VARCHAR(100),
  prix DECIMAL(10, 2),
  categorie VARCHAR(50),
  stock INT
);

CREATE TABLE Client (
  id_client INT PRIMARY KEY,
  nom VARCHAR(100),
  prenom VARCHAR(100),
```

2/5

```
JOIN Produit ON Détail_Commande.id_produit = Produit.id_produit
WHERE Détail_Commande.id_commande = 1;
```

6. Trouvez les clients qui n'ont jamais passé de commande.

```
sql

SELECT * FROM Client
WHERE id_client NOT IN (SELECT id_client FROM Commande);
```

7. Affichez les commandes qui ont été annulées, avec les informations des clients.

```
sql

SELECT Commande.id_commande, Client.nom, Client.prenom, Commande.date_commande
FROM Commande
JOIN Client ON Commande.id_client = Client.id_client
WHERE Commande.statut = 'Annulée';
```

Tout est complet ! Si tu as besoin d'explications ou de modifications, dis-moi. 😊

Is this conversation helpful so far?

5/5

```
email VARCHAR(100),
telephone VARCHAR(20),
ville VARCHAR(50)
);

CREATE TABLE Commande (
  id_commande INT PRIMARY KEY,
  id_client INT,
  date_commande DATE,
  statut VARCHAR(20),
  FOREIGN KEY (id_client) REFERENCES Client(id_client)
);

CREATE TABLE Détail_Commande (
  id_détail INT PRIMARY KEY,
  id_commande INT,
  id_produit INT,
  quantite INT,
  FOREIGN KEY (id_commande) REFERENCES Commande(id_commande),
  FOREIGN KEY (id_produit) REFERENCES Produit(id_produit)
);
```

2. Ajoutez des données d'exemple pour chaque table.

```
sql

INSERT INTO Produit (id_produit, nom, prix, categorie, stock) VALUES
(1, 'Ordinateur', 1500.00, 'Informatique', 10),
(2, 'Smartphone', 800.00, 'Téléphonie', 15),
(3, 'Imprimante', 200.00, 'Bureau', 8),
(4, 'Tablette', 500.00, 'Informatique', 12),
(5, 'Casque Audio', 150.00, 'Accessoires', 20);

INSERT INTO Client (id_client, nom, prenom, email, telephone, ville) VALUES
(1, 'Dupont', 'Jean', 'jean.dupont@example.com', '0123456789', 'Paris'),
(2, 'Martin', 'Sophie', 'sophie.martin@example.com', '0987654321', 'Lyon'),
(3, 'Durand', 'Paul', 'paul.durand@example.com', '0654321987', 'Marseille');
```

3/5