Proposition de correction de l'examen du cours *Systèmes d'information*

Fabien Coelho – MINES ParisTech

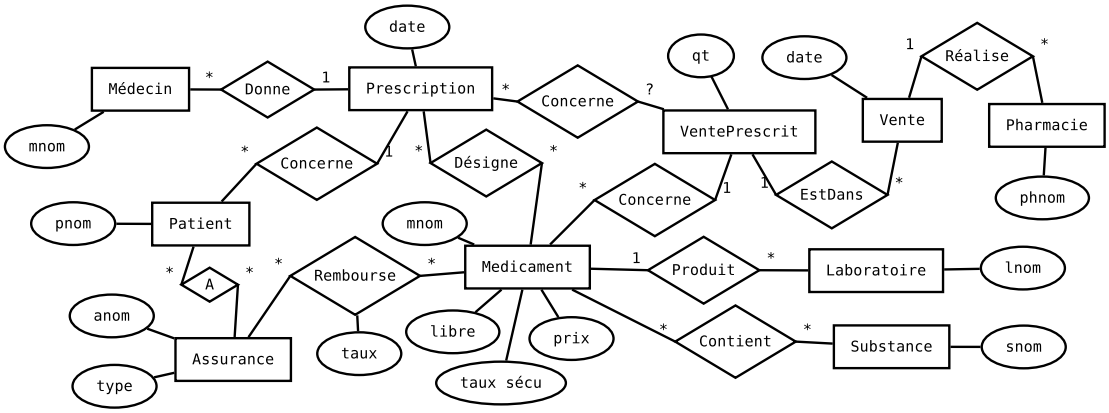
Juin 2018

Bravo Hobbes ! 20/20

1 Modélisation entité-association *6*/6

Des laboratoires produisent des médicaments qui sont vendus dans des pharmacies. Les médicaments sont prescrits aux patients par des médecins. Les médicaments prescrits et achetés peuvent être remboursés à différents taux par la sécurité sociale et éventuellement une mutuelle ou une assurance privée des patients. Les médicaments sont en vente libre ou uniquement sur ordonnance. Un médicament contient des substances.

Voici une proposition de modèle E/A pour cette situation :



Une prescription correspond à une ordonnance. Une vente peut concerner plusieurs ordonnances, ou aucune pour des médicaments en vente libre. Dans le cas d'une prescription, le médicament vendu doit être celui prescrit. On ne suit pas ici le détail des remboursements, juste le principe du taux de remboursement par un organisme. En réalité, la couverture dépend du contrat. Le type *d'un organisme d'assurance santé peut être sécu, mutuelle ou assurance privée.*

2 Traduction relationnelle *4*/4

À partir du modèle E/A précédent, construisez un modèle relationnel. Vous prendrez soin de bien préciser les champs utiles et les contraintes pertinentes sur vos relations. Vous commenterez les contraintes que vous ne pourriez exprimer directement dans le modèle.

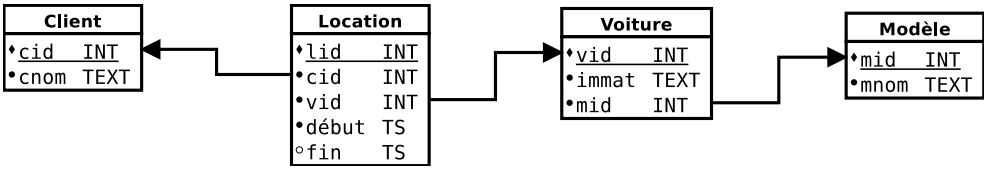
```
CT Médecin(mid SPK, mnom TUNN, ...);
CT Pharmacie(phid SPK, phnom TUNN, ...);
```

```
CT Assurance(aid SPK, anom TUNN, mutuelle BNN,  ...);
CT Patient(pid SPK, pnom TUNN, ...);
CT PatientAssuré(paid SPK, pid INNR Patient, aid INNR Assurance,
                 début DATE, fin DATE, U(pid,aid,...));
CT Laboratoire(lid SPK, lnom TUNN, ...);
CT Substance(sid SPK, snom TUNN, ...);
CT Médoc(oid SPK, mnom TUNN, libre BNN, prix MNN,
       taux_sécu FNN, lid INNR Laboratoire,
       CHECK(taux_sécu BETWEEN 0.0 AND 1.0));
CT PatientAssuré(paid SPK, pid INNR Patient, aid INNR Assurance, U(pid,aid));
CT Rembourse(rid SPK, aid INNR Assurance, oid INNR Médoc, tx FNN, U(aid,oid));
CT Prescription(prid SPK, date DNN, mid INNR Médecin, pid INNR Patient,
       U(date,mid,pid)?);
CT PrescriptionMédoc(poid SPK, prid INNR Prescription, oid INNR Médoc,
       U(prid,oid), posologie ..., ...);
CT Vente(vid SPK, phid INNR Pharmacie, date DNN, U(?) );
CT VentePrescrit(vpid SPK, vid INNR Vente, prid IR Prescription, oid INNR Médoc,
        qt INN, U(vid,pid,oid));
```

L'unicité de certaines relations n'est pas évidente avec les attributs présents.

