

GIS 2A 3^{ème} année

Devoir surveillé bases de données - documents papiers autorisés

© Polytech Lille - Olivier Caron

Durée 2 heures

1 Allocation de Bureaux (7 points)

Une grande entreprise souhaite avoir un système d'information pour optimiser l'affectation d'employés aux différents bureaux qu'elle dispose. Cette entreprise est structurée en services caractérisés par un nom. Les services gèrent des employés et se voit attribuer des bureaux. Les employés sont caractérisés par leurs nom et prénom et possèdent un statut. Le statut est caractérisé par un intitulé et une superficie minimum requise de travail (nombre de m²). Les employés sont affectés dans un bureau de leur service. Un bureau est identifié par un numéro et se caractérise par sa superficie (nombre de m²), le nombre d'employés qu'il peut accueillir et les différents statuts d'employés compatibles pour ce bureau. Grâce à ce système d'information, il sera possible de facilement savoir si un employé est affecté à un bureau ou pas, si le bureau est compatible avec le statut de l'employé, si la surface est suffisante, etc

Question 1 : Proposez un schéma conceptuel (4 points) décrit dans le formalisme UML qui permet de gérer l'ensemble de ces informations. Comme l'objectif est de créer une base de données relationnelle, précisez dans votre schéma UML à l'aide du stéréotype "<<Id>>", le ou les attributs UML qui joueront le rôle d'identifiant, ajoutez des attributs si nécessaire.

Question 2 : Écrire un schéma relationnel (3 points) dans le langage SQL conforme à ce schéma conceptuel.

2 Gestion des épidémies (13 points)

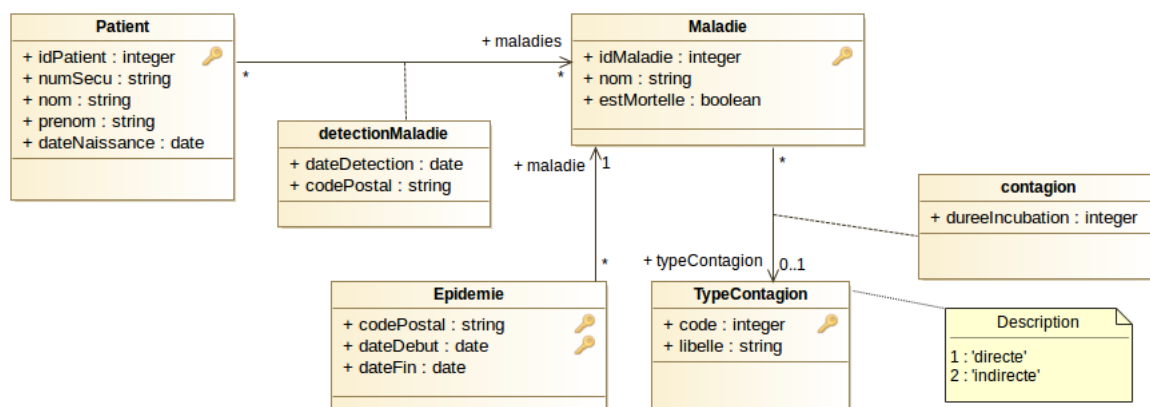


FIGURE 1 – Schéma conceptuel

Des établissements médicaux partagent à distance une même base de données afin de détecter au plus tôt l'apparition d'épidémies de maladies contagieuses. Pour cela, dès qu'une personne tombe malade, l'établissement médical enregistre le patient s'il n'est pas déjà présent dans la base puis enregistre la détection de sa maladie. Les données de la base sont simples (cf schéma conceptuel de la figure 1) :

- Toutes les maladies sont déjà référencées dans la table *Maladie*. Une maladie est identifiée par un entier (*idMaladie*) et se caractérise par son nom. On indique également si la maladie est mortelle. Pour les maladies contagieuses (champ *typeContagion* différent de NULL), on précise son type de contagion (contagion directe ou indirecte) ainsi que la durée d'incubation (en nombre de jours).

