



Base de Données

Cave à vins

Un caviste désire informatiser la gestion de ses bouteilles. Les données contenues dans la table ci-dessous représentent l'information qu'il désire conserver sur la qualité de ses vins.

Région	Appellation	Producteur	Millésime	Qualité
Bourgogne	Château Sapeur	Bert Encam	1994	**
Bourgogne	Château Fenouillard	Cunégonde Lainesse	1994	**
Bordeaux	Château Lagaffe	Gaston Ledormeur	1995	***
Bordeaux	Château Moucherot	Jérôme Chèvre	1995	***
Beaujolais	Château Blancsec	Adèle Matino	1996	*
Beaujolais	Château Hobbes	Calvin Lepetit	1996	*
Bordeaux	Château Lagaffe	Gaston Ledormeur	1996	**
Bordeaux	Château Moucherot	Jérôme Chèvre	1996	**
Beaujolais	Château Blancsec	Adèle Matino	1997	***
Beaujolais	Château Hobbes	Calvin Lepetit	1997	***
Bordeaux	Château Lagaffe	Gaston Ledormeur	1997	***
Bordeaux	Château du Vent	Isabelle Passager	1997	***
Touraine	Château Nickelé	Jack Forton	1997	*
Bordeaux	Château Moucherot	Jérôme Chèvre	1997	***
Touraine	Château Troy	Lignole Charleston	1997	*
Bourgogne	Château Sapeur	Bert Encam	1998	***
Bourgogne	Château Fenouillard	Cunégonde Lainesse	1998	****
Touraine	Château Fog	Jack Forton	1998	**
Touraine	Château Troy	Lignole Charleston	1998	**
Bourgogne	Château Sapeur	Bert Encam	1999	*
Bourgogne	Château Fenouillard	Cunégonde Lainesse	1999	*
Touraine	Château Fog	Jack Forton	1999	*

1. Quel est le schéma de cette table ? Quelle est sa cardinalité ? Quels sont les domaines de valeur des attributs « Région », « Millésime » et « Qualité » ?
2. Comment ces données ont-elles été triées ?
3. Quels liens peut-on identifier entre région, appellation et producteur ? (réponse attendue du type : « il y a un producteur par région » ou au contraire « il a plusieurs producteurs par région », etc.)
4. Quels liens peut-on identifier entre qualité et millésime ?
5. Peut-on éviter des redondances en divisant cette table en deux ?
6. Le couple d'attributs « région » & « appellation » pourrait-il servir de clé pour cette table ? Sinon, quelle clé proposeriez-vous ?

CAVE À VINS

1/ Schéma Vin (Région, Appellat°, Producteur, Millénisme, Qual)

◦ Card: 22

◦ Domaine valeur attribut: Région (4) CHAR(20)

Millésime (6) [1994; 1999] ^{nb entier} INTEGER

Qualité (4): {*, **, ***, ****}
CHAR(4)

2/ Les données sont triées par ordre croissant de millésime puis par ordre croissant prod

3/ Plusieurs appel par Région

2 appellat° par prod

Plusieurs product par Région

4/ Pas de lien

5/ Oui on pourrait normaliser la Bdd.

6/ Appellat° (Appellat°, Région, producteur)

Qualité (Appellat°, Millésime, qualité)

Coût de l'immobilier

Le tableau ci-dessous présente l'argus du logement à Paris (prix à l'achat au mètre carré).

Paris 1er Louvre

St Germain L'auxerois

Halles

S-2P

3P

4P

5P +

S-2P

3P

4P

5P +

Récent

Max

Min

46165

43700

37982

42741

33320

33796

35281

37937

Rénové

Max

Min

39257

37085

32233

36300

29329

29779

30984

33317

Rénové

Max

Min

36600

34541

30022

33823

27794

28235

29331

31540

Rénové

Max

Min

30997

29175

25358

28600

24557

24977

25846

27791

Ancien

Max

Min

28543

26740

23240

26226

23088

23497

24263

26090

Ancien

Max

Min

19168

18102

15733

17720

15638

15422

16987

18265

Paris 2 è Bourse

Gaillon

lettre

Vivienne

S-2P

C+L

3P

4P

5P +

S-2P

3P

4P

5P +

Récent

Max

Min

27463

28416

31407

38673

30524

32535

33878

32264

Récent

Max

Min

24322

15114

27757

34002

26580

28306

29447

28083

Rénové

Max

Min

23113

24001

26353

32206

25063

26679

27799

26475

Rénové

Max

Min

20263

21164

23392

28417

21863

23248

24237

23080

Rénové

Max

Min

19406

19949

22049

26697

20410

21690

22620

21543

Rénové

Max

Min

14649

15064

16650

20012

15188

15943

16729

15933

Quelle est la relation permettant d'enregistrer ce tableau ? Vous définirez pour chaque attribut de cette relation, son domaine de valeurs.

Clés

Dans chacune des deux tables ci-dessous, identifiez l'attribut pouvant servir de clé primaire.

T1

A	B	C	D
1	5	4	4
6	4	2	2
5	8	7	3
5	9	9	3
1	2	2	6

T2

E	F	G	H
4	2	9	2
9	0	4	1
9	3	5	3
8	6	5	7
7	0	2	6

Cherchez dans la table T2 la clé étrangère la liant à T1.

COÛT DE L'IMMOBILIER

Relat° Logement (NumLogement ↙ Arrondi, Quartier, Nb de pièce, Etat, Prix max, Px min)

- Arrondissement $\in [1; 20]$
- Quartier : `CHAR(20)`
- Nb de pièces : $\{5-2P, 3P, 4P, 5P\}$
- Etat {ancien, rénové}
- Prix max : `INTEGER`
- Prix min : `INTEGER`

CLÉS

$T_1(A, \underline{B}, C, D)$

$T_2(E, F, G^*, \underline{H})$

Gestion des fournisseurs

Imaginez que vous utilisez un tableur (comme Excel par exemple) pour gérer la base de données représentée par la table ci-dessous.

NuméroArticle	NomArticle	NomFournisseur	AdresseFournisseur
1	Clou	BHV	Rue de Rivoli
2	Vis	Mr Bricolage	Place d'Italie
3	Boulon	Bricolex	Avenue Ledru Rollin
4	Écrou	BHV	Rue de Rivoli

Indépendance des données :

1. Supprimez les clous et les boulons de votre base de données.

2. Retrouvez l'adresse de Bricolex. $\emptyset$

3. Quelle est la nature du problème ? Comment aurait-il fallu s'y prendre pour l'éviter ?

Cohérence : Faire plusieurs tableaux.

4. Créez le produit (5, Tournevis, bazar de l'Hôtel de Ville, Rivoli). Quel est le problème ?

unicité des saisies : 1 car nulle table Cohérence niv article

5. En repartant de la base de départ, créez cinq articles de votre choix vendus par Bricolex. Combien de fois avez-vous saisi l'adresse de ce fournisseur ? Est-ce normal ? Non, Redondance

6. Comment faudrait-il s'y prendre pour éviter d'avoir à re-saisir le nom et l'adresse du fournisseur au moment de la création d'un produit, lorsque celui-ci est déjà dans la base ?

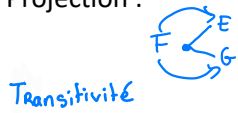
il faudrait faire 2 tables

Fournisseur	Nm Fournisseur	Adresse
1		
2	Bricolex	A.L
3		

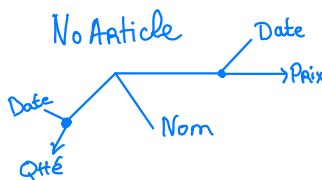
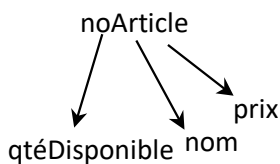
Dépendances fonctionnelles

1. **Propriétés des dépendances fonctionnelles** : remplacez les points d'interrogation de manière à définir les propriétés des dépendances fonctionnelles.

Additivité :	Si $A \rightarrow H$ et $A \rightarrow X$	Augmentation : Si $X \rightarrow T$
	Alors $A \rightarrow H, X$	Alors $\forall A : X, A \rightarrow T$
Projection :	Si $F \rightarrow E, G$	Pseudo transitivité : Si $E \rightarrow F$ et $FG \rightarrow H$
	Alors $F \rightarrow E$ et $F \rightarrow G$	Alors $E, G \rightarrow H$
Transitivité :	Si $Z \rightarrow A$ et $A \rightarrow Q$	
	Alors $Z \rightarrow Q$	

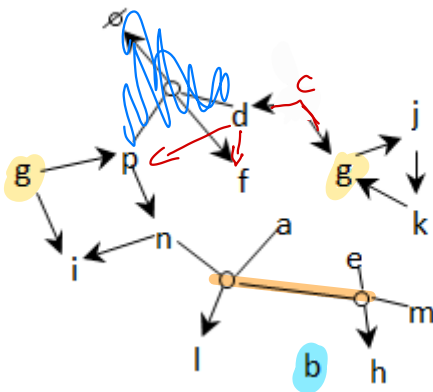


2. **Historisation** : L'extrait de graphe de dépendances fonctionnelles ci-dessous associe à tout article un nom, un prix et une quantité disponible. Définir une représentation historique de la quantité disponible et du prix des articles.

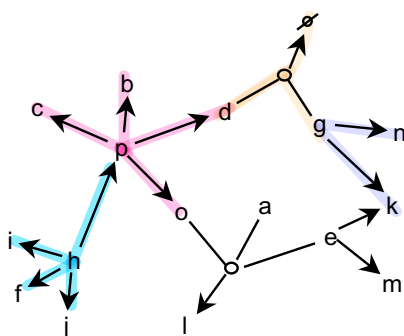


Pas de flèche pr date car c'est un attribut entre 2 date s'ajoute (modifier)

3. **Analyse de graphe de dépendances fonctionnelles** : Critiquez le graphe de dépendances fonctionnelles ci-dessous.



4. **Passage au relationnel** : Déduisez un schéma relationnel à partir du graphe de dépendances fonctionnelles ci-dessous.



- $R_1 (\underline{b}, i, j, \underline{p})$ clé secondaire
- $R_2 (\underline{p}, b, c, d, o)$
- $R_3 (\underline{d}, g)$
- $R_4 (\underline{g}, n, k)$

Les livres d'une bibliothèque

Une table a été définie pour enregistrer les données sur les livres que possède une bibliothèque.

NumLivre	Titre	Auteur	NumAuteur	ISBN	DateAchat	Emp
1030	L'humanité perdue	Finkelkraut A.	124563	2 02 033300 7	14/10/00	F3
1032	Mercure	Nothomb A.	26334	2 253 14911 X	14/10/00	G5
1045	Eva Luna	Allende I.	46A215	2 253 05354 6	22/02/01	F3
1029	L'humanité perdue	Finkelkraut A.	124563	2 02 033300 7	14/10/00	F3
1067	Mercure	Nothomb A.	26334	2 253 14911 X	24/02/01	G5
1022	Les combustibles	Nothomb A.	26334	2 125 04121 V	03/10/00	G6

Selon cette table, chaque livre a un numéro, un titre, un auteur, un code ISBN, une date d'achat et un emplacement dans les rayonnages.

1. Imaginez que vous utilisez un tableur (comme Excel par exemple) pour gérer la base de données.
2. Mettez à jour les noms d'auteurs en indiquant les prénoms de manière extensive selon la procédure suivante :
 - Modifiez le nom de l'auteur du livre 1030 ; l'auteur de « L'humanité perdue » est-il partout Alain Finkelkraut ? *Non*
 - Combien de fois faut-il répéter l'opération de modification du nom de l'auteur du livre « Mercure », et de manière générale ? *Avant de faire qu'il y a des lignes*
 - Lorsque vous modifiez le nom de l'auteur de « Mercure », le résultat est-il répercuté sur l'auteur de « Les combustibles » (c'est le même auteur) ? *Non*
3. Identifiez, à partir des données présentées ci-dessus, les dépendances fonctionnelles permettant de déduire une version de la table Livre normalisée en 3FN.
4. Présentez les tables résultant de cette normalisation avec leur contenu. Comment répondriez-vous à la question 2 après normalisation ?

Gestion du parc automobile

Il s'agit de la gestion du parc automobile d'une société. Voici les attributs qui ont été retenus :

1. Attributs :

numéro voiture	NOV	mot
marque voiture	MV	texte
nombre de kilomètres parcourus	KM	numérique positif
nombre de places passagers	PSG	entier(1,5)
nom chauffeur	CHAUFFEUR	texte
numéro chauffeur	NCH	entier(0,100)
nombre de kilomètres	NKM	numérique positif
numéro réparation	NOREP	entier(0,1000)
type réparation	TYPEREPA	mot
montant réparation	PX	numérique positif de FF
nombre de kilomètres au compteur	KMCPT	numérique positif
date du trajet	DATE	(entier(0, 5000), entier(1,12), entier(1,31))
heure de départ du trajet	HEURE	(entier(0,23), entier(0,59))
ville de départ	VILLEDEP	mot
ville d'arrivée	VILLEARR	mot
numéro trajet	NOTRAJ	entier
nombre de personnes transportées	NBPERSTER	entier
numéro de garage de réparation	NOG	entier(0,1000)
distance en kilomètres	NBKM	numérique positif de km (0, 500)

2. Relations

VOITURE(NOV, MV, KM, PSG)

À une voiture on associe son numéro de voiture NOV, sa marque MV, le nombre de kilomètres qu'elle a parcourus KM, le nombre de places disponibles de passagers PSG.

CH(NCH, CHAUFFEUR)

À un numéro de chauffeur NCH, on associe un seul nom du chauffeur CHAUFFEUR.

V-CH(NOV, NCH, NKM)

Le chauffeur de numéro NCH a conduit la voiture de numéro NOV pendant tant de kilomètres NKM depuis que la voiture est en service.

REPARATION(NOREP, NOV, NOG, TYPEREPA, PX, KMCPT)

La voiture de numéro NOV est menée au garage de numéro NOG pour une réparation de numéro NOREP et de type TYPEREPA ; elle a alors tant de kilomètres au compteur KMCPT. Cette réparation a coûté tant PX.

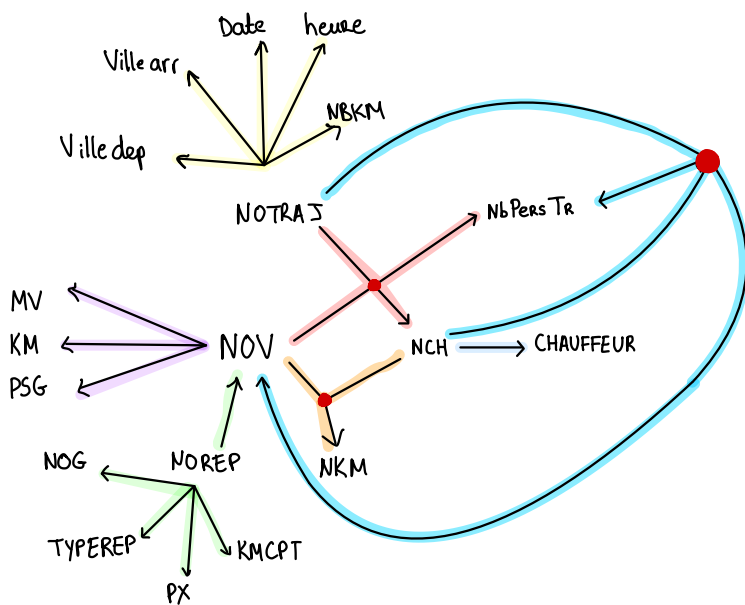
TRAJET(NOTRAJ, VILLEDEP, VILLEARR, DATE, HEURE, NBKM)

Un trajet de numéro NOTRAJ a été effectué à la date DATE, le départ a eu lieu à l'heure HEURE. Les villes de départ et d'arrivée sont respectivement VILLEDEP, VILLEARR ; le trajet est de tant de kilomètres NBKM.

TR-NOV(NOTRAJ, NOV, NCH, NBPERSTR)

La voiture de numéro NOV, conduite par le chauffeur de numéro NCH, a transporté tant de personnes NBPERSTR pour le trajet de numéro NOTRAJ.

- Voiture (Nov, NV, KM, PSG)
- CH (NCH, CHAUFFEUR)
- CH (Nov, NCH, NKM)
- Réparation (NOREP, NOV, NOG, TYPERE, PX, KMCPT)
- Trajet (Notraj, villedep, villearr, date, heure, nbkm)
- TR-NOV (NOTRAJ, NOV, NCH, NBPERS TR)



3. Règles d'intégrité

Nous indiquons ci-après seulement les règles d'intégrité du champ d'application qui s'expriment à l'aide de dépendances fonctionnelles :

NOV → MV	NOREP → NOV	NOTRAJ → VILLEDEP
NOV → KM	NOREP → NOG	NOTRAJ → VILLEARR
NOV → PSG	NOREP → TYPERE	NOTRAJ → DATE
	NOREP → PX	NOTRAJ → HEURE
NCH → CHAUFFEUR	NOREP → KMCPT	NOTRAJ → NBKM
NOV, NCH → NKM	NOTRAJ, NOV → NCH, NBPERSTR	NOTRAJ, NCH → NOV, NBPERSTR

Toutes ces dépendances fonctionnelles sont élémentaires.

Questions :

1. Construire le graphe des dépendances fonctionnelles.

2. Voici une liste d'affirmations concernant le même champ d'application. Vous indiquez si elles sont en accord ou en contradiction avec la modélisation proposée en justifiant votre réponse.

- C'est toujours le même chauffeur qui conduit la même voiture ; *Non pas de lien*
- Lors du trajet, c'est toujours la même voiture que conduit un chauffeur ; *Oui*
- L'organisation confie toutes ses voitures à réparer à un seul garage ; *NON*
- Il existe des trajets dont la distance dépasse mille cinq cents kilomètres ; *NON (0,500)*
- Un cadre de l'entreprise qui n'est pas un chauffeur peut quand même conduire une voiture pour un trajet ; *NON*
- Un trajet est assuré par une seule voiture ; *Oui/ Non*
- Il existe des voitures de plus cinq places pour les passagers. *NON (1,5)*

3. Voici une liste de questions concernant la modélisation. Veuillez-y répondre en justifiant votre réponse.

- Un chauffeur peut-il effectuer plusieurs fois le même trajet ? *NON*
- Deux chauffeurs différents peuvent-ils effectuer le même trajet ? *OUI*
- Est-ce qu'un chauffeur peut effectuer plusieurs trajets avec la même voiture ? *OUI*
- Est-ce qu'un trajet peut être effectué à deux dates différentes ? *NON*
- Un trajet peut-il être effectué par plusieurs voitures ? *OUI*
- Peut-on faire des trajets où les villes de départ et d'arrivée sont les mêmes ? *OUI*

Locations de voitures

On vous demande de modéliser les données utiles à la gestion des locations et des réservations d'une agence de location de voitures. Les locations sont instantanées, mais les réservations sont par anticipation. On réserve une voiture pour une période du futur. La réservation porte sur une catégorie de voitures tandis que la location porte sur une voiture.

Une voiture est caractérisée par son numéro d'immatriculation, sa puissance, sa marque et son nombre de kilomètres. Sa puissance la place dans une catégorie qui détermine le prix de location journalier et le prix au kilomètre. La location est faite pour une période qui peut être étendue. La base doit mémoriser la période prévue, mais aussi la date de retour réelle de la voiture au loueur. Le prix de location dépendant du nombre de kms parcourus, l'information mémorisée pour une voiture louée comporte le nombre de kms au début de la location et à la fin.

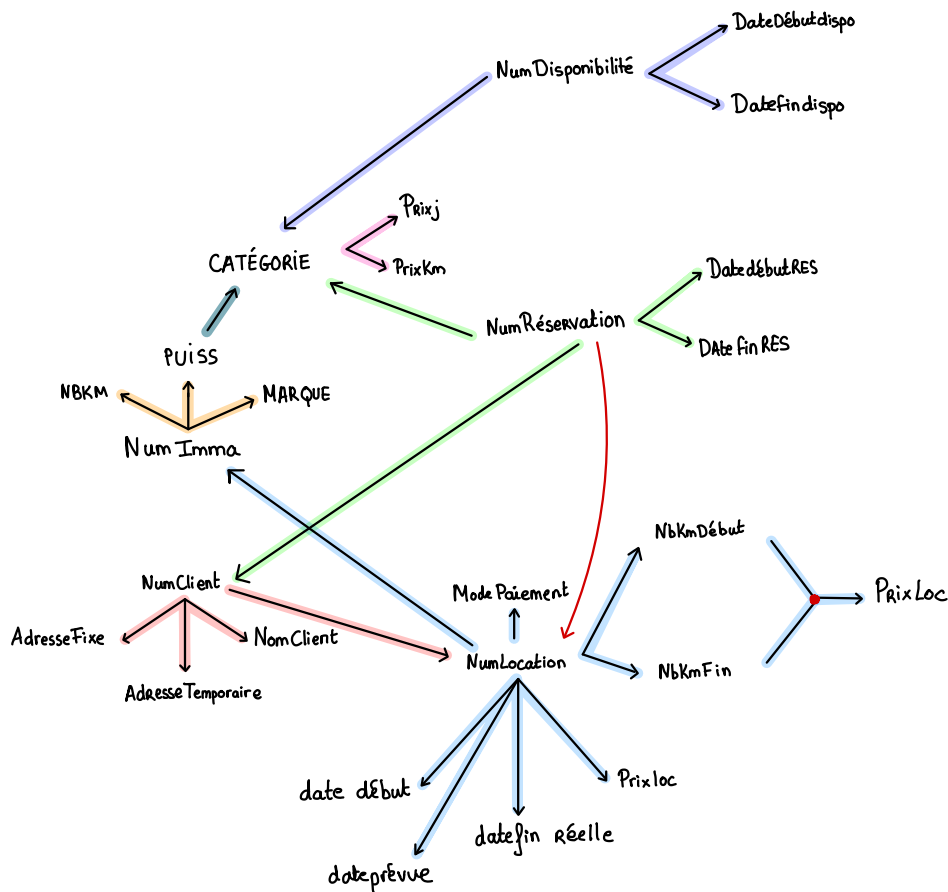
Une réservation porte sur une catégorie de voiture, pour une période donnée. La réservation comme la location sont faites par un client. Un client a un numéro, un nom, une adresse fixe, mais une adresse temporaire pour une location ainsi qu'un mode de paiement propre à la location. Lorsqu'une location est effectuée en réponse à une réservation, la réservation est effacée de la base.

Pour gérer au mieux les réservations du futur, la base de données maintient à jour les disponibilités par catégorie et unité de voiture dans chaque catégorie. Par exemple, l'unité référencée 1 de la catégorie 'haut de gamme' est disponible du 20/02 au 03/03 et du 04/04 au 30/05, etc.

On vous demande :

1. D'identifier les attributs et de leur donner une définition sommaire.
2. De construire le graphe des dépendances fonctionnelles élémentaires et directes.
3. D'en déduire la collection des relations en 3FN et de prouver que vos relations sont bien des 3FN.

Location de Voiture



- Location (numlocation, date début, date fin prévue, date fin réelle, nbkm début, nbkm fin, mode paiement, prix, numclient, numimmat)
- Voiture (num Imma, marque, puissance, nbkm)
- Puissance (puissance, catégorie)
- Catégorie (catégorie, prixj, prix km)
- Réservation (num Réservat, numcli, catégorie, date début res, date fin res)
- Disponibilité (num disponibilité, catégorie, date début dispo, date fin dispo)
- Client (num client, nom client, adresse fixe, adresse temporaire)

Vente de DVDs sur Internet

On vous demande de raisonner sur la structure de la base de données nécessaire à la mise en place d'une 'supply chain' de vente de DVDs sur Internet par une société appelée MSGelectronique.

Les clients accèdent par Internet au site Web de MSGelectronique et font leur commande de DVD. *On ne gère pas ici l'aspect des commandes des clients.*

Le site doit permettre l'accès aux informations sur les DVDs qui doivent être stockés dans la base : code barre du DVD, titre du film associé, date de sortie du film et prix de vente aux consommateurs. Le stock temporaire est géré par DVD de manière instantanée.

Pour alimenter son stock, MSGelectronique dispose d'un panel de distributeurs auquel elle fait appel pour être livrée des DVDs commandés par ses clients. La base doit contenir les informations relatives aux distributeurs (numéro, nom et email) et aux prix d'achat des DVD auprès de ces distributeurs, le même DVD pouvant être au catalogue de plusieurs distributeurs à des prix différents. En outre chaque distributeur a un ou plusieurs transporteur(s) avec qui il travaille et avec qui il a négocié le prix de transport d'un DVD. La base doit contenir tous ces renseignements de façon à ce que MSGelectronique choisisse au mieux le distributeur et le transporteur associé pour servir une commande de client. Chaque transporteur et chaque distributeur est identifié de manière unique et a un nom. Chaque distributeur a une adresse électronique.

