

Devoir Maison : “The Running Boutique”

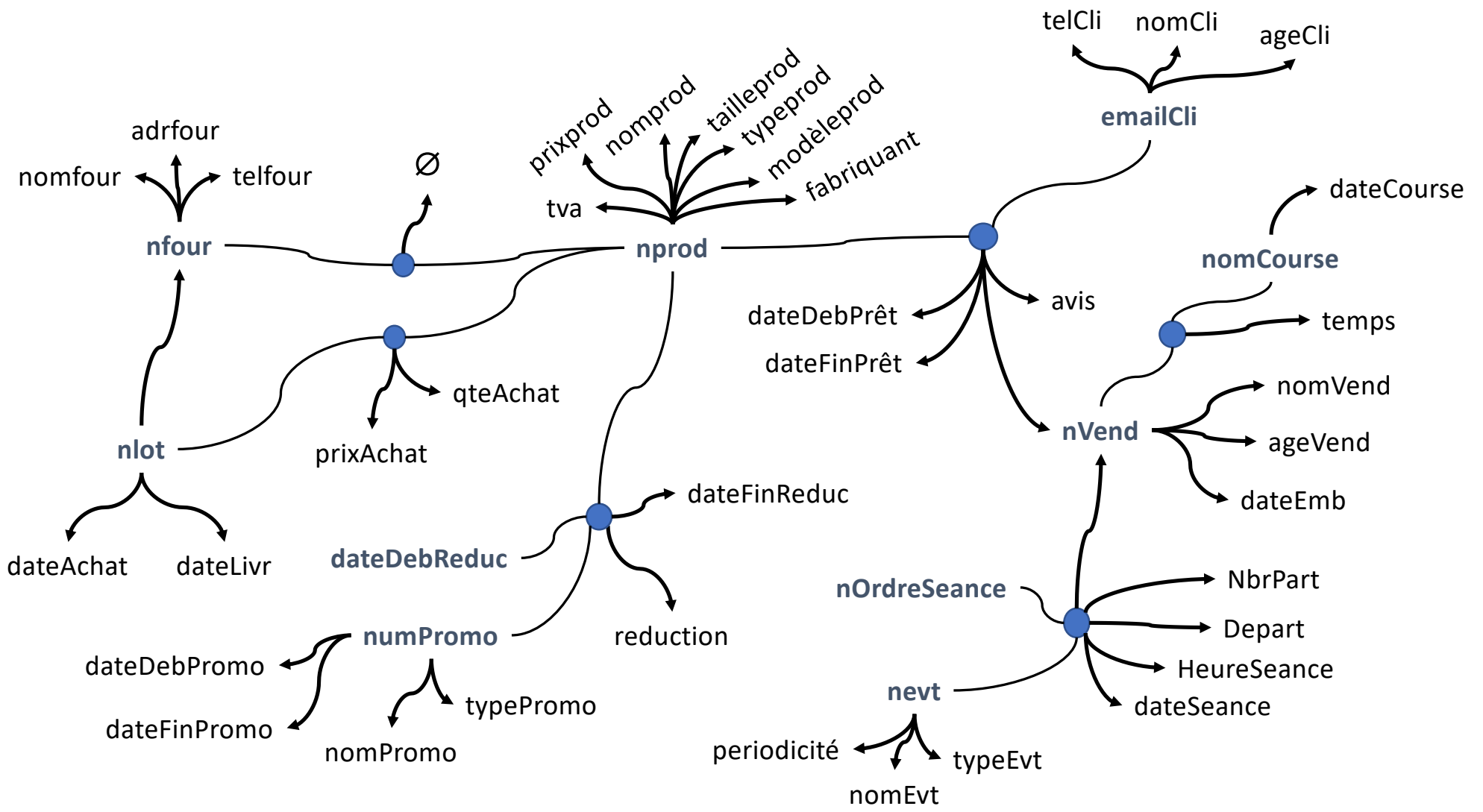
La boutique « The Running Boutique » est un magasin spécialisé dans la course à pieds. Les propriétaires souhaitent réorganiser leur base de données afin mieux gérer les périodes de promotion et les services proposés à la clientèle (tests de matériel, organisation d'événements...). Dans cette base de données, nous devons ainsi enregistrer les informations concernant les produits (essentiellement des chaussures). Chaque produit est identifié par un numéro unique (nprod). Il a un nom, un prix de vente HT, une valeur de TVA, un type, une taille, un modèle et un fabricant. Les produits sont achetés par lots. Chaque lot est identifié par un numéro de lot (nlot), correspondant à un achat de plusieurs produits réalisé auprès d'un même fournisseur. On doit connaître la date de l'achat, celle de livraison, mais également le fournisseur, lui-même identifié par un numéro unique (nfour). En outre, nous devons connaître le nom de ce fournisseur, son adresse et téléphone de contact. Un lot peut contenir plusieurs produits, qui sont achetés à une certaine quantité et à un prix d'achat donné. Par exemple, dans le lot n°1201, nous avons acheté 50 paires de Nike Vaporfly taille 38, et 25 paires d'Asics GT-2000 taille 37. Par ailleurs, il est important de souligner qu'un même produit peut être disponible chez plusieurs fournisseurs. On doit conserver la liste de fournisseurs proposant chaque produit.

