



**POLYTECHNIQUE  
MONTRÉAL**  
  
UNIVERSITÉ  
D'INGÉNIERIE

## **Solutionnaire Examen Final**

**INF3710**

**Sigle du cours**

<b>Sigle et titre du cours</b>		<b>Groupe</b>	<b>Trimestre</b>
<b>INF3710 - Fichiers et Bases de Données</b>		<b>Tous</b>	<b>20193</b>
<b>Professeur</b>		<b>Local</b>	<b>Téléphone</b>
<b>Coordonnateur : Giuliano Antoniol</b>		<b>M-4021</b>	<b>2027 / 5193</b>
<b>Jour</b>	<b>Date</b>	<b>Durée</b>	<b>Heures</b>
<b>Jeudi</b>	<b>12 décembre 2019</b>	<b>2h30</b>	<b>13h30-16h00</b>
<b>Documentation</b>		<b>Calculatrice</b>	
<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	<b>Les cellulaires, agendas électroniques ou téléavertisseurs sont interdits.</b>
<b>Directives particulières</b>			
<ul style="list-style-type: none"><li>- Deux feuilles 8/12 * 11 écrites à la main en recto – verso comme documentation autorisée.</li><li>- Écrivez lisiblement.</li><li>- <u>Ne pas détachez vos feuilles.</u></li><li>- <u>Écrivez vos brouillons à la page 13 du questionnaire.</u></li></ul>			
<b>Important</b>	Cet examen contient <b>5</b> exercices sur un total de <b>13</b> pages ( <b>excluant cette page</b> )		
	La pondération de cet examen est de <b>35</b> %		
	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 des relations (11 pts)

Un service de documentation technique dans une entreprise voudrait créer une base de données pour répertorier tous ses documents et gérer les prêts aux employés de l'entreprise. Dans cette application, un document revient à un département et un seul. Il peut être écrit par plusieurs auteurs et décrit par plusieurs mots clés. Un employé travaille dans un seul département.

Le schéma relationnel proposé pour cette base de données est le suivant :

**DOCUMENT** (iddoc, typedoc, titredoc, dateparution, iddept, idauteur, idmotclé)

**AUTEUR** (idauteur, nomauteur, prénomauteur)

**MOTCLE** (idmotclé, motclé)

**EMPLOYE** (idemp, nomemp, prenomemp, fonction, adresse, tel, iddept, nomdept)

**EMPRUNT** (idemp, iddoc, titredoc, dateemprunt, dateretourprévue, dateretoureffective)

**DEPARTEMENT** (iddept, nomdept, idemp, nomemp)

### 1.1- Précisez la forme normale de chacune des relations. Justifiez votre réponse. (4 pts)

#### Réponse 1.1 :

**DOCUMENT** n'est pas en 1FN, parce que idauteur et idmotclé sont multivalués pour la même valeur de la clé (iddoc). (0.75 pt)

**AUTEUR** est en 1FN, 2FN, 3FN et FNBC. (0.5 pt)

**MOTCLE** est en 1FN, 2FN, 3FN et FNBC. (0.5 pt)

**EMPLOYE** est en 1FN, en 2FN, mais pas en 3FN, parce que la DF (idemp → nomdept) n'est pas directe. Elle est déduite par transitivité des 2 DF (idemp → iddept et iddept → nomdept). (0.75 pt)

**EMPRUNT** est en 1FN, mais pas en 2FN parce que la DF (idemp, iddoc → titredoc) n'est pas élémentaire à cause de la DF (iddoc → titredoc). (0.75 pt)

**DEPARTEMENT** n'est en 1FN, parce que idemp et nomemp sont multivalués pour la même valeur de la clé (iddept) et il existe une relation (idemp, nomemp) dans la relation DEPARTEMENT. (0.75 pt)

**1.2- Pour les relations qui présentent toujours de la redondance, proposez une correction. (4pts)****Réponse 1.2 :**

La relation DOCUMENT sera décomposée en 3 relations : (0.25 pt)