MSGelectronique passe des commandes aux distributeurs tous les matins à 7 heures. Une commande a un numéro unique, une date, s'adresse à un distributeur pour un certain nombre de DVDs et mentionne le transporteur sélectionné pour la (ou les) livraison(s). Chaque ligne de commande indique le nombre de DVD ayant un code barre donné. La base doit garder trace des livraisons effectuées par les distributeurs en réponse aux commandes de MSGelectronique. Chaque livraison a un numéro unique, fait référence à une commande et à un ou plusieurs DVDs, livré chacun en une certaine quantité.

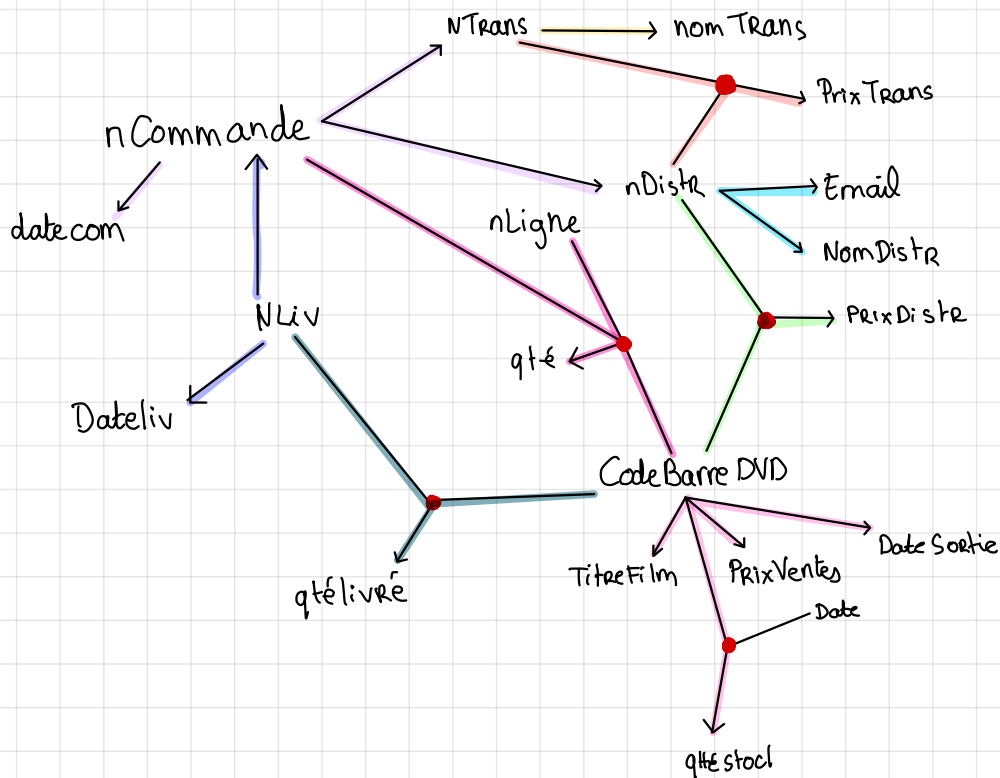
Vous pouvez vous aider de la liste d'attributs ci-dessous :

Ndist : Numéro de distributeur (unique)	Datefilm : Date de sortie du film d'un DVD
Nomdist : Nom de distributeur	Ncommande : Numéro unique d'ordre de livraison de MSGelectronique à un distributeur
Emaildist : Adresse mail de distributeur (unique)	DateComande : date de commande
CodeBarreDVD : Numéro unique de DVD	Nligne : numéro d'une ligne de commande
Prixdist : Prix d'un DVD à l'achat chez un distributeur donné	Qte : Quantité d'un DVD de code barre donné dans une ligne commande
Ntrans : Numéro unique de transporteur	Nliv : Numéro unique de livraison
Nomtrans : Nom de transporteur	Datliv : Date de livraison
Prixvente : Prix d'un DVD au catalogue de MSGelectronique	Qteliv : Quantité d'un DVD livrée dans une livraison donnée
Prixtrans : Prix du transport	Qtstk : quantité d'un DVD en stock
Titrefilm : Titre du film d'un DVD	

On vous demande de :

1. Construire le graphe des dépendances fonctionnelles élémentaires et directes
2. Projeter le graphe en une collection de relations en 3FN
3. Décrire les schémas de relations en SQL

Vente de DVDs sur Internet



- DVD (code barre DVD , titre film , date sortie , qté stock , prix vente)
- Distributeur (ndistr , nom distr , email)
- Prix distributeur (code barre DVD , ndistr , prix distr)
- Transporteur (ntrans , nom trans)
- Prix transporteur (ndistr , ntrans , prix trans)
- Commande (ncom , date com , ndistr , ntrans)
- Ligne de commande (ncom , nligne , code barre DVD , qté)
- Livraison (nliv , date liv , ncom)
- Qté livré (code barre DVD , nliv , qté liv)

Jeudi :
Lutins du FN
Résa hotelière

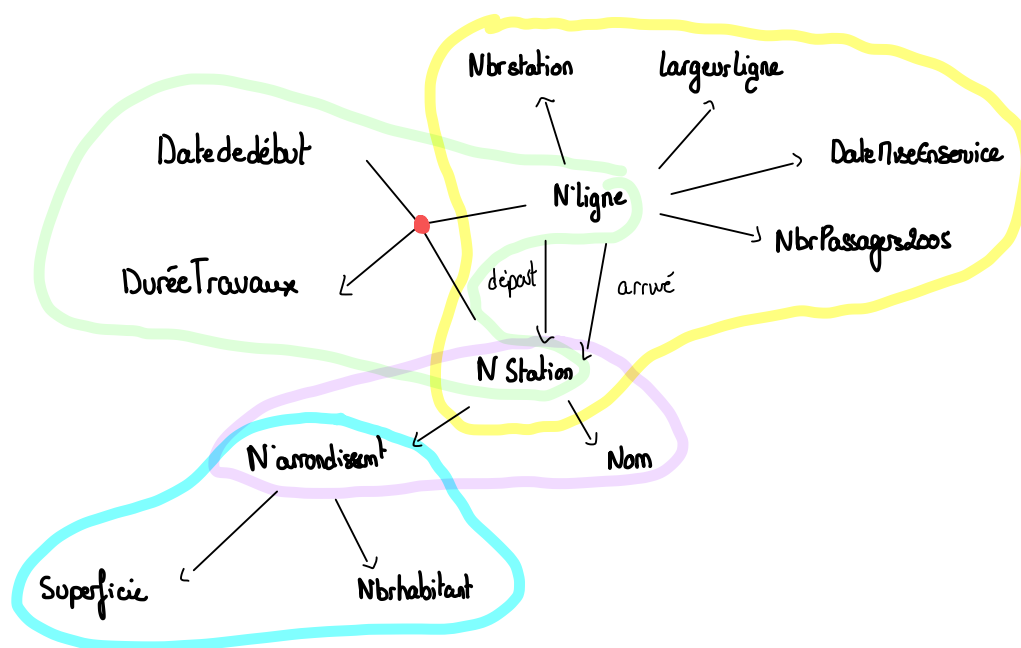
Le métro de Paris

On va mettre en place une base de données donnant les informations relatives aux différentes **lignes** et **stations** du Métro Parisien.

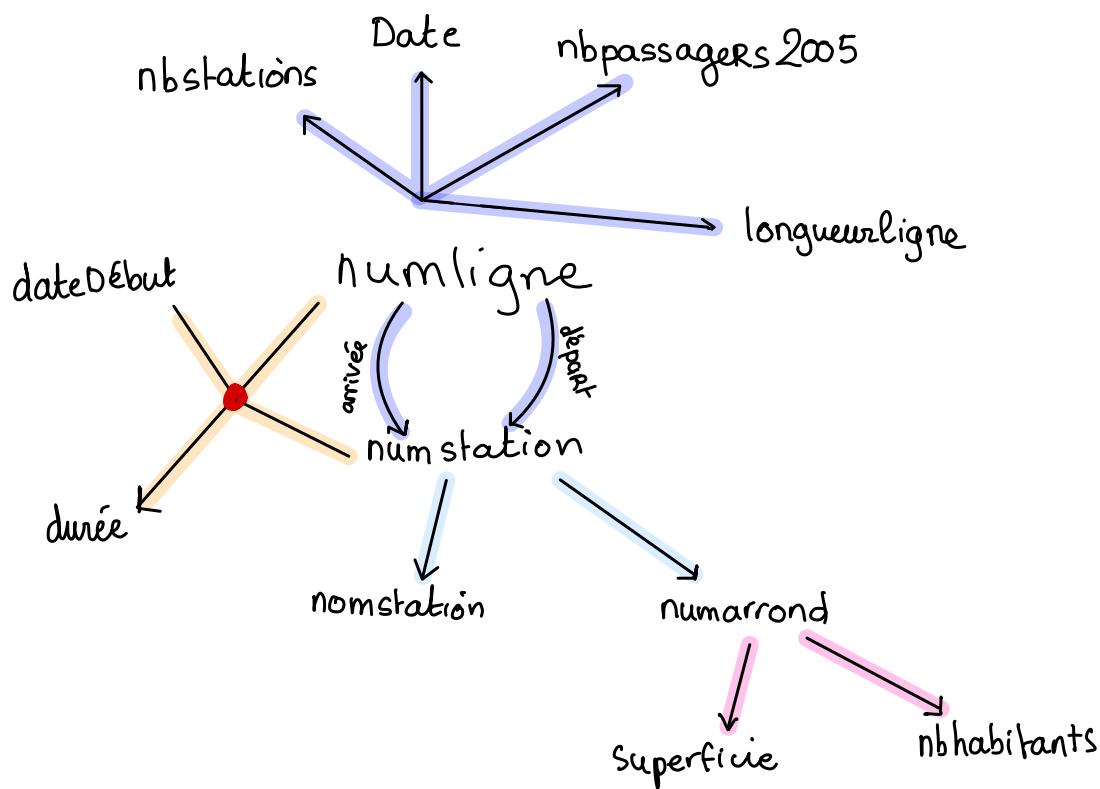
Pour chaque ligne, on connaîtra les stations de **départ** et d'arrivée, le nombre de stations. La **longueur** de la ligne et la **date** de mise en service seront également connues. Enfin, le **nombre de passagers** empruntant **chaque ligne**, recensé en 2005, sera intégré dans le système. Les informations concernant les stations sont : son **numéro** et son **nom**. On connaîtra également **l'arrondissement** dans lequel se situe une station. De plus, la donnée du **numéro d'un arrondissement** nous fournit automatiquement sa **superficie** et le **nombre d'habitants**.

Enfin, **tous les travaux effectués** dans les stations du métro de Paris seront recensés et rentré dans la base de données. On connaîtra la **date de début** et la **durée des travaux** ainsi que le **lieu de réalisation** (station + ligne). Il faut pouvoir planifier et organiser les différents travaux sur les différentes lignes et dans les stations, sachant qu'il ne peut y avoir qu'un seul chantier pour une station et une ligne de métro à une date de début **donnée**. En effet, cela entraînerait autrement un désagrément trop important pour les usagers. Le but de cette base de données est donc de gérer au mieux les travaux effectués dans une station et sur une ligne. Elle permettra aussi d'avertir les conducteurs afin qu'ils prennent en compte ces travaux et qu'ils préviennent les usagers.

Faire le graphe de dépendances fonctionnelles de cet énoncé et en déduire les relations en troisième forme normale.



- R_1 (N Ligne, NbrPassagers2005, Date mise en service, longueur ligne, Nbr station, départ, arrivée)
- R_2 (N Ligne, N Station, Date de début, durée Travaux)
- R_3 (N Station, N arrondissement, Nom)
- R_4 (N arrondissement, superficie, Nbr habitant)



● Ligne (numligne, nbstations, longueurligne, date, nbpassagers 2005, départ, arrivée)

● Station (numstation, nomstation, numarrond)

● Arrondissement (numarrond, superficie, nbhabitants)

● Travaux (numlignes, numstat, date début, durée)

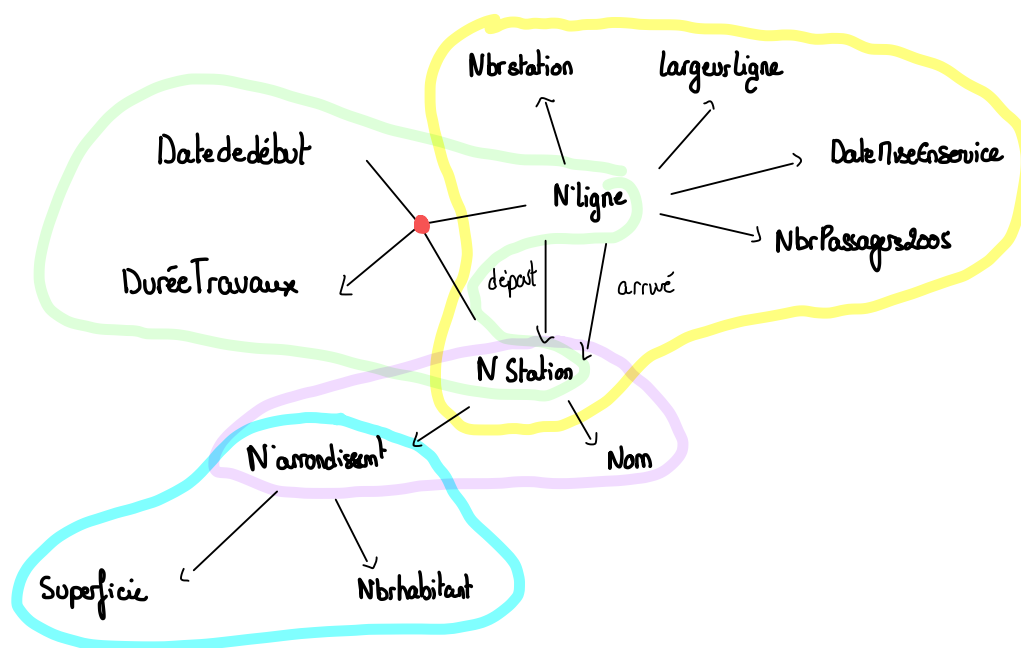
Le métro de Paris

On va mettre en place une base de données donnant les informations relatives aux différentes **lignes** et **stations** du Métro Parisien.

Pour chaque ligne, on connaîtra les stations de **départ** et d'arrivée, le nombre de stations. La **longueur** de la ligne et la **date** de mise en service seront également connues. Enfin, le **nombre de passagers** empruntant **chaque ligne**, recensé en 2005, sera intégré dans le système. Les informations concernant les stations sont : son **numéro** et son **nom**. On connaîtra également **l'arrondissement** dans lequel se situe une station. De plus, la donnée du **numéro d'un arrondissement** nous fournit automatiquement sa **superficie** et le **nombre d'habitants**.

Enfin, **tous les travaux effectués** dans les stations du métro de Paris seront recensés et rentrés dans la base de données. On connaîtra la **date de début** et la **durée des travaux** ainsi que le **lieu de réalisation** (station + ligne). Il faut pouvoir planifier et organiser les différents travaux sur les différentes lignes et dans les stations, sachant qu'il ne peut y avoir qu'un seul chantier pour une station et une ligne de métro à une date de début **donnée**. En effet, cela entraînerait autrement un désagrément trop important pour les usagers. Le but de cette base de données est donc de gérer au mieux les travaux effectués dans une station et sur une ligne. Elle permettra aussi d'avertir les conducteurs afin qu'ils prennent en compte ces travaux et qu'ils préviennent les usagers.

Faire le graphe de dépendances fonctionnelles de cet énoncé et en déduire les relations en troisième forme normale.



- R_1 (N ligne, NbrPassagers2005, Date mise en service, longueur ligne, Nbr station, départ, arrivée)
- R_2 (N ligne, N station, Date de début, durée Travaux)
- R_3 (N station, N arrondissement, Nom)
- R_4 (N arrondissement, superficie, Nbr habitant)

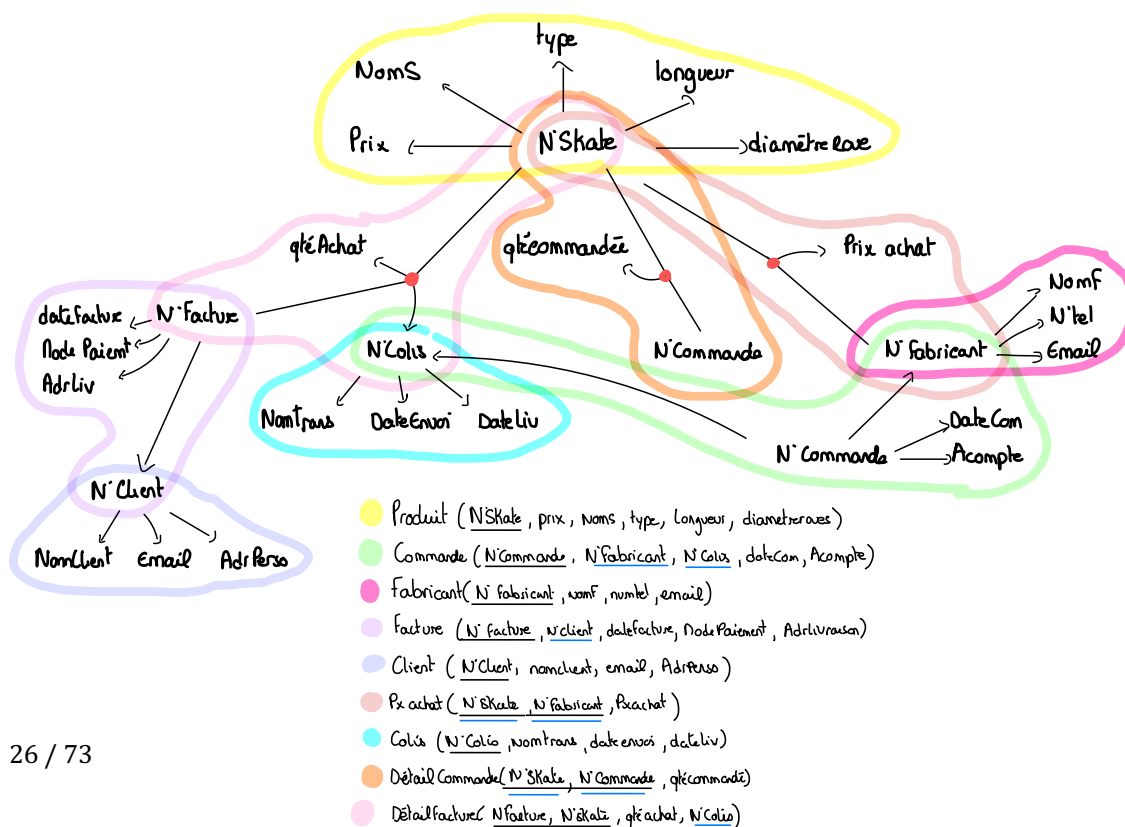
Le skateshop

Skateshop est une enseigne dédiée à la vente de skate ou autres planches de surf urbain (skate, cruiser, longboards, etc.). Elle prépare une nouvelle base de données pour mieux organiser les ventes et les livraisons des produits vendus.

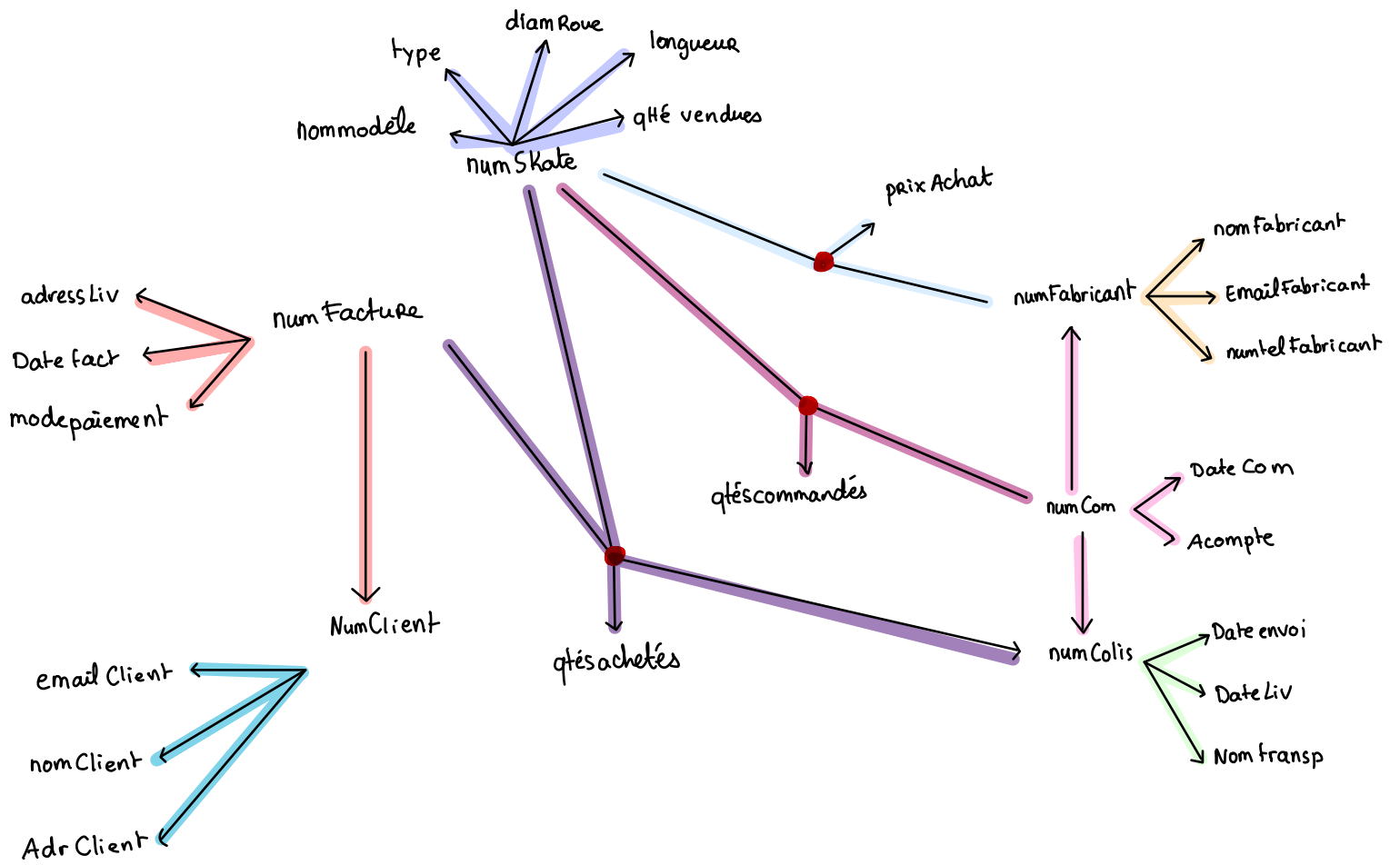
Chaque produit vendu par Skateshop est identifié par un numéro unique (le nskate). On doit connaître le nom attribué à ce modèle, son type (long, cruiser...), sa longueur, son prix, ainsi que le diamètre des roues. Chaque produit peut être fourni par un ou plusieurs fabricants, avec chacun un prix d'achat fixe. On doit connaître le nom, le numéro de téléphone et l'email de contact de chaque fabricant. Skateshop va passer régulièrement des commandes auprès de ces fabricants, pour leur commander des nouveaux produits. On doit garder une trace de ces commandes. Chaque commande est ainsi passée à une certaine date, auprès d'un fournisseur précis. On peut lui commander plusieurs produits, à différentes quantités. Pour chaque commande, Skateshop paie en avance une certaine valeur en guise acompte. Cette valeur sert ainsi de garantie à la commande et doit être enregistrée dans la base de données.

En plus des commandes réalisées auprès des fabricants, la base de données doit également enregistrer les factures des achats réalisés par les clients. Les clients peuvent acheter des produits présents dans la boutique ou précommander des articles qui seront bientôt disponibles. Chaque client est ainsi identifié par un numéro. On enregistre dans la base son nom, son email et une adresse. Cette adresse n'est pas forcément celle utilisée pour la livraison. A chaque achat, le client peut fournir une adresse de livraison qui peut être différente de son adresse personnelle. L'adresse de livraison est ainsi enregistrée avec la facture. Celle-ci est identifiée par un numéro. On doit connaître la date de chaque facture et le mode de paiement utilisé, ainsi que les produits qui ont été achetés et leurs quantités. Ces produits sont envoyés aux clients sur un ou plusieurs colis, en fonction de leur disponibilité dans la boutique. On doit garder une trace de ces colis. Chaque colis est identifié par un numéro de colis. On en enregistrera également le nom du transporteur, la date d'envoi et la date de livraison. La base de données enregistrera également les colis utilisés pour les livraisons des produits commandés auprès des fabricant. Dans ces cas, afin de simplifier la gestion du stock, on demande à chaque fabricant d'envoyer la totalité de la commande dans un seul colis, dont le numéro et les informations (transporteur, date d'envoi et de livraison) sont enregistrées dans la base de données.

