



MATRICULE NOM PRENOM Groupe

-----Partie 1 (10 : 1 x 10)-----

Indications : -1- Répondez en cochant les cases correctes.

-2- Certaines questions sont à réponses multiples.

-3- Une réponse correcte : +1, incorrecte : -1, aucune réponse : 0

-4- Attention aux ratures.

1. Que signifie le C de ACID ?

- ☐ Circonstance.
- ☐ Concurrence.
- ☐ Consistance.
- ☐ Confidence.

2. Une vue matérialisée est

- ☐ Une vue stockée sur un disque matériel.
- ☐ Une table temporaire.
- ☐ Une vue indexée.
- ☐ Une vue avec index cluster.

3. La durée de vie d'une table temporaire est

- ☐ 1 Semaine.
- ☐ La durée de vie de la base.
- ☐ La durée de vie de la session en cours.
- ☐ Jusqu'à sa suppression.

4. Les index sont utilisés pour

- ☐ Accélérer les mises à jour.
- ☐ Accélérer les recherches.
- ☐ Matérialiser les vues.
- ☐ Des besoins de contraintes d'intégrité.

5. Dans la requête suivante : SELECT * FROM client NATURAL JOIN commande ORDER BY nom desc. La clause qui consomme le plus de temps est :

- ☐ Le Select, parce qu'on a mis * et il doit chercher les champs dans la métabase avant d'exécuter la requête.
- ☐ La jointure naturelle, parce que le SGBD doit trouver les colonnes communes et effectuer des restrictions sur le produit cartésien.
- ☐ La clause order by, les algorithmes de tri sont connu pour leur lenteur.
- ☐ L'option desc du order by, car elle consomme du temps pour inverser l'ordre des lignes.

6. Dans quel cas le renommage est-il obligatoire

- ☐ Quand on mentionne une sous- requête au niveau du FROM.
- ☐ Quand il y'a plus d'une table au niveau du FROM.
- ☐ Quand on utilise les agrégats au niveau du SELECT.
- ☐ Dans les vues.

7. L'utilisation des sous-requêtes est à éviter parce que

- ☐ La requête est trop longue à rédiger.
- ☐ La requête est moins esthétique avec les sous-requêtes.
- ☐ Le SGBD trouve des difficultés à optimiser leur exécution
- ☐ On force le SGBD à les exécuter en premier ce qui n'optimise pas l'exécution.

8. Le nombre de mécanismes d'implémentations des contraintes d'intégrité en SQL est:___

9. Les curseurs sont utilisés dans SQL pour

- ☐ Afficher les lignes d'une table.
- ☐ Remplacer l'utilisation des boucles.
- ☐ Parcourir le contenu d'un flux de données.
- ☐ Parcourir la liste des champs d'une table.

10. Vrai ou faux (Mettez une croix sur celles qui sont Vraies uniquement)

- ☐ La mise à jour des données via une vue est impossible.
- ☐ Un SGBD qui répond au test ACID est un SGBD qui implémente les fonctions table.
- ☐ Les triggers sont considérés comme étant des procédures stockées spécifiques.
- ☐ Les procédures stockées acceptent uniquement des paramètres en entrée.

-----**Partie 2 (10 : 2+3+3+2)**-----

Soit le script suivant de la base de données scolarité.

```

1.  CREATE TABLE etudiant(idStd varchar(7) PRIMARY KEY, nomStd varchar(40), pnomStd
2.  varchar(40) CHECK (pnomStd=NULL), datenStd date DEFAULT getDate(), avgBacStd
3.  decimal(10,2))
4.  GO
5.  CREATE TABLE inscription(idAns varchar(9), idNiv varchar(4) idStd varchar(7),
6.  PRIMARY KEY(idAns,idNiv,idStd))
7.  GO
8.  CREATE TABLE matiere(idMat var char(4)PRIMARY KEY, libMat varchar(35))
9.  GO
10. CREATE TABLE matniv(idMat varchar(4) PRIMARY KEY, idNiv varchar(4) PRIMARY KEY,
11. CoefMat tinyint, FOREIGN KEY (idNiv) REFERENCES matiere(idNiv) ON DELETE
12. CASCADE ON UPDATE CASCADE)
13. GO
14. CREATE TABLE notes(idAns varchar(9),idStd varchar(7),idMat varchar(5), note
15. decimal(10, 2) DEFAULT 'ABSENT', PRIMARY KEY(idAns,idStd,idMat))
16. GO
17. CREATE FUNCTION TreatAns(@A varchar(9),@N int)
18. Returns varchar(9)
19. As
20. Begin
21.     Return Cast(Cast(Left(@A,4) as int)+@N As varchar)+
22.     Left(Right(@A,5),1)+Cast(Cast(Right(@A,4) as int)+@N As varchar);
23. End
24. GO
25. CREATE FUNCTION Deca (@A varchar(9),@N varchar(4),@T real)
26. RETURNS TABLE
27. As
28. RETURN (
29.     SELECT e.idStd, e.sexeStd, e.avgBacStd
30.     FROM etudiant e JOIN inscription i ON e.idStd=i.idStd
31.     WHERE i.idAns=@A And i.idNiv=@N And e.idStd IN( SELECT idStd FROM inscription
32.     si WHERE si.idAns=dbo.TreatAns(@A,-1) And si.idNiv=i.idNiv) And
33.     EXISTS ( SELECT DISTINCT idStd FROM notes sn WHERE sn.idAns=
34.     dbo.TreatAns(@A,-1) And sn.idStd=e.idStd) And (SELECT COUNT(idMat)
35.     FROM notes sn2 WHERE sn2.note<10 And sn2.idStd = e.idStd And
36.     sn2.idAns=dbo.TreatAns(@A,-1)) = ( SELECT Ceiling(@T * COUNT(idMat)/100)
37.     FROM matniv WHERE idNiv=@N)
38. )
39. GO
40. CREATE FUNCTION Stat(@A1 varchar(9),@A2 varchar(9),@N varchar(4),@T real)
41. RETURNS @R Table (Ans varchar(9), Sex varchar(20), NB int)
42. As
43. BEGIN
44.     DECLARE @A varchar(9)
45.     SET @A = @A1
46.     SET @A2 = dbo.TreatAns(@A2,1)
47.     WHILE @A<>@A2
48.     BEGIN
49.         INSERT INTO @R
50.         SELECT @A A, SexeStd, Count(F.idStd) NB
51.         FROM dbo.Deca(@A,@N,@T) F
52.         GROUP BY sexeStd
53.         SET @A = dbo.TreatAns(@A,1)
54.     END
55.     RETURN
56. END GO

```


This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There is no handwriting or other markings on the paper.

4. Dans la déclaration de l'une des trois fonctions précédentes, il existe un bloc inutile. Trouvez-le et expliquez pourquoi il est inutile.

[illegible]

Bonne chance.