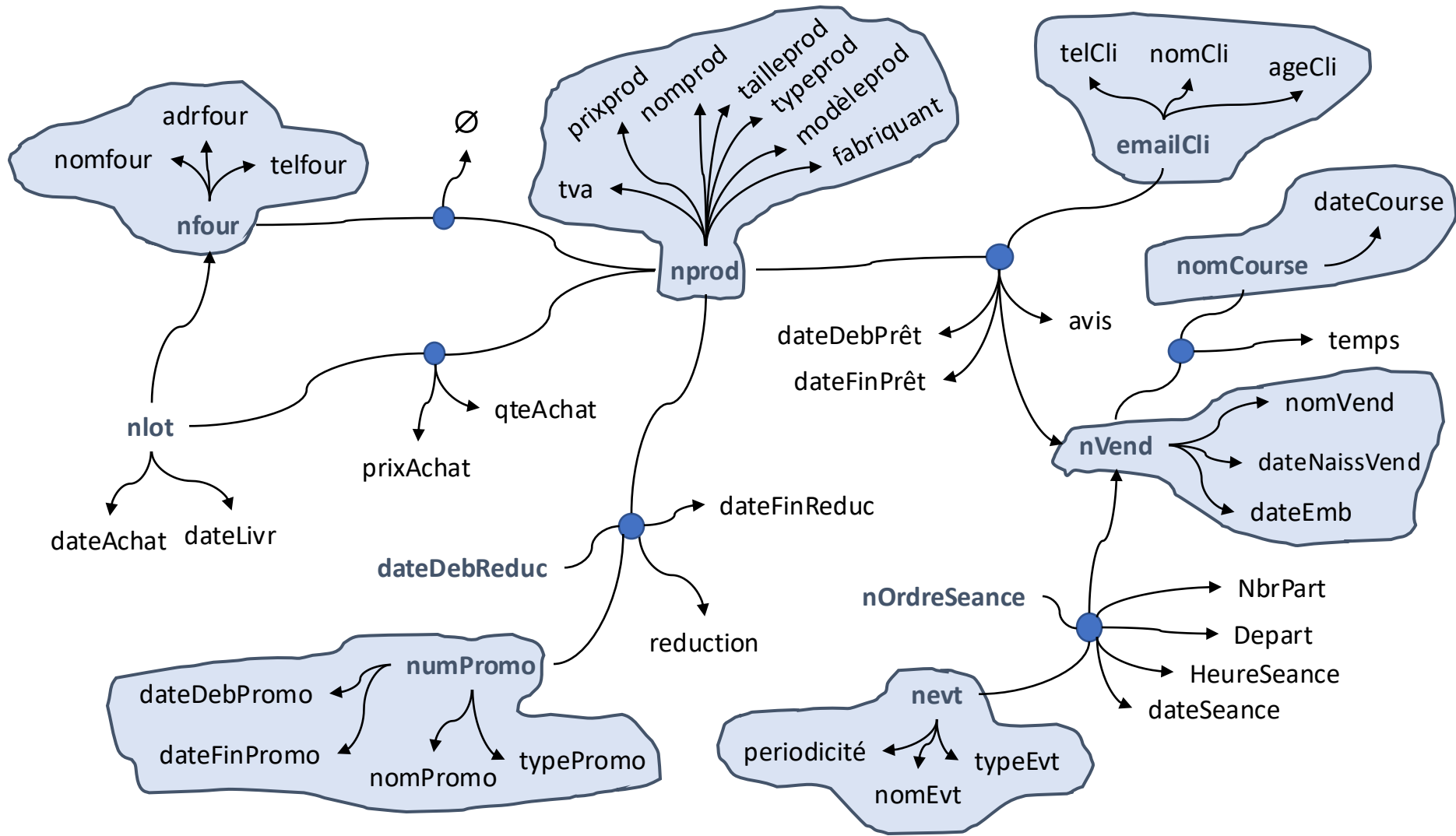
Les promotions sont identifiées par un numéro unique (numPromo). Chacune va avoir un nom et une période associée, en plus de son type (soldes, black Friday...). Les produits peuvent avoir différentes réductions pendant une promotion. Il est important de connaître l'évolution de ces réductions pendant une promotion. Par exemple, pendant le Black Friday 2019, qui a eu lieu du 29/11/2019 au 01/12/2019, les chaussures Nike Vaporfly ont eu 25% de réduction du 29 au 30/11/2019, puis 50% à partir du 01/12/2019.

