

<b>Commencé le</b>	mercredi 21 mai 2025, 09:05
<b>État</b>	Terminée
<b>Terminé le</b>	mercredi 21 mai 2025, 11:31
<b>Temps mis</b>	2 heures 26 min
<b>Note</b>	Pas encore évalué

## Description

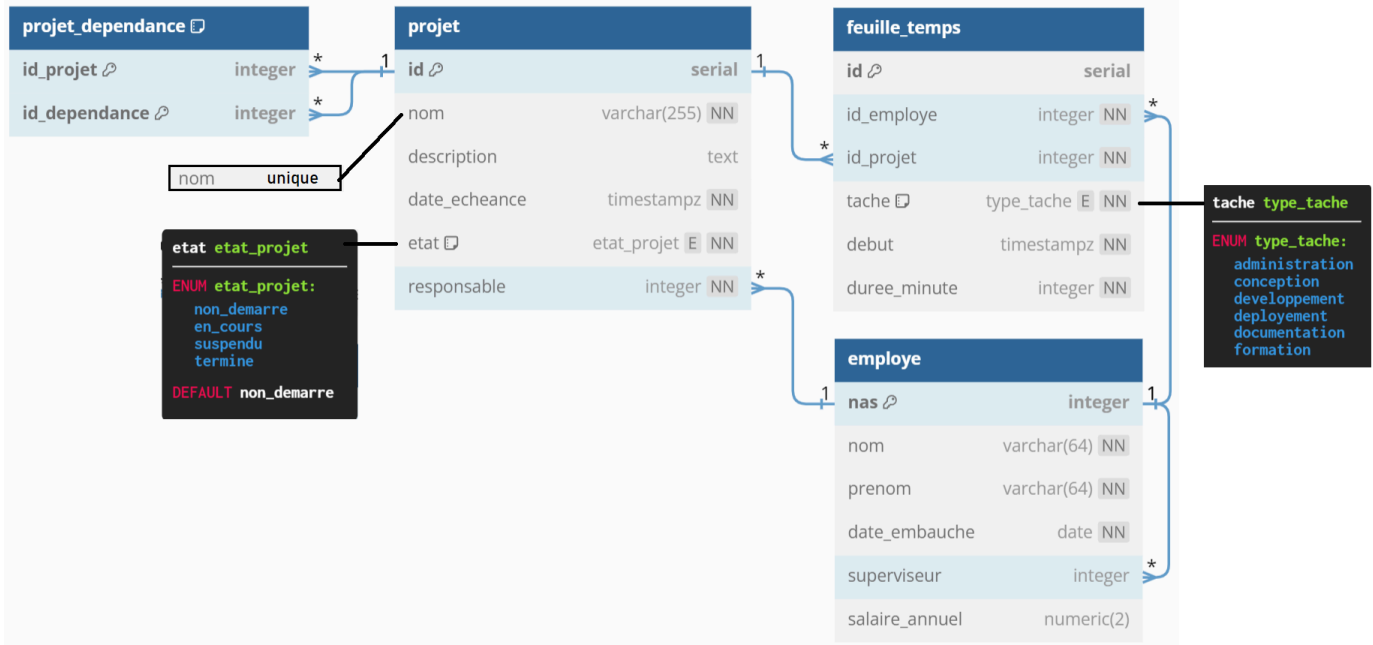
Vous avez la mise en situation suivante.

On désire gérer simplement des **employés**, des **projets** et les **feuilles de temps** décrivant les périodes de travail.

Les associations sont :

- un employé peut potentiellement avoir un superviseur parmi les autres employés
- un projet possède obligatoirement un employé responsable
- une entrée dans les feuilles de temps est toujours associée à un employé et un projet
- un projet peut avoir des dépendances provenant de certains projets (aucun ou plusieurs) où être préalable à d'autres projets (aucun ou plusieurs)

