

Exercice 1

La SA Bulot utilise la base de données **GestSalar** pour gérer son personnel. Elle vous communique le schéma relationnel relatif à la base de données :

Etablissement(id_etab, ville_etab)

Employe(id_emp, nom_emp, qualif_emp, date_embauch_emp, id_etab, id_poste)

Poste (id_poste, libelle_poste, id_cat)

Categorie(id_cat, libelle_cat)

1. Un établissement se situe-t-il dans une et une seule ville ? Justifie ta réponse.

Réponse

Puisque id_etab est la clé primaire de **Etablissement**, chacun d'entre eux ne peut apparaître qu'une fois dans cette table, avec par conséquent une unique valeur de ville_etab : un établissement ne peut se situer dans deux villes différentes.

2. Dans les faits, chaque employé occupe un poste de travail précis. Le schéma relationnel traduit-il bien ce fait ? Justifie ta réponse.

Réponse

Chaque employé apparaît une fois et une seule dans la relation **Employe** car id_emp est la clé primaire de cette relation. Ainsi chacun d'eux a un unique id_poste et donc un unique poste.

3. On voudrait ajouter une relation **Salaire**, qui permettrait d'intégrer les salaires de chaque employés **pour chaque mois** de la manière suivante :

- le salaire de base;
- les primes pour le mois;

Écrire la relation **Salaire**.

Réponse

Salaire(id_emp, annee, mois, base, prime)

La clé primaire est le triplet (id_emp, annee, mois) : on ne peut guère faire autrement pour stocker les informations de chaque employé, mois par mois.

id_emp est une clé étrangère, faisant référence à la valeur id_emp de **Employe**.

Exercice 2

Son professeur a demandé à Jules Le Men, étudiant en Licence d'Informatique, de produire un script SQL pour créer une base de données permettant de stocker des informations sur un étudiant et les diplômes qu'il prépare.

Voici ce que Jules a implémenté :

Code SQL

```
DROP TABLE IF EXISTS Etudiant;
DROP TABLE IF EXISTS Diplome;

CREATE TABLE Diplome
(
    id_dip INTEGER PRIMARY KEY,
    nom_dip TEXT
);

CREATE TABLE Etudiant(
    nom_etu TEXT PRIMARY KEY,
    date_naiss_etu TEXT,
    diplome_etu TEXT REFERENCES Diplome(id_dip)
);

INSERT INTO Diplome VALUES
    (1, "Licence d'Informatique"),
    (2, "BTS SIO");

INSERT INTO Etudiant VALUES
    ("Le Men", "2000-07-28",1),
    ("Desloges", "2002-11-20",2);
```

1. Explique pourquoi selon ce modèle, Antoine Le Men, le frère de Jules, ne peut pas préparer de diplôme.

Réponse

Malheureusement la clé primaire de **Etudiant** est le nom de l'étudiant, ce qui interdit à deux étudiants ayant le même patronyme d'être en même temps dans la BDD.

2. Explique pourquoi Killian Desloges ne peut pas s'inscrire pour préparer une licence de philosophie en plus de son cursus actuel.

Réponse

Telle que conçue, un étudiant, identifié de manière unique par son nom, ne peut être associé qu'à une valeur de `diplome_etu`, et donc ne peut préparer qu'un seul diplôme.

3. Reprend le schéma relationnel décrit par le SQL et modifie-le pour que les 2 problèmes précédents soient réglés. Tu dois donc écrire un modèle relationnel (pas besoin d'indiquer les types), pas un script SQL.

Réponse

Diplome(id_dip, nom_dip)

Etudiant(id_etu, nom_etu, date_naiss_etu)

Prepare(id_etu, id_dip)