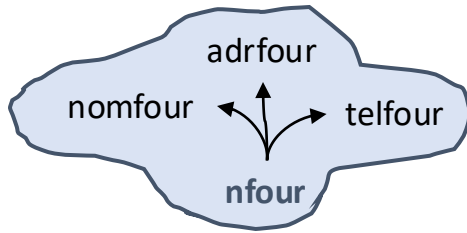
La boutique organise des nombreux événements (sorties running, entraînements collectifs...). Chaque événement possède un nom, un type (sortie, cours...) et une périodicité (ponctuel, hebdomadaire, mensuel...). Ainsi, un même événement peut avoir plusieurs séances, dont il faut connaître la date et l'heure de début, mais aussi le point de départ et le nombre de participants attendus. Attention, un même événement peut avoir plusieurs séances en parallèle. Par exemple, l'événement « running dimanche » a eu deux séances le 19/01/2019 à 9h30. Le départ de la 1^{ère} séance a été réalisé au Jardin de Luxembourg pour 10 participants, alors que celui de la 2^{ème} séance a eu lieu au Champs de Mars pour 20 participants. Pour des raisons de logistique (notamment du nombre de participants attendus), on pourrait avoir même plusieurs séances partant en parallèle d'un même point de départ (par exemple, une 3^{ème} séance de « running dimanche » au 19/01/2019 9h30 est possible en partant du Jardin de Luxembourg). Chacune de ces séances est accompagnée par un vendeur. Chaque vendeur est identifié par son numéro, et on doit connaître son nom, son âge ainsi que sa date d'embauche. S'il (ou elle) participe à des courses (autres que les événements organisés par la boutique). On aimerait connaître, pour chacune de ses courses (identifiées uniquement par leur nom), la date à laquelle elle a été réalisée et le temps obtenu par le vendeur.

Enfin, la boutique organise le prêt de matériel. Ces prêts permettent aux clients de tester un produit pendant une période. Le client récupère le produit qu'il souhaite tester auprès d'un vendeur à une date donnée et doit le rendre avant une certaine date. Pour chaque prêt, le client s'engage à donner un avis sur le produit testé. Cet avis doit être enregistré dans la base de données avec les informations sur le prêt. On doit enregistrer ces informations dans la base de données, sans pour autant utiliser un numéro de prêt, jugé trop lourd par l'équipe. Autre limitation demandée par l'équipe : chaque client ne peut tester un produit qu'une seule fois, et on doit absolument connaître le vendeur qui a effectué le prêt. Concernant les clients, peu d'informations sont jugées nécessaires : nom, âge, téléphone et email uniquement.