Le schéma relationnel ainsi que le code SQL suivant sont une conception et une implémentation satisfaisante du problème donné.



```
-- Enum: etat_projet
CREATE TYPE etat_projet AS ENUM (
    'non_demarre',
    'en_cours',
    'suspendu',
    'termine'
);

-- Enum: type_tache
CREATE TYPE type_tache AS ENUM (
    'administration',
    'conception',
    'developpement',
    'deploiement',
    'documentation',
    'formation'
);
```

```
-- Table: employe
CREATE TABLE employe (
  nas          INTEGER          PRIMARY KEY,
  nom          VARCHAR(64) NOT NULL,
  prenom       VARCHAR(64) NOT NULL,
  date_embauche DATE          NOT NULL,
  id_superviseur INTEGER       REFERENCES employe(nas),
  salaire_annuel NUMERIC(8,2)  NOT NULL
);

-- Table: projet
CREATE TABLE projet (
  id          SERIAL          PRIMARY KEY,
  nom         VARCHAR(255) NOT NULL UNIQUE,
  description  TEXT,
  date_echeance TIMESTAMPTZ  NOT NULL,
  etat        etat_projet    NOT NULL DEFAULT 'non_demarre',
  id_responsable INTEGER      NOT NULL REFERENCES employe(nas)
);

-- Table: projet_dependance
CREATE TABLE projet_dependance (
  id_projet    INTEGER REFERENCES projet(id),
  id_dependance INTEGER REFERENCES projet(id),

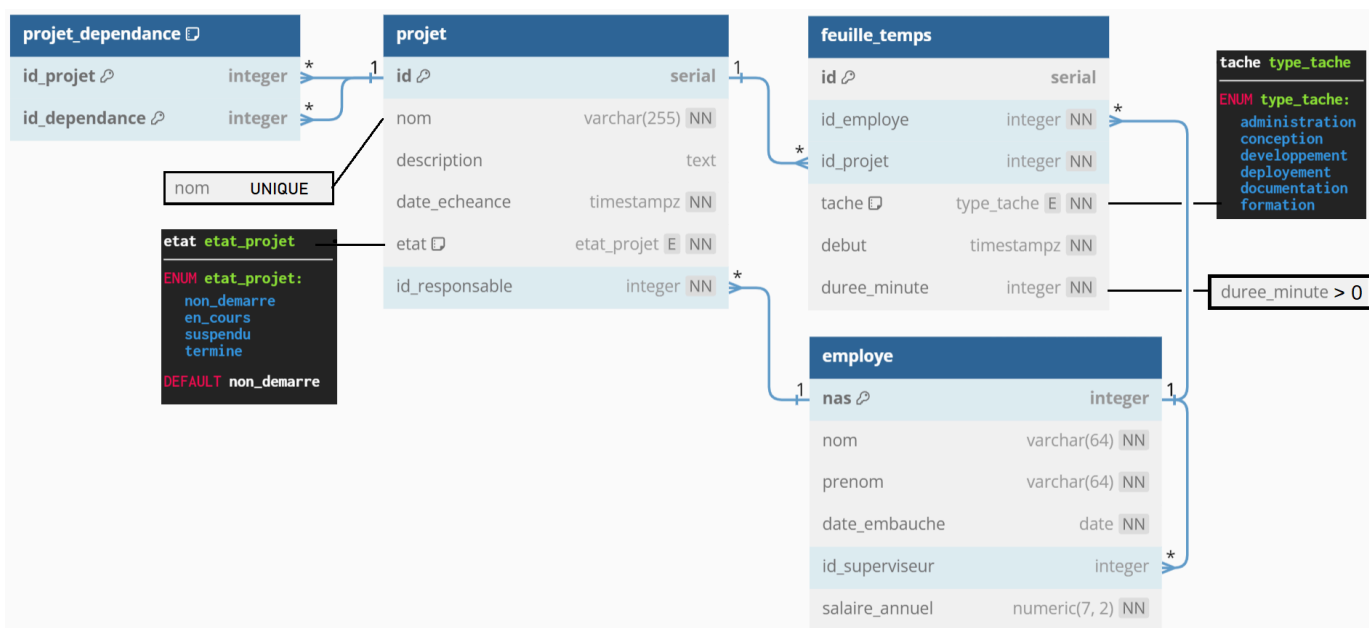
  PRIMARY KEY(id_projet, id_dependance)
);

-- Table: feuille_temps
CREATE TABLE feuille_temps (
  id          SERIAL          PRIMARY KEY,
  id_employe  INTEGER         NOT NULL REFERENCES employe(nas),
  id_projet   INTEGER         NOT NULL REFERENCES projet(id),
  tache       type_tache     NOT NULL,
  debut       TIMESTAMPTZ     NOT NULL,
  duree_minute INTEGER        NOT NULL CHECK (duree_minute > 0)
);
```

## Question 1

Terminé

Noté sur 10,00



Est-ce que la conception proposée présente une dépendance circulaire structurelle?

