

TP Noté

06 Nov 2024

Règles

- Vous avez accès au [cours](#) et à internet.
- Voici la répartition des questions et du barème de points :
 - 25% théorique (5 points)
 - 75% pratique
 - 33% sur MongoDB (5 points)
 - 66% sur SQL / Postgres (10 points)

VOUS POUVEZ RÉPONDRE DANS L'ORDRE QUE VOUS VOULEZ ;-)

Démarche pour structurer votre copie

Le TP se découpe en trois parties :

1. Des questions théoriques
2. Des requêtes SQL à écrire dans pgadmin
3. Des requêtes à écrire dans mongodb

La première partie sera à remplir directement dans le formulaire ci joint

<https://forms.gle/iDFUKxkyGeVTouT96>

La deuxième partie sera à rédiger dans un fichier au format *nom_prenom_pg.sql* et à insérer dans le formulaire.

La troisième partie sera à rédiger dans un fichier au format *nom_prenom_mongo.js* et à insérer dans le formulaire.

Notes additionnelles

Je vous invite à suivre cette approche pour structurer votre fichier de réponse.

1. Mettez le numéro de la question en commentaire. SVP présentez vos questions dans l'ordre même si vous êtes libre d'y répondre dans l'ordre que vous souhaitez.
2. Écrivez votre requête de manière la plus lisible. Si jamais vous avez des doutes, partagez les en commentaire. Comme on dit, plus on en met, plus on peut avoir de points ;-) (ceci dit, n'en faites pas trop non plus!)

Partie théorique

1. Partie Théorique (5 points)

- Qu'est-ce qu'une base de données NoSQL ? Qu'est-ce qui la différencie d'une base relationnelle ?
- Expliquez l'acronyme ACID et son importance. Quel est son impact dans le milieu NoSQL ?
- Quelles sont les principales différences entre OLTP et OLAP ?
- Qu'est-ce que la normalisation en SQL ? Citez les formes que vous connaissez

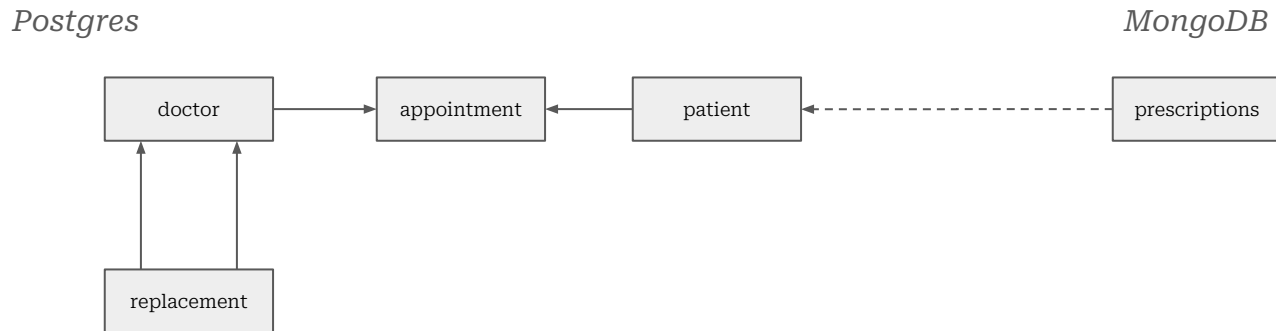
Partie pratique

Votre mission (si vous l'acceptez)

Vous êtes missionné en tant que data engineering pour intervenir dans un hôpital.

Dans un premier temps, vous serez amené à interagir avec différentes briques de leur système d'informations (SI).

- Voici le modèle de données



PostgreSQL
(10 points)

Mission PG1

Un nouveau médecin, John Smith, vient d'intégrer le service Cardiologie de l'hôpital. C'est un médecin titulaire et il a 8 ans d'expérience; ce n'est donc pas un interne.

Il est joignable par email john.smith@example.com.

Écrivez une requête permettant d'ajouter ce médecin dans la table *doctor*.

Mission PG2

Karen Phillips a fini son internat et a été embauché en tant que docteur au sein de l'hôpital. Son email n'est plus à jour. Elle est joignable à sa nouvelle adresse `karen.phillips@example.com`

Écrivez une requête matérialisant les changements nécessaire à faire en base de données.

Mission PG3

Après une interview avec les responsables de la DSI, vous apprenez qu'il y a des doublons dans la table des patients.

1. Écrivez une requête pour identifier les doublons.
2. Écrivez une requête pour supprimer le doublon. On gardera l'id le plus grand.

Mission PG4

Pour mieux vous représenter l'hôpital dans lequel vous venez d'arriver, nous allons faire un peu d'analyse.