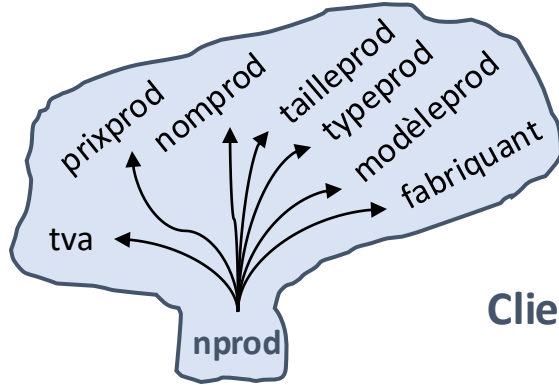
- 1) Proposez le graphe des dépendances fonctionnelles élémentaires et directes. (14 points)
- 2) Définissez le schéma de la base de données en 3^{ème} forme normale. (4 points)
- 3) Donnez la commande SQL nécessaire pour créer la table contenant les informations sur les prêts de matériel. (1 point)
- 4) Indiquez la commande SQL nécessaire pour insérer dans la table que vous venez de créer le prêt d'un Nike Vaporfly taille 38, réalisé le 5/11/2020, au client « Jean.Dupont@dupont.fr » par le vendeur « Toto ». (1 point)



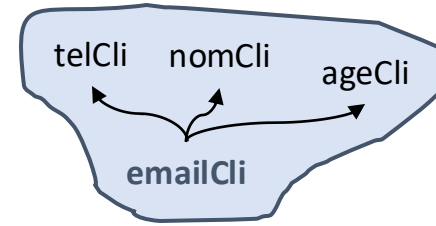




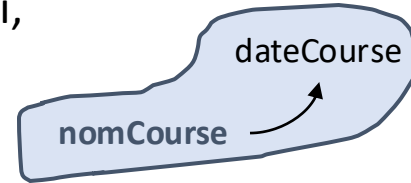
Fournisseur (nfour, nomFour, telFour, adrFour)



Produit (nprod, nomprod, typeprod, prixProd, modèleprod, fabricant, tva)



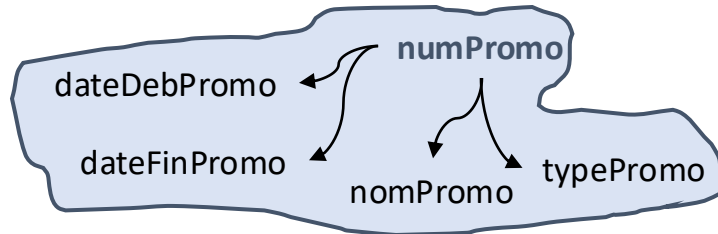
Client (emailcli, nomcli, telcli, agecli)



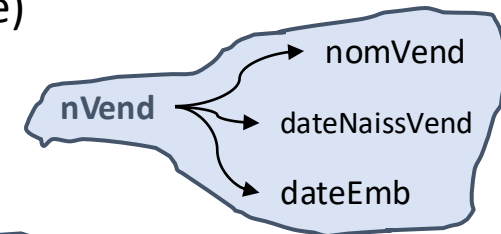
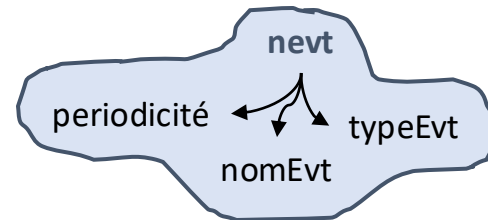
Course (nomCourse, dateCourse)