- 1) Proposer un graphe de dépendances fonctionnelles pour la base de données de la boutique Skateshop.
- 2) Indiquer les schémas des relations en 3FN pour cette base de données.



Le Skate Shop



● Skate (numSkate , nommodèle , diamRoue , longueur , prixvente , type)

● PrixAchat (numSkate , numFabr , prixAchat)

● Fabricant (numFabr , nomFabr , emailFabr , numtelFabr)

● Commandes (numCom , datecom , acompte , numFabr , numColis)

● Colis (numColis , dateenvoi , dateliv , modepaiement)

● Facture (numFacture , adressLiv , dateFact , modepaiement , numClient)

● Client (numClient , emailClient , nomClient , adressClient)

● Détail Commande (numSkate , numCom , qtécommande)

● Détail Facture (numSkate , numFacture , numColis , qtéachetees)

Les lutins du Père Noël

Comme chaque année, le Père Noël met toute une organisation en route pour permettre la gestion des cadeaux de Noël : prospection des vœux des enfants, choix des cadeaux en correspondance avec ces vœux, emballage, stockage... Il s'informatise de plus en plus et est même joignable maintenant directement sur Internet. Cependant, il gère encore le problème des livraisons à la main, ce qui prend énormément de temps et génère parfois des erreurs. Il a donc décidé d'informatiser la gestion des livraisons du 24 Décembre prochain à l'aide du système que vous allez lui modéliser ici. Les données à prendre en compte sont les suivantes.

Lors de la prospection des vœux des enfants, on enregistre plusieurs informations sur ceux-ci : leur nom, leur prénom, leur adresse (sinon, pas de livraison !), leur sexe, leur vœu et leur pourcentage de sagesse (s'il est trop bas, il n'y a pas le cadeau désiré, comme pour le petit Christophe qui aurait voulu un cheval de bois et qui n'aura qu'un cheval en plastique car il a un pourcentage de sagesse égal à 25%...). Pour simplifier la gestion, on les identifie par un numéro permettant de retrouver toutes les autres informations citées ci-dessus (par exemple, Franck, petit parisien qui aimerait avoir un camion de pompier, est identifié par le numéro 109 alors que Noémie, la jeune rouennaise qui rêve d'avoir une peluche d'unicorne, a le numéro 2302).

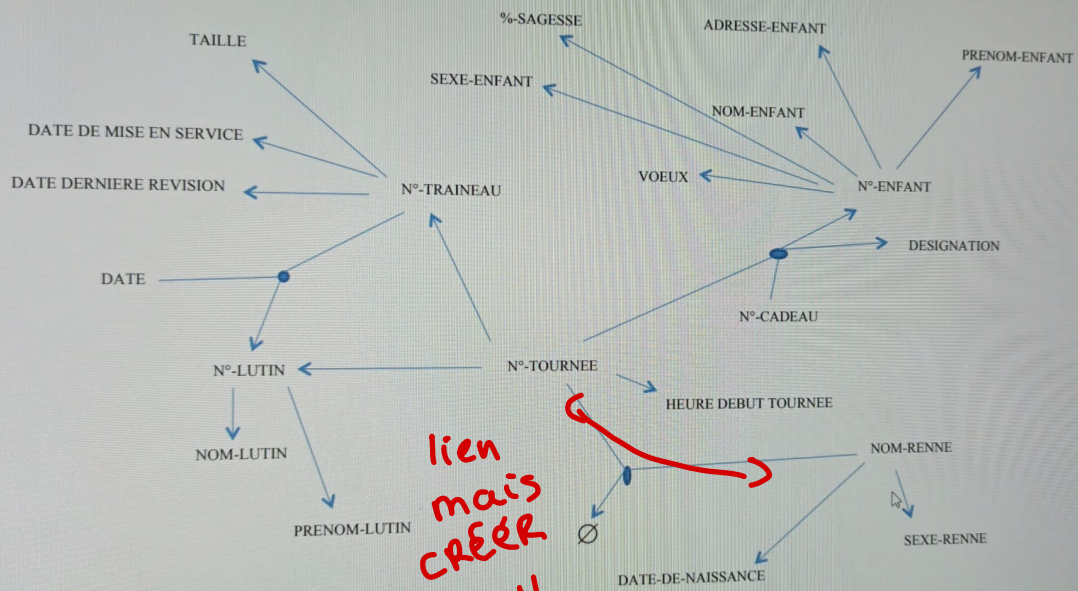
Comme le Père Noël ne peut pas se découper en milliers de morceaux, qu'il n'a pas le don d'ubiquité (contrairement à la légende populaire) et qu'il commence à ne plus être tout jeune, il se fait aider par toute une armée de petits lutins qui font les livraisons à sa place. Ces lutins ont un nom et un prénom, mais on les identifie par un numéro spécifique car les lutins ont souvent des homonymes (il y a en effet deux lutins s'appelant Tautie Hélène : les numéros 48 et 2013). Les lutins ont beau être des êtres fantastiques, ce ne sont toutefois pas des personnages très forts et ils utilisent donc des traîneaux pour porter tous les cadeaux. On veut pouvoir entrer les données concernant les traîneaux dans la base de données pour en permettre une meilleure gestion. Ils ont un numéro, une taille (le traîneau 14 a une taille de 3m alors que le 59 atteint les 4m) et une date de mise en service. Pour permettre une meilleure gestion et un meilleur entretien du parc des traîneaux, on veut également connaître la date de la dernière révision de chacun de ceux-ci (le 14^{ème} traîneau n'a pas été révisé depuis le 1^{er} Mars 2009 alors que le 59^{ème} a été révisé le 1^{er} Novembre de cette année). Chaque traîneau a un lutin qui en est responsable et l'on veut connaître l'historique des responsabilités (le traîneau 14 a eu deux responsables pour l'instant : tout d'abord le lutin 48 et, depuis le 1^{er} Décembre 2006, c'est le lutin Billano Pedro qui s'en est occupé). Il est à noter qu'un lutin peut être responsable de plusieurs traîneaux en même temps (par exemple, Billano Pedro s'occupe du traîneau 14 depuis le 1^{er} Décembre 2006 et du traîneau 59 depuis le 4 Juillet 2009).

Les cadeaux sont ensuite mis dans les traîneaux pour la livraison. C'est alors que l'on définit les tournées à effectuer. Celles-ci sont identifiées par un numéro unique qui permet de connaître le traîneau concerné, le lutin conducteur et l'heure de départ de la tournée (par exemple, la 4^{ème} tournée débute à 20h et c'est le lutin 48 qui conduit le traîneau 14). Les traîneaux, comme les lutins, pourront faire partis de plusieurs tournées dans la nuit de Noël (le traîneau 14 est aussi utilisé pour la 9^{ème} tournée à 2h30 que conduira le lutin 2013).

Pour ne pas confondre les cadeaux, on leur affecte un numéro unique au sein d'une tournée spécifique. Chaque cadeau possède une désignation et correspond à un enfant particulier (le 1^{er} cadeau de la 4^{ème} tournée est destiné au petit Franck qui va recevoir son camion de pompiers et le 4^{ème} cadeau de cette même tournée est pour Noémie qui va recevoir sa peluche d'unicorne).

Les traîneaux sont bien entendu tiré par des rennes (chacun sait que le Père Noël vit en Laponie, pays très peuplé de ces animaux). Chacun de ceux-ci est identifié dans la base de données par un nom particulier, ce qui permet de connaître également son sexe et sa date de naissance (Bouboule est un mâle né le 3 Avril 97, suivi de près par AhQueJohnny qui est né le 23 Juin de la même année). Chaque traîneau est tiré par un équipage de plusieurs rennes pour la tournée prévue. Bien entendu, un renne peut participer à plusieurs tournées différentes (Bouboule participe à la 4^{ème} tournée avec son copain AhQueJohnny et à la 9^{ème} tournée avec Starlette, née le 4 Janvier 99).

Proposer un graphe de dépendances fonctionnelles et l'ensemble de schéma de relations en 3FN pour la nouvelle base de données du Père Noël.



Réervations hôtelières

On envisage de créer un système centralisé de réservations de chambres d'hôtel dans une région de vacances qui englobe plusieurs stations ayant chacune plusieurs hôtels.

Tout demandeur peut téléphoner au système afin de réserver des chambres ; l'opérateur chargé de la réservation lui demande plusieurs renseignements : nom, prénom, adresse, numéro de téléphone et diverses indications sur la demande, soient la période de réservation, le nombre de chambres, la catégorie d'hôtel, la station demandée. Il affecte à chaque demande un numéro d'ordre.

Pour deux périodes distinctes, on considérera qu'il y a deux demandes distinctes.

L'opérateur est chargé de vérifier si la demande peut être satisfaite ; s'il n'y a pas de possibilité de la satisfaire, il sollicite le demandeur pour formuler éventuellement une nouvelle demande. S'il est possible de satisfaire la demande, il y a confirmation par l'opérateur avec création d'une réservation qui comporte tous les renseignements qui vont permettre de préparer ultérieurement la facture. Ce processus aboutit à attribuer au client un numéro qui servira d'identifiant pour celui-ci.

Chaque hôtel est décrit par son nom, son numéro, la station à laquelle il appartient, sa catégorie, le nombre de chambres disponibles, le prix unitaire de chaque chambre (prix haute saison, prix basse saison), les périodes de disponibilité pour chaque chambre ; chaque station est décrite par son nom et son numéro.

On considère :

- Qu'une demande est relative à un seul client, ainsi que la réservation
- Que chaque période est repérée par une date de début et une date de fin.

Le concepteur vous propose la collection de relations suivantes :

DEMANDE (ETATDEM, NOMDEM, ADRDEM, TELDEM, DATDEBDEM, DATFINDEM, NBCHDEM, CATHOTDEM, NOMSTADEM, NUMDEM, DATDEM, DATETDEM)

NOMDEM est le nom du demandeur, ADRDEM et TELDEM ses adresse et numéro de téléphone ; (DATDEBDEM, DATFINDEM) définit la période de réservation demandée ; CATHOTDEM est la catégorie d'hôtel demandée ; NOMSTADEM est le nom de la station et NBCHDEM est le nombre de chambres demandées. NUMDEM est le numéro d'ordre de la demande, DATDEM la date de sa formulation. On admet qu'une demande puisse être mise en attente. À cet effet, ETATDEM caractérise l'état, et DATETDEM la date de cet état.

RESSOURCE (NOMSTA, NUMSTA, NOMHOT, NUMHOT, ADRHOT, CATHOT, NUMCHA, CATCHA, DATDEBDIS, DATFINDIS)

NOMSTA et NUMSTA sont respectivement le nom et le numéro de la station ; NOMHOT, NUMHOT, ADRHOT et CATHOT sont les caractéristiques de l'hôtel ; NUMCHA et CATCHA correspondent au numéro et à la catégorie de la chambre. DATDEBDIS et DATFINDIS caractérisent les dates de début et de fin d'une période de disponibilité de chambre.

SAISON (TYPSTAI, DATSAIDEB, DATSAIFIN, NUMSTA)

TYPSAI définit le type de saison, les deux dates caractérisent le début et la fin de la saison.

RESERVATION (ETATRES, NUMRES, DATDEBRES, DATFINRES, NUMDEM, DATRES, NUMHOT, NUMCHA, NUMCLI, NOMCLI, ADRCLI)

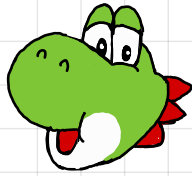
ETATRES détermine l'état de la réservation (OK, ANNULE...). NUMRES est le numéro identifiant la réservation définie pour la période (DATDEBRES, DATFINRES) à la date DATRES et en réponse à la demande de numéro NUMDEM ; la réservation est passée par un client, identifié par NUMCLI (numéro) et caractérisé par NOMCLI (nom) et ADRCLI (adresse).

TARIF (NUMCHA, NUMHOT, NUMSTA, DATSAIDEB, DATSAIFIN, PRIXCHA)

PRIXCHA est le prix de nuitée de la chambre de numéro NUMCHA dans l'hôtel de numéro NUMHOT appartenant à la station de numéro NUMSTA, entre les dates DATSAIDEB de début de saison et DATSAIFIN de fin de saison

On vous demande :

1. D'étudier chacune des relations proposées, de préciser les dépendances fonctionnelles qui existent entre leurs attributs et de les transformer, si besoin est, en 3e forme normale.
2. De vérifier que la collection des relations en 3e forme normale obtenue précédemment est cohérente par rapport à l'énoncé et la compléter si nécessaire.
3. D'ajouter une collection de contraintes d'intégrité.



NumDem

numcli

nomcli

Adr Cli

telCli

Club de TaeKwonDo

Le TaeKwonDo est un art martial coréen, devenu pratique olympique depuis les Jeux de Sydney en 2000. Le club de TaeKwonDo MSG Tae souhaite organiser sa base de données. Dans cette base, le club stocke les informations concernant les licenciés et les entraîneurs, ainsi que les groupes d'entraînements et les commandes d'accessoires.

Chaque personne liée au club (licencié ou entraîneur) est identifiée par un numéro unique (npers). Elle a un nom, un prénom, une adresse et un téléphone. Par ailleurs, chaque licencié est aussi identifié par son numéro de licence (nlic). Il a une cotisation (cotisation), dont chaque licencié doit s'acquitter annuellement. Un entraîneur est identifié par un numéro unique (nens) fourni par le club. Un entraîneur est obligatoirement un licencié. Par ailleurs, les entraîneurs sont souvent de bénévoles. Ils ont une activité professionnelle (profession) ailleurs.

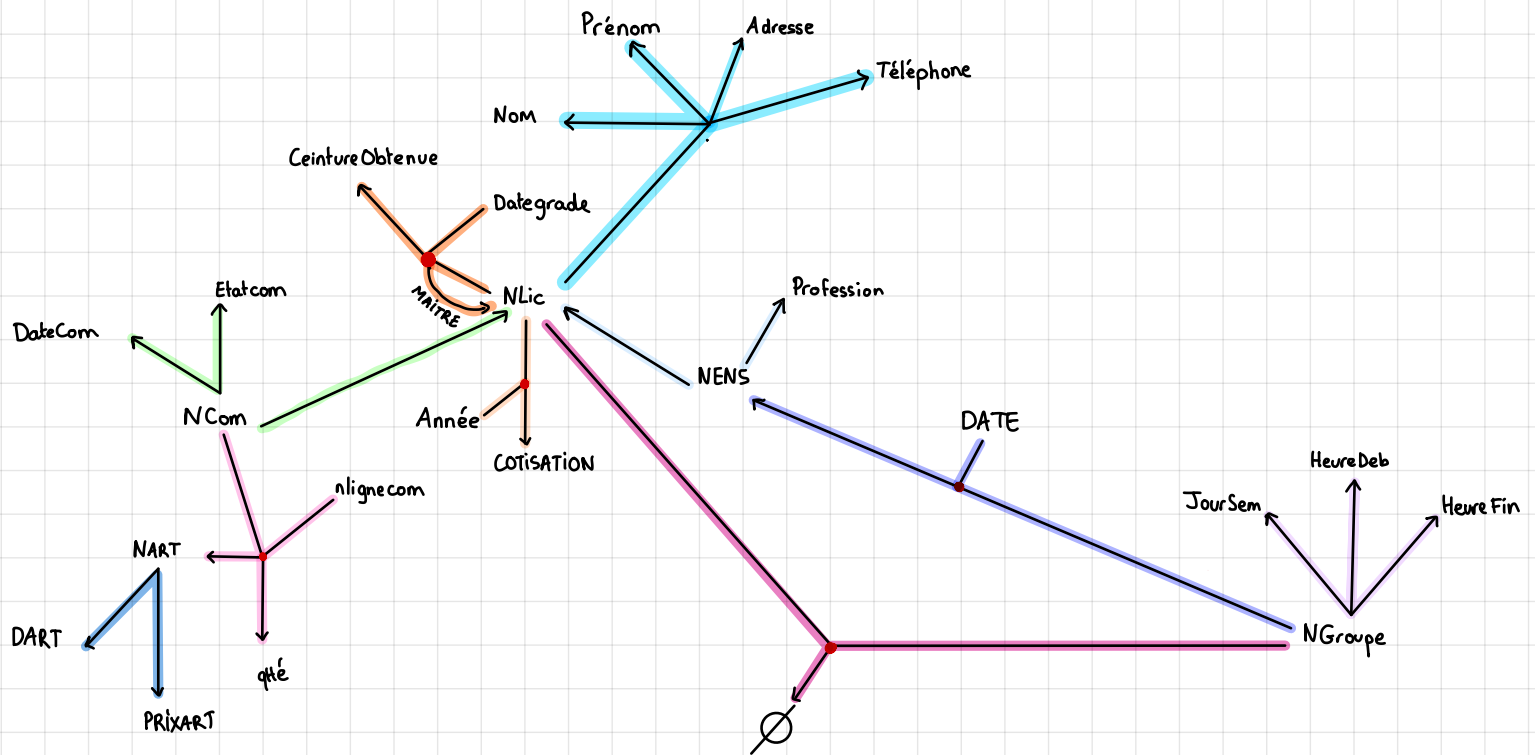
Les entraînements s'organisent en groupes. Chaque groupe est identifié par un numéro (ngroupe), par le jour de la semaine (joursem), l'heure de début (heuredeb) et l'heure de fin (heurefin) pendant lesquels les entraînements sont réalisés. Chaque groupe a un entraîneur responsable, qui doit, en théorie, assurer les entraînements. Cependant, dans la pratique, il arrive qu'un autre entraîneur réalise l'entraînement d'un groupe à une certaine date. Le club souhaite garder une trace de ces remplacements, en gardant particulièrement les dates et les entraîneurs qui ont effectivement réalisé l'entraînement d'un groupe. De plus, les licenciés peuvent participer à différents groupes d'entraînements. Le club souhaite savoir précisément quels licenciés sont inscrits à quels groupes.

Par ailleurs, le club souhaite garder l'historique de l'évolution de chaque licencié. L'évolution d'un licencié se fait par l'attribution d'une ceinture colorée (blanche pour les débutants, noir pour les plus expérimentés). L'attribution se fait par un maître (également licencié dans le club, mais pas forcément un entraîneur) à la suite d'un examen technique. Le club souhaite garder toutes ces informations (le licencié, la ceinture obtenue, la date de son obtention et le maître qui l'a attribué).

Enfin, le club organise également la commande d'accessoires pour la pratique sportive (dobok, ceintures, protections, etc.). Chaque commande concerne forcément un licencié et peut concerner plusieurs articles, chacun avec une quantité précise (par exemple, 1 dobok, 3 t-shirts, 1 ceinture bleue). Une commande est identifiée par un numéro de commande (ncom). Elle est passée à une date donnée (datcom) et a un état (« en attente », « payée »...). Chaque article est identifié par un numéro (nart) et une description (dart). Il a aussi un prix de référence (prixart).

A partir des informations ci-dessous, on vous demande :

- a) Définir le graphe des dépendances fonctionnelles en 3FN
- b) Définir l'ensemble de relations en 3FN correspondant. Décrire le schéma de la base de données correspondant en précisant les contraintes de clé primaire et de clé étrangère
- c) Indiquer la commande SQL nécessaire pour la création de la table contenant l'évolution des licenciés (les dates de passages de ceinture, avec la ceinture obtenue et le maître).
- d) Ajouter à la table récemment créée le passage de ceinture de l'élève Jean Dupont. Ce licencié a obtenu sa ceinture jaune le 1/09/2020. Il a été évalué par le maître Thomas.



- Licencié (Nlic, / Nom, Prénom, adresse, téléphone)
- Cotisation (nlic, année, cotisation)
- Entraîneur (NENS, nlic, profession)
- Groupe (NGroupe, joursem, heureDeb, heureFin)
- Remplacement (ngroupe, dategroupe, NENS)
- Inscrit (ngroupe, nlic)
- PassageGrade (nlic, DateGrade, ceintureObtenue, Maître)
- Commande (ncom, datecom, etatcom, nlic)
- DétailCom (ncom, nlinecom, nart, qté)
- Article (NART, DART, PrixArt)

CREATE TABLE Passage Grade (

nlic INTEGER, REFERENCES Licencié (nlic),
 dategrade DATE,
 couleurceinture VARCHAR(20),
 maître INTEGER, REFERENCES Licencié (nlic),
 PRIMARY KEY (nlic, dategrade),
 FOREIGN KEY (nlic) REFERENCES Licencié (nlic),
 FOREIGN KEY (maître) REFERENCES Licencié (nlic)
);

INSERT INTO PassageGrade (nlic, dategrade, couleurceinture, maître)
 VALUE (26, '01/09/2020', 'jaune', 55);

Salle de sport

La salle de sport « NewGame » souhaite réorganiser sa base de données pour les cours collectifs et les adhérents. Dans cette base, NewGame souhaite enregistrer les informations suivantes sur les adhérents : leur numéro d'adhérent, leur nom et prénom, date de naissance et sexe. On souhaite garder les mêmes informations par rapport aux coaches. Par contre, pour ces derniers, on doit également garder une trace des certificats médicaux et des diplômes obtenus. En effet, tous les ans, chaque coach doit être en mesure de présenter à la salle un certificat médical signé par son médecin traitant lui autorisant la pratique sportive. On doit enregistrer la date de ces certificats et le nom du médecin l'ayant autorisé. Du même pour les diplômes : les coaches passent régulièrement des nouveaux diplômes et certifications. On doit enregistrer dans la base l'intitulé des diplômes obtenus, leur date d'obtention ainsi que l'organisme ayant délivré le diplôme au coach.

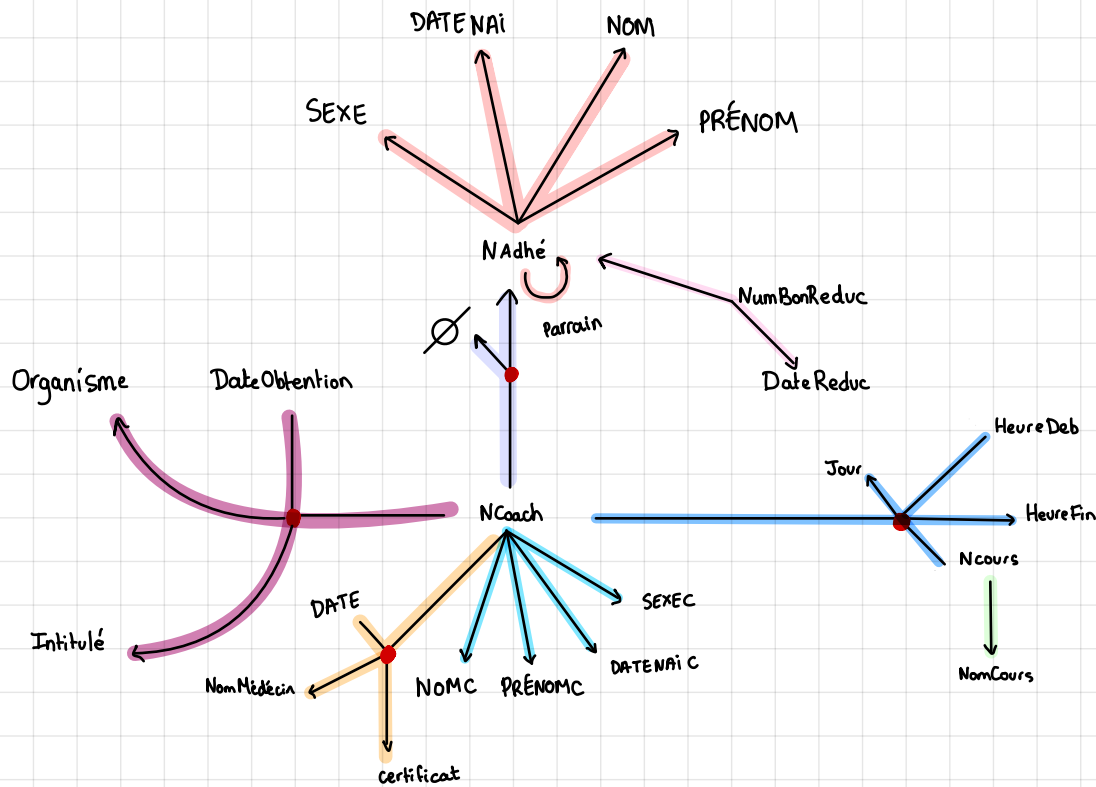
On souhaite mettre en place une politique de parrainage pour les adhérents. L'idée est qu'un nouvel adhérent peut être parrainé par un autre adhérent, qui lui aurait suggéré de venir s'inscrire à la salle. Un adhérent ne peut avoir qu'un parrain, mais il peut parrainer à son tour des nombreux adhérents. Au bout de 5 parrainages, un adhérent gagne un bon de réduction de 15€ qu'il peut utiliser auprès des boutiques partenaires. On doit enregistrer dans la base le numéro identifiant chaque bon de réduction offert, ainsi que l'adhérent l'ayant reçu et la date d'émission du bon.