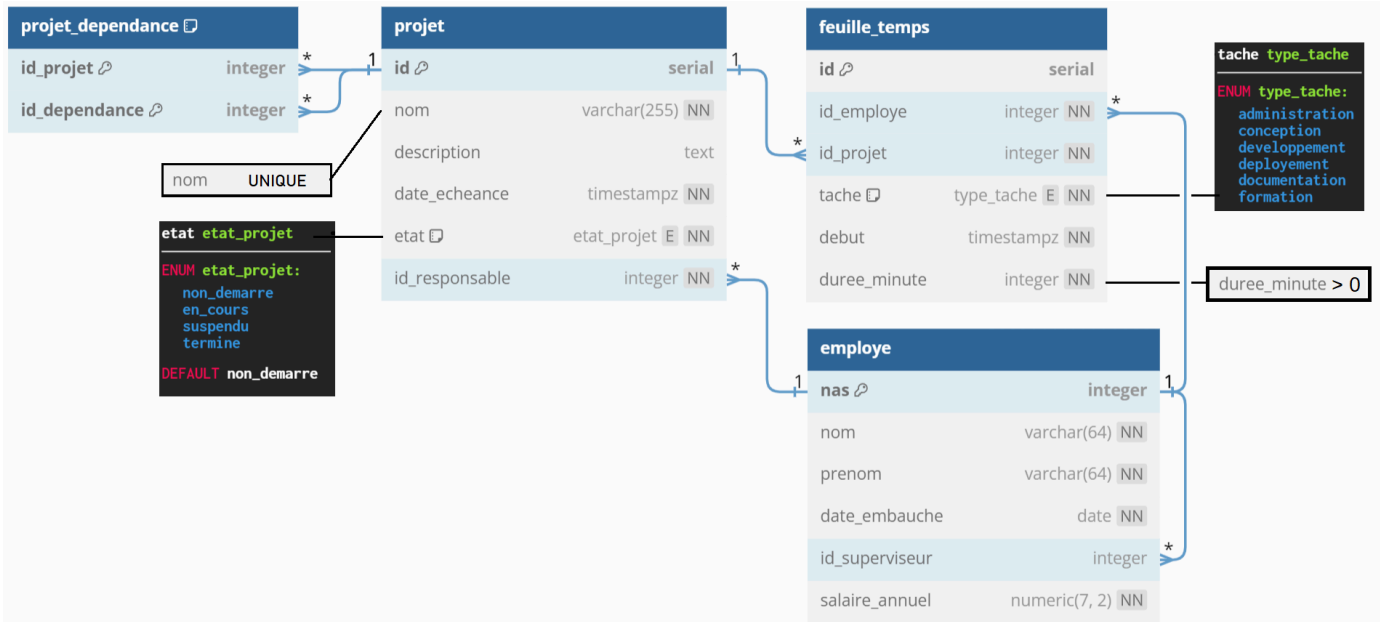
- Si oui, expliquez ce qui détermine cette dépendance.
- Si non, donnez l'ordre de création des tables évitant une telle problématique.

non il n'y aura pas d'erreur de dépendance circulaire, lors de l'insertion la table employe le superviseur pointe vers nas, ensuite projet dépend du nas d'employe, projet dépendance depend de l'id de projet et id\_emp et id\_p de feuille\_temps depend de employe et projet. Toutes les tables sont créées dans le bonne ordre pour éviter

**Question 2**

Terminé

Noté sur 20,00



Rédigez le script permettant de réaliser les opérations du DML suivantes.