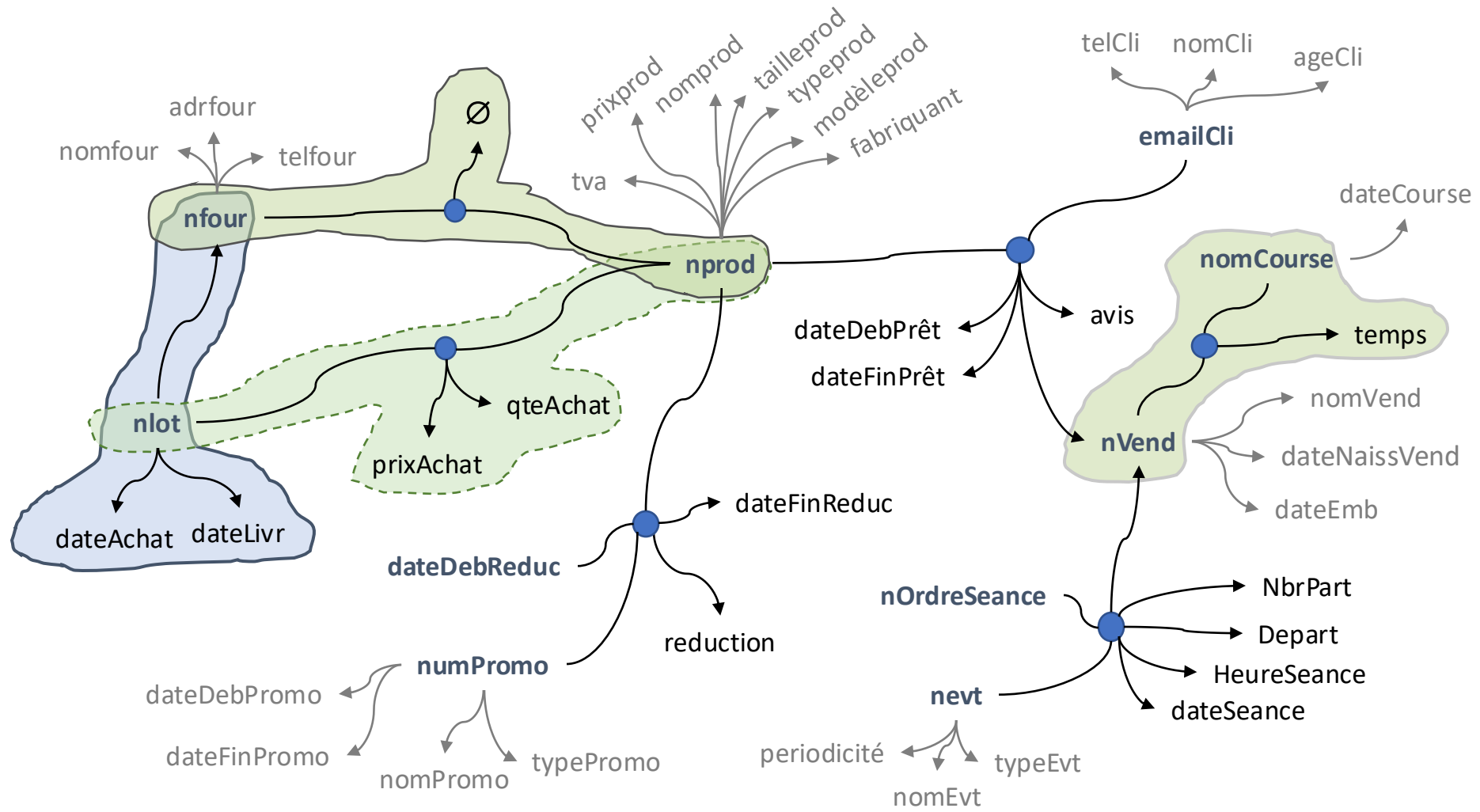
Vendeur (nvend, nomVend, dateEmb, dateNaissVend)

Promotion (numPromo, nomPromo, typePromo, dateDebPromo, dateFinPromo)