Les cours collectifs proposés par la salle sont identifiés par un numéro unique. Plusieurs créneaux peuvent être proposés par un même cours. Par exemple, le cours de Zumba a lieu le mercredi et le jeudi de 18h à 19h, alors que le cours de Stretching a lieu le mardi de 10h à 10h30 et le samedi de 11h à 11h45. On doit connaître ces créneaux ainsi que le coach qui doit assurer chaque créneau (tous les créneaux d'un cours ne sont pas toujours assurés par le même coach).

Enfin, les adhérents ont droit à 3 séances de coaching avec les coaches de leur choix. Chaque adhérent peut ainsi choisir un à plusieurs coaches pour réaliser ces séances. On doit enregistrer dans la base de données les coaches qui ont été choisis par chaque adhérent.

- 1°) Proposez le graphe des dépendances fonctionnelles élémentaires et directes
- 2°) Définissez le schéma relationnel de la base de données en 3ème forme normale
- 3°) Donnez la commande SQL nécessaire pour créer la table contenant les informations sur les adhérents.
- 4°) Donnez la commande SQL nécessaire pour insérer un nouvel adhérent « Jean Paul », née le 1/10/1990, qui a été parrainé par l'adhérent « Marie José » (adhérent n° 45).

Salle de Sport



- Adhèrent (Nadhé, Nom, Prénom, Sexe, DateNai, PARRAIN)
- Coaching (Nadhé, Ncoach)
- Coach (Ncoach, NomC, PrénomC, SexeC, DateNaiC)
- Certificat Médical (Ncoach, Date, NomMédecin, Certificat)
- Diplôme (Ncoach, Dateobtention, Intitulé, Organisme)
- Bon Réduc (NumBonRéduc, Nadhé, DateRéduc)
- Cours (Ncours, NomCours)
- Créneaux (Ncours, heureDeb, jour, Ncoach, heureFin)

```

3) CREATE TABLE Adhèrent (
  NAdh  INTEGER PRIMARY KEY,
  Nom   VARCHAR(50),
  Prénom VARCHAR(50),
  Sexe  VARCHAR(10),
  DATENAI DATE,
  Parrain INTEGER REFERENCES Adhèrent (Nadhé)
);
  
```

```

4) INSERT INTO ADHÉRENT (nlig, prénom, sexe, datenai, parrain)
VALUES (18, Jean Paul, 'homme', '01/10/1990', 45);
  
```

Vente en ligne (Examen 2014-2015)

Le site de vente en ligne « Foire au Numérique » vous demande de concevoir sa nouvelle base de données. Dans celle-ci devront être enregistrées les informations concernant les adhérents, leurs commandes, leurs avis sur les articles ainsi que leur « wish list » (liste d'envies).

Chaque adhérent est identifié par son numéro d'adhérent. Plusieurs informations sur les adhérents doivent être enregistrées : leur nom, prénom, sexe, email, adresse postale, date de naissance et date d'adhésion. Les adhérents peuvent également cumuler plusieurs codes de réduction. A chaque code, correspondent une valeur (en euros) et une période de validité. Par ailleurs, un code de réduction ne peut être associé qu'à un adhérent précis.

Les adhérents passent leurs commandes via le site Internet. Chaque commande est identifiée par un numéro de commande et contient une date et une adresse de livraison (qui peut être différente de l'adresse de l'adhérent). Chaque commande peut contenir plusieurs articles, en différentes quantités. Chaque article est identifié par un numéro et comporte un prix. Ce prix évolue au fil du temps, et on doit garder une trace de cette évolution.

Le magasin souhaite offrir à ses adhérents la possibilité de mensualiser les paiements de leurs commandes, sans frais additionnels. Par exemple, une commande d'un montant total de 120€ peut être payée en 12 mensualités de 10€. Pour chaque mensualité, on doit connaître le montant à payer et la date de paiement.

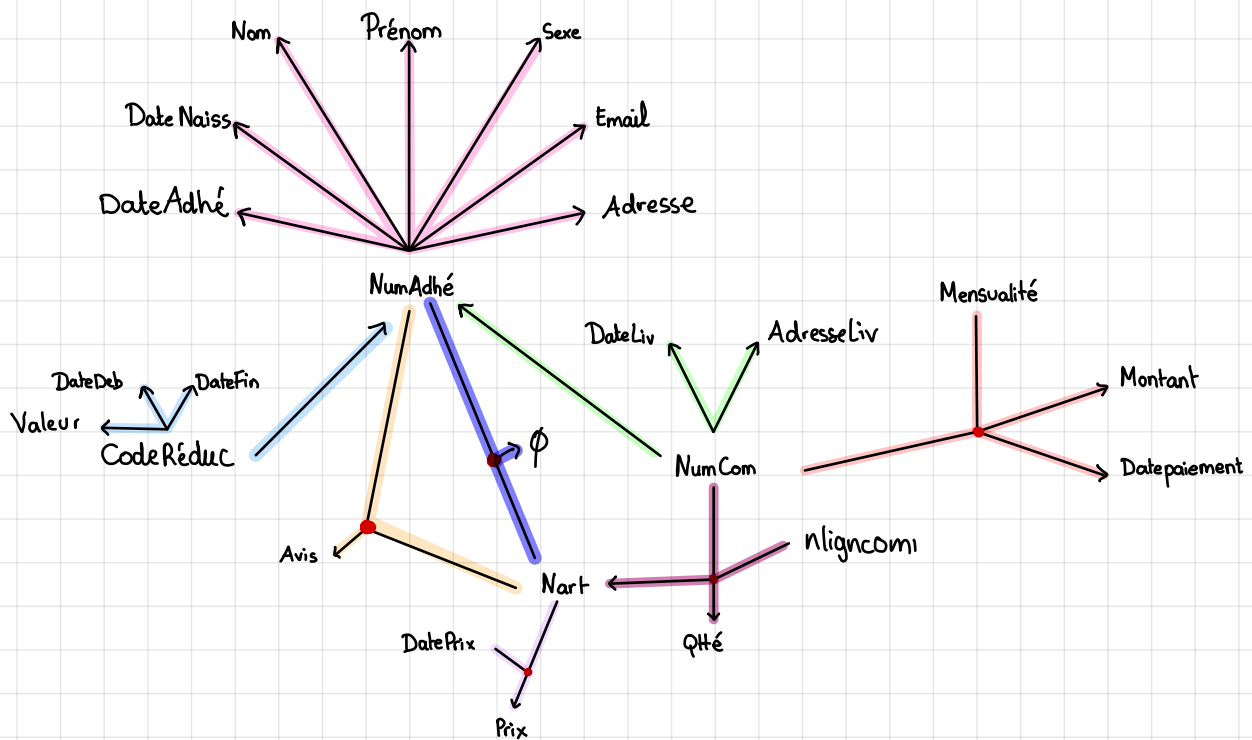
Un adhérent peut également exprimer sur le site Internet son avis sur les articles (sur tous les articles et pas seulement ceux qu'il a déjà achetés). Pour chaque article, l'adhérent peut donner au maximum un avis qui est enregistré dans la base de données et affiché sur le site. En plus des avis, les adhérents ont une liste d'envie (appelée « wish list »), dans laquelle ils ajoutent les articles qui les intéressent (mais qu'ils n'ont pas forcément achetés).

1°) Proposez le graphe des dépendances fonctionnelles élémentaires et directes (6 points)

2°) Définissez le schéma relationnel de la base de données en 3^{ème} forme normale (3 points)

3°) Donnez la commande SQL nécessaire pour créer la table contenant le contenu des commandes (1 point)

Vente en ligne



- Adhérent (numadhé, dateadhé, date Naiss, nom, prénom, sexe, email, adresse)
- Réduction (code reduc, num adhé, date début, date fin, valeur)
- Mensualité (ncom, mensualité, date paiement, montant)
- Commande (ncom, date com, adresse liv)
- Détail commande (ncom, nligne com, nat, qté)
- Article (nat, date prix, prix)
- Wishlist (nadhé, nat)
- Avis (nadhé, nat)

```
CREATE TABLE DETAIL COMMANDE (
```

ncom	INTEGER,
nlinecom	INTEGER,
hart	INTEGER,
glté	INTEGER,

PRIMARYKEY (ncom, nlignecom)

FOREIGN KEY (ncom) REFERENCES COMMANDE

FOREIGN KEY (nart) REFERENCES ARTICLE

)

Personal Trainer

L'autoentrepreneur Tiberius Tot, plus connu sur le surnom de « Toto », est un personal trainer reconnu dans son métier. Il dispose d'une large clientèle qui le sollicite pour une mise en forme personnalisée. Il vous a sollicité pour lui créer une base de données lui permettant de mieux organiser les informations sur ses clients, leurs séances et les groupes d'entraînements qu'il anime.

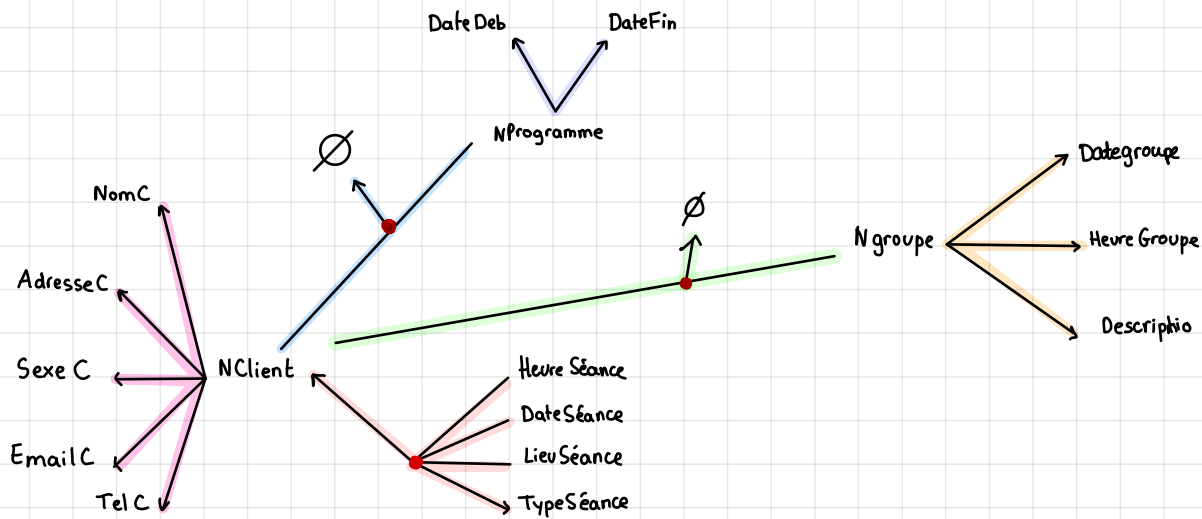
La base de données devra ainsi enregistrer des informations personnelles permettant à Toto de contacter ses clients : nom, adresse, téléphone, sexe et email notamment. Les clients peuvent payer les séances individuelles ou s'inscrire à un programme de remise en forme. Chaque programme est valable pour une période (entre 2 à 4 semaines), et un client peut s'inscrire plusieurs fois à ces programmes. Par exemple, la cliente Gisèle s'est déjà inscrite à deux programmes de mise en forme, un premier programme du 1/06/2020 au 1/07/2020 et un second programme allant du 10/10/2020 au 30/10/2020. On doit garder une trace dans la base de données de tous les programmes inscrits par les clients.

On doit également enregistrer les séances de remise en forme individuelles réalisées par chaque client. Chaque séance est programmée à une date et heure données, et se déroule à une adresse précise (le plus souvent la résidence du client, mais pas forcément, certaines séances, notamment celles de running, peuvent se dérouler dans un parc). Chaque séance va ainsi avoir un type (« running », « renforcement musculaire », « cardio », etc.). Toto demande à ce que ces séances ne soient pas identifiées par un numéro ou un quelconque code, car il a dû mal à s'organiser avec des codes...

Enfin, Toto organise également des groupes d'entraînements. Chaque groupe va avoir un numéro qui l'identifie de manière unique, ainsi qu'une description et des séances à des dates et des heures fixes. Par exemple, le groupe n° 1 est un groupe de « running » (description). Il va avoir sa 1^{ère} séance le 25/10/2020 à 9h, la 2^{ème} 27/10/2020 à 18h et la 3^{ème} le 30/10/2020 à 18h. Les clients peuvent s'inscrire à un plusieurs de ces groupes.

- 1°) Proposez le graphe des dépendances fonctionnelles élémentaires et directes
- 2°) Définissez le schéma relationnel de la base de données en 3^{ème} forme normale
- 3°) Donnez la commande SQL nécessaire pour créer la table contenant les inscriptions des adhérents aux groupes.
- 4°) Donnez la commande SQL nécessaire pour insérer l'inscription de l'adhérent Gisèle au groupe n° 1.

Personal Trainer



- Client (Nclient, nom, adresse, sexe, email, tel)
- Programme (nprogramme, datedeb, datefin)
- ClientProgramme (Nclient, nprogramme)
- Groupe (ngroupe, dategroupe, heuregroupe, description)
- ClientGroupe (Nclient, ngroupe)
- Séance (HeureSéance, DateSéance, LieuSéance, Nclient, typeSéance)

```

CREATE TABLE ClientGroupe (
    nclient INTEGER,
    ngroupe INTEGER,
    PRIMARY KEY (nclient, ngroupe)
    FOREIGN KEY (nclient) REFERENCES CLIENT,
    FOREIGN KEY (ngroupe) REFERENCES GROUPE
)
    
```

4) INSERT INTO ClientGroupe (nclient, ngroupe)
VALUES (112, 1);

Intranet (Examen 2008-2009)

Un important cabinet d'études parisien souhaite mettre en œuvre son *intranet* (site internet corporatif à usage interne). Le responsable du projet vous demande de faire la *normalisation* de la base de données relationnelle prévue pour le site avant la mise en place effective de celui-ci. Cette base des données doit contenir les informations concernant les *pages* Web disponibles dans le site, les *rubriques* qui regroupent les pages du site, les *mots-clés* qui classifient ces ressources (pages et rubriques), ainsi que les informations concernant les *utilisateurs* du site. Ces informations sont représentées au travers des attributs suivants :

IdPage : numéro identifiant de manière unique une page Web dans le site	URLPage : lien (URL) vers une page Web
TitrePage : titre d'une page Web	ContenuPage : contenu d'une page Web
DateModifPage : date de modification d'une page Web	DateCreationPage : date de création d'une page Web
IdRubrique : numéro identifiant de manière unique une rubrique du site	DateCreaRub : date à laquelle une rubrique a été créée
TitreRubrique : titre associé à une rubrique	DateExpiration : date à partir de laquelle une rubrique n'est plus valable
IdUser : numéro identifiant de manière unique un utilisateur du site	NomUser : nom de l'utilisateur
PrenomUser : prénom de l'utilisateur	EmailUser : email d'un utilisateur
AdrUser : adresse de l'utilisateur	TelUser : téléphone de l'utilisateur
Statut : statut de l'utilisateur	Motcle : mot-clé pour l'indexation des pages Web
DateStatut : date à laquelle un utilisateur a acquis un statut	

Par ailleurs, le responsable du projet vous fournit les renseignements complémentaires suivants :

- Chaque page Web a un responsable (un utilisateur qui est responsable par son contenu). Cependant, plusieurs utilisateurs peuvent contribuer à la page et modifier son contenu. On souhaite garder trace de ces modifications de manière à pouvoir reconstruire l'historique des modifications et de ceux qui les ont faites pour chaque page ;
- Une rubrique est proposée par un utilisateur qui en est responsable.
- Une page Web peut ne pas avoir de titre, mais elle a toujours un contenu et un URL. En revanche, une rubrique possède toujours un titre ;
- Une rubrique regroupe plusieurs pages Web auxquelles elle permet d'accéder. Inversement, une même page Web peut être accessible par différentes rubriques ;
- On peut indiquer une date d'expiration (optionnelle) à partir de laquelle la rubrique et les pages Web qu'elle contient ne sont plus affichées dans le site (même si elles restent stockées dans la base des données) ;
- Chaque page Web est indexée par différents mots-clés. Par exemple, toutes les pages Web qui parlent des activités sportives proposées aux employés sont indexées avec le mot-clé « sport ».
- Les adresses emails sont individuelles, mais un utilisateur peut avoir plusieurs adresses email. Par ailleurs, un utilisateur possède à un moment donné différents statuts (« observateur », « contributeur », « rédacteur », « rédacteur-chef », « responsable ») pour différentes pages. Ainsi, un même utilisateur peut être « rédacteur » d'une page donnée et « observateur » d'une autre page. On souhaite garder l'évolution du statut des utilisateurs pour chacune des pages.

A partir des informations ci-dessous, on vous demande :

- Définir le graphe des dépendances fonctionnelles en 3FN.
- Définir l'ensemble de relations en 3FN correspondant. Décrire le schéma de la base de données correspondant en précisant les contraintes de clé primaire et de clé étrangère.
- Proposer une liste de contraintes d'intégrité supplémentaires.
- Donner la commande SQL nécessaire pour créer la table contenant les informations sur les pages et celle contenant les informations sur les mots clés, avec les contraintes.
- Donner la commande SQL permettant d'ajouter à la base de données la page n° 10 (idPage), intitulée « Aide MS Access » (titrePage), dont l'URL est « <http://intranet.cabinet.fr/aideAccess> » (page créée le 31/1/2009). Ajouter également les mots clés « informatique » et « access » associés à cette page.

Prêts bancaires

On vous soumet les deux relations suivantes :

Client (ncli, ncompte, nomcli, nagence, adrcli, telcli, responsable, ntypecompte, libeltypecompte, legislationcompte, solde, datesolde, adragence, telagence)

OperationBancaire (nop, nvirement, montntop, typeop, natureop, dateeffetop, jourdumois, comptesource, comptecible)

Les données permettent de gérer les comptes des clients d'une banque parisienne composée de plusieurs agences. On s'intéresse dans cette base de données à mémoriser les opérations bancaires effectuées sur les comptes de la banque ainsi que la déclaration associée à ces comptes.

Un client a un numéro unique (ncli), un nom (nomcli), une adresse (adrcli) et un téléphone (telcli). Un client peut avoir plusieurs comptes.

Un compte est identifié de manière unique par le numéro de compte (ncompte) et le numéro d'agence (nagence). Une agence est définie au sein de la banque par un numéro (nagence), une adresse (adragence) et un téléphone (telagence). Un compte appartient à un type de compte (compte courant, compte épargne, codévi, plan d'épargne logement, etc.). Le type de compte est caractérisé par un code (ntypecompte), un nom (libeltypecompte) et un texte de législation du compte (legislationcompte). Un compte peut être partagé par plusieurs clients.

La base de données doit mémoriser les différents soldes d'un compte (solde, datesolde).

Les opérations bancaires effectuées sur un compte sont mémorisées. Une opération bancaire est définie par un numéro unique (nop), un montant (montantop) avec une date d'effet (dateeffetop), un type d'opération (typeop) dont la valeur est « débit » ou « crédit » et une nature d'opération (natureop) permettant de mentionner si l'opération est faite par un chèque, des espèces, un virement ou la carte bleue.

Une opération bancaire peut être générée automatiquement à partir d'une déclaration de virement effectué par le client. Une déclaration de virement est définie par un numéro unique (nvirement) et mentionne deux comptes de la banque (comptesource, comptecible). Le client doit déterminer le jour d'effet dans le mois du virement (jourdumois).

On vous demande :

1. De construire le graphe des dépendances fonctionnelles élémentaires et directes.
2. D'en déduire la collection des relations en 3 FN.
3. De donner la description SQL de la collection en intégrant les contraintes standards (de nullité, de référence et d'unicité).
4. D'adjoindre les contraintes d'intégrité autre que les contraintes standards en les libellant en français (par relation)

Les Cadeaux de Noël (Examen 2009-2010)

Un ami qui a toujours du mal à s'organiser pour Noël vous a demandé de l'aider en créant une base de données pour préparer ses achats. Votre ami souhaite garder dans cette base les informations concernant les cadeaux qu'il achète, ainsi que ceux qu'il a reçus, ainsi que les magasins et les commandes qui ont été utilisés pour acheter les cadeaux.

Un cadeau est un article identifié de manière unique par un numéro (nart). Chaque article dispose d'un nom, d'une description et d'un prix de référence. Par ailleurs, un article possède une thématique principale (sport, cinéma, etc.) et plusieurs thématiques secondaires. Par exemple, un t-shirt officiel du PSG aura 'sport' comme thématique principal, et 'foot' et 'vêtement' comme thématiques secondaires.

Les articles sont vendus par les magasins. Un magasin est identifié de manière unique par un numéro (nmag). Il a un nom, un téléphone, une adresse physique, ainsi qu'une adresse internet unique. Un magasin propose, pour un article donné, un prix, un certain délai de livraison, et une disponibilité ('en stock', 'disponible en 48h', 'en réapprovisionnement', 'indisponible', etc.).

On souhaite garder dans la base les informations concernant les commandes qui ont été passée aux différents magasins. Chaque commande est identifiée par son numéro (ncom). Elle concerne un magasin et elle a été passée à une date précise. Chaque commande est composée de plusieurs lignes de commande (ligcom). Une ligne de commande concerne un article commandé et la quantité commandée dans une commande donnée.

Chaque article commandé peut être livré de manière indépendante, c'est-à-dire que différents articles d'une même commande peuvent être livrés à des dates distinctes. Ainsi, chaque article, appartenant à une commande précise, aura sa date de réception (date à laquelle il a été livré).

Par ailleurs, un article acheté par le biais d'une commande est destiné à quelqu'un à qui on va l'offrir. Attention : un même article peut être acheté plusieurs fois, auprès différents magasins, à travers plusieurs commandes distinctes, afin d'être offert à différentes personnes. Une personne (un ami) est identifiée par un numéro unique (nami). Elle a un nom, un prénom, un surnom, une adresse, un téléphone et plusieurs adresses email. On associe à chaque ami un plafond indicatif : la valeur d'un article offert à cette personne ne doit pas dépasser ce plafond.

Chaque personne peut aussi offrir des articles en cadeau. On souhaite stocker dans la base les articles qui ont ainsi été reçus, sachant qu'un article peut nous être offert par différentes personnes, et qu'une même personne peut offrir plusieurs articles. Enfin, on souhaite garder l'historique d'une évaluation annuelle des magasins. Ainsi, chaque année, chaque magasin à qui l'on a fait appel reçoit une évaluation ('excellent', 'trop cher', 'pas de service après-vente', 'retard de livraison'...). On souhaite garder l'historique de ces évaluations.

1. Faites le graphe des dépendances fonctionnelles élémentaires et directes
2. Donnez le schéma relationnel de la base de données correspondante en 3^{ème} forme normale en précisant les clefs primaires, clefs étrangères et contraintes de non nullité

Gestion de troupe de théâtre (Examen 2010-2011)

La troupe de théâtre Phiasco vous demande de concevoir une base de données pour gérer les informations concernant les pièces de théâtre que la troupe met en œuvre, leurs auteurs, ainsi que les acteurs qui y jouent et les techniciens qui y participent. Une pièce est identifiée par un identifiant unique (npiece). Chaque pièce comporte une description (descpiece) et l'année dans laquelle elle a été écrite (anneepiece). Une pièce est rédigée par un ou plusieurs écrivains, dont le nom (nomecr) et l'année de naissance (anneenascecr) sont aussi enregistrés dans la base, accompagnés d'un identifiant unique (necr). Par ailleurs, chaque pièce a un responsable, habituellement le directeur de la pièce (dirpiece). Le responsable de la troupe préfère qu'on enregistre dans la base des données un seul responsable.

Chaque pièce compte plusieurs acteurs qui y jouent des rôles. Un rôle est, en fait, un nom de personnage (nompers) dans une pièce. Le responsable de la troupe Phiasco nous fait remarquer qu'un même nom de personnage peut apparaître dans différentes pièces (mais pas deux fois dans une même pièce), et qu'un même acteur peut jouer différents rôles dans différentes pièces (le responsable ne souhaite pas qu'un rôle ou un personnage soient identifiés par des codes, car cela rendrait difficile son travail d'attribution des rôles).

