
EXAMEN

Élément SMINFL5B - Bases de Données 2

L3 Informatique

6 décembre 2023 - De 10H à 11H30

Documents autorisés

Université de Caen Normandie

- Donner aux réponses le maximum de précision possible.
- Les réponses aux questions sont à donner uniquement sur les emplacements prévus dans ce document. Pour chaque question, le nombre de points (approximatif) est donné dans le cadre situé à droite et au début de la question.
- Le candidat indiquera sur ce document (voir emplacement en bas de cette page) son numéro de place. En outre, il devra en début d'épreuve remplir une copie sur laquelle il indiquera ***aussi*** son numéro de place. Il écrira également son nom dans le coin de la copie qu'il cachera par collage après signature de la feuille d'émargement.
- Cet examen comporte 14 questions.

Bon travail.

Numéro de place :

1 Bases de données classiques

On souhaite modéliser le système d'information d'un ensemble de musées.

- Un musée a un nom et une adresse. Il possède des salles dans lesquelles des œuvres sont exposées.
- Toute œuvre a un titre, un auteur et une date d'acquisition. Elle est exposée dans l'une des salles du musée.
- Une salle a un nom et un numéro.

Question 1.

Pour gérer ce système d'information, proposez un diagramme UML qui modélise la vie d'un musée telle que décrite ci-dessus.

1

Réponse:

Question 2.

Proposez un schéma logique pour votre modèle conceptuel. N'oubliez pas d'indiquer les clés primaires et clés étrangères.

1

Réponse:

Question 3.

Écrivez la requête SQL qui permet d'obtenir l'auteur et la date d'acquisition de l'œuvre intitulée 'La promenade'.

1

Réponse:

Question 4.

Écrivez la requête SQL qui permet d'afficher la liste des titres de toutes les œuvres dans la salle 2 du musée Louvre dont l'auteur est Monet.

1

Réponse:

Question 5.

Écrivez la requête SQL qui indique le nom de toutes les salles avec leur numéro de musée où une œuvre y est exposée avec une date d'acquisition ultérieure au 1^{er} janvier 2000.

1

Réponse:

2 Bases de données non traditionnelles

Question 6.

Expliquez pourquoi les données ont besoin d'être dénormalisées dans les systèmes NO-SQL comme MongoDB ?

1

Réponse:

Question 7.

On souhaite gérer des données relatives à un ensemble de musées à l'aide de MongoDB. Considérons l'exemple suivant de données dénormalisées :

1

```
[
  {
    5: "Le Louvre",
    salles : [
      {
        92 : "Salle des États",
        oeuvres: [
          {
            "INV 779": "La Joconde",
            "auteur": "Léonard de Vinci"
          },
          {
            "INV 142": "Les noces de Cana",
            "auteur": "Veronese"
          }
        ]
      },
      {
        345: "Salle de la Grèce classique et hellénistique",
        oeuvres: [
          {
            "INV 453", "Venus de Milo",
            "auteur": "Milo"
          },
          ...
        ]
      },
      ...
    ]
  },
  {
    12: "Beaubourg",
    salles: [
      ...
    ]
  }
]
```

Cette écriture des données vous semble-t-elle correcte ? Pourquoi ? Si non, proposez une modélisation correcte, éventuellement en barrant les lignes incorrectes et en indiquant clairement par quelles lignes les remplacer.

Réponse:

Question 8.

On suppose disposer dans MongoDB d'une collection `musee`.

Proposez une requête pour afficher le nom des musées qui contiennent des œuvres de Léonard de Vinci.

1

Réponse:

Question 9.

Expliquez pourquoi on ne peut pas écrire de requête permettant de compter les salles contenant des œuvres de Léonard de Vinci.

1

Réponse:

Question 10.

Proposez un schéma **GraphQL** pour gérer le système d'information d'un ensemble de musées. Vous vous baserez par exemple sur les données de la question 7.

1

Réponse:

Question 11.

Proposez une requête **GraphQL** pour obtenir les données équivalentes à celles présentées à la question 7.

1

Réponse:

Question 12.

En supposant que la partie **query** du schéma **GraphQL** est comme ci-dessous, expliquer pourquoi on ne peut pas effectuer une requête indiquant le nombre de musées dans la base de données.

1

```
type Query {  
  musees: [Musee]  
  salles: [Salle]  
  oeuvres: [Oeuvre]  
}
```

Réponse:

3 Administration PostgreSQL

Question 13.