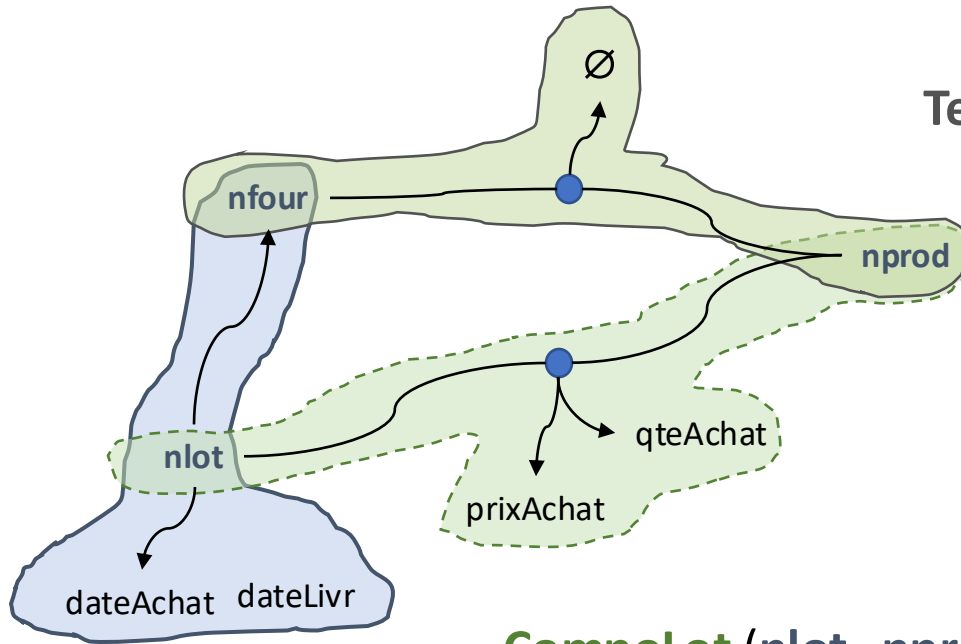


Evénement (nevt, nomEvt, typeEvt, périodicité)

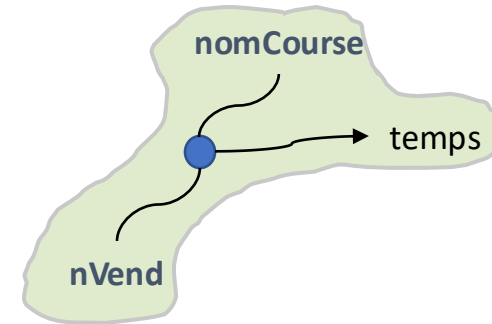




CatalogueFournisseur (nfour, nprod)

