

Documents de cours autorisés. Répondre de manière claire et concise. Le barème est indicatif.

**Répondre exclusivement sur les feuilles réponses fournies avec le sujet.**

## I. Contexte

Un directeur de théâtre désire informatiser son système de réservation de places pour les spectacles qui se déroulent dans son théâtre au cours d'une saison. Les données de l'application décrites ci-après ont été simplifiées.

### LESSPECTACLES (NUMS, NOMS)

$\{(ns, nms) \in \text{LESSPECTACLES} \Leftrightarrow \text{Un spectacle est identifié par son numéro } ns \text{ et on connaît son nom } nms.\}$

### LESREPRESENTATIONS (NUMS, DATEREP)

$\{(ns, d) \in \text{LESREPRESENTATIONS} \Leftrightarrow \text{Une représentation identifiée par son numéro } ns \text{ débute à un moment donné } d \text{ exprimé à la granularité de l'heure (par exemple 28/11/2023 20H). Un spectacle fait généralement l'objet de plusieurs représentations proposées à des moments différents.}\}$

### LESCATEGORIES (NUMS, CATEGORIE, PRIX)

$\{(ns, nc, p) \in \text{LESCATEGORIES} \Leftrightarrow \text{Le tarif de base associé à la catégorie } nc \text{ pour le spectacle identifié par } ns \text{ est fixé au prix } p.\}$

### LESPLACES (NOPPLACE, NORANG, CATEGORIE)

$\{(np, nr, c) \in \text{LESPLACES} \Leftrightarrow \text{Une place identifiée par son numéro } np \text{ dans son rang } nr \text{ est de catégorie } c \text{ (3 catégories possibles)}\}$

### LESTICKETS (NUMSERIE, NUMS, DATEREP, NOPPLACE, NORANG, DATEEMISSION, NODOSSIER)

$\{(n, ns, d, np, nr, de, do) \in \text{LESTICKETS} \Leftrightarrow \text{Chaque place vendue fait l'objet de l'émission d'un ticket identifié par un numéro de série } n, \text{ appartient à un dossier } do, \text{ et estampillé par la date } de \text{ au moment de la transaction. Un ticket concerne un spectacle de numéro } ns \text{ à la date de représentation } d. \text{ Il indique le positionnement dans la salle par le numéro de place } np \text{ au rang } nr.\}$

### LESDOSSIERS (NODOSSIER, MONTANT)

$\{(nd, m) \in \text{LESDOSSIERS} \Leftrightarrow \text{Chaque achat se traduit par la création d'un dossier identifié par son numéro } nd \text{ auquel est associé le prix global des places } m. \text{ Un achat peut concerner plusieurs places.}\}$

domaine(dateRep, dateEmission) = date      domaine(NOMS) = Chaine de caractères

domaine(categorie) = {'orchestre', 'balcon', 'poulailleur'}

domaine(noPlace, noRang, numS, noDossier, NumSerie) = entier > 0

domaine(prix, montant) = réels > 0

LesPlaces[categorie]  $\subseteq$  LesCategories[categorie]

LesRepresentations[numS] = LesSpectacles[numS]

LesTickets[numS, dateRep]  $\subseteq$  LesReprésentations[numS, dateRep]

LesTickets[noPlace, noRang]  $\subseteq$  LesPlaces[noPlace, noRang]

LesTickets[noDossier]  $\subseteq$  LesDossiers[noDossier]

## II. Requêtes (8 points)

**Q1)** Quelles sont les spectacles qui n'ont vendu aucun ticket (toutes représentations confondues) ? Le résultat attendu est une liste de NOMS. Vous donnerez uniquement la version algébrique de cette requête.

**Q2)** Pour quelles représentations de spectacles tous les tickets correspondant à des places de catégorie 'orchestre' ont-ils été vendus? Le résultat est un couple (NUMS, DATEREP). Vous écrirez la requête en la considérant comme une division.