Par ailleurs, pour chaque acteur, on souhaite garder un historique des rôles qu'il (ou elle) a joué dans les pièces. On souhaite aussi garder les informations concernant chaque acteur : nom (nomact), prénom (prenomact), téléphone (telact) et email (mailact). Un acteur est identifié par un numéro unique (nact) attribué par le responsable de la troupe.

Outre les acteurs, une pièce mobilise aussi un certain nombre d'éléments scénographiques et de décors, identifiés par un code (codeelem) et une description (descrelem). Chaque élément est réutilisable (il peut être utilisé sur différentes pièces) et comporte aussi un état (etatelem) à une date donnée (utile pour la réalisation de l'inventaire et pour comptabiliser les coûts de chaque pièce)

Enfin, la troupe gère également son équipe technique. Pour chaque technicien, on enregistre dans la base des données les mêmes informations que pour les acteurs (nom, prénom, téléphone et email). Un technicien travaille pour une mission pendant une certaine période. Dans cette mission, il a une fonction et travaille pour une pièce donnée. Enfin, un technicien est soumis à, au maximum, un chef d'équipe, lui-même technicien.

1°) proposez le graphe des dépendances fonctionnelles élémentaires et directes en réponse à ce cahier des charges (6 points)

2°) déduisez-en le schéma de base de données relationnel en 3^{ème} forme normale en spécifiant les contraintes de clef primaire et de clef étrangère (4 points)

Gestion des Ressources Humaines (Examen 2011-2012)

On vous demande de concevoir la base de données d'un système de gestion des ressources humaines. Le système permettra de gérer les données relatives aux emplois, aux salaires, et à l'évaluation des compétences.

Les employés sont identifiés de manière unique par un numéro (numEmp). On doit pouvoir retrouver leur nom, leur adresse et leurs numéros de téléphones personnel et professionnel dans la base de données, ainsi que la date de leur première embauche. Chaque employé appartient à un département (identifié par numDep), dont on retient le nom et l'adresse. Chaque département est dirigé par un employé.

On souhaite conserver l'information relative à la rémunération des employés. Le salaire se décompose en deux parties : le fixe (mensuel) et le variable. En ce qui concerne le salaire variable, on distinguera le montant prévu du réel, sachant que la rémunération d'un employé pourra comporter plusieurs variables (primes, intéressement, etc.). Il faudra retenir pour chaque variable la date et le type de salaire. Pour le salaire fixe, on ne souhaite que conserver le montant mensuel, mais son évolution doit être historisée.

Chaque emploi sera documenté par une fiche de poste indiquant un intitulé (par exemple directeur, assistant, ingénieur, etc.), le niveau (cadre supérieur, cadre intermédiaire, ouvrier, etc.), et un descriptif textuel. Par ailleurs, la fiche de poste doit faire référence à une liste de compétences. Chaque compétence sera enregistrée avec un numComp unique, un label et une description. Enfin, une grille de salaires est définie pour la fiche emploi. Cette grille de salaires fixe indique une tranche de salaire (i.e., le salaire minimum et le salaire maximum) par ancienneté minimale. Par exemple, à partir de 0 année d'ancienneté, la tranche de salaire fixe pour le poste P1 sera [30.000, 50.000], à partir de 3 ans, la tranche est [35.000, 57.000], à partir de 5 ans, la tranche est [40.000, 60.000], etc. Un numFichePoste est défini pour identifier chaque fiche de poste de manière unique.

Les départements offrent des emplois. Une offre d'emploi correspond à la publication d'une fiche de poste par un département pour une période donnée. Un employé appartient au département qui a offert l'emploi qu'il occupe, pendant la période à laquelle il l'occupe. On souhaite conserver l'historique des emplois occupés par les employés.

Les bilans de compétences sont réalisés chaque année par le directeur de chaque département pour chaque employé du département. Les bilans annuels de compétences de chaque employé (identifiées par un numBilan unique) seront enregistrés avec la date de réalisation, et une appréciation générale. Par ailleurs, une note est donnée de 1 à 7 est enregistrée pour chaque compétence définie dans la fiche de poste de l'emploi occupé.

1°) Proposez le graphe des dépendances fonctionnelles élémentaires et directes (6 points)

2°) Définissez un schéma relationnel en 3ème forme normale (4 points)

Service de Scolarité (Examen 2012-2013)

Une université parisienne vous demande de l'aider à concevoir une base de données pour son service de scolarité. Ce service gère les inscriptions pédagogiques des étudiants aux formations, leur répartition en groupe de TD et leurs notes.

Ainsi, on vous explique qu'un étudiant est identifié par son numéro d'étudiant (ine). Pour chaque étudiant, on garde plusieurs informations utiles au service de scolarité : nom, prénom, adresse, téléphone contact, téléphone portable et email personnel. Chaque année, les étudiants s'inscrivent à une formation précise. Une formation est identifiée par son code (codefor) et par son intitulé (nomfor). Elle correspond également à un niveau (L3, M1...), est maintenue par une UFR et compte un enseignant responsable. Les étudiants s'inscrivent également, chaque année, à un groupe de TD, identifié par son numéro (ngroupe), et dont un des étudiants en est le délégué.

Chaque formation est composée de plusieurs modules, qui peuvent être mutualisés entre les formations (par exemple, le module d'Informatique est partagé entre les formations L3 Gestion et L3 Finance d'Entreprise). Chaque module est identifié par son code (codemod). Il a un intitulé, un volume horaire en CM et un volume en TD. Par contre, son emploi du temps varie en fonction du groupe de TD. Un groupe de TD peut avoir, pour un même module, plusieurs séances dans la semaine, identifiées par leur numéro d'ordre (nseance). Par exemple, le groupe 3603 effectue deux séances de TD d'Informatique par semaine : la première séance a lieu tous les lundis à 10h, tandis que la seconde a lieu les mardis à 8h. Pour chaque séance d'un module, pour un groupe donné, on souhaite savoir la salle qui lui est attribuée, l'heure de début et de fin, et le jour de la semaine. La participation d'un étudiant à un module est attestée, chaque année, par les notes qu'il obtient à ce module. Chaque étudiant ne peut avoir qu'une note de partiel (notePart) pour un même module chaque année. Par contre, il peut avoir plusieurs notes de contrôle continu (noteCC) par module. Pour chacune de ces notes, on souhaite connaître la date de réalisation de l'épreuve et son type (devoir maison, devoir sur table, interrogation...).

Enfin, le service de scolarité souhaite connaître les enseignants qui interviennent dans chaque module, un module pouvant impliquer plusieurs enseignants. Pour chaque enseignant, identifié par son numéro d'enseignant (numen), on souhaite connaître son nom, son prénom, ainsi que son email, son adresse et son téléphone (personnel et professionnel). Un enseignant est lui aussi attaché à une UFR. Chaque UFR est identifiée par son numéro (nUFR) et son nom (nomUFR). Elle a également un directeur, lui aussi enseignant à l'UFR.

1°) Proposez le graphe des dépendances fonctionnelles élémentaires et directes (6 points)

2°) Définissez le schéma de base de données relationnel en 3^{ème} forme normale (3 points)

3°) Donnez la commande SQL nécessaire pour créer la table définissant les formations (1 point)

La maison intelligente (Examen 2013-2014)

Vous êtes chargé de concevoir la base de données pour une maison intelligente qui permet aux habitants de régler la température et l'éclairage à l'aide d'un système informatique.

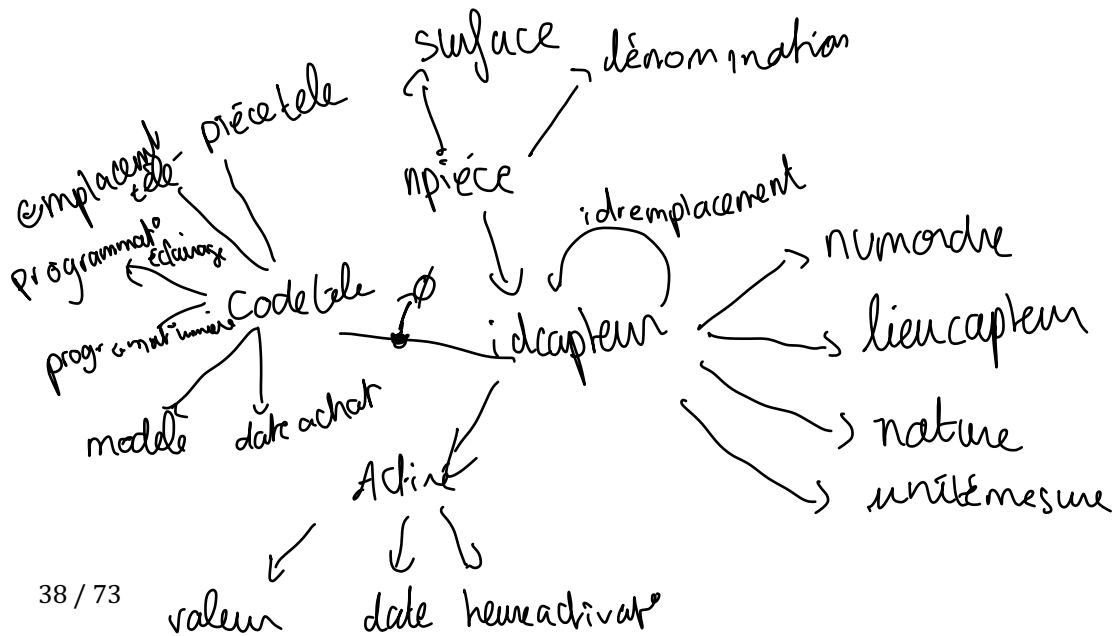
Dans cette maison, des capteurs sont disposés dans les différentes pièces de la maison. Les pièces ont une dénomination (salle de bain, salon...), une surface et possèdent chacune un numéro de pièce (npièce) unique. On doit savoir où le capteur se situe exactement dans la pièce : chaque emplacement dans une pièce est décrit par un numéro d'ordre et un bref texte (p.ex. « au plafond »). Chaque capteur est identifié par un numéro (idcapteur). On doit connaître sa nature précise (température, luminosité, etc.) et l'unité de mesure propre à ce capteur (celsius, lumen...). Par ailleurs, en cas de défaillance, un capteur doit pouvoir être remplacé par un autre capteur de même nature. L'identification de ce capteur de remplacement doit être connue dans la base de données.

Un capteur peut être désactivé ou activé et prend des mesures lorsqu'il est activé : pour chaque mesure réalisée, on dispose d'une valeur et d'une trace temporelle (date et heure) indiquant le moment où la mesure a été réalisée. Cependant, ces mesures étant réalisées très souvent (plusieurs mesures par heure), on ne souhaite pas qu'elles soient identifiées par un quelconque identificateur unique, car elles sont bien trop nombreuses. On souhaite également pouvoir historiser l'état d'un capteur, afin de pouvoir réaliser des statistiques sur ses périodes d'activité.

Les capteurs peuvent être contrôlés par des télécommandes disposées dans la maison. Chaque télécommande est identifiée par un code (`codetele`) et peut servir à programmer différents capteurs. On doit être capable d'identifier quelle télécommande peut être utilisée pour régler un capteur donné et, inversement, quels capteurs une télécommande donnée peut contrôler. Certaines de ces télécommandes sont fixes et placées dans une pièce, alors que d'autres sont mobiles. Pour les télécommandes fixes, on souhaite connaître leur emplacement exact et la pièce dans laquelle elles trouvent. Par ailleurs, on doit connaître le modèle de chaque télécommande et sa date d'achat.

Les télécommandes servent aussi à programmer la température et l'éclairage dans chaque pièce. En effet, les habitants peuvent programmer, pour chaque pièce, la température et l'éclairage qu'ils considèrent comme idéale, selon la plage horaire et le jour de la semaine. Par exemple, pour le salon, on peut programmer une température de 20°C de 18h à 22h de lundi à vendredi, et une température de 18°C de 10h à 12h le samedi et le dimanche. Il y en va de même pour l'éclairage : l'éclairage de l'entrée doit s'allumer automatiquement de 18h30 à 23h de lundi à vendredi. La programmation de chacun de ces paramètres (température et éclairage) doit être conservée dans la base de données.

- 1°) Proposez le graphe des dépendances fonctionnelles élémentaires et directes (6 points)
- 2°) Définissez le schéma relationnel de la base de données en 3^{ème} forme normale (3 points)
- 3°) Donnez la commande SQL nécessaire pour créer la table définissant les capteurs (1 point)



Capteur (idcapteur, nature, unité mesure (idcapteur, n ordre, lieu) idremplacement)
 pièce (npiece, dénominat°, surface)

Emplacement

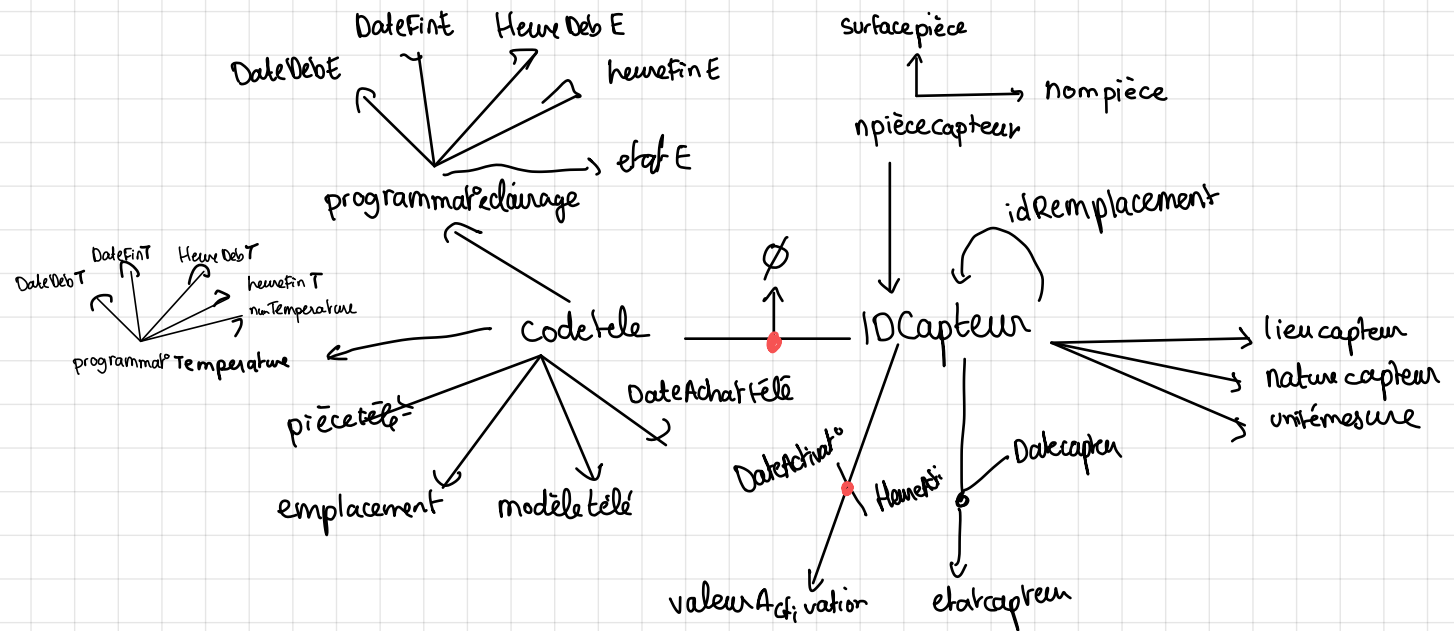
Activation (active, valeur, dateactivation°, heureactivation)

Etat (idcapteur, date, état capteur)

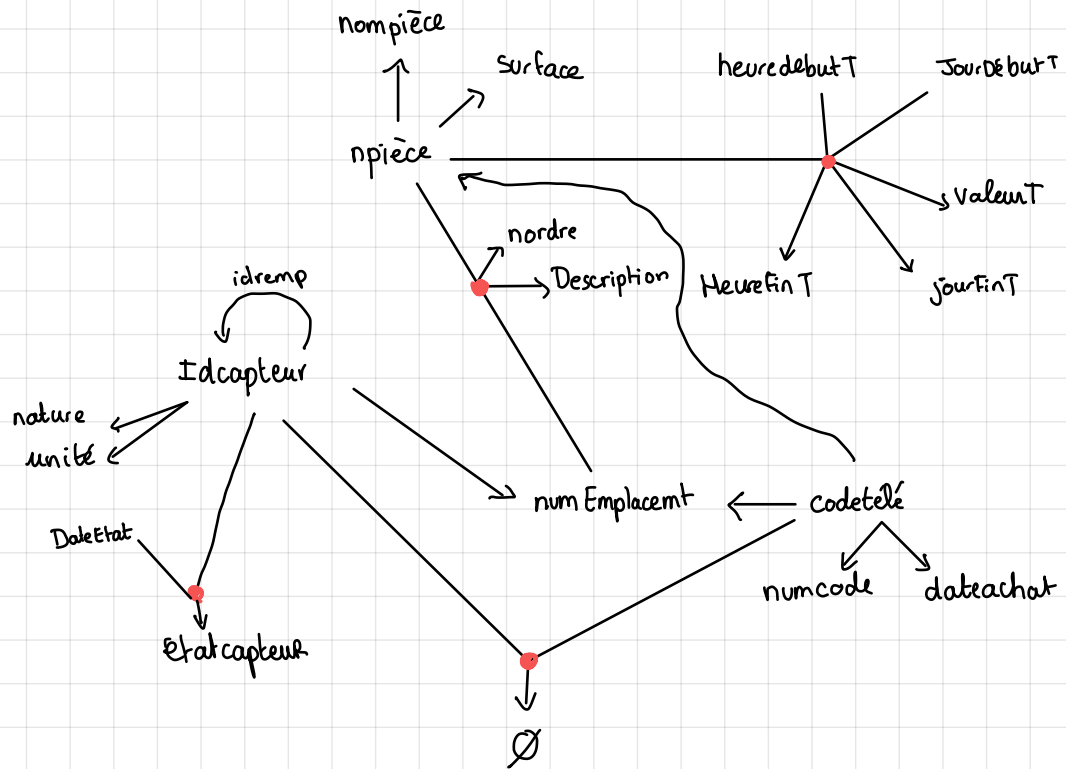
Contrôle Capteur (codetele, idcapteur)

Télécommande (codetele, fixe, mobiles, modèle télé, dateachat)

Fixe (fixe, emplacement, lieu télé,)



Capteur (idcapteur, idrempl, nature, unite, npiece, nempl,)



Create Table Capteur (

idcapteur	INTEGER	Primary Key
idrempl	INTEGER	References Capteur(idcapteur),
unite	VARCHAR(15),	
nature	VARC(1A (25)	
numEmpl	INTEGER,	REFERENCES (Emplacement) .

);

Vente en ligne (Examen 2014-2015)

Le site de vente en ligne « Foire au Numérique » vous demande de concevoir sa nouvelle base de données. Dans celle-ci devront être enregistrées les informations concernant les adhérents, leurs commandes, leurs avis sur les articles ainsi que leur « wish list » (liste d'envies).

Chaque adhérent est identifié par son numéro d'adhérent. Plusieurs informations sur les adhérents doivent être enregistrées : leur nom, prénom, sexe, email, adresse postale, date de naissance et date d'adhésion. Les adhérents peuvent également cumuler plusieurs codes de réduction. A chaque code, correspondent une valeur (en euros) et une période de validité. Par ailleurs, un code de réduction ne peut être associé qu'à un adhérent précis.

Les adhérents passent leurs commandes via le site Internet. Chaque commande est identifiée par un numéro de commande et contient une date et une adresse de livraison (qui peut être différente de l'adresse de l'adhérent). Chaque commande peut contenir plusieurs articles, en différentes quantités. Chaque article est identifié par un numéro et comporte un prix. Ce prix évolue au fil du temps, et on doit garder une trace de cette évolution.

Le magasin souhaite offrir à ses adhérents la possibilité de mensualiser les paiements de leurs commandes, sans frais additionnels. Par exemple, une commande d'un montant total de 120€ peut être payée en 12 mensualités de 10€. Pour chaque mensualité, on doit connaître le montant à payer et la date de paiement.

Un adhérent peut également exprimer sur le site Internet son avis sur les articles (sur tous les articles et pas seulement ceux qu'il a déjà achetés). Pour chaque article, l'adhérent peut donner au maximum un avis qui est enregistré dans la base de données et affiché sur le site. En plus des avis, les adhérents ont une liste d'envie (appelée « wish list »), dans laquelle ils ajoutent les articles qui les intéressent (mais qu'ils n'ont pas forcément achetés).

1°) Proposez le graphe des dépendances fonctionnelles élémentaires et directes (6 points)

2°) Définissez le schéma relationnel de la base de données en 3^{ème} forme normale (3 points)

3°) Donnez la commande SQL nécessaire pour créer la table contenant le contenu des commandes (1 point)

Les commerciaux de AtoB (Examen 2015-2016)

Vous venez d'être embauché par la société AtoB et on vous demande de créer une base de données permettant de mieux gérer le payement des commerciaux. Les commerciaux de AtoB agissent pour le compte d'autres entreprises. Celles-ci représentent les clients de AtoB. Elles sont identifiées par un numéro unique et ont un nom et une raison sociale, ainsi qu'une adresse postale. Chaque entreprise transmet à AtoB un ensemble d'offres commerciales que AtoB se charge de vendre auprès des consommateurs lors des différentes foires et événements auxquels elle participe. Chaque offre est envoyée par une seule entreprise. On doit garder pour chaque offre la date à laquelle elle a été envoyée et sa période de validité. En effet, une entreprise peut transmettre à AtoB une offre bien avant que celle-ci puisse être commercialisée (par exemple, une offre de fin d'année, valable du 1/12 au 31/1, mais déjà transmise à B2B en novembre). A chaque offre, on associera un texte de contrat et une description, ainsi que le montant correspondant à chaque option disponible. Par exemple, l'offre de Noël de l'entreprise SuperCableTélé propose une première option à 19€, puis une seconde à 39,99€, et enfin une troisième à 49,99€. Chaque option de l'offre a, en plus de son montant, un texte contenant ses conditions. On associera également aux offres une valeur de commission qui sera payée aux commerciaux à la signature d'un contrat correspondant à cette offre. Une offre peut aussi proposer des bonus aux commerciaux en fonction du nombre de contrats signés. Ces bonus sont organisés en plusieurs tranches : à chaque tranche [nombre minimum, nombre maximum] correspond un pourcentage. Par exemple, l'offre de Noël de SuperCableTélé paie un bonus de 10% supplémentaires pour les commerciaux ayant signé entre 5 et 10 contrats et 15% pour ceux qui ont signé de 10 à 15 contrats. Il est important de souligner que ces tranches sont propres à chaque offre : différentes offres auront différents paliers de bonus. La base de données doit aussi enregistrer chaque contrat qu'un commercial aura fait signer. Un contrat est identifié par un numéro unique et correspond à une offre. Pour chaque contrat, on doit connaître la date de signature, le montant total du contrat et le commercial qui l'a fait signer. Le montant du salaire de chaque commercial est calculé en fonction des contrats qu'il a fait signer. Un commercial a ainsi un salaire de base auquel s'ajoutent les différentes commissions. Pour des raisons d'audit, on souhaite garder une trace de la valeur de salaire réellement payé à chaque commercial par mois. On enregistre aussi le nom, le téléphone, le mail et l'adresse de chaque commercial. Enfin, ces commerciaux interviennent lors de différents événements. Chaque événement a un nom et se déroule à un endroit précis (local) pendant une période. On doit connaître quels commerciaux participent à chaque événement. Lors d'un événement, on aura à tout moment un « chef de stand » : un commercial responsable du stand AtoB pendant une période donnée d'une journée. On peut avoir plusieurs chefs de stand le même jour durant un événement, mais on n'aura pas deux commerciaux chef de stand en même temps.

1°) Proposez le graphe des dépendances fonctionnelles élémentaires et directes (6 points)

2°) Définissez le schéma de base de données relationnel en 3^{ème} forme normale (3 points)

3°) Donnez la commande SQL nécessaire pour insérer le commercial J.Dupont (tél 0123456789, mail jdp@atob.com) dans la base de données (1 point)

Les prêts immobiliers (Examen 2016-2017)

La banque EMS *banking* offre des prêts à des taux avantageux pour des jeunes diplômés. Suite à une refonte de son Système d'Information, EMS *banking* revoit sa base de données dédiée aux prêts. Vous êtes en charge de la conception de cette nouvelle base.