- Les patients sont référencés dans la table `Patient`, on précise leur numéro de sécurité sociale, leur prénom, nom et date de naissance. Lors de l'insertion d'une détection de maladie d'un patient, on précise la date de détection ainsi que le code postal de la ville où s'est déclarée la maladie.
- Enfin, une table `Epidemie` recense toutes les épidémies détectées qu'elles soient en cours (champ `dateFin` égal à `NULL`) ou passées. Une épidémie est identifiée par une maladie, une ville (son code postal) et la date de début de l'épidémie. La détection d'une épidémie se fera grâce aux règles décrites dans la question 4.

Le schéma relationnel correspondant est le suivant :

```
CREATE TABLE Patient (
    idPatient INTEGER PRIMARY KEY,
    numSecu char(15) UNIQUE NOT NULL,
    prenom VARCHAR(30) NOT NULL,
    nom VARCHAR(50) NOT NULL,
    dateNaissance DATE NOT NULL
);
CREATE TABLE TypeContagion (
    code INTEGER PRIMARY KEY,
    libelle CHAR(10) UNIQUE NOT NULL
);
CREATE TABLE Maladie (
    idMaladie INTEGER PRIMARY KEY,
    nom VARCHAR(80) UNIQUE NOT NULL,
    estMortelle BOOLEAN NOT NULL DEFAULT FALSE,
    typeContagion INTEGER REFERENCES TypeContagion,
    dureeIncubation INTEGER
);
CREATE TABLE DetectionMaladie (
    refPatient INTEGER NOT NULL REFERENCES Patient,
    refMaladie INTEGER NOT NULL REFERENCES Maladie,
    dateDetection DATE NOT NULL DEFAULT now(),
    codePostal CHAR(5) NOT NULL,
    PRIMARY KEY(refPatient, refMaladie, dateDetection)
);
CREATE TABLE Epidemie (
    refMaladie INTEGER NOT NULL REFERENCES Maladie,
    codePostal CHAR(5) NOT NULL,
    dateDebut DATE NOT NULL default now(),
    dateFin DATE DEFAULT NULL,
    PRIMARY KEY(refMaladie, codePostal, dateDebut)
);
```

Question 3 : Donnez les requêtes SQL répondant aux questions suivantes (6 points) :

1. La liste des noms de patients et les noms des maladies qu'ils ont contractés.
2. Pour chaque maladie (on affiche le nom) qui a provoqué des épidémies, le nombre d'épidémies provoquées.
3. La liste de toutes les maladies de la base, contagieuses ou pas. On affichera le nom de la maladie, le libellé du type de contagion ('directe' ou 'indirecte') et la durée d'incubation. Pour les maladies non contagieuses, ces deux dernières colonnes auront la valeur nulle.

Question 4 : Écrire la fonction PHP `ajoutDetectionMaladie` (7 points). La signature de cette fonction est la suivante :

```
function ajoutDetectionMaladie($base, $idPatient, $idMaladie, $codePostal)
```

La base de données distante est déjà connectée et est accessible via la variable PHP `$base`. Cette fonction réalise deux points :

- Enregistrement dans la base d'une nouvelle détection d'une maladie donnée (`$idMaladie`) pour un patient donné (`$idPatient`) dans une ville donnée (`$codePostal`). Pour simplifier, on considère que le patient et la maladie existent déjà dans la base. La date de détection sera la date du jour.
- Détection et enregistrement d'une épidémie : dans le cas d'une maladie contagieuse, s'il apparaît que pour la même maladie, on recense plus de 5 maladies détectées à la date du jour dans la même ville alors on enregistre dans la base une nouvelle épidémie à la condition qu'il n'y ait pas déjà une épidémie similaire (même maladie, même ville) **en cours** (`dateFin` égale à `NULL`).

Si une erreur SQL intervient, un message d'erreur est affiché et la fonction s'arrête en retournant le code -1. En fonctionnement normal, cette fonction retourne 1 si une épidémie a été insérée dans la base, 0 sinon.