R1 = DOCUMENT (iddoc, typedoc, titredoc, dateparution, iddept.) (0.5 pt)

R2 = DOCAUTEUR (iddoc, idauteur) (0.25 pt)

R3 =DOCMOTCLE (iddoc, idmotclé) (0.25 pt)

La relation EMPLOYE sera décomposée en 2 relations : (0.25 pt)

R4 = EMPLOYE (idemp, nomemp, prenomemp, fonction, adresse, tel, iddept) (0.25 pt)

R5 = DEPARTEMENT (iddept, nomdept) (redondante à éliminer) (0.25 pt)

La relation EMPRUNT sera décomposée en 2 relations : (0.25 pt)

R6 = EMPRUNT (idemp, iddoc, dateemprunt, dateretourprévue, dateretoureffective) (0.5 pt)

R7 = DOCUMENT (iddoc, titredoc) (redondante à éliminer) (0.25 pt)

La relation DEPARTEMENT sera décomposée en 3 relations : (0.25 pt)

R8 = DEPARTEMENT (iddept, nomdept) (0.25 pt)

R9 = DEPEMP (iddept, idemp) (0.25 pt)

R10 = EMPLOYE (idemp, nomemp) (redondante à éliminer) (0.25 pt)

**1.3- Donnez le schéma relationnel final. (3pts)****Réponse 1.3 :**

R1 = DOCUMENT (iddoc, typedoc, titredoc, dateparution, iddept.) (0.5 pt)

R2 = DOCAUTEUR (iddoc, idauteur) (0.25 pt)

R3 =DOCMOTCLE (iddoc, idmotclé) (0.25 pt)

AUTEUR (idauteur, nomauteur, prénomauteur) (0.25 pt)

MOTCLE (idmotclé, motclé) (0.25 pt)

R4 = EMPLOYE (idemp, nomemp, prenomemp, fonction, adresse, tel, iddept) (0.5 pt)

**R6 = EMPRUNT** (idemp, iddoc, dateemprunt, dateretourprévue, dateretoureffective) (0.5 pt)

**R8 = DEPARTEMENT** (iddept, nomdept) (0.25 pt)

**R9 = DEPEMP** (iddept, idemp) (0.25 pt) (puisque iddept existe au niveau de Employe, alors cette relation n'est pas obligatoire de l'avoir au niveau du résultat final et +0.25 au niveau de Employe)

## Exercice 2 – Algèbre relationnelle et SQL (10 points)

On considère le schéma relationnel suivant :

Œuvre (idœuvre, titre)

Morceau (idmorceau, durée, iddisque, idœuvre)

Disque (iddisque, marque, type, dateparution, prix)

Execute (idmorceau, idinterprète)

Interprete (idinterprète, nom, adresse)

### 2.1- Écrivez les requêtes suivantes en algèbre relationnelle. (3 points)

- a) Affichez l'identifiant de l'interprète (idinterprète) et le nom de l'interprète (nom) pour les interprètes qui ont exécuté des morceaux de durée de 2 minutes. (1.5 pt)

**Réponse 2.1-a :**

$\pi$  [idinterprète, nom] (0.5 pt) ( $\sigma$  [durée = 2] (0.5 pt) (Interprete  $\bowtie$  Execute  $\bowtie$  Morceau) (0.5 pt))

- b) Affichez l'identifiant du morceau (idmorceau), la durée du morceau (durée) et la marque du disque (marque); pour les morceaux qui sont enregistrés sur les disques dont leur date de parution (dateparution) est comprise entre février 2019 et juillet 2019. (1.5 pt)

**Réponse 2.1-b :**

$\pi$  [idmorceau, durée, marque] (0.5 pt) ( $\sigma$  [dateparution >= '2019-02-01' (0.25 pt)  $\wedge$  (0.25 pt) dateparution <= '2019-07-31' (0.25 pt)] (Morceau  $\bowtie$  Disque) (0.25 pt))

