

EXAMEN BLANC DE BASES DE DONNEES :

On considère la Base de Données suivante :

ASSURE	(num_assu, nom_assu, adr_assu)
CONTRAT	(code_contrat, num_assu, code_bien, intitulé, date_début, date_fin)
EMPLOYE	(matricule, nom_employé)
GESTION	(matricule, code_contrat, annee_gestion)
BIEN	(code_bien, libellé, type, valeur)

L'utilisateur exécute la requête suivante :

Caractéristique des contrats concernant des biens de type Véhicule gérés par l'employé Hochet en 1998.

Une caractéristique comprend : intitulé, date début et fin du contrat, nom de l'assuré et libellé des biens.

QUESTIONS :

1. Construire l'arbre algébrique optimisé correspondant à la requête ?

2. Quels sont les index intéressants pour l'optimiseur pour exécuter cet arbre ? Justifier.

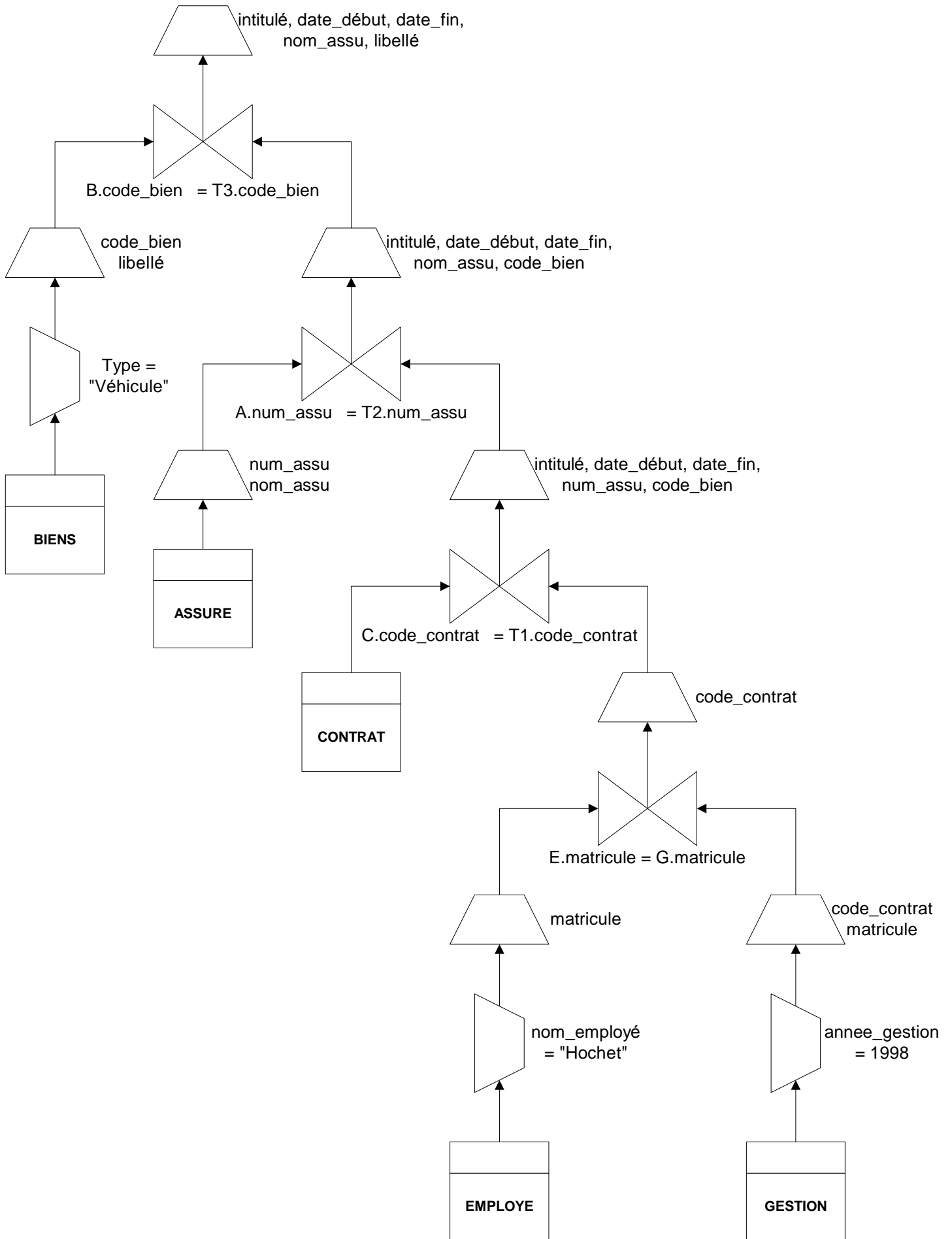
3. On définit les contraintes d'intégrité suivantes :