3

Pour chacune de ces commandes spécifier son niveau d'exécution (**terminal** ou **PostgreSQL**) et ses **cas d'utilisation** (à quoi sert la commande). Donner toutes les informations à votre disposition sur l'instance PostgreSQL de connexion (hôte / port / rôle / base de données). Puis, **justifier son effet** précis.

1. `admin@rge3df$ sudo systemctl stop postgresql@14-replicat.service`

Réponse:

2. `gpub=# CREATE UNIQUE INDEX data_githubrepo_uniq_idx
ON data_githubrepo (account_id, repo_id);`

Réponse:

3. `gtricot@debian$ pg_restore -U admin -j5 -p 5436 -d db_anon prod_anon_20231015.dump`

Réponse:

Question 14.

Soit le plan de requête suivant :

5

QUERY PLAN

```
-----
GroupAggregate (cost=175972.06..175977.56 rows=200 width=48)
  (actual time=1297.587..1336.764 rows=20 loops=1)
  Group Key: uniq_entities.entity_env_type, uniq_entities.is_protected
  Buffers: shared hit=21209, temp read=5008 written=5022
  -> Sort (cost=175972.06..175972.56 rows=200 width=90)
    (actual time=1291.824..1315.161 rows=138200 loops=1)
    Sort Key: uniq_entities.entity_env_type, uniq_entities.is_protected
    Sort Method: external merge Disk: 14048kB
    Buffers: shared hit=21209, temp read=5008 written=5022
  -> Finalize GroupAggregate (cost=175658.81..175962.42 rows=200 width=90)
    (actual time=546.424..1218.086 rows=138200 loops=1)
    Group Key: leaf_entities_account.entity_uid
    Buffers: shared hit=21209, temp read=3252 written=3261
  -> Gather Merge (cost=175658.81..175952.42 rows=800 width=90)
    (actual time=546.416..1146.403 rows=196927 loops=1)
    Workers Planned: 4
    Buffers: shared hit=21209, temp read=3252 written=3261
  -> Partial GroupAggregate (cost=174658.75..174857.07 rows=200 width=90)
    (actual time=508.869..570.755 rows=39385 loops=5)
    Group Key: leaf_entities_account.entity_uid
    Buffers: shared hit=21209, temp read=3252 written=3261
  -> Sort (cost=174658.75..174698.01 rows=15706 width=90)
    (actual time=508.860..547.917 rows=50667 loops=5)
    Sort Key: leaf_entities_account.entity_uid
    Sort Method: external merge Disk: 5536kB
    Buffers: shared hit=21209, temp read=3252 written=3261
    Worker 0: Sort Method: external merge Disk: 5008kB
    Worker 1: Sort Method: external merge Disk: 5080kB
    Worker 2: Sort Method: external merge Disk: 5264kB
    Worker 3: Sort Method: external merge Disk: 5128kB
  -> Index Scan usinng l_e_acc_112 leaf_entities_account
    (cost=2588.51..173564.12 rows=15706 width=90)
    (actual time=8.082..46.188 rows=50667 loops=5)
    Index Cond: ((cluster_id)::text
                  = ANY ('1582439, 238830284, 6822395')::text[]))
    Filter: ((NOT is_deleted) AND (account_id = '001500a86AAC'))
    Rows Removed by Filter: 3482
    Buffers: shared hit=21177
Planning Time: 0.553 ms
Execution Time: 1341.816 ms
(42 rows)
```

1. Quels mots clés ont permis d'obtenir le plan d'exécution de cette requête ?

Réponse:

(Suite du problème page suivante)

*** Ces questions portent sur le plan d'exécution de la page précédente ***

3. Combien de noeuds sont présent dans le plan d'exécution ? **Justifiez.**

Réponse:

4. Quelle est la durée prise par l'exécution de cette requête par le serveur de base de données ?
Où trouve-t-on cette information ?

Réponse:

5. Quel est le nombre de lignes retournées par la requête ? Justifiez.

Réponse:

6. Ce nombre était-il attendu ? Justifiez.

Réponse:

(Suite du problème page suivante)

*** Ces questions portent sur le plan d'exécution deux pages ci-dessus ***

7. Quel noeud d'opération a pris le plus de temps ? Justifiez.

Réponse:

8. Les cardinalités ont-elles bien été estimées ? Justifiez.

Réponse:

9. Le paramètre *work_mem* est-il bien paramétré ? Justifiez.

Réponse: