



POLYTECHNIQUE
MONTRÉAL
LE GÉNIE
EN PREMIÈRE CLASSE

Questionnaire Examen Final

INF3710

Sigle du cours

Identification de l'étudiant(e)		
Nom :	Prénom :	
Signature :	Matricule :	Groupe :

Sigle et titre du cours		Groupe	Trimestre
INF3710 - Fichiers et Base de Données		Tous	2018-3
Professeur		Local	Téléphone
Franjeh EL KHOURY		B-415/B-418	
Jour	Date	Durée	Heures
Samedi	15 décembre 2018	2h30	9h30-12h00

Documentation	Calculatrice
<input checked="" type="checkbox"/> Aucune <input type="checkbox"/> Toute <input checked="" type="checkbox"/> Voir directives particulières	<input checked="" type="checkbox"/> Aucune <input type="checkbox"/> Toutes <input type="checkbox"/> Non programmable Les cellulaires, agendas électroniques ou téléavertisseurs sont interdits.

Directives particulières
Deux feuilles A4 écrites à la main en recto/verso comme documentations autorisées.

Important	Cet examen contient 5 exercices sur un total de 12 pages (excluant cette page)
	La pondération de cet examen est de 35 %
	Vous devez répondre sur : <input checked="" type="checkbox"/> le questionnaire <input type="checkbox"/> le cahier <input type="checkbox"/> les deux
	Vous devez remettre le questionnaire : <input checked="" type="checkbox"/> oui <input type="checkbox"/> non

L'étudiant doit honorer l'engagement pris lors de la signature du code de conduite.

Exercice 1 – Normalisation (10 points)

Considérons le schéma relationnel suivant pour une compagnie de vente de produits pour des clients selon des factures déterminées.

Client (idclient, nomclient, adresseclient, nofacture)

Facture (nofacture, idclient, datefacture, idproduit, quantité, prix)

Produit (idproduit, nomproduit, idtype, descriptiontype)

Type (idtype, descriptiontype)

Les contraintes logiques sont les suivantes :

- Chaque facture est relative à un seul client et doit contenir plusieurs produits.
- Chaque produit doit exister une seule fois dans la facture elle-même.
- Chaque produit doit avoir un seul type.

1.1. Vérifiez chacune des relations mentionnées ci-dessus si elle est en forme normale en précisant le niveau. Justifiez votre réponse dans le cas où une fonction n'est pas en forme normale pour le niveau trouvé. (4 points)

La relation « Client » n'est pas en 1FN, parce qu'il y a plusieurs valeurs de « nofacture » pour la même valeur de la clé « idclient ».

La relation « Facture » n'est pas en 1FN, parce qu'il y a plusieurs valeurs de « idproduit », « quantité » et « prix » pour la même valeur de la clé « nofacture ».

La relation « Produit » est en 1FN, en 2FN, mais elle n'est pas en 3FN, parce que la DF (idtype → descriptiontype) n'est pas directe.

La relation « Type » est en 1FN, en 2FN, en 3FN et en FNBC.

Réponse de l'exercice 1 :

1.2. Décomposez les relations qui ne sont pas en forme normale. (3 points)

La relation « Client » sera décomposée en deux relations :

R1 = Client (idclient, nomclient, adresseclient)

R2 = ClientFacture (idclient, nofacture)

La relation « Facture » sera décomposée en deux relations :

R3 = Facture (nofacture, idclient, datefacture)

R4 = FactureProduit (nofacture, idproduit, quantité, prix)

La relation « Produit » sera composée en deux relations :

R5 = Produit (idproduit, nomproduit, idtype)

R6 = Type (idtype, descriptiontype)

1.3. Quel sera le schéma relationnel après normalisation des relations. (3 points)

R1 = Client (idclient, nomclient, adresseclient)

R3 = Facture (nofacture, idclient, datefacture)

R4 = FactureProduit (nofacture, idproduit, quantité, prix)

R5 = Produit (idproduit, nomproduit, idtype)

Type (idtype, descriptiontype)

Exercice 2 – Algèbre relationnel, requête SQL et vue (9 points)

Considérons le schéma relationnel suivant pour un système de gestion de projets.

Employe (idemp, nomemp, salaire, date_de_naissance, iddep)

Departement (iddep, nomdep)

Projet (idprojet, descriptionproj, date_debut, date_fin)

Travail (idemp, idprojet, nombre_heures)

Les contraintes logiques sont les suivantes :

- Chaque employé travaille dans un seul département.
- Chaque produit doit être réalisé par plusieurs employés.
- Chaque employé travaille sur un projet donné un certain nombre d'heures.

2.1. Écrivez les requêtes suivantes en algèbre relationnel. (4 points)

a- Affichez l'identifiant des employés, le nom des employés et la description du projet pour tous les employés qui travaillent dans le département 'Génie Logiciel'. (2 points)

$\Pi_{\text{idemp}, \text{nomemp}, \text{descriptionproj}}((\text{Employe} \bowtie (\sigma_{[\text{nomdep} = \text{'Génie Logiciel'}]}(\text{Departement}))) \bowtie \text{Travail} \bowtie \text{Projet})$

b- Affichez l'identifiant des départements et le nom des départements contenant des employés qui ont travaillé moins de 6 heures sur un projet donné. (2 points)

$\Pi_{iddep, nomdep} (Departement) \bowtie (Employe \bowtie (\sigma_{[nombre_heures < 6]} (Travail)))$

2.2. Écrivez les requêtes suivantes en SQL. (5 points)

a- Supprimez les projets dont leur « date_debut » n'est pas dans l'année 2018. (2 points)

Delete From Projet

Where date_debut Not Between '01-01-2018 and '31-12-2018';

b- Créez une vue « V_employe » contenant l'identifiant de l'employé « V_idemp », le nom de l'employé « V_nomemp » et le total de nombre d'heures « V_TotNbHeure » sur tous les projets qu'il a travaillé, dont la « date_fin » est dans les années 2017 et 2018, le nom de l'employé contient la lettre « H » et dont leur salaire est plus grand que 550, trié par ordre décroissant selon le nom de l'employé. **(3 points)**

```
create or replace view V_employe(V_idemp,V_nomemp,V_TotNbHeure) as
select E.idemp,E.nomemp,sum(T.nombre_heures)
from Employe E, Travail T, Projet P
where E.idemp = T.idemp
and T.idprojet = P.idprojet
and year(P.date_fin) in (2017,2018)
and E.nomemp Like '%H%'
and E.salaire > 550
group by E.idemp,E.nomemp
order by E.nomemp;
```


Exercice 3 – Conception d'une Base de Données relationnelle : Gestion du Système d'Information de la chaîne de pharmacies (6 points)

Une chaîne de pharmacies vous demande de concevoir une base de données représentant des informations sur les pharmacies, les compagnies pharmaceutiques, les médicaments, les clients des pharmacies ou patients et leurs médecins.

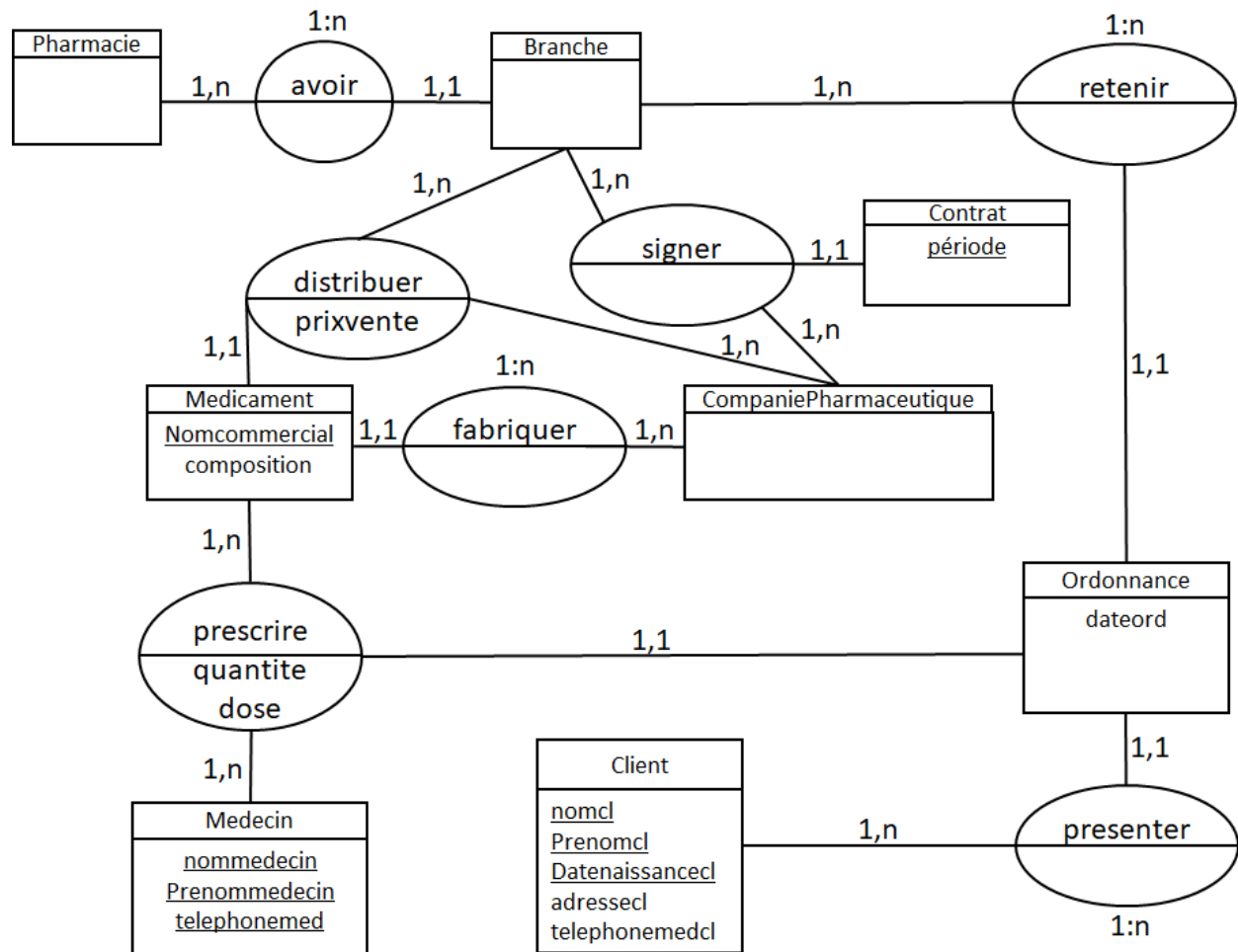
La chaîne comporte plusieurs pharmacies dont chacune peut avoir différentes branches. Ces branches peuvent signer, indépendamment les unes des autres, des contrats, pour des périodes déterminées, avec des compagnies pharmaceutiques qui fabriquent et distribuent des médicaments. Notez qu'un médicament est fabriqué par une seule compagnie pharmaceutique qui le distribue sur différentes branches de pharmacies ayant signé des contrats avec la compagnie pharmaceutique qui fixe le prix de vente de ce médicament.

Pour acheter un médicament, un client doit se présenter avec une ordonnance. Pour chaque ordonnance, la branche de la pharmacie retient le nom, prénom, date de naissance, adresse et téléphone du patient (ou client de la pharmacie) ; le nom, prénom et téléphone du médecin ayant prescrit l'ordonnance ; la date de l'ordonnance, les médicaments prescrits avec une quantité et une dose pour chacun.

Concernant les médicaments, outre le prix de vente, chacun est caractérisé par un nom commercial. Un médicament est divisé en deux catégories : sirop et comprimé. Le sirop est caractérisé par le nombre de jours que le patient puisse utiliser le même flacon. Le comprimé est caractérisé par sa dose maximale par jour.

3.1. Proposez un schéma conceptuel dans le modèle "entité-association" (MCD). Spécifiez clairement sur votre schéma les différents types d'associations existants et leurs cardinalités ainsi que les identifiants des entités-types et des associations-types s'ils existent.

Réponse de l'exercice 3 :



Pour le reste des identifiants, des attributs et des types d'entités, ça dépend de la pensée de chaque étudiant; même pour les relations tertiaires.

Exercice 4 – Transactions (5 points)

Supposons que nous avons la table Client suivante :

IDClient	NomClient	AdresseClient
1	Jean	Montréal
2	Sébastien	Laval
3	Armand	Montréal

Six transactions seront exécutées simultanément sur cette table Client par deux utilisateurs : Utilisateur1 et Utilisateur2 comme suites :

- 1- **Utilisateur1:** Delete From Client
Where IDClient = 3;
- 2- **Utilisateur 2:** Commit;
- 3- **Utilisateur 2:** Select *
From Client
Where ClientID = 3;
- 4- **Utilisateur 1:** Commit;
- 5- **Utilisateur 1:** Select *
From Client
Where ClientID = 3;
- 6- **Utilisateur 2:** Select *
From Client
Where ClientID = 3;

4.1. Expliquez la fonctionnalité de chacune de ces transactions en précisant si elle sera mise en place au niveau physique ou au niveau logique de la table Client. Affichez au niveau de chacune de ses quatre transactions mentionnées ci-dessus le résultat obtenu (s'il y a lieu) selon l'utilisateur en question.

Transaction #1 : Utilisateur1 supprime logiquement l'enregistrement correspondant au client avec IDClient = 3 de la table Client.

Transaction #2 : Utilisateur2 sauvegarde physiquement les changements effectués sur la table Client, mais dans ce cas nous obtenons un message « aucun changement effectué ».

Transaction #3 : Utilisateur2 collecte de la table Client les informations correspondant au client avec IDClient = 3 et les charge en mémoire comme suit :

IDClient	NomClient	AdresseClient
3	Armand	Montréal

Nous remarquons que la transaction#1 n'a pas pris effet sur la table Client pour l'Utilisateur2 puisqu'il travaille dans une session séparée de l'Utilisateur1.

Transaction #4 : Utilisateur1 confirme physiquement la suppression logique de l'enregistrement correspondant au client avec IDClient = 3 de la table Client.

Réponse de l'exercice 4 :

Transaction #5 : Utilisateur1 collecte de la table Client les informations correspondant au client avec IDClient = 3 et les charge en mémoire comme suit :

IDClient	NomClient	AdresseClient
----------	-----------	---------------

Transaction #6 : Utilisateur2 collecte de la table Client les informations correspondant au client avec IDClient = 3 et les charge en mémoire comme suit :

IDClient	NomClient	AdresseClient
----------	-----------	---------------

Nous remarquons que la transaction#1 a pris effet sur la table Client pour l'Utilisateur2 après la Transaction #4 effectuée par Utilisateur1.

Exercice 5 – Déclencheurs (5 points)

Supposons que nous avons les deux tables suivantes :

Etudiant (matricule, nom_etudiant, adresse_etudiant, courriel_etudiant)

BulletinNote (matricule, session, cours, crédits, note)

5.1. Écrivez un déclencheur qui supprime tous les bulletins de notes dans la table « BulletinNote » de l'étudiant après la suppression des informations de cet étudiant dans la table « Etudiant ».

Create or Replace Function deleteBulletinNoteTrigger() Returns trigger AS \$\$

Begin

 Delete from BulletinNote

 Where BulletinNote.matricule = New.matricule;

 Return New;

End;

\$\$ language plpgsql;

Create Trigger af_del_etudiant

After Delete On Etudiant

For each row

When (Etudiant.matricule = New.matricule)

Execute procedure deleteBulletinNoteTrigger ();

Bon travail!