Afin d'accorder un prêt, la banque sollicite plusieurs informations de la part des demandeurs : nom, date de naissance, profession, salaire mensuel moyen et le taux d'endettement approximatif. Une fois ces informations renseignées, une demande de prêt est créée. Celle-ci est identifiée par un numéro unique et n'est valable que pendant une période donnée. En outre, une demande a un état ('en analyse', 'accordé', 'refusé'...) dont on souhaite garder une trace de l'évolution. On doit aussi enregistrer la durée minimale et maximale souhaitée, le montant demandé et l'apport qui sera fourni. Chaque demande concerne un bien précis, dont on doit enregistrer quelques informations : l'adresse, le type (nouveau ou ancien), le nombre de pièces et la surface (en m²). Par contre, une demande peut concerner plusieurs demandeurs (un couple marié, par exemple). Chaque demandeur doit être enregistré dans la base de données séparément et être associé à la demande de prêt.

Si la demande est acceptée, le prêt sera accordé (aucun prêt n'est accordé sans l'étude au préalable de la demande). EMS *banking* travaille uniquement avec des prêts à taux fixe, dont il faut connaître la valeur du taux et le montant total des intérêts. Pour chaque prêt, on va enregistrer également les dates de toutes les échéances. Pour chaque échéance, on enregistre le montant de la mensualité, des intérêts et l'amortissement sur le montant du prêt.

Enfin, chaque prêt doit être assuré par une police d'assurance, dont on doit connaître une rapide description et les garanties qu'elle couvre. En effet, une police peut couvrir différentes garanties (décès, interruption de travail, etc.) sous certaines conditions (couverture total, franchise 90 jours, etc.). Par exemple, le contrat numéro « XYZ234 » couvre une 1^{ère} garantie « décès » avec une couverture totale (condition), puis une 2^{ème} garantie « ITT/IPT/IPP » avec une franchise de 90 jours (condition). En plus des garanties, on doit aussi connaître les montants des cotisations et les dates auxquelles ces cotisations doivent être versées.

1°) Proposez le graphe des dépendances fonctionnelles élémentaires et directes (6 points)

2°) Définissez le schéma de la base de données relationnelle en 3^{ème} forme normale (3 points)

3°) Donnez la commande SQL nécessaire pour insérer dans la base de données un prêt au taux de 1,2%, d'une durée de 20 ans et d'un montant total d'intérêt de 20 000€. (1 point)

Vidéo à la demande (Examen 2017-2018)

Vous venez d'être engagé par NetEMSix, nouvel opérateur de vidéo à la demande spécialisé dans les films, et on vous demande de concevoir leur toute nouvelle base de données. Celle-ci doit gérer les informations des abonnés, leur facturation ainsi que les films proposés.

La plateforme NetEMSix est accessible uniquement par abonnement. Un abonné est identifié par un numéro unique. Il a aussi un email et un mot de passe. On doit également connaître son nom, sa date de naissance et son adresse. Un abonné peut définir plusieurs profils. Pour chaque profil, on doit connaître l'âge de son utilisateur (utilisé pour limiter l'accès à certains films) et une petite icône (utilisée pour représenter le profil). Chaque profil défini par l'abonné est identifié par un pseudo (*unique dans le cadre de cet abonné*). Par exemple, deux abonnés différents peuvent avoir chacun un profil nommé « maman », mais un abonné X ne peut avoir qu'un seul profil « maman ».

Les films disponibles sur la plateforme sont identifiés par un numéro unique. On enregistre également leur titre, une description, leur année de sortie au cinéma, la date de sortie sur la plateforme, ainsi que l'âge minimal pour le regarder et son prix. Chaque film est fourni par un fournisseur qui en a les droits de diffusion. Chaque fournisseur est identifié par son numéro et on enregistre son nom, son email et un téléphone de contact.

Un film peut appartenir plusieurs catégories. Par exemple, le film « Star Wars VI : Le retour du Jedi » est classé en 'science-fiction' et 'action'. Chaque catégorie est identifiée uniquement par son nom. Les catégories sont également organisées de manière hiérarchique, une catégorie pouvant appartenir à une autre catégorie dite 'parent'. Par exemple, les catégories 'science-fiction' et 'fantasy' appartiennent à la catégorie 'fiction'.

Les films sont organisés en bouquets. Un bouquet est identifié par un numéro de bouquet. Il a un nom et un prix. Chaque bouquet peut contenir plusieurs films et faire aussi partie d'un autre bouquet dit 'parent'. Par exemple, le bouquet 'SciFi' contient tous les films de la saga 'Star Wars' et fait partie du bouquet « Fiction ». Il est important de souligner qu'un même film peut faire partie de plusieurs bouquets. Un abonné peut souscrire à plusieurs bouquets. Pour chaque bouquet, l'abonné décide de la date de début et de fin de la souscription, sachant qu'il peut à tout moment l'arrêter puis recommencer après. Par exemple, l'abonné X a souscrit le bouquet 'SciFi' du 17/12/2016 au 8/1/2017, puis du 16/12/2017 au 7/1/2018.

Les abonnés peuvent choisir les films à la carte. Ils seront alors facturés en conséquence. Ainsi, une fois par mois, chaque abonné reçoit une facture avec les films qu'il a vus. Chaque facture a un numéro et contient un montant total, la date d'émission et la date de paiement. Chaque facture peut comporter plusieurs lignes, représentant les films facturés. Dans chaque ligne, on indique le film visionné, la date et l'heure du visionnage, ainsi que le profil utilisé. Si un film est visionné à plusieurs reprises (par exemple, par plusieurs profils de l'abonné), il doit être mentionné sur autant de lignes différentes, avec les dates, les heures et les profils utilisés à chaque fois.

- 1) Proposez le graphe des dépendances fonctionnelles élémentaires et directes
- 2) Définissez le schéma de la base de données en 3^{ème} forme normale
- 3) Donnez la commande SQL nécessaire pour créer la table classant les films dans les catégories
- 4) Donnez la commande SQL nécessaire pour insérer dans la base de données un nouveau bouquet « Romance », de 15€, faisant partie du bouquet « Comédie »

Partie II – Langages de Requêtes

La bibliothèque

Soit le système d'information décrit par les relations en 3e forme normale suivantes :

Livre (nlivre, titrelivre, datelivre, langue, nbpages, nediteur, ntheme)

Éditeur (nediteur, nomediteur, adrediteur)

Theme (ntheme, nomtheme)

Index (mot-cle, nlivre)

Auteur (nlivre, numecr)

Écrivain (numecr, nomecr, datenais, pays)

Bibliothèque (nbibli, adrbibli, telbibli)

Ouvrage (nbibli, nlivre, nordre, dateachat, état)

Emprunteur (nemp, nomemp, adrempr, telempr, dateadhe, nbemp, nbretards)

Emprunts (nbibli, nlivre, nordre, dateemp, daterendu, nemp)

L'état d'un ouvrage peut être BON, MOYEN ou MAUVAIS

nemp est le nombre d'emprunts effectués depuis l'adhésion (dateadhe); nbretards est le nombre de retards de restitution d'emprunts enregistrés depuis l'adhésion.

Rédiger les requêtes en langage algébrique répondant aux questions suivantes :

1. Liste des numéros de livres publiés chez EYROLLES.
2. Liste des noms des thèmes des livres publiés chez EYROLLES.
3. Liste des numéros de thèmes traités par l'écrivain OLYMPE DE GOUGES.
4. Liste des livres (tous les attributs de chaque livre) de la bibliothèque située à l'adresse 15, rue Cujas, 75005 PARIS).
5. Liste des mots-clés qu'on ne trouve jamais chez l'éditeur MAGNARD.
6. Liste des mots-clés associés uniquement au thème INFORMATIQUE.
7. Liste des livres (nlivre) que la bibliothèque dont le numéro de téléphone est « 12345678 » est la seule à proposer.
8. Liste des livres indexés par POÉSIE et par ALGORITHMIQUE (les deux mots-clés simultanément).
9. Liste des noms et adresses des emprunteurs ayant emprunté « La peste » de Camus ET « Le monde des non-A » de Van Vogt.
10. Liste des éditeurs publiant dans toutes les langues présentes dans le système.
11. Noms et adresses des emprunteurs n'ayant jamais emprunté « La peste » de Camus ni « Le monde des non-A » de Van Vogt.

Les courses de bateaux

La base de données comporte les relations suivantes :

Bateau (nbat, nombat, sponsor)

Course (ncomp, nomcomp, datcomp, prixcomp)

Résultat (nbat, ncomp, score)

Un bateau a un numéro unique (nbat), un nom (nombat) et un sponsor (sponsor). Une course a une clé (ncomp), un nom (nomcomp), une date (datcomp) et un montant de prix pour le gagnant (prixcomp). Un tuple de la relation résultat décrit l'association entre un bateau et une course et le score de ce bateau dans la course (score).

On vous demande de répondre aux questions suivantes en utilisant le langage algébrique. Trouver :

1. les noms des courses auxquelles a participé le bateau nommé « Ville de Paris » et dont le prix est supérieur à 200 000 Fr.
2. les noms des bateaux qui ont été classés 1^o (score = 1) au moins une fois en 1999 et classés au moins une fois troisième en 1998.
3. les bateaux qui n'ont participé à aucune course en 1999.
4. les bateaux qui ont le même sponsor que le bateau nommé « Ville de Paris ».
5. les numéros des bateaux qui ont toujours été classés avant le rang 4.
6. les numéros des bateaux qui ont participé à toutes les courses.
7. les numéros des bateaux qui ont participé à toutes les courses auxquelles « Ville de Paris » a participé.
8. les numéros des bateaux qui n'ont jamais été ni premiers ni derniers dans une course ainsi que ceux qui n'ont jamais couru.
9. la dernière course à laquelle le bateau « Ville de Paris » a participé.

Les courses de chevaux

Soit le Système d'Informations hippiques décrit par les relations en 3FN suivantes :

Cheval (NUMCH, DATNAIS, NOMCH, NUMCHPERE, NUMCHMERE)

Ecurie (NUMEC, NOMPROP, NOMECH)

Couleur (NUMCH, COULEUR)

Propriétaire (NUMCH, date-entrée-écurie, NUMEC)

Haras (NUMH, NOMH, ADRH, TELH)

Entraînement (NUMCH, date-entrée-haras, NUMH)

Jockey (NUMJOCK, NOMJOCK, ADRJOCK)

Course (NUMCOUR, NOMCOUR, DATECOUR, HEURECOUR, TYPECOUR)

Résultat (NUMCH, NUMCOUR, NUMJOCK, SCORE)

JockeyEcurie (NUMJOCK, date-entrée-écurie-jock, NUMEC)

Contraintes d'intégrités :

Domaine (NUMCHPERE) = domaine(NUMCH)

Domaine (NUMCHMERE) = domaine(NUMCH)

En utilisant les langages, définir les 10 ensembles suivants (si cela est possible):

1. Liste des jockeys (nom et adresse).
2. Liste des courses de type « handicap-1600m » (nom, date, heure).
3. Liste des résultats de la course « Prix de Houdan » du 6 mars 1987 (n° cheval, n° jockey, score).
4. ID à 3, mais : (nom du cheval, nom du jockey, et score).
5. Liste des vainqueurs (score=1) du "Prix de l'Arc de Triomphe" depuis 1950 (nom du cheval et date).
6. Liste des chevaux (numéros) ayant toujours appartenu à l'écurie n° 357.
7. Liste des chevaux (noms) ayant toujours été dans les 4 premiers à l'arrivée des courses de type « steeple-chase-3500m » auxquelles ils ont participé.
8. Liste des chevaux (noms) n'ayant jamais eu la couleur jaune, ni la couleur bleue dans les couleurs des écuries les ayant possédés.
9. Liste des chevaux (noms), avec le nom de leur propriétaire actuel (nom de l'écurie).

Les Auteurs

Les informations concernant une base d'articles scientifiques sont stockées dans le schéma relationnel suivant :

ARTICLE (numarticle, titre, annee)

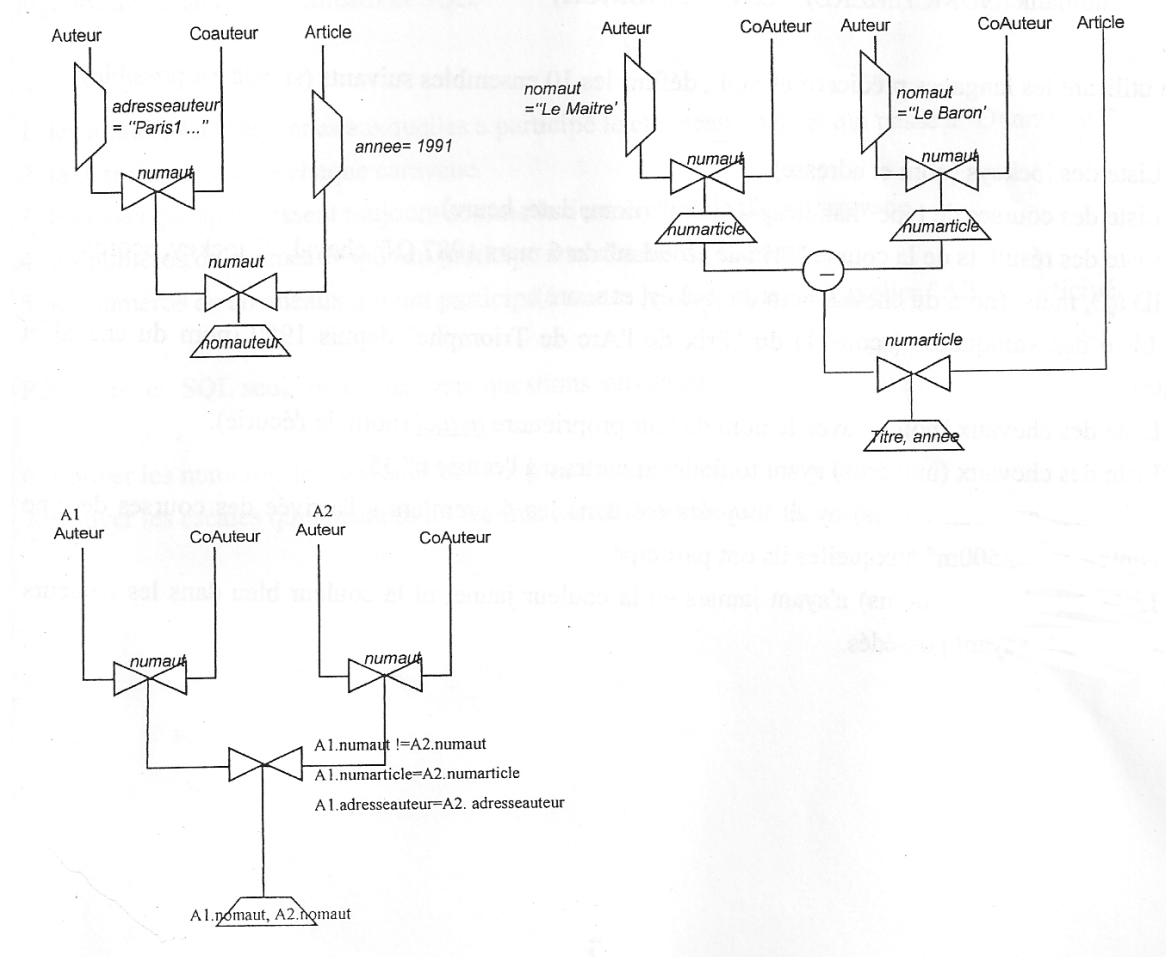
COAUTEUR (numarticle[#], numaut[#])

AUTEUR (numaut, nomaut, prenomaut, adresseauteur)

On vous demande d'écrire les requêtes suivantes en SQL :

1. Quels sont les auteurs qui n'ont écrit aucun article.
2. Donner les titres des articles dont les auteurs ne sont pas référencés dans la BD.
3. Donner les noms des auteurs qui n'ont jamais écrit un article seul.
4. Donner les titres des articles rédigés par un seul auteur.

On vous demande de traduire les requêtes algébriques suivantes en SQL :



Festival de Cannes

La base de données comporte les relations suivantes :

Festivaliers(numeroF, nom, adress, tel, profession, nationalité, typef)

Film(NomF, titre, nationalité, producteur)

Séance(numeroS, nomF, dateséance, heureséance, salle)

Fest-scéance(numeroF[#], numeroS[#])

Attributs : typef est le type de festivalier (membre du jury, représentant d'un film, professionnel, amateur)

On supposera que le festival ne programme aucune séance en parallèle.

On vous demande de répondre aux questions suivantes en utilisant successivement le langage algébrique et SQL. Trouver :

- 1) les festivaliers « membre du jury » en mentionnant leur nom, leur nationalité ainsi que leur profession.
- 2) les festivaliers qui ont vu tous les films du festival.
- 3) les festivaliers « membre de jury » qui n'ont pas vu tous les films.
- 4) Les festivaliers qui ont vu que des films de nationalité « française » ou qui n'ont vu aucun film de nationalité « américaine ».
- 5) Les films qui ont été vu par tous les membres du jury.
- 6) Le film qui a été vu en dernier.

Répondre en SQL seulement, aux questions suivantes :

- 7) Donner le nombre de festivaliers par nationalité.
- 8) Donner le nombre de séances par film.
- 9) Donner le nombre de festivaliers par film.

Windsurf-club de la côte de Rêve

Soit les relations de la base de données du Windsurf-club de la côte de Rêve :

Spot (numspot, nomspot, exposition, type, note)

Planchiste (numpers, nom, prenom, niveau)

Matériel (nummat, marque, type, longueur, volume, poids)

Vent (date, numspot[#], direction, force)

Navigue (date, numpers[#], nummat[#], numspot[#])

Les spots sont les bons coins pour faire de la planche à voile. Chaque spot est décrit dans la relation *Spot* par un nom, l'exposition principale par exemple « sud ouest », le type par exemple « slalom » ou « vague » et une note d'appréciation comprise entre 0 et 20.

Les planchistes sont les membres du club et les invités. On notera pour chaque planchiste son niveau variant de « débutant » à « compétition » en passant par « confirmé ». La relation *Matériel* référence la description des planches utilisées (pour simplifier voiles, ailerons, mats, wishbone ne sont pas représentés). Chaque matériel est décrit par la marque de la planche, le type de planche (slalom, vague, vitesse ...), la longueur, le volume et le poids de la planche.

La relation *Vent* garde trace, pour chaque jour de l'année, de la force et de la direction du vent constatées sur chaque spot. La relation *Navigue* enregistre chaque sortie d'un planchiste sur un spot à une date donnée avec le matériel utilisé. Pour simplifier, on suppose qu'un planchiste n'effectue qu'une sortie par jour et ne change ni de matériel ni de spot dans une journée.

Questions :

Partie 1

Exprimez les requêtes suivantes à l'aide de l'algèbre relationnelle et du langage SQL.

- 1) Donner le nom et le niveau des planchistes qui ont toujours navigué avec un vent de force supérieur à 5.
- 2) Donner pour chaque planchiste (numpers) le descriptif du matériel utilisé (marque et type) lors de sa dernière sortie.
- 3) Donner le nom des planchistes « confirmés » qui ont déjà navigué à « L'Almanarre » ou à « La Torche ».
- 4) Donner le nom et le prénom des planchistes qui ont déjà navigué avec toutes les marques de planches.
- 5) Donner le nom et le prénom des planchistes qui ont navigué sur les mêmes spots que le planchiste « Dupond ».

Partie 2

Exprimez les requêtes suivantes en SQL.

- 6) Donner pour chaque planchiste de niveau « compétition » (numpers), le nombre de jours de navigation et le nombre de marques de planches utilisées durant l'année 1999.
- 7) Donner pour chaque spot (numspot) le nombre de jours (de l'année 1999) où le vent dépassait force 5 et la force moyenne pour l'année 1999.
- 8) Donner le nom des planchistes « confirmés » qui ont effectué plus de 30 jours de navigation durant l'année 1999 en utilisant plus de 2 marques de planches.

Les aventures de Vil Coyote



Vous connaissez tous les terribles aventures de Vil Coyote et de Bip-Bip. Dans ce dessin animé, le coyote (Carnivorous Vulgaris) essayait toujours de s'offrir un bon repas en attrapant le road runner (Accelerati Incredibilus). Celui-ci était tellement rapide qu'il réussissait toujours à s'échapper en faisant son célèbre cri « Beep Beep ! ». Pour tenter de l'attraper, Vil Coyote commandait un certain nombre de pièges à l'usine Acmé mais ceux-ci se retournaient systématiquement contre lui. Nous avons inventorié ici les différents épisodes de la série, les pièges qui y sont utilisés et les accessoires faisant partie de ceux-ci. Les auteurs sont également répertoriés dans la base de données.

Piege (npiege, nompiege, descriptionP, efficacité-relative)

Apparition (npiege[#], nepisode[#], nordre, efficacité-réelle)

Épisode (nepisode, date-premiere-diffusion, accueil-public)

Accessoire (nacc, nomacc, descriptionA)

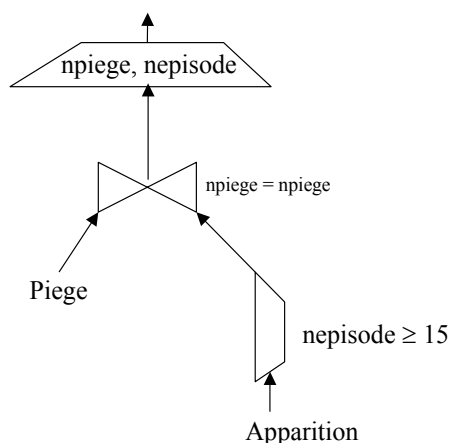
Utilisation (npiege[#], nacc[#])

Créateur (ncreateur, nomcreateur, prenom, age)

Auteur (nepisode[#], ncreateur[#])

Interprétez les requêtes suivantes en exprimant leur signification en langage naturel. Corrigez-les si nécessaire.

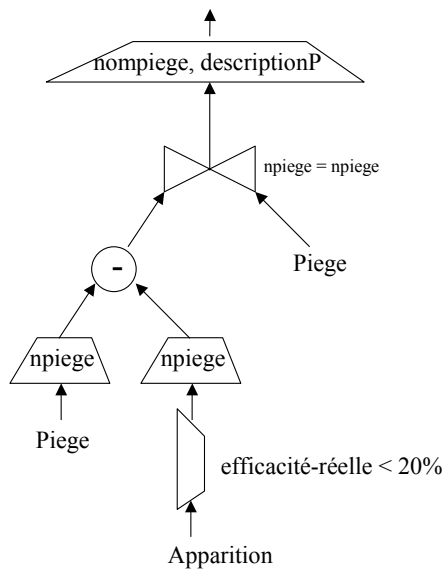
1.



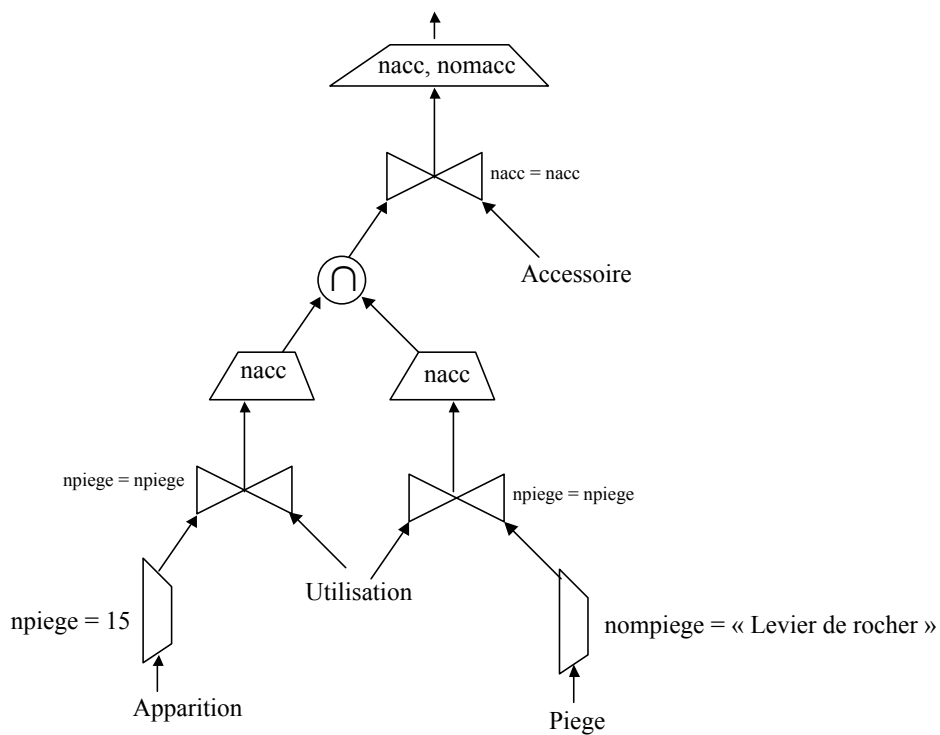
2.

