Donne les requêtes d’insertion permettant de remplir la base de données dont le modèle relationnel est le suivant :

**Employe = (Matricule *INT***, nom\_emp *VARCHAR(50)*, prenom\_emp *VARCHAR(50)*, fonction\_emp *VARCHAR(50)*, service\_emp *VARCHAR(50), #Matricule\_supervise***);**

**Molecule = (code\_mol *INT***, Nom\_scientifique *VARCHAR(50)*, nom\_commercial *VARCHAR(50)***);**

**projet\_recherche = (code\_projet\_recherche *INT***, titre\_projet *VARCHAR(50)*, responsable *VARCHAR(50), #code\_mol***);**

**chercheur = (id\_chercheur *INT***, nom\_chercheur *VARCHAR(50)*, prenom\_chercheur *VARCHAR(50), #code\_projet\_recherche***);**

**partenaire = (nom\_partenaire *VARCHAR(50)***, type\_partenaire *VARCHAR(50)*, duree\_partenariat *VARCHAR(50)***);**

**essai\_clinique = (code\_essai *INT***, phase\_clinique *INT*, date\_debut *DATE*, date\_fin *DATE, #code\_projet\_recherche***);**

**patient = (id\_patient *INT***, date\_naissance *DATE*, sexe *VARCHAR(1)*, nom\_patient *VARCHAR(50)*, prenom\_patient *VARCHAR(50), #code\_essai***);**

**medecin\_investigateur = (id\_medecin *INT***, nom\_medecin *VARCHAR(50)*, centre\_hospitalier *VARCHAR(50), #id\_patient***);**

**medicament = (num\_AMM *INT***, date\_AMM *DATE*, nom\_marque *VARCHAR(50)*, forme\_galenique *VARCHAR(50)***);**

**lot\_production = (num\_lot *INT***, date\_fabrication *DATE*, date\_peremption *DATE***);**

**site\_fabrication = (code\_site *INT***, localisation *VARCHAR(50)***);**

**pays = (nom\_pays *VARCHAR(50)***, ventes *INT***);**

**rapport\_scientifique = (code\_rapport\_scient *INT***, date\_rapport\_scient *DATE*, resultats\_essai *VARCHAR(50), #code\_essai***);**

**investissement\_R\_D = (*#code\_projet\_recherche*, code\_projet\_R\_D *VARCHAR(50)***, budget\_alloué *DECIMAL(15,2)*, budget\_consommé *DECIMAL(15,2)***);**

**rapport\_réglementaire = (*#num\_AMM*, code\_rapport\_regle *INT***, date\_rapport\_regle *DATE***);**

**brevet = (num\_brevet *INT***, date\_de\_depot *DATE*, date\_dexpiration *DATE, #num\_AMM, #code\_mol***);**

**collaborer = (*#Matricule, #code\_projet\_recherche, #nom\_partenaire*);**

**production = (*#num\_AMM, #num\_lot, #code\_site*);**

**est\_distribué = (*#num\_AMM, #nom\_pays*);**

Les clés primaires correspondent aux id, sauf si autre chose est précisé (quand c'est un attribut composé) les clés étrangères sont identifiées par les #, et ont le même nom que les clés primaires auxquelles elles font référence.

Les clés étrangères doivent faire référence aux clés primaires existantes : donne les lignes en commençant par remplir les tables dans lesquelles il n'y a pas de clés étrangères, puis les tables dans lesquelles les clés étrangères font références à des clés primaires des tables déjà remplies.

Il doit y avoir : 30 lignes pour chacune des tables ne contenant pas de cle etrangere et 20 lignes pour chacune des tables qui en contiennent.

J’aimerais que tu t’inpires de cas réels avec de vrais noms par exemple de molecules et médicaments.  
  
Voici un résumé clair et structuré des contraintes de validation que je voudrais avoir:

* **chk\_employe\_supervise (Employe)**  
  Condition : Matricule different de Matricule\_supervise
* **chk\_titre\_projet (projet\_recherche)**  
  Condition : trim(titre\_projet) <> ''
* **chk\_nom\_chercheur (chercheur)**  
  Condition : trim(nom\_chercheur) <> ''
* **chk\_type\_partenaire (partenaire)**  
  Condition : type\_partenaire IN ('Université','Entreprise','ONG','Autre')
* **chk\_phase (essai\_clinique)**  
  Condition : phase\_clinique BETWEEN 1 AND 4
* **chk\_dates\_essai (essai\_clinique)**  
  Condition : date\_debut < date\_fin
* **chk\_sexe (patient)**  
  Condition : sexe IN ('M','F','O')
* **chk\_lot\_dates (lot\_production)**  
  Condition : date\_peremption > date\_fabrication
* **chk\_ventes (pays)**  
  Condition : ventes >= 0
* **chk\_budget\_nonneg (investissement\_R\_D)**  
  Condition : budget\_alloué >= 0 AND budget\_consommé >= 0
* **chk\_budget\_consomme (investissement\_R\_D)**  
  Condition : budget\_consommé <= budget\_alloué
* **chk\_brevet\_dates (brevet)**  
  Condition : date\_de\_depot < date\_dexpiration

Sur la base de ceci, propose un script d’insertion sql prêt a être execute