## 1. Transaction 1

- L'entreprise démarre un nouveau projet et embauche un nouvel employé pour le superviser.
- Ajouter l'employé suivant :
  - nas : 123
  - nom, prénom : Deschamps, Godefroy
  - date d'embauche : 2025-03-30 (année-mois-jour)
  - superviseur : Marcel L'Italien avec le nas 789 (cet employé existe déjà dans la base de données)
  - salaire annuel : 120 000 \$
- Ajouter le projet suivant :
  - nom : Coolvision
  - description : Outil d'inspection oculaire.
  - date d'échéance : 2026-07-01
  - état : non démarré
  - responsable : Godefroy Deschamps (nas 123)
- Ajouter une nouvelle feuille de temps :
  - projet : Coolvision
  - employé : Godefroy Deschamps (nas 123)
  - tâche : administration
  - date : 2025-03-30
  - durée : 4 heures
- L'employé suivant change de superviseur pour Godefroy Deschamps (nas 123) (l'employé mentionné existe déjà dans la base de données) :
  - Frédéric Lavoie (nas 900)

## 2. Transaction 2

- Ajouter la feuille de temps suivante:
  - Frédéric Lavoie (nas 900) a fait 5 heures développement et 1 heure d'administration sur le projet *Coolvision* le 1 avril 2025 à partir de 8h.

## 3. Transaction 3

- Vous devez identifier et rédiger toutes les opérations du DML nécessaires à la transaction des événements suivants.  
Les employés mentionnés existent déjà dans la base de données.

- Godefroy Deschamps est remercié et est retiré de la base de données.
- **Toutes** les dépendances (FK) associées à Godefroy Deschamps (nas 123) sont transférées à Marc-André Thériault (nas 800).

Begin

```
DEFERRABLE SET CONSTRAINTS ALL DEFERRED;
```

```
Insert Into employe (nas, nom, prenom, date_embauche, id_superviseur  
                    ,salaire_annuel)
```

```
Values (123, 'Deschamps', 'Godefroy', '2025-03-30', 789, 120000)
```

```
Insert Into projet( nom, description, date_echeance, etat,  
                   id_respondable)
```

```
Values('Coolvision', 'Outil d'inspection oculaire', '2026-07-01',  
      'non demarre', 123)
```

```
Insert Into feuille_temps ( id_employe, id_projet, tache,debut,  
                           duree_minute)
```

```
Values ( 123, (SELECT nas FROM projet WHERE nom = 'Coolvision'),  
      'administration', '2025-03-30',240)
```

```
UPDATE employe
```

```
SET superviseur = 123 WHERE nas = 900
```

```
Insert Into feuille_temps ( id_employe, id_projet, tache,debut,  
                           duree_minute)
```

```
Values( 900,(SELECT nas FROM projet WHERE nom = 'Coolvision'),  
      'developpement','2025-05-1 8:00:00', 360)
```

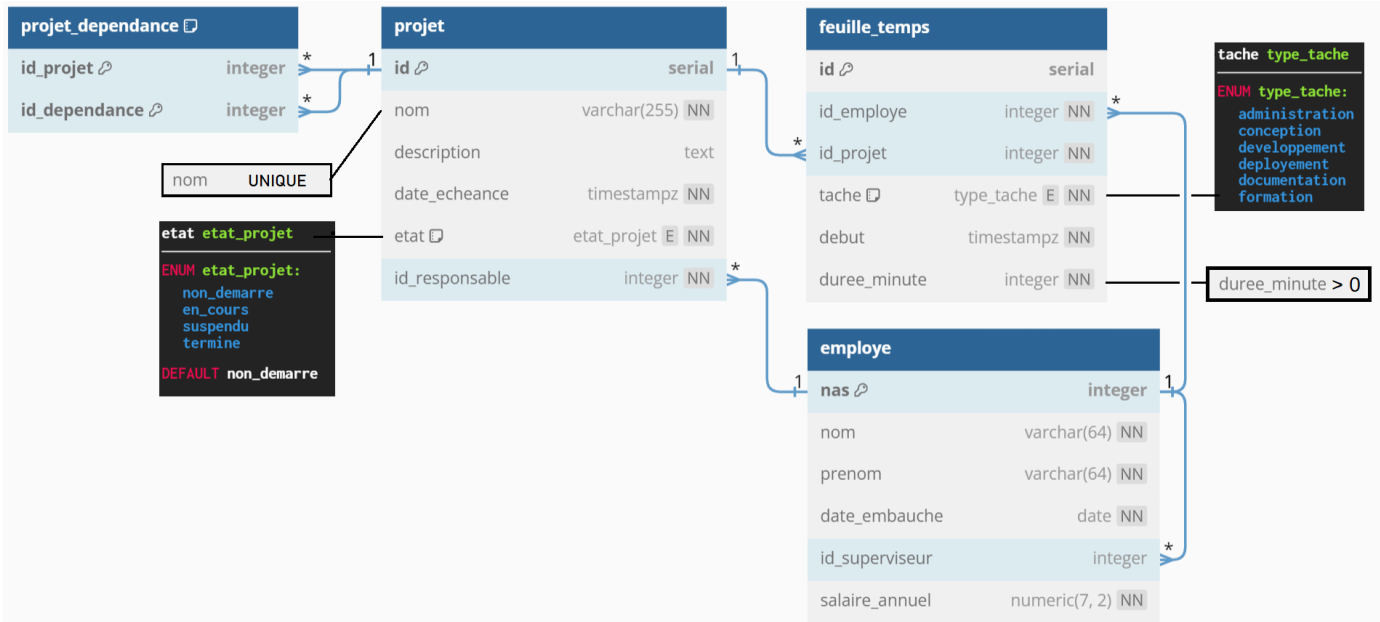
```
UPDATE employe
```

```
SET superviseur = 800 WHERE superviseur = 123
```