3 Requêtes *6*/6

On considère le modèle relationnel suivant, qui représente des clients qui louent des voitures de différents modèles (les attributs soulignés sont les clefs primaires, les points noirs désignent des attributs `NOT NULL`, les flèches sont des clefs étrangères) :



1. Quels clients ont loué des voitures de modèle *Clio* en *janvier 2018* ? (La location commence au cours du mois de janvier).

```
SELECT DISTINCT c.cnom
FROM Client AS c
JOIN Location AS l USING (cid)
JOIN Voiture AS v USING (vid)
JOIN Modele AS m USING (mid)
WHERE m.mnom = 'Clio'
  AND l.debut BETWEEN '2018-01-01' AND '2018-01-31';
```

   Pour cette requête uniquement, suggérez des index (hors clefs primaires simples) potentiellement utiles pour en améliorer les performances.

```
-- modèle -> voiture -> location -> client
CREATE INDEX modele_nom ON Modele(mnom);
CREATE INDEX voiture_mid ON Voiture(mid);
CREATE INDEX location_vid ON Location(vid);
-- location -> ...
CREATE INDEX location_debut ON Location(debut);
```

2. Quelle sont les trois plus longues durées cumulées de location des voitures du parc ? (On calculera la durée cumulée de location de chaque voiture).

```sql
SELECT v.immat, SUM(COALESCE(l.fin, NOW())-l.debut) AS duree
FROM Location AS l
JOIN Voiture AS v USING (vid)
WHERE fin IS NOT NULL
GROUP BY v.immat
ORDER BY 2 DESC
LIMIT 3;
```

3. Quelles voitures n'ont pas du tout été louées ?

```sql
SELECT v.immat
FROM Voiture AS v
EXCEPT
SELECT v.immat
FROM Voiture AS v
JOIN Location AS l USING (vid);
```

```sql
SELECT v.immat
FROM Voiture AS v
LEFT JOIN Location AS l USING (vid)
WHERE l.lid IS NULL;
```

4. Quelles voitures sont louées en même temps à deux clients différents ? (C'est louche...)

```sql
-- question pas bien posée : douteux si même client.
-- cherche ici les locations d'une même voiture en même temps
SELECT DISTINCT v.immat, l1.*, l2.*
FROM Voiture AS v
JOIN Location AS l1 USING (vid)
JOIN Location AS l2 USING (vid)
WHERE l1.lid < l2.lid -- évite les symétries
  AND (l1.debut, COALESCE(l1.fin, NOW())) OVERLAPS
      (l2.debut, COALESCE(l2.fin, NOW()));
```

5. Trouver les plus longues locations suivies (les locations s'enchaînent directement, la fin de la précédente est le début de la suivante) d'une même voiture à un même client.

```sql
WITH RECURSIVE
  LocationsSuivies(cid, vid, debut, fin) AS (
    -- locations individuelles
    SELECT l.cid, l.vid, l.debut, l.fin
    FROM Location AS l
    UNION
    -- chaînage direct
    SELECT ls.cid, ls.vid, ls.debut, l.fin
    FROM LocationsSuivies AS ls
    JOIN Location AS l ON (ls.fin=l.debut AND ls.cid=l.cid AND ls.vid=l.vid)
  )
  -- on oublie pas les locations pas encore finies
  SELECT c.cnom, v.immat, MAX(COALESCE(ls.fin, NOW()) - ls.debut) AS duree
  FROM LocationsSuivies AS ls
  JOIN Client AS c USING (cid)
  JOIN Voiture AS v USING (vid)
  GROUP BY c.cnom, v.immat
  ORDER BY duree DESC;
```

4 Questions de cours *4/4*

Choisissez un thème parmi les deux tirés aléatoirement en début d'examen dans la liste *Postgres, Relationnel, Transaction, MVCC, Optimisation, Droits, PL/pgSQL, Injection SQL, Décisionnel, SIG, Systèmes distribués*, et expliquez en moins de 100 mots ce que vous en avez retenu.

***Thème** : ACID (Advanced Chanting In Databases)*

Ce thème est très passionnant, j'ai appris plein de choses super intéressantes qui éclairent parfaitement le fonctionnement de cet aspect des bases de données à la fois théorique et pratique, et me permettent de bien comprendre les caractéristiques générales et particulières des multiples facettes de ce thème dans une perspective transversale de mise en application concrète du modèle relationnel sur des problèmes réels, tout en gardant, au delà du simple niveau fonctionnel, une maîtrise généraliste des aspects opérationnels sur Postgres qui gère le stockage à un prix compatible avec toutes les bourses, une bonne nouvelle pour nos budgets !

Citez les noms de trois scientifiques ayant obtenu le prix Turing pour leurs travaux de recherche sur les bases de données.

Il y a 4 prix Turing liés aux recherches sur les bases de données :

1. *Charles William (Charlie) Bachman (USA 1924-2017) – 1973*
2. *Edgard Franck (Ted) Codd (UK 1923-2003) – 1981*
3. *James Nicholas (Jim) Gray (USA 1944-2007/2012) – 1998*
4. *Michael Ralph Stonebraker (USA 1943-) – 2014*

Document composé avec LaTeX, révision 3854.