- a. Vous compléterez la version algébrique,
- b. Et la version SQL de cette requête.

**Q3)** Quel(s) spectacle(s) a(ont) vendu le plus de tickets ? Le résultat attendu est une liste de NUMS. Donner la requête SQL qui utilise un pattern spécifique sans clause LIMIT et ORDER BY.

**Q4)** Pour chaque spectacle, donner le nombre de tickets vendus de la catégorie 'balcon'. Le résultat sera de la forme (NUMS, NB\_BALCON). Dans le cas où un spectacle n'a vendu aucune place de cette catégorie le résultat ((NUMS, 0) sera renvoyé. Cette requête SQL doit utiliser obligatoirement une jointure externe.

## III. Normalisation (4 points)

Les informations dans le dossier sont enrichies de nouvelles données et le schéma suivant est proposé :

LESDOSSIERS2 (NODOSSIER, MONTANT, NOMC, VILLE, CP, MOYENP)

$\{(nd, m, nc, v, cp, mp) \in \text{LESDOSSIERS2} \Leftrightarrow \text{Chaque achat se traduit par la création d'un dossier identifié par son numéro } nd \text{ auquel est associé le prix global des places } m. \text{ Un achat peut concerner plusieurs places. Le nom } nc \text{ du client ayant réalisé la transaction, sa ville de résidence } v \text{ ainsi que le code postal associé } cp \text{ sont ajoutés au dossier. Le moyen de paiement } mp \text{ utilisé par le client est également précisé.}\}$

Avec les dépendances fonctionnelles suivantes :

NODOSSIER → MONTANT, NOMC, MOYENP      NOMC → VILLE, CP

NODOSSIER, NOMC → MOYENP      CP → VILLE

**Q5)** Donner le graphe minimal de dépendance et préciser la (ou les) clé(s) de la relation LESDOSSIERS2.

**Q6)** Donner en justifiant la forme normale de LESDOSSIERS2.

**Q7)** Proposer une décomposition optimale de cette relation (aucune table résultante ne peut être décomposée).

## IV. SPRING - JPA (8 points)

**Q8)** Placer les annotations JPA (on ignorera les imports nécessaires) sur la classe Spectacle ci-dessous pour qu'elle persiste dans la table énoncé dans le contexte.

```
public class Spectacle {
    private Long num;
    private String nom;
    // ...
}
```

**Q9)** Mapper la relation representation de cette même classe Spectacle. Pour simplifier, la date d'une Representation ne fait plus partie de la clé primaire mais on lui ajoute sa propre clé primaire (cf. Q10). Si un spectacle est supprimé alors toutes ses représentations le sont aussi. On peut créer un spectacle et ses représentations en une fois.

```
public class Spectacle {
    // ...
    private Set<Representation> representations;
}
```

**Q10)** En tenant compte des contraintes de la question Q8 et de l'implémentation de `Representation` ci-dessous (on fait l'hypothèse qu'elle est annotée correctement pour être persistante).

```
//...
public class Representation {
    private Long id;
    private Instant date;
    public int hashCode() {
        return Objects.hash(date);
    }
}
```

En considérant l'extrait du code de test suivant :

```
@Test
void testRepresentations() {
    var s = new Spectacle();
    s.setNom("Bernard Minet au Casino de Paris");
    var r1 = new Representation(Instant.now().plus(1,ChronoUnit.DAYS));
    var r2 = new Representation(Instant.now().plus(3,ChronoUnit.DAYS));
    var r3 = new Representation(Instant.now().plus(3,ChronoUnit.DAYS));
    s.addRepresentations(r1, r2, r3);
    entityManager.persist(s);
}
```

Donner le nombre de lignes inscrite dans la table “LESREPRESENTATIONS” et justifier votre réponse.

**Q11)** Compte tenu du modèle de la question précédente, écrire la requête JPQL permettant de récupérer les spectacles dont les représentations ont eu lieu avant l'année 2020.