### Question 3

Terminé

Noté sur 10,00



Dans la conception actuelle, il est possible de définir un responsable unique pour le projet. Toutefois, si on suppose que le projet s'écoule sur plusieurs années et que plusieurs responsables sont nommés au fil du temps, il est impossible de représenter une telle situation.

1. Pourquoi?
2. Comment le schéma actuel pourrait être modifié pour vous permettre de représenter ces informations et quels outils devriez-vous utiliser pour y arriver.

Décrire votre solution en présentant de façon détaillée les modifications, additions ainsi que les outils utilisés pour y arriver (*automatisation...*)

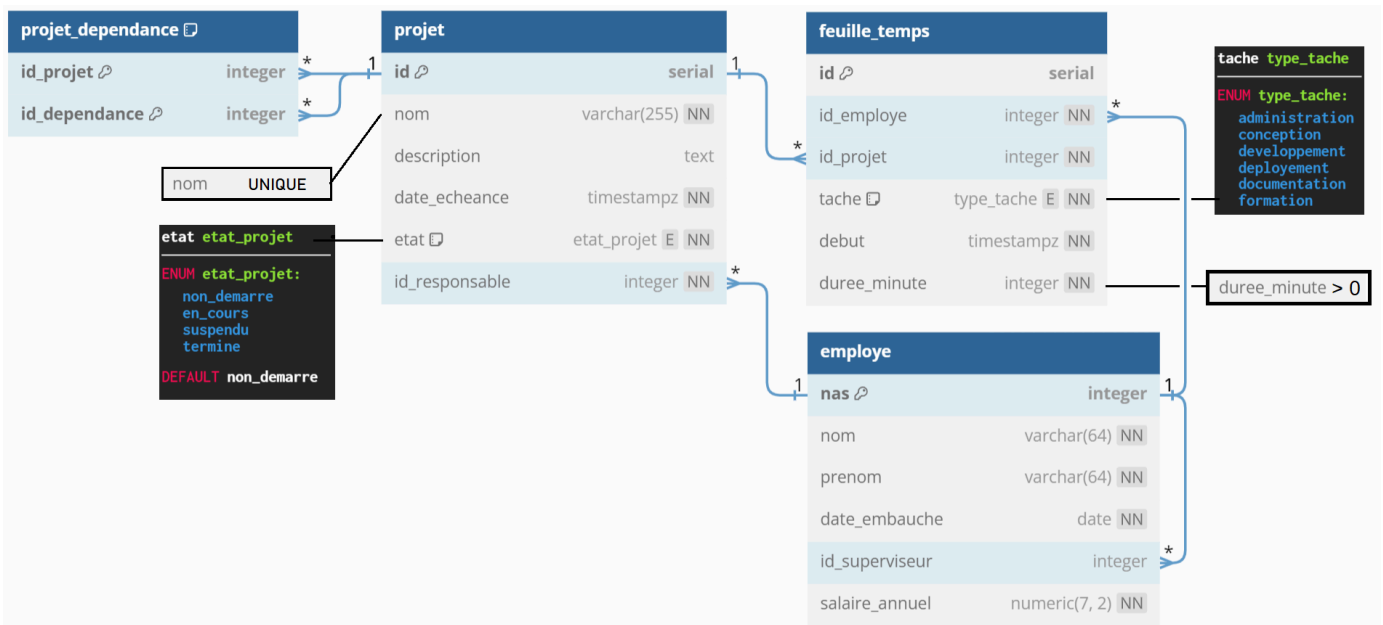
