

Examen du 26 mai 2025 –Base de Données

Cette épreuve est composée de deux parties : partie I BD, partie II Spring/JPA. durée totale 60 mn.
Documents interdits, sauf 1 feuille A4 recto-verso

I. Partie BD (10 points)

1 Contexte

Planet Express est une entreprise de livraison intergalactique. Le Professeur Hubert Farnsworth à créer une base de données pour gérer l'entreprise et qui permet de suivre les **employés**, leurs **missions**, les **planètes** desservies, les **vaisseaux** utilisés, les **colis** gérés, les **comptes rendus** d'accidents.

MISSION (mission_id, title, description, mission_date, planet_id, ship_id)

$\{(id, t, d, md, p, s)\} \Leftrightarrow$ une livraison spatiale est identifiée par id , intitulée t , décrit par d , a lieu à la date md , concerne la planète p , et en utilisant le vaisseau s .

EMPLOYEE (employee_id, name, grade, species, money)

$\{(id, n, g, s, m)\} \Leftrightarrow$ Un employé est identifié par id , possède le nom n , le grade g , est de l'espèce s . Son salaire est m brouzoufs.

SHIP (ship_id, name, model)

$\{(id, n, m)\} \Leftrightarrow$ Un vaisseau est identifié par id , possède un nom n et est du modèle m .

PLANET (planet_id, name, type)

$\{(id, n, t)\} \Leftrightarrow$ Une planète est identifiée par id , possède un nom n et un type t .

PACKAGE (package_id, content, weight, mission_id, warehouse_id)

$\{(id, c, w, m, wh)\} \Leftrightarrow$ Un colis est identifié par id , possède un poids cw , est rattaché à la mission m , et est stocké dans l'entrepôt wh . Une description du colis est donné dans c .

WAREHOUSE (warehouse_id, location, capacity)

$\{(id, l, c)\} \Leftrightarrow$ Un entrepôt est identifié par id , possède une localisation l , et une capacité en nombre de colis de c .

DELIVERY_RATING (rating_id, score, comment, mission_id)

$\{(id, s, c, m)\} \Leftrightarrow$ Un rating est identifié par id concerne la mission m , caractérisé par un score s et un commentaire c .

IMPLANT (implant_id, employee_id, serial_number, function)

$\{(id, e, s, f)\} \Leftrightarrow$ Un implant biométrique est identifié par id , est attaché à un employé e , possède un numéro de série s et précise sa fonction anatomique f .

ACCIDENT_REPORT (accident_id, accident_date, employe_id, mission_id, description)

$\{(id, d, e, m, d)\} \Leftrightarrow$ Un rapport d'accident est identifié par id , concerne l'employé e dans la mission m , et contient une description d de l'accident avec sa date d .

MISSION_CREW (mission_id, employee_id)

$\{(m, e)\} \Leftrightarrow$ Une affectation est identifiée par la mission m , et l'employé concerné e .

domaine(*_id) = long integer >0

domaine(description, comment, content, title, location) = varchar(255)

domaine(name, model, grade, species, function, serial_number) = varchar(255)

domaine(score, capacity) = integer domaine(mission_date, accident_date) = date

domaine(weighth, money) = double

$\pi_{\text{planet_id}} \text{MISSION} \subseteq \pi_{\text{planet_id}} \text{PLANET}$

$\pi_{\text{ship_id}} \text{MISSION} \subseteq \pi_{\text{ship_id}} \text{SHIP}$

$\pi_{\text{mission_id}} \text{PACKAGE} \subseteq \pi_{\text{mission_id}} \text{MISSION}$

$\pi_{\text{warehouse_id}} \text{PACKAGE} \subseteq \pi_{\text{warehouse_id}} \text{WAREHOUSE}$

$\pi_{\text{mission_id}} \text{DELIVERY_RATING} \subseteq \pi_{\text{mission_id}} \text{MISSION}$

$\pi_{\text{employe_id}} \text{IMPLANT} \subseteq \pi_{\text{employe_id}} \text{EMPLOYE}$

$\pi_{\text{mission_id}} \text{ACCIDENT_REPORT} \subseteq \pi_{\text{mission_id}} \text{MISSION}$

$\pi_{\text{employe_id}} \text{ACCIDENT_REPORT} \subseteq \pi_{\text{employe_id}} \text{EMPLOYE}$

$\pi_{\text{mission_id}} \text{MISSION_CREW} \subseteq \pi_{\text{mission_id}} \text{MISSION}$

$\pi_{\text{employe_id}} \text{MISSION_CREW} \subseteq \pi_{\text{employe_id}} \text{EMPLOYE}$

2 Algèbre (3.5 points)

Vous écrirez en algèbre relationnelle (et pas en SQL) les requêtes 1 à 3. Lorsqu'une projection sur des attributs n'est pas explicite, cela signifie que tous les attributs d'une entité sont requis. On désire privilégier l'opération de semi-jointure lorsque cela est possible.

- Q1) Lister les employés qui n'ont pas fait de rapport d'accidents parmi les missions auxquels ils ont participés mais où un rapport d'accident a été émis.
- Q2) Lister les employés qui ont accédé à toutes les planètes répertoriées lors de leurs différentes missions.
- Q3) On s'intéresse aux missions réalisées par chaque vaisseau spatial, et cela même si un vaisseau n'a réalisé aucune mission. Les résultats sont de la forme (ship_id, mission_id) ou (ship_id, NULL). Indiquer la ou les proposition(s) de requêtes correcte(s).