- a. La valeur d'un bien doit être comprise entre 1 000 € et 1 000 000 €.
- b. Tous les biens sont de type : Véhicule, Immobilier ou foncier.
- c. Un contrat commence toujours avant de se terminer.

Ecrire le schéma SQL des relations concernées par ces contraintes d'intégrité en implémentant les contraintes a, b et c de clés primaires et référentielle.

4. Optimiser le schéma de tel sorte que la requête n'utilise que la relation Gestion \Leftrightarrow Contrat, et construire l'arbre algébrique optimisé correspondant à la requête à partir du schéma optimisé.

5. L'optimisation de la question 4 peut être gérée de manière transparente à l'aide de triggers. Ecrire les triggers correspondants.



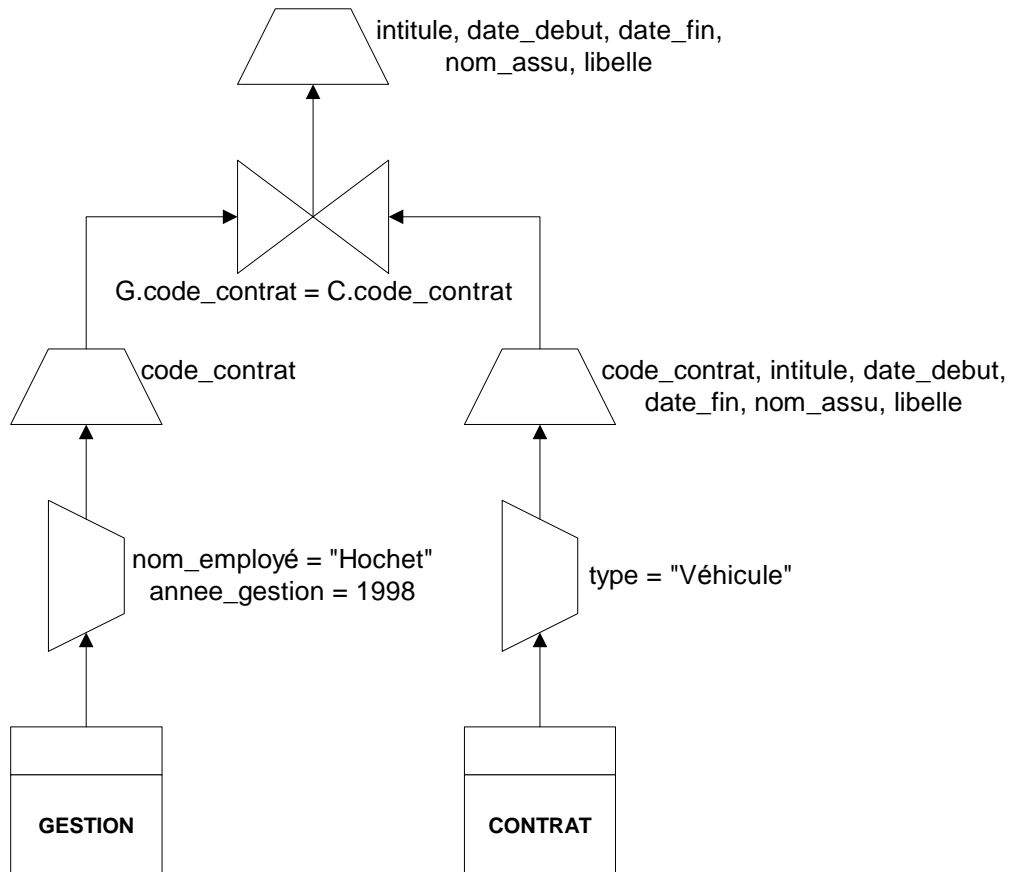
Index intéressants	Accélérer la sélection	Accélérer la jointure
BIEN	type	code_bien
EMPLOYE	nom_employé	matricule
GESTION	annee_gestion	code_contrat code_bien
CONTRAT		code_contrat num_assu code_bien
ASSURE		num_assu

```
CREATE TABLE CONTRAT (
  code_contrat INT NOT NULL,
  num_assu INT NOT NULL,
  code_bien INT NOT NULL,
  intitule CHAR (50),
  date_debut DATE,
  date_fin DATE,
  CHECK date_debut < date_fin,
  PRIMARY KEY code_contrat,
  FOREIGN KEY num_assu REFERENCES ASSURE,
  FOREIGN KEY code_bien REFERENCES BIEN )
```

```
CREATE TABLE BIEN (
  code_bien INT NOT NULL,
  libelle CHAR (25),
  type CHAR (10),
  valeur INT,
  CHECK type ( "Véhicule","Immobilier", "Foncier" ),
  CHECK valeur BETWEEN ( 1 000 AND 1 000 000 )
  PRIMARY KEY code_bien )
```