**(aucun code n'est attendu).**

Si on veut que plusieurs responsables se succèdent sur un projet qui dure plusieurs années il faudrait que lors de la création des tables on indique dans les foreign key On update cascade pour que lors de la modification d'un responsable toutes les autres foreign key prennent la nouvelle donné.

**Question 4**

Terminé

Noté sur 10,00



Écrivez une requête SQL pour obtenir les **trois projets** ayant la date d'échéance la plus rapprochée. Affichez le nom du projet, la description. Triez le tout par ordre décroissant de temps restant.

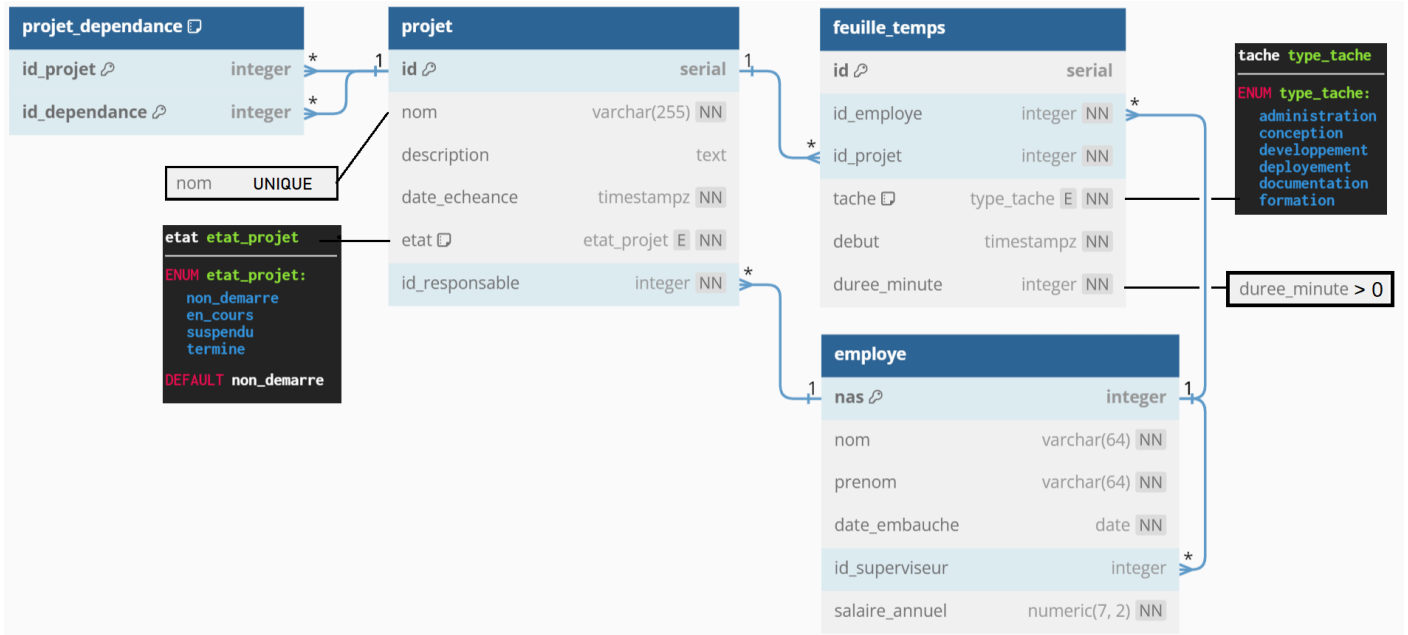
```
SELECT nom, description
FROM projet
ORDER BY date_echeance DESC
LIMIT 3
```



**Question 5**

Terminé

Noté sur 15,00



Créer une fonction en langage **SQL** retournant le id d'un projet à partir du nom du projet.

