université de BORDEAUX	ANNEE UNIVERSITAIRE 2020-2021 SESSION1 DE PRINTEMPS 2021 **MENTION** : MASTER BIOINFORMATIQUE **Code UE : 4TBI803U** **Intitulé de l'épreuve : BASES DE DONNÉES : FONDEMENTS ET APPLICATIONS** **Date :** 03/05/2021 **Heure : 14h30** **Durée :** 1h30 Documents : autorisés Épreuve de M : Bruno Pinaud	**Collège Sciences et technologies** **Masters**

Exercice 1 – NoSQL vs Relationnel (2 points)

Expliquer avec vos mots (sans paraphraser le cours) les limites du modèle relationnel qui ont amené depuis une bonne décennie à l'émergence des bases de données avec le paradigme NoSQL. Vous pouvez déjà commencer par rappeler une définition d'une base de données relationnelles et d'un système NoSQL.

Exercice 2 – Implémentation d'une base de données (8 points)

Soit le schéma relationnel partiel du Windsurf-Club de la Côte de Rêve (WCCR) :
SPOTS (NUMSPOT, NOMSPOT, EXPOSITION, TYPE, NOTE)
PLANCHISTES (NUMPERS, NOM, PRENOM, NIVEAU)
MATOS (NUMMAT, MARQUE, TYPE, LONGUEUR, VOLUME, POIDS)
VENT (DATE, NUMSPOT, DIRECTION, FORCE)
NAVIGUE (DATE, NUMPERS, NUMMAT, NUMSPOT)

La base comprend cinq relations. Les SPOTS sont les bons coins pour faire de la planche, avec un numéro, leur nom, l'exposition principale au vent par exemple 'Sud-Ouest', le type par exemple 'Slalom' ou 'Vague', et une note d'appréciation globale. Les PLANCHISTES sont les membres du club et les invités, les niveaux varient de 'Débutant' à 'Compétition' (5 niveaux au total). Le MATOS (jargon planchiste pour matériel) comprend la description des planches utilisées (pour simplifier les voiles, ailerons, etc., ne sont pas représentés). Le VENT décrit la condition moyenne d'un spot pour une date donnée. Enfin NAVIGUE enregistre chaque sortie d'un planchiste sur un spot à une date donnée avec le matos utilisé. On suppose qu'un planchiste ne fait qu'une sortie par jour et ne change pas de matos ni de spot dans une même journée.

Exercice 1.1 – Création de la base de données (3 points)

On souhaite implémenter la base de données du WindSurf-Club dans un SGBD.

1) Indiquez les clés étrangères éventuelles de chaque relation et toutes autres contraintes pertinentes à implémenter dans la base de données. Indiquez aussi un domaine de valeur pour chaque attribut.
2) En utilisant la description de la base de données et vos réponses à la question précédente, représenter le schéma relationnel par un graphe.

Exercice 1.2 – Algèbre relationnelle et requêtes SQL (5 points)

Écrivez les expressions suivantes en utilisant l'algèbre relationnelle (si possible sinon indiquez que c'est impossible) et le langage SQL :

a) Nom des planchistes de niveau 'Confirmé' qui ont navigué le '20/07/2020' sur un spot où le vent moyen était supérieur à force 4 sur une planche de moins de 2,75 m.
b) Nom des planchistes qui ne sont pas sortis le '20/07/2020'.
c) Nom des planchistes qui ont essayé tous les types de planches de la marque 'FANA-BIC'. On suppose que tous les types sont dans la relation MATOS.
d) Pour chaque spot de la base, en indiquant son nom, donner le nombre de jours de vent au moins de force 4 pour l'année 2021.

Exercice 3 – DF et formes normales (4 points)

Soit la relation suivante : T[E, R, D, P] et ses dépendances fonctionnelles (DF) associées {E → D, D → R}.
A partir de cette relation, quatre décompositions sont envisagées :

Décomposition 1 : R1[E, D] avec la DF {E → D}
R2[D, R] avec la DF {D → R}

Décomposition 2 : R1[E, D] avec la DF {E → D}
R3[E, R, P] avec la DF {E → R}

Décomposition 3 : R2[D, R] avec la DF {D → R}
R4[E, D, P] avec la DF {E → D}

Décomposition 4 : R1[E, D] avec la DF {E → D}
R2[D, R] avec la DF {D → R}
R5[E, P] avec aucune DF.

1) Comment vérifier si la décomposition d'une relation est en 3e forme normale et sans perte.
2) Est-ce que chaque décomposition est en 3e FN et sans perte ? Justifiez votre réponse.

Exercice 4 – Création d'une base de données en SQL (6 points)

Soit la création de deux tables en SQL pour modéliser les départements français et leur population dans le temps:

```
CREATE TABLE Villes (
    -- nom de la ville
    nom varchar(80) CONSTRAINT nomPK PRIMARY KEY,
    -- département
    departement varchar(2) CONSTRAINT populationFK
      REFERENCES Departements(departement),
    -- chef-lieu du département
    CheflieuDepartement boolean DEFAULT FALSE,
    -- nombre d'habitants
    population int CONSTRAINT check_population
      CHECK (population>0) CONSTRAINT popnull NOT NULL
);

CREATE TABLE Departements (
    -- nom du département
    nom varchar(80),
    -- numéro du département
    departement varchar(2) CONSTRAINT depPK PRIMARY KEY
);

CREATE TABLE Etablissements (
    -- ville
    idVille varchar(80) REFERENCES Villes(nom),
    -- école
    idEcole int NOT NULL,
    -- type de l'établissement
    typeEcole int REFERENCES Types(id),
    PRIMARY KEY(idVille, idEcole)
);

CREATE TABLE Types (
    -- identifiant du type
    id serial PRIMARY KEY,
    -- nom du type
    nom varchar(20) CHECK
      (nom IN ('maternelle', 'primaire', 'collège', 'lycée'))
);
```

1) Quel est le degré de chaque relation ? (1 point)

2) Pourquoi lors de la déclaration d'une table en SQL, est-il recommandé de déclarer des contraintes d'intégrité et de les nommer avec le mot le mot-clé optionnel constraint ? Ajoutez une contrainte d'intégrité pertinente dans le schéma (1 point).

3) Que signifie le mot-clé not null ? Pourquoi ce mot-clé n'est pas utilisé pour l'attribut nom de la table département ? (1 point)

4) Les 3 instructions suivantes sont à exécuter dans l'ordre proposé. Vont-elles déclencher des erreurs ? Si oui, lesquelles et pourquoi (3 points)?

   a) INSERT INTO villes(nom,departement) VALUES ('Lyon','69');

   b) INSERT INTO villes(nom, population, departement) VALUES ('Marseille', '120000','13');

   c) INSERT INTO villes(nom, population, departement) VALUES ('Lyon', '150084','69');

5) L'ordre proposé pour la création des tables est-il correct ? Si non, proposez et justifiez un ordre correct.