Dans un premier temps, rédiger une requête qui me donne le nombre total de rdv par médecin.

Mission PG5

Les médecins peuvent avoir des internes comme remplaçant.

Plusieurs internes peuvent remplacer plusieurs médecins.

Écrivez une requête pour connaître le nombre de remplaçants par médecin.

Mission PG6

Une mission un peu plus compliquée vous attend.

Écrivez une requête pour connaître le nombre de rdv par remplaçant pour un médecin particulier (id à préciser en dur dans la requête) sur une la période de remplacement.

Mission PG7

Intéressons-nous un peu aux patients.

Écrivez une requête pour connaître le/la patient•e qui a le plus de RDVs.

Mission PG8

Maintenant, cherchons quels sont les médecins les plus occupés.

Écrivez une requête pour connaître le TOP 3 des médecins qui consultent le plus.

Mission PG9

On souhaite désormais s'intéresser à l'activité des internes.

Écrivez une requête permettant de connaître le nombre de rdv fait par un interne lors de sa période de remplacement.

Mission PG10

Intéressons-nous aux différentes spécialités au sein de l'hôpital.

Écrivez une requête permettant de connaître le nombre de rdv et de médecins par spécialité médicale.

Mission PG11

Écrivez une requête permettant de connaître l'âge moyen et le nombre de patients par spécialité.

Mission PG12

On s'intéresse aux médecins ayant le plus de jours de remplacements au cours de l'année 2024.

Écrivez une requête permettant de connaître les médecins, leur spécialité, le nombre de remplacement et le nombre de jours total remplacés.

MongoDB
(5 points)

Description

La base de données MongoDB est utilisée pour stocker les ordonnances (prescriptions en anglais).

A droite, une description du schéma de la table, en suivant la syntaxe *json schema*.

A noter que seul le champ *doctor* est obligatoire.

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "doctor": {
        "type": "object",
        "required": ["last_name", "first_name", "address", "specialty"],
        "properties": {
          "last_name": { "type": "string" },
          "first_name": { "type": "string" },
          "address": { "type": "string" },
          "specialty": { "type": "string" }
        }
      },
      "medication": { "type": "string" },
      "medication_type": { "type": "string" },
      "prescription_type": {
        "type": "string",
        "enum": ["consultation"]
      },
      "specialist": { "type": "string" },
      "details": {
        "type": "object",
        "properties": {
          "boxes": { "type": "integer" },
          "dosage": { "type": "string" },
          "duration": { "type": "string" },
          "route": {
            "type": "string",
            "enum": ["oral", "injection", "inhalation", "topical"]
          },
          "urgency": {
            "type": "string",
            "enum": ["urgent", "semi-urgent", "scheduled"]
          },
          "reason": {
            "type": "string",
            "enum": [
              "initial consultation",
              "follow-up visit",
              "acute symptoms",
              "chronic condition management"
            ]
          }
        }
      },
      "required_documents": {
        "type": "array",
        "items": {
          "type": "string",
          "enum": [
            "medical history",
            "imaging results",
            "blood work",
            "previous prescriptions"
          ]
        }
      }
    },
    "required": ["doctor"]
  }
}
```

Mission MG1

Commençons notre aventure avec MongoDB de manière simple et agréable.

Dans un premier temps, écrivez la commande pour compter le nombre total de documents dans la collection *prescriptions*.

Mission MG2

Écrivez la commande permettant de trouver toutes les prescriptions urgentes.

Vous pourrez utiliser le champ *urgency* de l'objet *details*.

Mission MG3

En vous inspirant de la requête précédente,

Écrivez les commandes permettant de compter toutes les prescriptions urgentes.

Vous ferez deux requêtes :

- Une avec la fonction *countDocuments()*
- Une autre avec la logique *aggregate([])* (avec les operateurs *\$match* et *\$count*)

Mission MG4

Écrivez une commande permettant de lister toutes les médications uniques.

Utiliser la fonction *distinct()*

Mission MG5

Place à un peu d'analyse (avec la fonction *aggregate([])*).

Écrivez une commande listant le top 5 des médicaments les plus prescrits.

Vous pourrez utiliser le champ *medication* ainsi que les opérateurs suivant :

\$match, *\$exist* / *\$group*, *\$sum* / *\$sort* / *\$limit*

Mission MG6

Place à un peu d'analyse (avec la fonction *aggregate()*).

Écrivez une commande comptant le nombre de prescriptions par *type* d'antibiotique et *route* d'admission.

Vous avez atteint la fin du sujet

Pensez à bien indenter vos requêtes afin de faciliter la lecture et **ensuite** à soumettre le form.

Merci de votre participation !