La fonction doit avoir la signature suivante : `get_id_projet(p_nom projet.nom%TYPE)`

Son retour sera du type: `projet.id%TYPE`

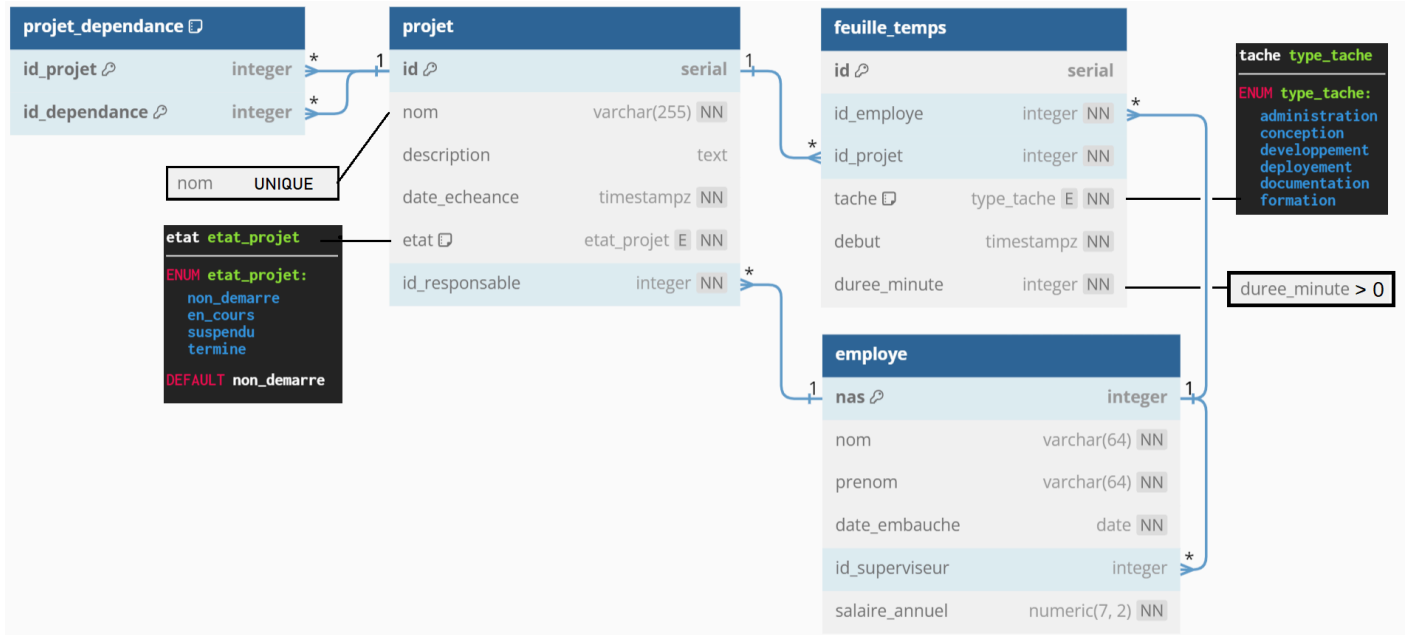
Ensuite, rédiger une requête SQL utilisant votre fonction qui répond à l'énoncé suivant : retourner le nombre de projets dépendant du projet nommé `abc_xyz`.

```
CREATE OR REPLACE FUNCTION get_id_projet(p_nom projet.nom%TYPE)
RETURNS projet.nom%TYPE
AS $$
DECLARE iid INTERGER
BEGIN
    SELECT id INTO iid
    FROM projet
    WHERE nom = p_nom
    return iid
END;
```

## Question 6

Terminé

Noté sur 20,00



On désire mettre en place la contrainte suivante : toutes les entrées faites dans la table des feuilles de temps doivent être antérieures à la date d'échéance du projet concerné.

