

BDR
23 Juin 2020

Durée : 1h20

Conception BDD répartie

La base de données d'une entreprise d'informatique, NetInfo, a le schéma global suivant:

DEPARTMENTS (#service, nom, chef, site)

PROJECTS (#projet, service, nom, chef, budget, durée, début)

EMPLOYEES (#emp, service, nom, prénom, fonction, salaire, prime, datenaiss, adresse)

NetInfo est localisée sur trois sites de la région : CasaCentre, CasaSud, et CasaNord. Le site de CasaCentre tient également lieu de siège pour l'entreprise.

On souhaite proposer une répartition de la base NetInfo en se basant sur les informations suivantes :

- L'attribut site prend une des valeurs suivantes: "centre", "sud", et "nord".
- L'attribut nom de DEPARTMENTS prend une des valeurs : "commercial", "financier", "technique", "maintenance".
- Un projet appartient à un département.
- Les employés sont affectés à un site donné, sauf pour les employés du département de maintenance qui interviennent dans tous les sites.
- Les données concernant la rémunération ainsi que les informations personnelles des employés sont centralisées au siège de l'entreprise.

1. Quel type de fragmentation faut-il opérer sur la relation DEPARTMENTS
 - Verticale
 - Horizontale
 - Hybride
 - Horizontale dérivée
2. Quel type de fragmentation faut-il opérer sur la relation PROJECTS
 - Verticale
 - Horizontale
 - Hybride
 - Horizontale dérivée
3. Quel type de fragmentation faut-il opérer sur la relation EMPLOYEES
 - Verticale
 - Horizontale
 - Hybride
 - Horizontale dérivée
4. Combien de fragments contient la base répartie NetInfo

BDR

- 11
 - 13
 - 15
 - 17
5. Parmi les fragments précédents, il y a combien de répliques (ou dupliques) au total ?
- 3
 - 4
 - 5
 - 6

BDD réparties Oracle

6. Sur une bdd base1, quel ordre permet de créer un lien de base de données entre base1 et base2
- CREATE PUBLIC DATABASE LINK dblink
CONNECT TO myuser IDENTIFIED BY "mdp"
USING base1
 - CREATE PUBLIC DATABASE LINK dblink
CONNECT TO myuser IDENTIFIED BY "mdp"
USING base2
 - CREATE PUBLIC DATABASE LINK dblink
CONNECT TO base2 IDENTIFIED BY "mdp"
USING base1
 - CREATE PUBLIC DATABASE LINK dblink
CONNECT TO base1 IDENTIFIED BY "mdp"
USING base2
7. Comment assurer la transparence vis-à-vis de la localisation des données (plusieurs réponses):
- Les vues
 - Les synonymes
 - Les fragments
 - La clause driving_site
 - les database link
8. Pour bénéficier du rafraîchissement automatique des répliques, on peut utiliser :
- Les vues
 - Les vues matérialisées
 - L'ordre : SET AUTOTRACE TRACE EXPLAIN STAT
 - Les synonymes
9. L'opérateur qui permet de faire des fragmentations horizontales dérivées est :
- La semi-jointure
 - La projection
 - La sélection
 - L'auto-jointure

BDR

- L'union

MongoDB

1. MongoDB permet de gérer le partitionnement :
 - A) Horizontal
 - B) Vertical
 - C) Hybrid
2. L'utilisateur doit créer une collection de données avant de pouvoir y insérer des documents.
 - A) Vrai
 - B) Faux
3. Le concept correspondant à une table SQL en MongoDB est :
 - A) une collection
 - B) un document
 - C) un attribut clé:valeur
4. Pour se connecter au serveur MongoDB on peut utiliser le client ligne de commandes :
 - A) mongo
 - B) mongod
 - C) mongodb
5. En MongoDB, les données sont bien structurés et on ne peut pas modifier la structure prédéfinie d'un document. En d'autre termes tous les documents d'une même collection ont la même structure.
 - A) Vrai
 - B) Faux
6. Les données MongoDB sont décrites sous le format:
 - A) JSON
 - B) Javascript
 - C) XML
 - D) Java
7. mongoimport est :
 - A) Une commande du shell mongodb
 - B) Un client Mongodb
 - C) Un serveur Mongodb
8. En MongoDB, avec des programmes javascript on peut réaliser des «jointures» :
 - A) Coté Serveur
 - B) Coté Client
 - C) Coté Client et Serveur
9. L'architecture MongoDB est composée de
 - A) Serveurs routeurs et serveurs de shards
 - B) Serveurs routeurs, serveurs de shards, et serveur de configurations
 - C) Serveurs routeurs, serveurs de shards, serveur de configurations, et serveurs de données

BDR

D) Serveurs routeurs et serveurs de données

Requêtes Mongodb

10. `db.movies.find({"year": { $gte: "2000", $lte: "2005" } }, {"title": 1})` pour effectuer des recherches par intervalle sur les années des films entre 2000 et 2005.

- A) Vrai
- B) Faux

11. `db.movies.remove("title":"Alien")` pour supprimer les films de la collection movies dont le titre est Alien

- A) Vrai
- B) Faux

12. `db.movies.find({"summary": {$exists: true}}, {"title": 1})` pour chercher les titres des films qui ont un résumé

- A) Vrai
- B) Faux

13. `db.basetest.movies.find()` pour afficher tous les films de la collection movies

- A) Vrai
- B) Faux

14. Pour la jointure par boucles imbriquées, sous le shell de MongoDB :

```
var lesFilms = db.movies.find() ;
while (lesFilms.hasNext()) {
    var film = lesFilms.next();
    var mes = db.artists.findOne({"_id": film.director._id});
    printjson(film.title);
    printjson(mes);
}
```

- A) Vrai
- B) Faux

15. `db.movies.find({"actors.last_name": "Al Pacino"}, {"actors": 1, "summary": 1})` est une requête correcte

- A) Vrai
- B) Faux

16. `db.artists.drop({})` pour supprimer la collection artists

- A) Vrai
- B) Faux

17. `db.movies.remove({})` pour supprimer tous les documents de la collection movies

- A) Vrai
- B) Faux