GESTION (matricule, code_contrat, annee_gestion, *nom_employe*)

CONTRAT (code_contrat, num_assu, code_bien, intitule, date_debut, date_fin, *nom_assu*, *type*, *libelle*)



5.

a. Quand on crée un contrat ou quand on modifie un assuré ou le bien d'un contrat. Il faut récupérer le nom de l'assuré, le type et le libellé du bien, avant l'insertion ou la modification du t-uple dans contrat.

```
CREATE TRIGGER premier
  BEFORE INSERT OR UPDATE OF num_assu, code_bien ON contrat
  FOR EACH ROW
  BEGIN
    SELECT nom_assu INTO :new.nom_assu FROM assure WHERE num_assu=:new.num_assu
    SELECT libelle INTO :new.libelle FROM bien WHERE code_bien=:new.code_bien
    SELECT type INTO :new.type FROM bien WHERE code_bien=:new.code_bien
  END
```

b. Quand on modifie le nom d'un assuré, il faut récupérer cette modification dans tous les t-uples contrat de cet assuré.

```
CREATE TRIGGER second
  AFTER UPDATE OF nom_assure ON assure
  FOR EACH ROW
  BEGIN
    UPDATE contrat SET nom_assu=:new.nom_assu WHERE num_assu=:new.num_assu
  END
```

c. Quand on modifie le libellé ou le type d'un bien, il faut récupérer cette modification dans tous les t-uples contrat de ce bien.

```
CREATE TRIGGER troisieme
  AFTER UPDATE OF libelle, type ON bien
  FOR EACH ROW
  BEGIN
    UPDATE contrat SET libelle=:new.libelle, type=:new.type WHERE code_bien=:new.code_bien
  END
```

d. Quand on crée un t-uple dans GESTION, il faut récupérer le nom de l'employé avant l'insertion du t-uple dans GESTION.

```
CREATE TRIGGER quatrieme
  BEFORE INSERT ON gestion
  FOR EACH ROW
  BEGIN
    SELECT nom_employe INTO :new.nom_employe FROM employe WHERE matricule=:new.matricule
  END
```