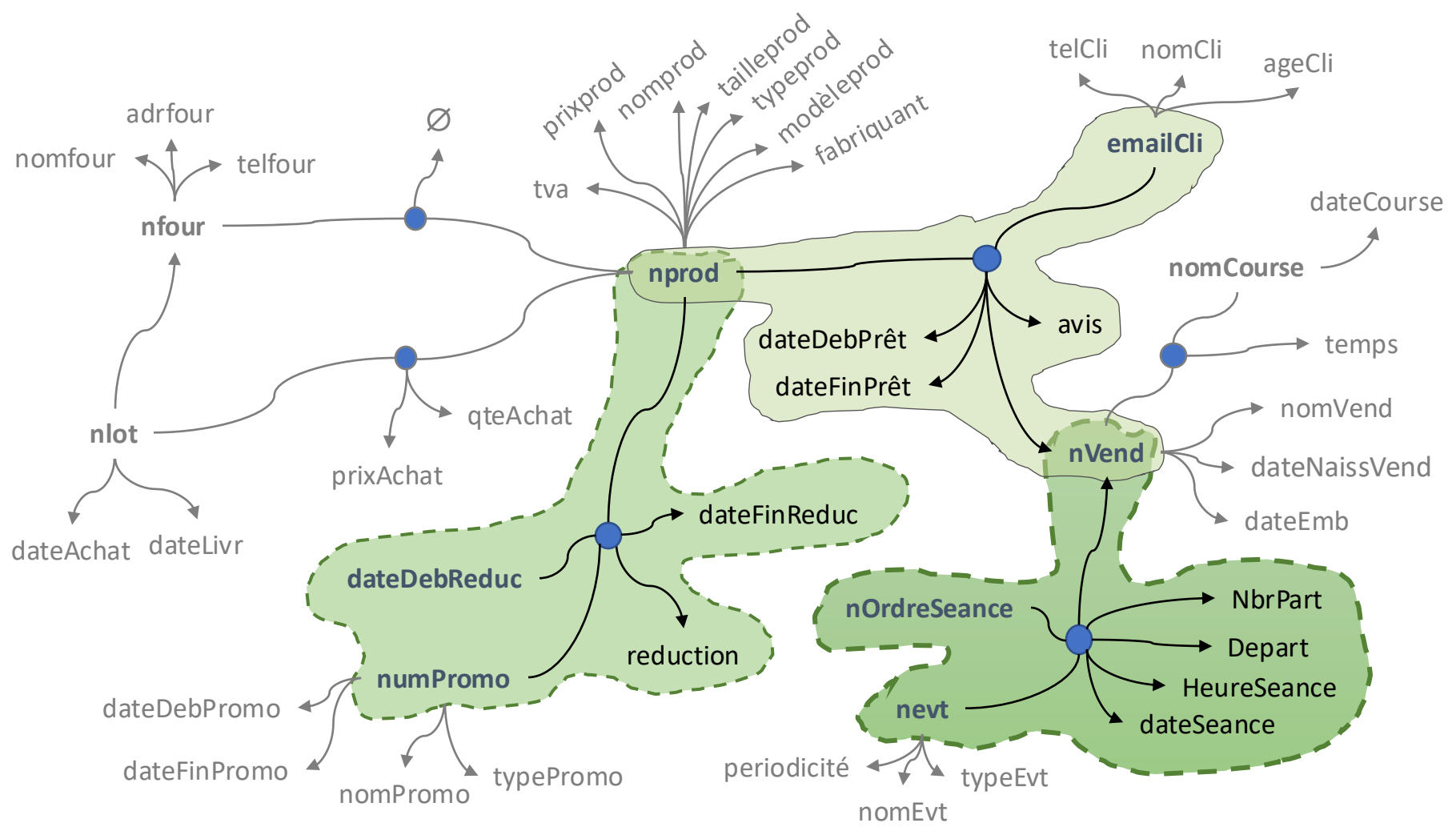


TempsCourse (nvend, nomCourse, temps)