Select A.nacc, A.nomacc
 From Accessoire A, Utilisation U, Piege P
 Where A.nacc = U.nacc
 And U.npiege = P.npiege
 And P.nompiege = « tunnel sous montagne » ;

3.



5.



6.

Select C.nomcreateur, C.prenomcreateur

From Createur C, Auteur A

Where C.ncreateur = A.ncreateur

And A.nepisode = 46

And C.ncreateur in

(Select A. ncreateur

From Auteur A, Apparition App, Utilisation U, Accessoire Acc

Where A.nepisode = App.nepisode And U.npiege = App.npiege And U.nacc = 489);

4.

Select A.*

From Accessoire A

Where A.nacc not in

(Select U.nacc

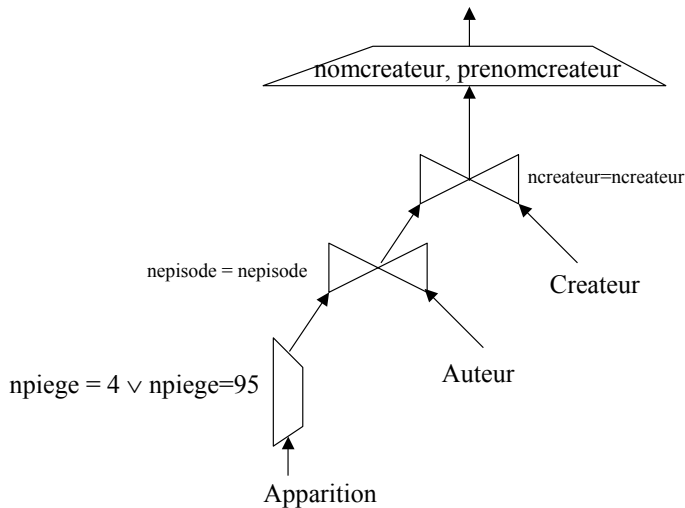
From Utilisation U, Apparition A, Episode E

Where U.npiege=A.npiege

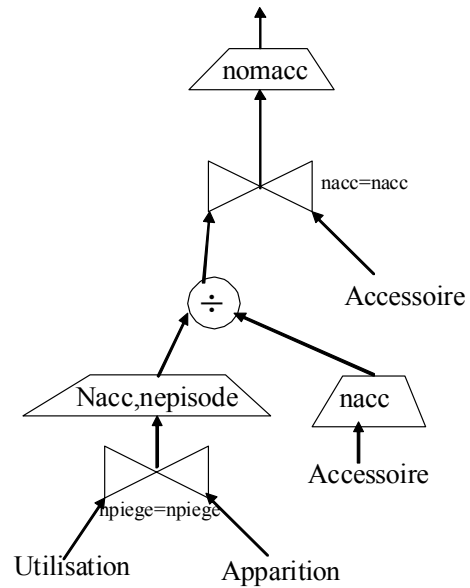
And A.nepisode=E.nepisode

And E.date-premiere-diffusion < 01/05/1990) ;

7.



8.



9.

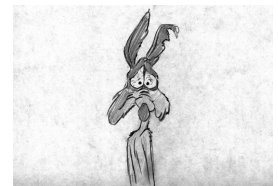
Select P.nompiege
 From Piege P
 Where not exists
 (Select E.*
 From Episode E
 Where E. accueil-public $\geq$ 75%
 And not exists
 (Select A.*
 From Apparition A
 Where A.nepisode = E.nepisode
 And A.npiege = P.npiege));

11.

Select P.nompiege
 From Piege P, Episode E, Apparition A
 Where E.date-premiere-diffusion $\geq$ 01/01/2002
 And E.nepisode = A.nepisode
 And A.npiege = P.npiege
 Order by P.nompiege Desc;

10.

Select A.nepisode, count (distinct U.nacc)
 From Utilisation U, Apparition A
 Where U.npiege=A.npiege
 Group by A.nepisode;



12.

Select P.npiege
 From Piege P, Apparition A
 Where P.efficacité-relative=75%
 And A.npiege = P.npiege
 Group by P.npiege
 Having avg (A.efficacité-réelle)=5%;

Les marchés

Soit les relations suivantes :

Commerçant (ncom, nomcom, type-produit)

Marché (nmar, nommarché, ville)

Emplacement (nmar[#], nemp, surface)

Location (ncom[#], nmar[#], nemp[#], date, prix)

Un commerçant peut louer un ou plusieurs emplacements dans un marché à une date donnée. Les numéros d'emplacements sont relatifs au marché.

1. Lister les noms et numéros des commerçants qui ont vendu des produits de type « fromage » sur le marché parisien de nom « Turbigo ».
2. Lister les noms, numéros et type-produits des commerçants n'ayant jamais effectué de location.
3. Lister les noms des commerçants ayant effectué au moins une location à la fois au marché « Turbigo » ET au marché « Aligre », ces deux marchés sont à Paris.
4. Lister les emplacements (avec le nom du marché et la ville où ils se trouvent) qui n'ont jamais été loués par des commerçants vendant des produits de type « fromage » ou « viande ».
5. Lister les marchés de la ville de « Lyon » dont des emplacements ayant une surface supérieure à 20 m n'ont pas été loués le 20/10/95, mais ont été loués le 25/10/95.

Pour les questions 6 à 10, utilisez seulement le langage SQL.

6. Calculer la surface totale des emplacements du marché parisien « Turbigo ».
7. Calculer le nombre de marchés parisiens pour lesquels il y a eu plus de 10 locations le 20/10/95.
8. Calculer la moyenne du prix de location des emplacements du marché parisien « Turbigo ».
9. Lister les noms des commerçants ayant effectué au moins une location dans tous les marchés parisiens.
10. Augmenter de 10 % les prix de tous les emplacements des marchés parisiens ayant une surface supérieure à 20 m.

Gestion de commandes

La base de données relationnelle "GESCOM" est décrite par les schémas de relations suivantes :

Client (CODECLI, NOMC, CATC, VILC)

Article (CODEART, NOMA, COULEUR, QTESTK)

Commande (NUMCOM, CODECLI, DATECOM)

DetailCo (NUMCOM[#], CODEART[#], QTECOMD)

Les données gérées par l'entreprise contiennent des informations concernant :

- Les clients sont identifiés de manière unique par leurs codes,
- Les articles sont identifiés de manière unique par leurs codes,
- Les commandes sont identifiées de manière unique par leurs numéros,
- Le détail des commandes, chaque ligne de la table DetailCo représente le numéro d'une commande, le numéro de l'article commandé et la quantité commandée. La combinaison (NUMCOM, CODEART) permet d'identifier chacune des lignes de manière unique.

Dictionnaire de Données :

CODECLI	Code du client	CARACTÈRES(4)
NOMC	Nom du client	CARACTÈRES(10)
CATC	Catégorie du client	NUMÉRIQUE(1)
VILC	Ville du client (son adresse)	CARACTÈRES(10)
CODEART	Code de l'article	CARACTÈRES(4)
NOMA	Nom de l'article	CARACTÈRES(10)
COULEUR	Couleur de l'article	CARACTERES(10)
QTESTK	Quantité en stock de l'article	NUMÉRIQUE(3)
NUMCOM	Numéro de la commande	NUMÉRIQUE(6)
DATECOM	Date de la commande	DATE
QTECOMD	Quantité commandée	NUMÉRIQUE(3)

On vous demande d'écrire les requêtes suivantes :

1. Donnez la liste des codes (CODECLI) et des catégories (CATC) de tous les clients.
2. Donnez la liste des clients (CODECLI, NOMC, CATC, VILC) dont la catégorie est 3.
3. Donnez la liste des numéros des clients (CODECLI) qui habitent Paris, dont la catégorie est supérieure à 1 et qui ont commandé des clous.
4. Donnez la liste des articles (CODEART, NOMA) de couleur mauve qui n'ont jamais été commandés.
5. Donnez la liste des noms des clients (NOMC) ayant fait au moins une commande en septembre 1994.
6. Donnez la liste des noms des articles (NOMA) qui figurent sur la commande de numéro 940817 en quantité supérieure à 10.
7. Donnez la liste des noms des articles (NOMA) qui figurent sur les commandes du mois de septembre 1994, mais pas sur celles du mois d'août.
8. Donnez la liste des noms des articles (NOMA) commandés par les clients de Paris.
9. Donnez la liste des codes des clients (CODECLI) qui n'ont pas fait de commande en septembre 1994.
10. Donnez la liste des clients ayant effectué plus de deux commandes ayant chacune plus de 20 articles différents commandés
11. Donnez la liste des noms des clients (NOMC) ayant effectué au moins une commande en août et en septembre 1994.
12. Donnez la liste des codes des clients (CODECLI) parisiens qui ont commandé plus de 300 clous au total en 1994.
13. Donnez la liste des dernières commandes effectuées par les clients parisiens
14. Donnez la liste des noms des clients (NOMC) ayant commandé tous les articles rouges.
15. Donnez la liste des noms de clients (NOMC) n'ayant jamais commandé d'article rouge
16. Donnez le numéro de la dernière commande de clou effectuée par le client de numéro C003.
17. Donnez la liste des produits commandés par tous les clients
18. Donnez la moyenne des quantités en stock pour l'ensemble des articles.
19. Donnez la liste des noms des articles (NOMA) dont la quantité en stock est supérieure à la moyenne.
20. Donnez le nom du client qui a commandé le plus de clous en 1994.

Gestion de news (Examen 2008-2009)

Le système Intranet que vous avez conçu devra en plus intégrer un système de gestion de nouvelles (news) qui seront affichés dans le bandeau déroulant en haut des pages Web gérées par votre base de données. Le schéma relationnel est le suivant :

Dictionnaire(motCle)

News (idNews, dateCréation, Motcle#, idNewsRemplacement#)

SequenceAffichage (idSequence, DateDébutValidité, DateFinValidité)

OrdreAffichage (idSequence#, numOrdre, idNews#)

Rédigez les requêtes suivantes :

- 1°) Liste des news de sport affichées dans la séquence valide le 23 Janvier 2009
- 2°) Liste des mots-clés du dictionnaire qui ne sont utilisés pour indexer aucune news
- 3°) Liste des news qui sont utilisées pour le remplacement de plusieurs autres (deux au moins)
- 4°) Liste des séquences qui affichent des news indexées par tous les mots-clés commençant par la lettre s
- 5°) Nombre de news affichées par séquence

Gestion de Vœux (Exercice inspiré de l'examen 2009-2010)

Soit le schéma relationnel d'une base de données que vous avez conçue pour gérer les vœux que vous envoyez à vos contacts pour diverses fêtes :

Personne (npers, nom, contact)

Voeux (npers, dateEnvoi, nFete, message)

Fête (nFête, date, fréquence, occasion)

avec

fréquence = "unique", "mensuel", "annuel", "tous les 10 ans"

occasion = "anniversaire", "Noël", "Hanouka", "Aïd", "jour de l'an"

Questions initiales de l'examen : Exprimez en langage algébrique et SQL les requêtes suivantes :

- 1) Occasion et date d'envoi des vœux envoyés à Jean Dupont
- 2) Liste des occasions de fêtes pour lesquelles aucun vœu n'a jamais été envoyé
- 3) Nom des personnes à qui l'on a envoyé des vœux pour toutes les fêtes
- 4) Nom des personnes à qui l'on a envoyé des vœux pour toutes les fêtes que l'on a souhaitées à Jean Dupont
- 5) Date du vœu le plus récent
- 6) SQL seul : Nombre de personnes différentes à qui l'on a envoyé des vœux par occasion de fête

Questions supplémentaires pour vous exercer :

- 7) Nom des personnes qui ont reçu des vœux pour Noël et pour le Nouvel An
- 8) Nom des personnes qui ont reçu des vœux pour Noël ou pour le Nouvel An
- 9) Occasion du vœu le plus récent
- 10) SQL seul : Occasions de fêtes pour lesquelles on a envoyé des vœux à + de 20 personnes

Réservation de places de théâtre (Examen 2010-2011)

On vous propose maintenant de travailler avec la base de données des réservations de places de théâtre suivante :

Séance (nPiece, dateSéance, prixUnitairePlace)

Client (nClient, nomClient, telClient)

Réservation (nClient, nPiece, dateSéance, nombrePlaces, montantRégulé)

NB : un numéro de téléphone n'est pas un chiffre !

1°) Donnez les instructions SQL de création des 3 tables correspondantes

2°) Quelle instruction SQL qui permet de garantir la persistance du schéma que vous venez de créer ?

Veuillez donner les requêtes qui permettent d'obtenir :

3°) Liste des réservations (tous les attributs) dont le montant réglé est encore inférieur au prix des places réservées

4°) Liste des clients (tous les attributs) qui ont fait une réservation pour toutes les pièces présentées en 2010

5°) Liste des clients (tous les attributs) qui n'ont jamais fait de réservation de plus de 2 places

6°) Montant moyen des réservations par client dont le numéro de téléphone commence par "01"

Base d'Annonces d'Emplois (Examen 2011-2012)

On vous propose maintenant de travailler avec le schéma de bases de données ci-dessous. Celui-ci décrit une base d'annonces d'offres d'emplois publiés dans des journaux et les candidats qui répondent à ces offres.

Annonce (numAn, numOffre, numJour, datPubli, duree, texte)

Journal (numJour, titre, adrWeb)

Offre (numOffre, desc, poste)

Candidat (numCand, nom, email, age, ville, pageWeb)

Reponses (numCand, numAn, datRecpCV)

Répondre aux questions ci-dessous utilisant le langage algébrique et le langage SQL :

- 1) Lister les noms et les emails des candidats qui ont répondu aux offres pour le poste de directeur publiées le 01/12/2011 (2,0 pts)
- 2) Lister les journaux (titre) qui n'ont publié aucune offre entre le 01/09/2011 et le 01/12/2011 (2,0 pts)
- 3) Lister le numéro, le nom et l'âge du plus jeune candidat qui a répondu à une annonce (2,0 pts)
- 4) Lister les offres (poste et description) qui ont été publiées dans tous les journaux disponibles (2,0 pts)

Répondre aux questions ci-dessous utilisant le langage SQL :

- 5) Lister le nombre d'offres publiées par journal (1,0 pt)
- 6) Donner la commande SQL nécessaire pour la création de la table Offre, avec les contraintes qui s'y appliquent (1,0 pts)

Gestion des stages (Examen GEE 2012-2013)

Vous êtes responsable de la gestion des stages des étudiants de l'université Paris 1.

Les données dont vous disposez sont stockées dans les tables suivantes :

ETUDIANT (nEtu, nomEtu, dateNaiss, formation)

ENTREPRISE (codeEntr, nomEntr, ville, directeur)

STAGE (nEtu, codeEntr, dateDebStage, numOffre, dateFinStage)

OFFRE (numOffre, descrPoste, salMensuel)

Les clés primaires sont soulignées en trait plein, les clés étrangères en pointillés.

Un étudiant possède un numéro d'étudiant, un nom, une date de naissance. Il suit une formation.

Une entreprise possède un code, un nom, un directeur et est implantée dans une ville.

Un stage démarre à la date de début de stage, pour un étudiant donné, dans une entreprise particulière et se termine à la date de fin de stage. Chaque stage correspond à une offre de stage spécifique.

Une offre de stage est décrite par un numéro d'offre, et indique le poste et le salaire mensuel du stagiaire attendu.

Traduire les requêtes suivantes (en algèbre relationnelle et/ou en SQL selon les consignes pour chaque question) :

- 1) **En algèbre relationnelle et en SQL** : Noms des étudiants de L3 Gestion qui ont effectué un stage entre le 1^{er} Juillet 2011 et le 31 Août 2011.
- 2) **En algèbre relationnelle et en SQL** : Numéros des offres de stage qui n'ont pas été pourvues en 2011.
- 3) **En algèbre relationnelle et en SQL** : Noms des étudiants qui ont effectué un stage à Alcatel et à Thalès.
- 4) **En algèbre relationnelle et en SQL** : Noms des étudiants qui ont effectué un stage dans toutes les entreprises.
- 5) **En algèbre relationnelle et en SQL** : Nom de l'entreprise (ou des entreprises) qui propose(nt) le salaire mensuel le plus élevé.
- 6) **En SQL** : Noms des étudiants, par ordre alphabétique, qui ont déjà obtenu un salaire mensuel supérieur à la moyenne des stages proposés.
- 7) **En SQL** : Nombre de stages proposés pour chaque formation.

Produits innovants (Examen 2013-2014)

On a stocké dans une base de données les informations relatives à des produits innovants, conçus par des chercheurs et faisant l'objet d'un ou plusieurs brevets.

Chercheur (n°SS, nom, dateNaiss, nationalité)

Produit (n°prod, nomProd, prix, catégorie)

Brevet (n°brev, intitulé, date, pays)

Invention (n°brev, n°prod)

Propriétaire (n°brev, n°SS)

Les clés primaires sont soulignées par un trait plein, les clés étrangères en pointillés.

Un chercheur est identifié par son numéro de sécurité sociale, et est propriétaire d'au moins un brevet et éventuellement plusieurs.

Chaque brevet, identifié par un numéro de brevet, est valable dans un pays spécifique. Un brevet peut s'appliquer à plusieurs produits, et réciproquement, un produit peut faire l'objet de plusieurs brevets.

Un produit est identifié par un numéro de produit, et appartient à une catégorie (par exemple : électronique, décoration, cosmétique, etc.).

Exprimez en algèbre relationnelle et en SQL les requêtes suivantes :

- 1) Nom des produits cosmétiques de prix supérieur à 200 euros, ayant fait l'objet d'au moins 1 brevet (2 pts)
- 2) Nom des chercheurs ayant contribué à des produits de décoration ou à des produits cosmétiques (2 pts)
- 3) Nom des chercheurs ayant contribué à la fois à des produits de décoration et à des produits cosmétiques (2 pts)
- 4) Nationalité des chercheurs n'ayant déposé que des brevets américains (USA) (2 pts)
- 5) Nom des chercheurs ayant contribué à tous les produits de bricolage (3 pts)
- 6) Intitulé des brevets du chercheur le plus jeune (3 pts).

Exprimez en SQL uniquement les requêtes suivantes :

- 7) Nombre de brevets déposés par le chercheur Géo Trouverien (2 pts)
- 8) Nom et prix du (ou des) produit(s) le(s) moins cher(s) protégé(s) par un brevet français, triés par ordre alphabétique (2 pts)
- 9) Prix moyen des produits de chaque catégorie, pour les catégories comportant plus de 50 produits (2 pts)

Projets de lois (Examen GEE 2013-2014)

On considère dans cet exercice une base de données permettant de stocker les informations sur les votes relatifs aux projets de loi à l'Assemblée Nationale. Un projet de loi, constitué d'articles (numérotés 1, 2, 3 etc.), est proposé à une date donnée par un ministre. Lors du vote d'un projet, les députés présents en séance votent « pour », « contre » ou « abstention ». Un député appartient à une circonscription (numérotée 1, 2, 3 etc.) d'un département. Un ministre peut être en charge de plusieurs ministères au cours de sa carrière et on conserve l'historique de ses fonctions (avec sa date de nomination).

Les tables de la base sont les suivantes :

ProjetLoi (numProjet, intitulé, dateProjet, numMinistre)

Article (numProjet, numArticle, contenu)

Ministre (numMinistre, nomMinistre, dateNaissance)

HistMinistre (numMinistre, dateNomin, nomMinistère)

Député (numDéputé, nomDéputé, Circonscription, Département)

VoteDéputé (numDéputé, numProjet, vote)

Les clés primaires sont soulignées d'un trait plein, les clés étrangères sont soulignées en pointillés.

Traduire en algèbre relationnelle et en SQL les requêtes suivantes :

1. Nom et date de naissance des ministres qui ont été à la fois ministre des finances et ministre de la culture lors de leur carrière politique **(1 pt)**.
2. Nom du député de la 3^e circonscription de Seine-Saint-Denis **(1 pt)**.
3. Nom des députés qui n'ont jamais voté « contre » à un projet **(1 pt)**.
4. Nom du ministre de l'écologie nommé le 2 avril 2014 **(1 pt)**.
5. Nom du ministre ayant proposé le projet de loi le plus récent, et intitulé de ce projet **(1,5 pt)**.
6. Département des députés ayant participé au vote sur tous les projets proposés en 2013 **(1,5 pt)**.

Traduire en SQL uniquement, les requêtes suivantes :

7. Nom des ministres dont le nom du ministère contient le mot « éducation », triés par date de nomination décroissante **(1 pt)**.
8. Nombre d'articles par projet de loi **(1 pt)**.
9. Nom des députés ayant participé au moins à 10 votes **(1 pt)**.

Livraison de colis (Examen 2014-2015)

On considère le schéma relationnel suivant, qui décrit la base de données d'un service de livraisons.

Facture (nfact , datefact , nomClient , prix , adrfact , villefact, adrlivr, villelivr)

Colis (ncolis , nfact , nlivr , type, prixAssurance)

Livraison (nlivr , nemp , dateLivr)

Employé (nemp , nomEmp , dateEmbauche)

Les attributs « adrfact » et « villefact » décrivent l'adresse et la ville de facturation, alors que les attributs « adrlivr » et « villelivr » décrivent l'adresse et la ville de livraison. Ces 2 adresses sont spécifiées lors de la commande et apparaissent sur la facture.

L'attribut « type » décrit le type de colis (colissimo, chronopost, etc.).

L'attribut « prixAssurance » indique le montant de l'assurance de chaque colis. Cette valeur est égale à 0 si aucune assurance n'a été contractée.

Répondez aux questions ci-dessous en utilisant **le langage algébrique et le langage SQL** :

- 1) Nom des clients ayant déjà commandé un colissimo avec une adresse de livraison différente de l'adresse de facturation. (1,0 point)
- 2) Nom de l'employé ayant réalisé la dernière livraison de colissimo (1,5 pts)
- 3) Date d'embauche des employés ayant livré des colis de tous les types (2,0 pts)
- 4) Nom des employés n'ayant livré que des chronopost en 2014 (2,0 pts)
- 5) Nom des employés qui ont effectué des livraisons à Paris et à Marseille. (2,0 pts)

Répondez aux questions ci-dessous en utilisant **le langage SQL uniquement** :

- 6) Nombre de livraisons par employé (nom et numéro de l'employé), trié par ordre alphabétique (0,5 pts)
- 7) Prix moyen des assurances par facture comportant plus de 2 colis (1 point)

Achat de vêtements au détail (Examen 2015-2016)

Vous êtes responsable d'une marque de vêtements, qui possède plusieurs boutiques en France. Vous gérez la base de données où sont stockées toutes les informations relatives aux achats et à l'état des stocks dans les différentes boutiques.

Chaque boutique est identifiée par un numéro, et on connaît son adresse, sa ville et son département.

Un vêtement possède un numéro (celui qui apparaît sur le code barre), ainsi qu'un type (robe, jupe, pantalon, veste, chemise, etc.), une couleur, une taille (par exemple XS, S, M, etc.) et un prix unitaire.

Pour chaque boutique, on connaît le nombre d'exemplaires de chaque vêtement (indiqué par `qttéStock`). Cette quantité peut être nulle pour un vêtement et une boutique donnés, si la boutique ne possède à cet instant aucun exemplaire du vêtement en question.

L'historique des achats effectués par les clients dans les différentes boutiques de la chaîne est également enregistré dans la base de données. Il est possible d'acheter plusieurs exemplaires d'un même vêtement lors d'un même achat : le nombre d'exemplaires est alors indiqué dans l'attribut `qttéAchat`.

Les tables de la base sont les suivantes :

Vêtement (`nVet`, type, couleur, taille, prix)

Boutique (`nBout`, adr, ville, département)

Stock (`nBout`, `nVet`, `qttéStock`)

Client (`nCli`, `nomCli`, `dateNaiss`, `villeCli`)

Achat (`nCli`, `nVet`, `nBout`, `date`, `qttéAchat`)

Les clés primaires sont soulignées d'un trait plein, les clés étrangères sont soulignées en pointillés.

Traduire en algèbre relationnelle et en SQL les requêtes suivantes :

1. Ville des boutiques ayant en stock au moins un pantalon bleu et une jupe noire **(1 pt)**.
2. Ville de résidence des clients ayant effectué en octobre 2015 des achats dans les boutiques de Paris ou de Lyon **(1 pt)**.
3. Ville et département des boutiques n'ayant vendu que des vêtements de taille XL **(1 pt)**.
4. Type et couleur des vêtements achetés par le client le plus jeune (on suppose que le client le plus jeune a bien effectué des achats) **(1,5 pts)**.
5. Numéros et prix des vêtements qui sont disponibles dans toutes les boutiques de Marseille **(1,5 pts)**.