1. Comment est-il possible d'imposer une telle contrainte? Expliquer en vos propres mots votre solution.
2. Donner le code nécessaire pour réaliser votre solution.

```
on peut utiliser un trigger qui vérifiera si les dates sont antérieurs
on devra mettre en place une fonction qui retournera un trigger
puis mettre en place un trigger pour l'appeler

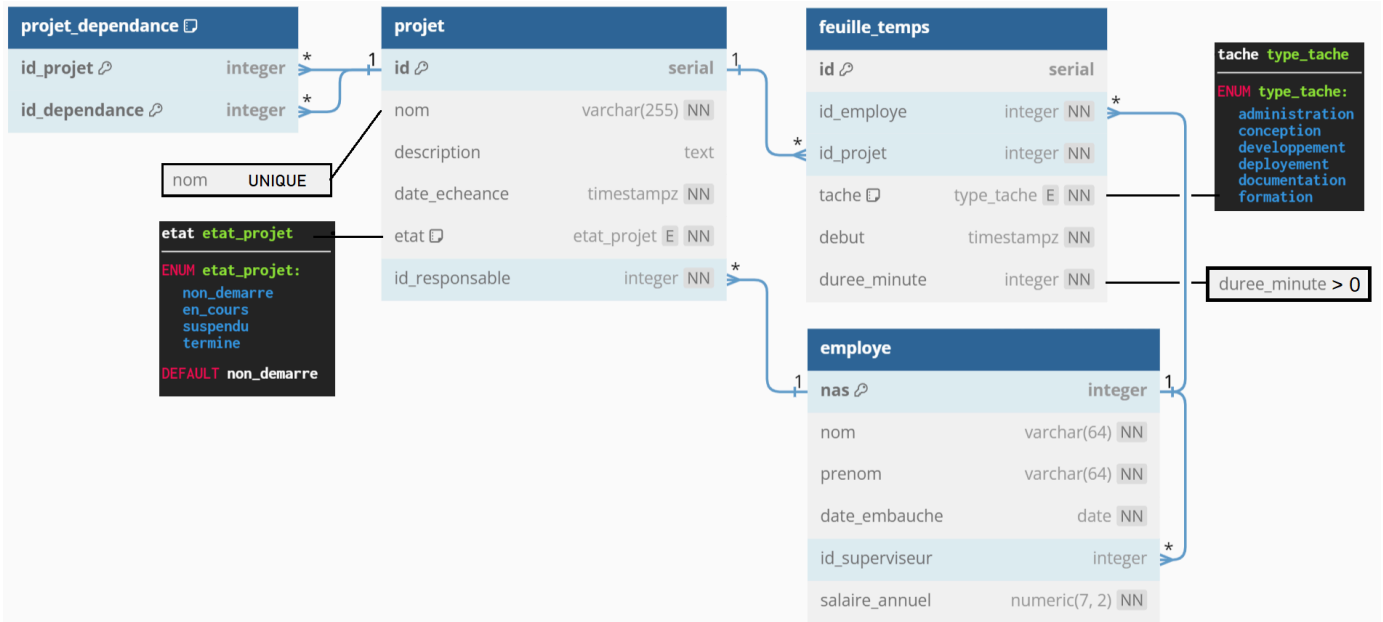
CREATE OR REPLACE FUNCTION verif_date_projet()
RETURNS TRIGGER AS $
DECLARE
    date_debut timestampz
BEGIN
    SELECT date_echeance INTO date_debut
    FROM projet
    WHERE New.id_projet = id

    IF date_debut < New.debut
        RAISE EXCEPTION 'la date de début doit être antérieures à la date d'échéance du projet concerné.'
    END IF;
    RETURN New;
END;
$ LANGUAGE plpgsql;
```

## Question 7

Terminé

Noté sur 15,00



Supposons qu'il existe un projet nommé *abc\_xyz*, rédiger la requête SQL permettant d'afficher les trois colonnes suivantes :

- nom du projet
  - le nombre total d'heure passé sur le projet par tous les employés en heures
  - le coût total du projet pour la main d'oeuvre pour chaque employé. Vous devez considérer son salaire appliqué aux heures qu'il a passé sur ce projet.
- Il y a 1950 d'heures de travail par année.**

```
SELECT p.nom, (SUM(ft.duree_minute) / 60) as "Nombre d heures",
--1 Version du calcul main d'oeuvre
SUM((emp.salaire_annuel / 1950) * (ft.duree_minute) / 60))
--2 Version du calcul main d'oeuvre
--(SUM(emp.salaire_annuel / 1950) * (SUM(ft.duree_minute) / 60))
FROM projet
JOIN feuille_temps ft ON ft.id_projet = p.id
JOIN employe emp ON emp.nas = ft.id_employe
WHERE p.nom = 'abc_xyz'
GROUP BY p.nom
```