

Compte rendu INF403

Conception d'une base de données SQLite et intégration avec Python

I. Position du problème :

Nous souhaitons conserver une base de données pour suivre les prix de l'électricité dans une région donnée, afin de fournir une interface aux consommateurs pour le suivi de leurs contrats.

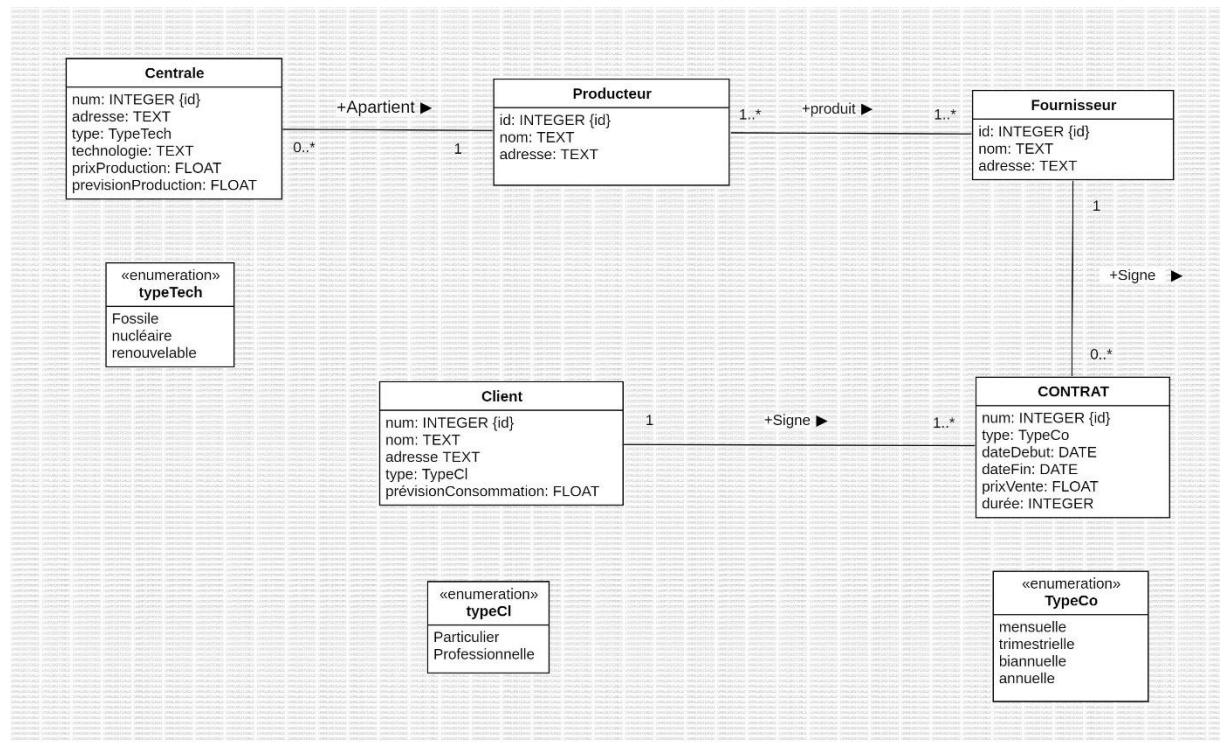
Un client est caractérisé par un numéro client, un nom et une adresse, et peut être un particulier ou une entreprise. Chaque client peut avoir plusieurs contrats avec son fournisseur d'électricité. Ces contrats sont définis par une date de début et une date de fin, ainsi qu'un prix de vente fixe décidé lors de la signature du contrat et une prévision de consommation par mois en KWH.

Les fournisseurs d'électricité sont identifiés par un identifiant unique, un nom et une adresse. Un fournisseur peut acheter son électricité auprès de plusieurs producteurs.

Les producteurs possèdent un identifiant, un nom et une adresse. Le prix de vente de l'énergie est déterminé par le producteur en fonction de la centrale d'où elle provient, et ce prix est fixe pour tous les fournisseurs auxquels il fournit de l'électricité. Les producteurs ont au minimum un fournisseur.

Chaque centrale de production appartient à un seul producteur or un producteur peut en posséder plusieurs. Elles sont caractérisées par un identifiant, une adresse, une date de construction, l'identifiant du producteur, ainsi que la technologie et le type de technologie utilisés pour produire l'électricité et une prévision de production par mois en KWH.

II. Modèle Conceptuelle :



III. Schéma de relations :

Fournisseurs (idFournisseur, nomFournisseur, adresseFournisseur)

/<i>i, n, a> ∈ Fournisseurs ⇔ un fournisseur de nom n possède un identifiant i et une adresse a*/*

Producteurs (idProducteur, nomProducteur, adresseProducteur)

/<i>i, n, a> ⇔ un producteur de nom n possède un identifiant i une adresse a*/*

Clients (numClient, nomClient, adresseClient, typeClient)

/<i>n, m, a, t ∈ Clients ⇔ un client de nom m a un numéro unique n, une adresse a et un type t*/*

Contrats (numContrat, idFournisseur, idClient, typeContrat, dateDebut, dateFin, prixVente, dureeContrat, previsionConsommation)

/<i>n, f, c, t, d, i, j, p> ∈ Contrats ⇔ le contrat de numéro n, a comme fournisseur f, client c, type de contrat t, date de début d et date de fin de contrat i d'une durée j calculable en mois avec une prévision de consommation p par mois en Kwh*/*

Centrales (numCentrale, idProducteur, adresseCentrale,
typeTechnologie, technologie, prixProduction, previsionProduction)
/<n, p, a, t, h, r, v> ∈ Centrales ⇔ une centrale de numéro n, a pour
producteur p, à l'adresse a, qui utilise une technologie r de type h avec un prix de
production r au kWh avec une prévision de production v par mois en Kwh*/*

Productions (idProducteur, idFournisseur)
/<p, f> ∈ Productions ⇔ le producteur p produit de l'énergie pour le
fournisseur f*/*

- Domaines associés :

Domaine(dateDebut) = domaine(dateFin) = Date

*Domaine(idFournisseur) = domaine(idProducteur) = domaine (numClient) =
domaine(numContrat) = domaine(numCentrale) = entier*

*Domaine(adresseFournisseur) = domaine(adresseProducteur) =
domaine(adresseCentrale) = domaine(adresseClient) =
domaine(nomFournisseur) = domaine(nomProducteur) = domaine(nomClient)
= domaine(typeClient) = domaine(typeContrat) = chaine de caractères*
Domaine(prixVente) = domaine(prixProduction) = réel

Domaine (typeContrat) ∈ {mensuelle, trimestrielle, biannuelle, annuelle}

Domaine (typeTechnologie) ∈ {fossile, nucléaire, renouvelable}

Domaine(typeClient) ∈ {particulier, professionnelle}

- Contraintes d'intégrités référentielles :

Contrats[idFournisseur] ⊆ Fournisseurs[idFournisseur]

Contrats[idClient] ⊆ Clients[idClient]

Centrales[idProducteur] ⊆ Producteurs[idProducteur]

Productions[idProducteur] = Producteurs[idProducteur]

Productions[idFournisseur] = Fournisseurs[idFournisseur]