Traduire en SQL uniquement les requêtes suivantes :

6. Nombre de villes différentes dans lesquelles M. Dupont a acheté des chemises **(1 pt)**.
7. Nom des clients ayant acheté des vêtements dont la couleur contient le mot rose (par exemple rose vif, rose pâle, vieux rose), triés par prix décroissant de ces vêtements **(1,5 pt)**.
8. Montant total dépensé par chaque client parisien (uniquement pour les clients ayant acheté au total plus de 10 vêtements) **(1,5 pt)**.

Réseau d'agences immobilières (Examen 2016-2017)

Vous êtes responsable de la base de données d'un réseau d'agences immobilières dédiées à la location de logements.

Chaque agence est identifiée par un numéro et possède un nom, et on connaît son adresse, sa ville et son numéro de téléphone.

Les logements, identifiés par un numéro, sont caractérisés par un type (appartement, maison, parking...), une adresse, une ville, une surface et le montant du loyer mensuel associé. Les agences ont des mandats sur les logements afin de pouvoir assurer les visites des clients.

Les clients sont identifiés de manière unique par un numéro, et on stocke leur nom, leur nationalité et leurs coordonnées (téléphone et email).

Chaque visite de logement par un client est organisée par une agence particulière, à une date et une heure données.

La base de données contient également les informations relatives à chaque bail de location associant un client à un logement et indiquant la date de début et de fin du bail.

Les tables de la base sont les suivantes :

Logement (nLog, type, adrLog, villeLog, surf, loyer)

Agence (nAg, nomAg, adrAg, villeAg, telAg)

Client (nCli, nomCli, nationalite, emailCli, telCli)

Mandat (nAg, nLog)

Visite (nCli, nLog, date, nAg, heure)

Bail (nCli, nLog, datedeb, dateFin)

Les clés primaires sont soulignées d'un trait plein, les clés étrangères sont soulignées en pointillés.

Traduire en algèbre relationnelle et en SQL les requêtes suivantes :

1. Nom des locataires à Bordeaux dont le bail se terminera le 31 décembre 2017 ou le 31 décembre 2018 **(1 pt)**.
2. Adresse et ville de l'agence qui a un mandat sur l'appartement le plus cher **(1,5 pts)**.
3. Email des clients qui ne font des visites que le matin (avant midi) **(1,5 pts)**.
4. Nom agences parisiennes qui ont un mandat pour chacune des maisons de plus de 100 m2 situées à Deauville **(1,5 pts)**.
5. Surface des parkings pour lesquels il y a des mandats dans des agences de Lille et dans des agences de Roubaix **(1,5 pts)**.

Traduire en SQL uniquement les requêtes suivantes :

6. Nombre de visites pour chaque appartement dont la ville finit par « Seine » (Vitry-sur-Seine, Ivry-sur-Seine, ...) **(1 pt)**.
7. Surface des maisons de plus de 200 m2 dont le loyer mensuel est inférieur à la moyenne **(1 pt)**.
8. Créez une requête permettant d'afficher le planning des visites de l'agence « SuperImmo » le 14 janvier 2017. On veut afficher cinq colonnes, intitulées « heure de visite », « adresse_du_logement », « ville_du_logement », « nom_du_client », « téléphone_du_client ». Le planning doit respecter l'ordre des visites **(1 pt)**.

Les séries télé (Examen 2017-2018)

Vous êtes responsable de la base de données d'un grand site spécialisé sur les séries télé.

Dans cette base, chaque série est identifiée par un numéro. Elle peut contenir plusieurs saisons, diffusées pendant une certaine période. Chaque saison contient plusieurs épisodes, dont la date de sortie exacte et la durée (en minutes) sont indiquées dans la base.

La base de données contient également les informations sur les créateurs de ces séries et sur les acteurs qui jouent dedans. On garde ainsi le casting pour tous les épisodes, c'est-à-dire le personnage interprété par chaque acteur, dans chaque épisode auquel il a participé.

Les tables de la base sont les suivantes :

Serie (nserie, titre, descr, datDeb, datFin, ncrea#, catégorie)

Createur (ncrea, nom, datNas, pays)

Saison (nserie#, nsai, datDeb, datFin)

Episode (nserie#, nsai#, nepisode, datSortie, duree, titre)

Acteur (nact, nom, datNas, pays)

Casting (nact#, nserie#, nsai#, nepisode#, personnage)

Les clés primaires sont soulignées, les clés étrangères sont indiquées par un « # ».

Traduire en algèbre relationnelle et en SQL les requêtes suivantes :

- 1) Date de sortie et durée de tous les épisodes de 'GoT' sortis en 2015
- 2) Nom des acteurs qui ont participé à des épisodes de 'GoT' et aussi à des épisodes de 'Dr. Who'
- 3) Titre des séries ne comptant aucun acteur britannique (dont le pays est UK)
- 4) Nom et date de naissance des acteurs qui ont participé à tous les épisodes de 'GoT'
- 5) Titre et nom de la série proposant l'épisode le plus court

Traduire en SQL uniquement les requêtes suivantes :

- 6) Durée moyenne des épisodes sortis en 2016
- 7) Nom des acteurs ayant joué dans une série dont le nom commence par « Docteur », triés par ordre alphabétique
- 8) Titre des séries dont la durée totale, en comptant tous les épisodes, est supérieure à 700 minutes.

Vacances au ski (Examen 2018-2019)

Vous êtes administrateur de la base de données d'une station de ski où sont donnés des cours par une équipe de moniteurs.

Chaque piste de la station est identifiée par un numéro et possède un nom de piste, et on connaît sa couleur (qui indique la difficulté), ainsi que sa longueur en kilomètres et son dénivelé (pente). L'historique des pistes empruntées par les skieurs est enregistré dans la table Randonnée.

Des cours de ski sont proposés, identifiés par un numéro. Chaque cours est adapté à un niveau de ski (1ère étoile, 2^e étoile...), un support (par exemple ski, snowboard...) et il est associé à un créneau (matin ou après-midi) et à un prix. La participation à un cours nécessite une inscription de la part des skieurs, en précisant le numéro de la semaine concernée (entre 1 et 52). Le numéro du moniteur associé est également enregistré.

Les skieurs sont identifiés par un numéro, et on connaît leur nom, prénom, âge et ville de résidence.

Les moniteurs sont identifiés par un numéro, et on connaît leur nom, prénom et leur date de recrutement.

Les tables de la base sont les suivantes :

Piste (nPiste, nomPiste, couleur, longueur, dénivelé)

Randonnée (nSkieur, nPiste, date)

Cours (nCours, niveau, support, prix, créneau)

Inscription (nCours, nSkieur, nSemaine, nMoniteur)

Skieur (nSkieur, nomS, prenomS, âge, villeS)

Moniteur (nMon, nomM, prenomM, dateRecr)

Les clés primaires sont soulignées d'un trait plein, les clés étrangères sont soulignées en pointillés.

Traduire en algèbre relationnelle et en SQL les requêtes suivantes :

10. Longueur et dénivelé des pistes noires qui ont été empruntées uniquement par des adultes. **(1,5 points)**
11. Nom et prénom des moniteurs recrutés depuis le 1^{er} janvier 2016 qui ont donné des cours les matins en 2^e et en 3^e étoile. **(1,5 points)**
12. Age du skieur qui emprunté la piste la plus longue de la station. **(1,5 points)**
13. Nom des skieurs grenoblois qui ont emprunté toutes les pistes noires en 2017. **(1,5 points)**
14. Numéro des semaines où le moniteur Bob Machin a donné des cours de snowboard ou de monoski. **(1 point)**

Traduire en SQL uniquement les requêtes suivantes :

15. Prix total des cours payés par chaque skieur (seulement pour les skieurs ayant pris au moins 2 cours) **(1 point)**
16. Liste des numéros de semaine et des créneaux des cours de ski suivis par les skieurs dont le prénom commence par « Jean », de la plus récente à la plus ancienne. **(1 point)**
17. Longueur moyenne des pistes vertes empruntées par des enfants de moins de 10 ans. **(1 point)**

Running

On considère la base de données ci-dessous, consacrée aux courses organisées en 2019 en Ile de France.

Chaque course est identifiée par un numéro. On associe à chacune un nom et sa distance en km.

Les participants sont identifiés par un numéro, et on connaît leur nom, leur prénom, leur sexe, leur âge et leur ville de résidence.

Les résultats sont enregistrés dans une table contenant le temps réalisé pour chaque participant à une course donnée.

Les participants peuvent courir de manière individuelle ou bien en équipe. Chaque équipe est identifiée par un numéro et possède un nom ainsi qu'une catégorie (par exemple « duo », « trio », « famille », etc.). L'appartenance des participants à une (ou plusieurs) équipe(s) est enregistrée dans la table Membre.

Enfin, on enregistre la date et la durée de tous les entraînements réalisés par les participants.

Les tables de la base sont les suivantes :

Course (nCourse, nomC, dist)

Participant (nPart, nom, prénom, sexe, âge, ville)

Résultat (nPart, nCourse, temps)

Equipe (nEq, nomEq, catégorie)

Membre (nPart, nEq)

Entraînement (nPart, date, durée)

Les clés primaires sont soulignées d'un trait plein, les clés étrangères sont soulignées en pointillés.

Traduire en algèbre les requêtes suivantes :

18. Nom et prénom des participants recensés dans la base de données.
19. Nom des courses de 10 km ou plus.
20. Date des entraînements qui ont duré 45 minutes.
21. Nom et sexe des participants habitant Nice ou Marseille.
22. Nom des courses auxquelles Jean Machin a participé, ainsi que temps réalisé à chaque course.
23. Nom et ville des coureurs masculins qui ont déjà couru en duo.
24. Prénom et âge des membres de l'équipe « Team Runners »
25. Age des participants (tous sexes confondus) qui ont terminé le semi-marathon de Paris ou les 20 km de Paris en moins d'1h30.
26. Ville des participants qui ne sont dans aucune équipe
27. Prénom des participants de plus de 30 ans qui n'ont jamais couru en équipe duo.
28. Nom des équipes ayant au moins un membre ayant participé à la course Odyseea et au moins un membre ayant participé à la course La Parisienne.
29. Nom et distance des courses avec des participants parisiens et toulousains.
30. Ville des femmes majeures qui ont terminé toutes les courses de 10 km en moins de 45 minutes.
31. Nom et distance des courses dont les participants ont fait les entraînements les plus longs (en durée).

Casting de films

Vous êtes responsable de la base de données d'un grand site spécialisé sur les films sortis au cinéma et en particulier sur les acteurs qui y ont participé.

Dans cette base, chaque film est identifié par un numéro. On enregistre son titre, son réalisateur, sa catégorie (suspense, action, comédie, ...) et sa date de sortie dans les salles de cinéma. Un film est découpé en séquences, qui sont numérotées (chaque film possède une séquence 1, 2, etc.). On connaît la durée et la description de chaque séquence. La durée totale du film correspond à la somme des durées de chaque séquence.

La base de données contient également les informations relatives au casting, c'est-à-dire aux acteurs qui jouent dans les films. On garde ainsi en mémoire le personnage joué par les acteurs dans toutes les séquences des films auxquelles ils ont participé.

Les tables de la base sont les suivantes :

Film (nFilm, titre, réalisateur, catégorie, dateSortie)

Sequence (nFilm, nSequence, durée, description)

Acteur (nAct, nom, dateNaiss, pays)

Casting (nAct, nFilm, nSequence, personnage)

Les clés primaires sont soulignées en trait plein, les clés étrangères en pointillés.

Traduire en algèbre relationnelle les requêtes suivantes :

- 9) Titre et catégorie des films recensés dans la base de données.
- 10) Nom des acteurs américains.
- 11) Titre des films sortis en 2019.
- 12) Réalisateurs ayant réalisé des films d'action ou des comédies.
- 13) Description et durée de toutes les séquences du film 'Avengers'.
- 14) Nom des acteurs ayant joué le personnage « maigret » ou « poirot » dans un film policier.
- 15) Nom des acteurs qui ont joué dans le film 'Minority report' et également dans 'Mission impossible'.
- 16) Date de naissance des acteurs ayant joué dans des films réalisés par Steven Spielberg et dans des films réalisés par Woody Allen.
- 17) Date de sortie et catégorie des films ne comptant aucun acteur américain.
- 18) Nom et pays des acteurs ayant joué dans des comédies mais pas dans des drames.
- 19) Nom et date de naissance des acteurs qui ont participé à tous les films d'action sortis en 2017.-
- 20) Description de la séquence la plus longue du film 'Alien'.

Karaoké

Le propriétaire d'un bar-karaoké fait appel à vous pour l'étude des usages de ses clients (en matière de karaoké). Cette étude lui permettra de mieux comprendre leurs habitudes.

Afin de réaliser cette étude, vous disposez de la base de données ci-dessous, qui contient 3 tables :

Interprétation (nInter, date, heureDébut, nCli, nChanson)

Personne (nPers, nom, prénom, dateNaiss, langue)

Chanson (nChanson, titreChanson, typeMusique, nPro, durée)

Les clés primaires sont soulignées d'un trait plein, les clés étrangères sont soulignées en pointillés.

La table Interprétation enregistre chaque interprétation réalisée par un client individuel (les chansons chantées en groupe ne sont pas enregistrées dans la base de données). On connaît la date et l'heure de l'interprétation, ainsi que l'identifiant du client et celui de la chanson.

L'attribut nCli est une clé étrangère dans la table Interprétation, qui fait référence à l'attribut nPers, clé primaire de la table Personne. On connaît le nom, le prénom, la date de naissance et la langue maternelle de chaque personne.

Enfin, des détails sont également stockés concernant les chansons disponibles sur le catalogue du karaoké. Chaque chanson est identifiées par un numéro, et on connaît son titre (par exemple « Imagine », « Happy », ...), son type de musique (par exemple soul, rap, disco, rock,...), sa durée, ainsi que l'attribut nPro, qui est le numéro de la personne correspondant au chanteur professionnel qui en est l'interprète original. nPro est une clé étrangère qui fait référence à l'attribut nPers de la table Personne, ce qui signifie que l'on a également des informations sur les chanteurs professionnels dans la base de données (nom, prénom, date de naissance et langue maternelle).

Attention donc à ne pas confondre les attributs nCli et nPro :

- nCli est l'identifiant de la personne qui interprète une chanson en tant que client durant le karaoké
- nPro est l'identifiant d'une personne qui est chanteur professionnel, interprète original d'une chanson

Exprimer en algèbre relationnelle et en SQL les requêtes suivantes :

1. Prénom et date de naissance des chanteurs amateurs francophones qui ont chanté en karaoké toutes les chansons de Michael Jackson.
2. Prénom et titre des chansons chantées par les chanteurs amateurs qui ne chantent jamais après 21h en karaoké
3. Prénom des chanteurs amateurs anglophones qui ont chanté des chansons de David Bowie le 25/03/2020
4. Titre et type des chansons interprétées par le chanteur professionnel le plus jeune.
5. Date de naissance des chanteurs professionnels qui sont les interprètes originaux de chansons de type disco ou rap.
6. Langue maternelle des chanteurs amateurs qui ont chanté la chanson intitulée « Imagine » le 1^{er} janvier 2020 et le 1^{er} janvier 2019.

Exprimer en SQL uniquement les requêtes suivantes :

7. Nombre de chansons associées à chaque chanteur professionnel (uniquement pour ceux qui possèdent plus de 10 chansons dans leur répertoire)
8. Prénom, date de naissance et langue maternelle des chanteurs amateurs qui ont chanté en 2019 une chanson dont le titre finit par « forever », triés du plus jeune au plus âgé.
9. Nombre de chansons chantées en karaoké par le chanteur amateur Jean Dupont à chacun de ses karaokés. On indiquera, pour chaque date où il a participé à un karaoké, le nombre de chansons interprétées.

1- Division

```

SELECT PRÉNOM, dateNaiss
FROM Personne P1
WHERE P1.langue = 'français'
AND NOT EXISTS (
    SELECT *
    FROM Chanson C, Personne
    WHERE C.npers = P2.npers
    AND P2.nom = 'Jackson' AND P2.prénom = 'Michael'
    AND NOT EXISTS (
        SELECT *
        FROM INTERPRETATION I
        WHERE I.nchanson = C.nchanson
        AND I.ndli = P1.npers
    )
)

```

```

SELECT prénom, titreChanson
FROM Personne P, Chanson C, Interprétation I
WHERE P.npers = I.ndli
AND C.nchanson = I.nchanson
AND NOT EXISTS (
    SELECT *
    FROM Interprétation I2
    WHERE heureDeb > 21h
    AND I.ndli = I2.ndli
)

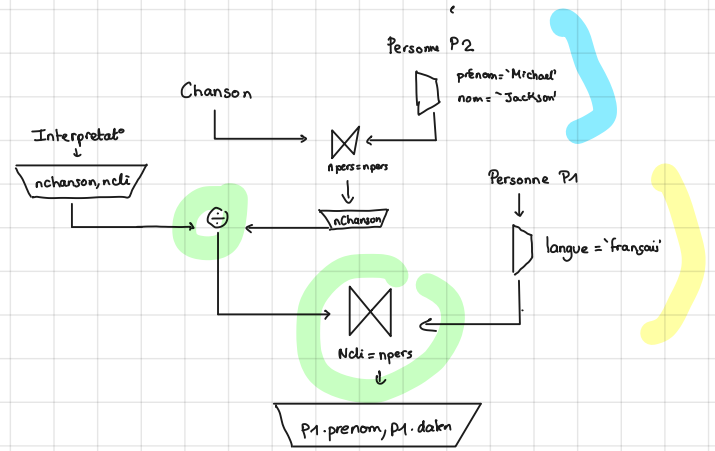
```

```

3- SELECT P2.prénom
FROM Personne P1, Personne P2, Interprétation I, Chanson C
WHERE P1.prénom = 'David'
AND P1.nom = 'Bowie'
AND I.date = '25/03/2020'
AND P2.langue = 'anglais'
AND C.npro = P1.npers
AND I.nchanson = C.nchanson
AND I.ndli = P2.npers

```

personne pr lequel
il y a pas de chanson MS pr lesquels il n'existe pas d'nb

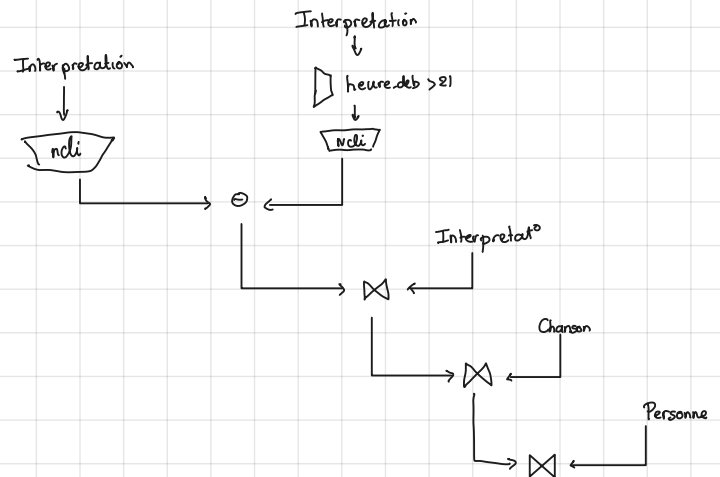


AFIN de réaliser cette étude, vous disposez de la base de données ci-dessous

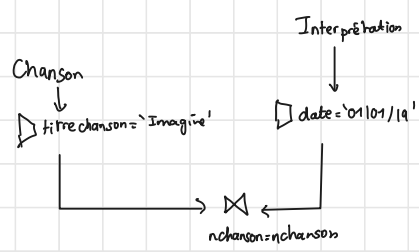
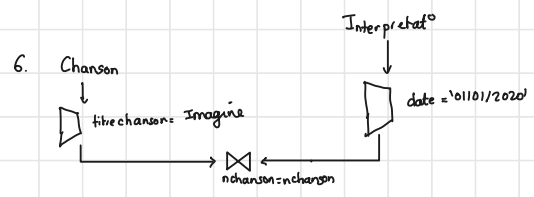
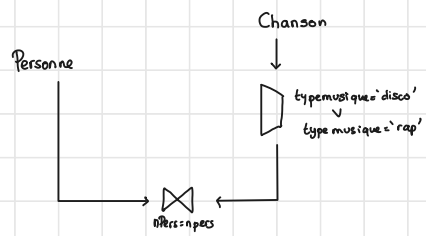
Interprétation (nInter, date, heureDébut, nCli, nChanson)

Personne (nPers, nom, prénom, dateNaiss, langue)

Chanson (nChanson, titreChanson, typeMusique, nPro, durée)

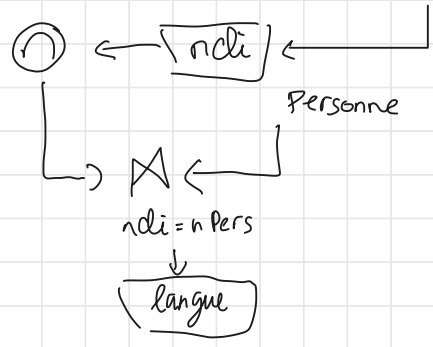


5. **SELECT** datenais
FROM Personne P, chanson C
WHERE (type musique = 'disco' OR type musique = 'rap')
AND P.npers = C.npers



SELECT langue
FROM Personne P, Chanson C, Interpretation I
WHERE C.titrechanson = 'Imagine'
AND I.date = '01/01/2020'
AND C.nchanson = I.nchanson
AND I.ndli = P.npers
AND EXISTS (SELECT #

FROM Chanson C2, Interpretation I2
WHERE C2.titrechanson = 'Imagine'
AND I2.nchanson = C2.nchanson
AND I2.date = '01/01/2019'
AND I2.ndli = P.npers)



7) **SELECT** nPro **COUNT** (nchanson)
FROM chanson
GROUP BY nPro
HAVING COUNT (nchanson) > 10

8) **SELECT** personne, datenav, langue
FROM Personne P, Chanson C, Interpretat° I
WHERE I.date BETWEEN '01/01/2019' AND '31/12/2019'
AND C.titre_chanson LIKE '%foreign'
AND I.nchanson = C.nchanson
AND I.ncli = P.npers
ORDER BY datenav DESC

9) **SELECT COUNT** (DISTINCT nchanson).date
FROM Interpretat° I, Personne P
WHERE P.prenom = 'jean'
AND P.nom = 'Dupon'
AND P.npers = I.ncli
GROUP BY date

Choix d'émission de télévision

Vous êtes chargé d'effectuer des statistiques sur les choix des émissions télévisées effectués par les téléspectateurs.

Afin de réaliser cette étude, vous disposez de la base de données ci-dessous, qui contient 3 tables :

Choix (nChoix, date, heure, nPers, nEmiss)

Personne (nPers, nom, prénom, âge)

Emission (nEmiss, nomEmiss, typeEmiss, durée)

Les clés primaires sont soulignées d'un trait plein, les clés étrangères sont soulignées en pointillés.

La table Choix enregistre les différents choix effectués par les personnes enregistrées dans la base de données. Chaque choix est identifié par un numéro unique. On connaît la date et l'heure de début de l'émission choisie, ainsi que l'identifiant de la personne qui a fait ce choix et l'identifiant de l'émission choisie.

Chaque personne est identifiée par un numéro unique. On connaît son nom, son prénom et son âge.

Enfin, des détails sont également stockés concernant les émissions diffusées à la télévision. Chaque émission est identifiée par un numéro unique. On connaît pour chacune le nom de l'émission (par exemple « Friends », « Top Chef », « Capital », « Journal TV »,...), son type (par exemple « série », « divertissement », « jeu », « information », « musique », ...) et sa durée.

Exprimer en algèbre relationnelle et en SQL les requêtes suivantes :

7. Noms des émissions de divertissement démarrant à 17h ou à 20h, ainsi que nom et prénom des personnes les ayant sélectionnées
8. Prénom des personnes de moins de 15 ans qui ont déjà choisi l'émission « Capital » le 16 mars 2020
9. Nom des personnes qui ont déjà choisi des émissions de type série et des émissions de type information
10. Nom des personnes ayant choisi l'émission la plus longue diffusée le 31/12/2019
11. Prénom et âge des personnes qui n'ont jamais choisi d'émissions avant 23h
12. Age des personnes adultes qui ont choisi toutes les émissions de type information en 2019

Exprimer en SQL uniquement les requêtes suivantes :

10. Nombre d'émissions de type jeu choisies chaque jour par Bob Martin. On indiquera, pour chaque date où il a choisi une émission, le nombre d'émissions qu'il a choisies
11. Nombre d'émissions de chaque type, uniquement lorsqu'il y a au moins 5 émissions de ce type. On indiquera, pour chaque type d'émission, le nombre d'émissions associées
12. Âge des personnes qui ont choisi une émission dont le nom commence par « Journal ». Les résultats seront triés du plus âgé au plus jeune