3 SQL (3.5 points)

Vous écrirez en SQL uniquement (pas d'algèbre) les requêtes 4 à 6 :

- Q4) Lister les employés qui ont accédé à toutes les planètes répertoriées lors de leurs différentes missions.
- Q5) Lister les employés qui ont notifié le plus de rapport d'accident. Le résultat sera de la forme (employee_id). Choisir les requêtes correctes parmi les propositions.
- Q6) Compléter la requête qui décrit "*Les employés avec le rôle de capitaine qui n'ont pas de rapport d'accident sur leur mission*".

4 Normalisation (3 points)

Soit la table suivante qui modélise les commanditaires d'une mission :

MISSION_CLIENT(mission_id, clientName, clientEmail, companyName, companyAddress)
 $\{(id, t, n, e, c, a) \Leftrightarrow \text{La mission identifiée par } id \text{ a été commanditée par le client de nom } n, \text{ d'email } e, \text{ appartenant à l'entreprise } c \text{ située à l'adresse } a. \text{ Une même mission peut avoir plusieurs commanditaires.}\}$

Avec les dépendances fonctionnelles :

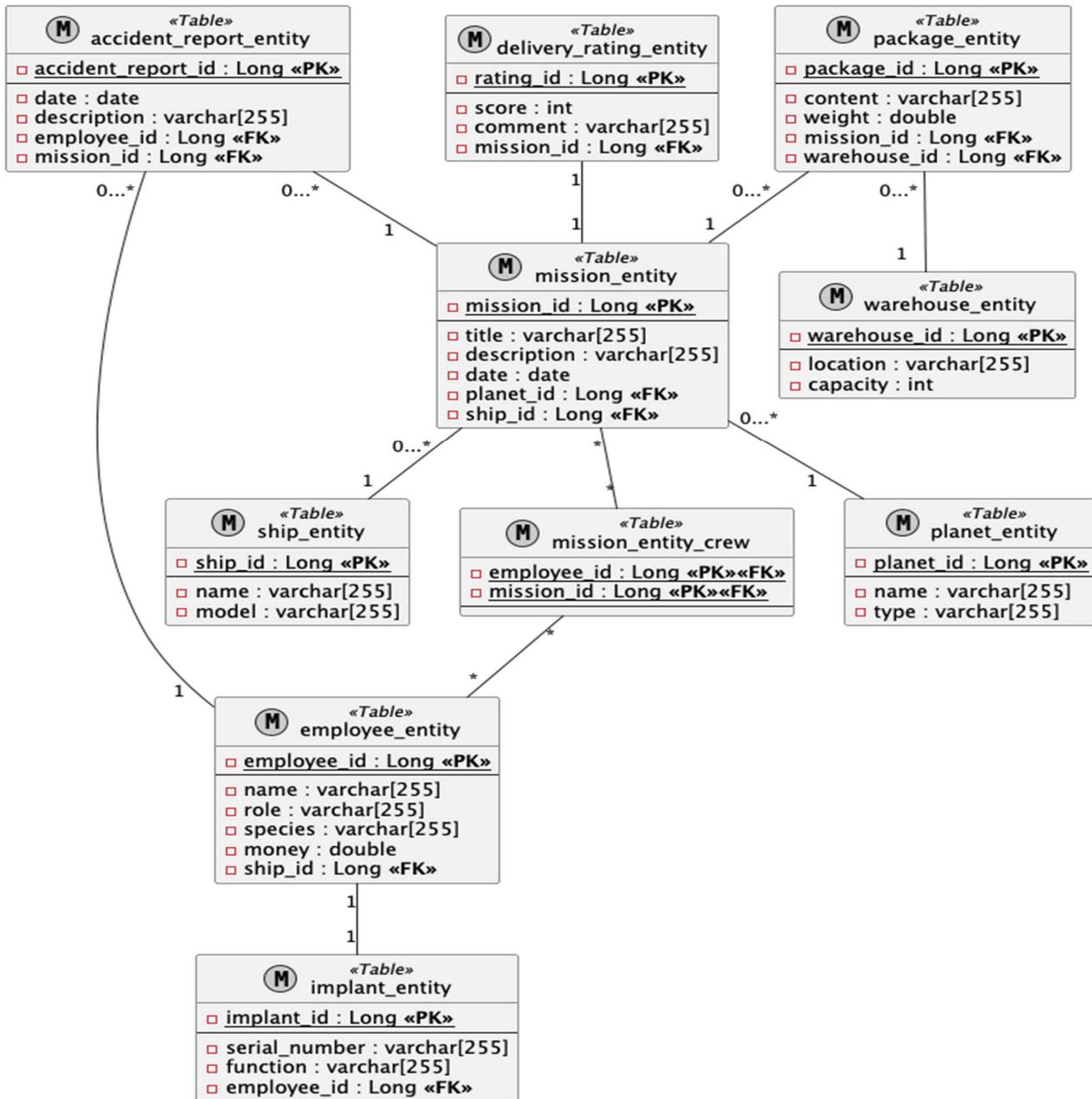
ClientEmail → clientName, companyName
companyName → companyAddress
mission_id, clientEmail → clientName, companyName, companyAddress

- Q7) Donner le graphe Minimum de dépendance fonctionnelle et en déduire le ou les identifiants de MISSION_CLIENT.
- Q8) Donner en justifiant la forme normale de R.
- Q9) Si la forme normale de R n'est pas optimale, proposer une décomposition au moins en 3^{eme} forme normale.

II. Mapping JPA & Repositories (10 points)

Le système actuel n'est pas très pratique. Bender vous demande de l'aider pour réaliser un mapping JPA.

4 JPA



Q10) Bender à fait une partie du schéma, mais a dit que les relations ce n'est pas son truc. Ajouter les relations manquantes dans les entités.

5 Repositories

Hermès veut récupérer la liste des membres des missions pour une planète (par exemple la « Samplanète »).

Q11) Faire le repository permettant ce type d'accès avec sa requête associée.