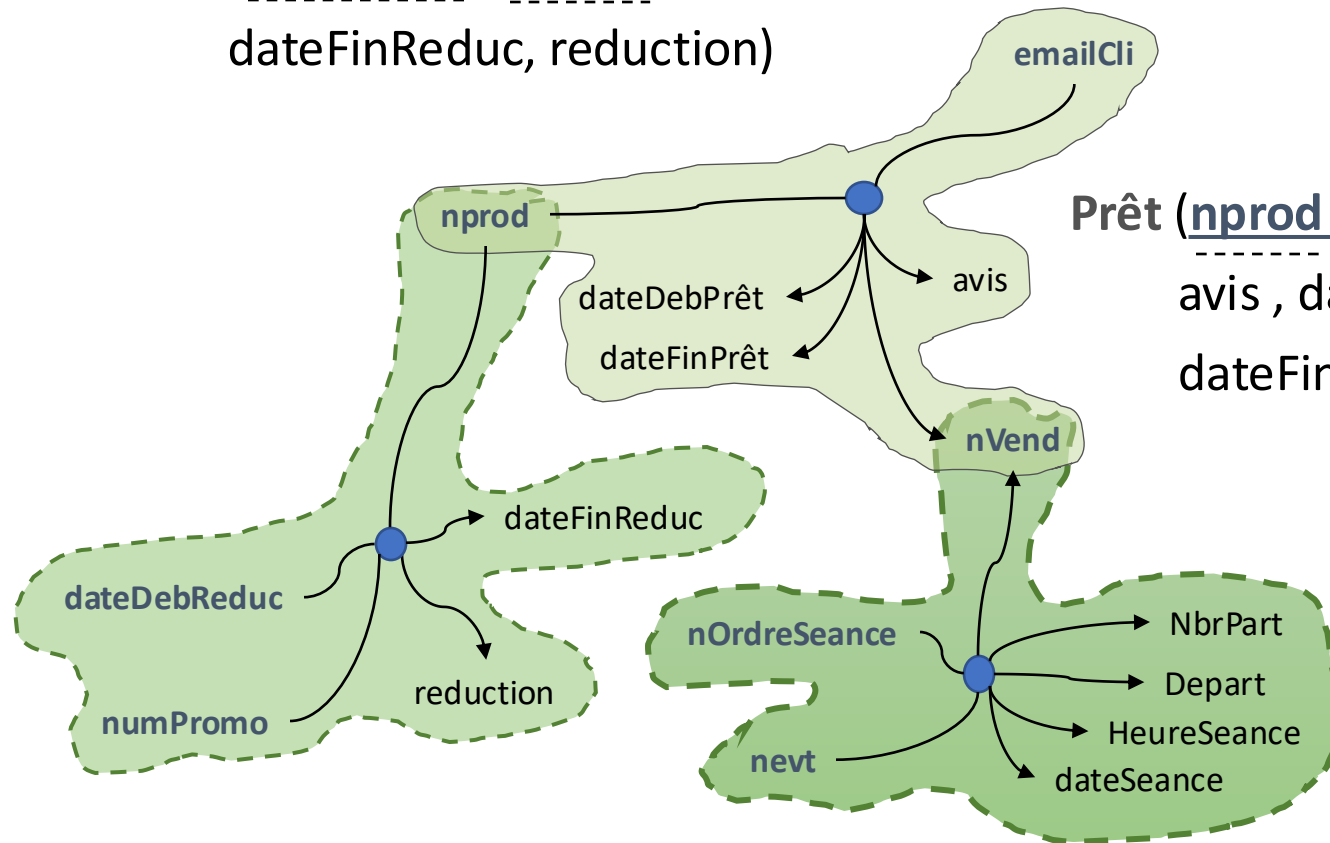


CompoLot (nlot, nprod, qteAchat, prixAchat)

Lot (nlot, dateAchat, dateLivr, nfour)



Réduction (numPromo, nprod, dateDebReduc,
dateFinReduc, reduction)



Prêt (nprod, emailCli, nvend
avis, dateDebPrêt,
dateFinPrêt)

Séances (nevt, nOrdreSeance, dateSeance, heureSeance,
départ, nbrPart, nvend)

Révision : Running Boutique

- Produit (nprod, nomprod, typeprod, modèleprod, fabricant, prixProd, tva)
- Fournisseur (nfour, nomFour, telFour, adrFour)
- Lots (nlot, dateAchat, dateLivr, nfour[#])
- Client (emailcli, nomcli, telcli, agecli)
- Courses (nomCourse, dateCourse)
- Événement (nevt, nomEvt, typeEvt, périodicité)
- Vendeur (nvend, nomVend, dateNaissVend, dateEmb)
- CatalogueFournisseur (nfour[#], nprod[#])
- TempsCourse (nvend[#], nomCourse[#], temps)
- Prêt (nprod[#], emailCli[#], nvend[#], dateDebPrêt, dateFinPrêt, avis)
- CompoLot (nlot[#], nprod[#], qteAchat, prixAchat)
- Réduction (numPromo[#], nprod[#], dateDebReduc, dateFinReduc, reduction)
- Séances (nevt[#], nOrdreSeance, dateSeance, heureSeance, départ, nbrPart, nvend[#])
- Promotion (numPromo, nomPromo, typePromo, dateDebPromo, dateFinPromo)

CREATE TABLE

Prêt (nprod, emailCli, nvend#, avis,
dateDebPrêt, dateFinPrêt)

```
CREATE TABLE Prêt (  
  nprod INTEGER REFERENCES Produits,  
  emailCli VARCHAR(255)  
                REFERENCES Clients,  
  nvend INTEGER REFERENCES Vendeur,  
  dateDebPrêt DATE,  
  dateFinPrêt DATE,  
  avis VARCHAR (1024),  
  PRIMARY KEY (nprod, emailcli)  
);
```

```
CREATE TABLE Prêt (  
  nprod INTEGER ,  
  emailCli VARCHAR(255) ,  
  dateDebPrêt DATE ,  
  dateFinPrêt DATE ,  
  avis VARCHAR (1024) ,  
  nvend INTEGER ,  
  PRIMARY KEY (nprod, emailcli) ,  
  FOREIGN KEY (nprod) REFERENCES Produits ,  
  FOREIGN KEY (emailcli) REFERENCES Clients ,  
  FOREIGN KEY (nvend) REFERENCES Vendeur  
);
```

Attention **PRIMARY KEY** :
Invariablement à la fin, car **clé**
composée de plusieurs attributs.

NOT NULL possible pour **nvend** et **dateDebPrêt**.

INSERT INTO

Prêt (nprod, emailCli, nvend#, avis , dateDebPrêt, dateFinPrêt)

```
INSERT INTO Prêt (nprod, emailCli, dateDebPrêt, nvend )  
VALUES (1234 , "jean.dupont@dupont.fr", '5/11/2020' , 345 ) ;
```

Valeurs de clés étrangères
choisies au hasard.

Attention :

Les valeurs doivent correspondre aux types de données et aux contraintes indiquées.

Le **nom du vendeur** (« Toto »), ni celui du **produit** (« Nike Vaportfly ») ne doivent pas apparaître sur le INSERT INTO, uniquement des **valeurs de clés**.