**2.2. Écrivez les requêtes suivantes en SQL. (7 points)**

- a) Créez une vue v\_marque\_disque qui contient la liste des marques (vmarque) et le nom des interprètes (vnom), pour toutes les marques qui produisent des disques qui contiennent des morceaux interprétés par des interprètes dont le nom contient la lettre 'a' et la lettre 'r' à la quatrième position de la fin et qui résident à Montréal et à Toronto; triez par ordre alphabétique selon le nom de l'interprète. **(2.5 pts)**

**Réponse 2.2-a :**

Create View v\_marque\_disque (vmarque, vnom) As (0.25 pt)

Select D.marque, I.nom (0.25 pt)

From Disque D, Morceau M, Execute E, Interprete I (0.25 pt)

Where D.iddisque = M.iddisque (0.25 pt)

And M.idmorceau = E.idmorceau (0.25 pt)

And E.idinterprete = I.idinterprete (0.25 pt)

And I.nom Like '%a%' (0.25 pt)

And I.nom Like '%r...' (0.25 pt)

And I.Adresse in ('Montréal', 'Toronto') (0.25 pt)

Order By I.nom; (0.25 pt)

- b) Affichez pour chaque interprète son nom, le nombre de ses morceaux avec la moyenne de leurs durées. (2 pts)

**Réponse 2.2-b :**

**Select I.nom, (0.25 pt) Count(E.idmorceau) (0.25 pt), AVG(M.durée) (0.5 pt)**

**From Interprete I, Execute E, Morceau M (0.25 pt)**

**Where I.idinterprète = E.idinterprète (0.25 pt)**

**And E. idmorceau = M. idmorceau (0.25 pt)**

**Group By I.nom; (0.25 pt)**

- c) Parmi les interprètes qui ont exécuté des morceaux de durée supérieure à 5 minutes, affichez les identifiants des interprètes (idinterprète) et le nombre de morceaux de ceux qui ont exécuté plus de 5 morceaux et qui ne sont pas résidents à Montréal. La minute est l'unité de mesure de la durée. (2.5 pts)

**Réponse 2.2-c :**

**Select I.idinterprète, Count(E.idmorceau) (0.25 pt)**

**From Interprete I, Execute E, Morceau M (0.5 pt)**

**Where I.idinterprète = E.idinterprète (0.25 pt)**

**And E.idmorceau = M.idmorceau (0.25 pt)**

**And M.durée > 5 (0.25 pt)**

**And I.adresse <> 'Montréal' (0.25 pt)**

**Group By I.idinterprète (0.25 pt)**

**Having Count(E.idmorceau) > 5; (0.5 pt)**

### Exercice 3 – Conception d’une base de données relationnelle : Gestion de parc de voitures (7 points)

Un parc de voitures décide de réorganiser son système d’information en rassemblant les informations suivantes dans une base de données comme suit :

Chaque voiture est identifiée par un numéro de série et caractérisée par un nom. Une voiture est achetée par un seul propriétaire : soit par un particulier, soit par une entreprise. Chaque propriétaire est caractérisé par un identifiant propriétaire, un nom, une adresse et téléphones.

Un particulier est identifié par un numéro de sécurité sociale (NSS).

Une entreprise est identifiée par un numéro de registre.

Si le propriétaire est une entreprise, nous devons connaître le nom de la personne à contacter.

Nous devons également connaître la date d'achat de la voiture par un propriétaire.

Chaque voiture a un type. Ce type spécifie la marque de la voiture (BMW, Volkswagen, etc.), son numéro de modèle et le type de carrosserie (Berline, Wagon, Limousine, etc.).

Un certain nombre de conducteurs participent au parc. Chaque conducteur est caractérisé par un nom, une adresse, un numéro de téléphone et le type de voiture qu’il conduit, ainsi que le nombre total d’heures de conduite qu’il a effectué sur chacun de ces types de voiture.

L’entretien des voitures est fait par des techniciens appartenant au parc. La réparation peut être effectuée par un ou plusieurs techniciens. Pour toute réparation, nous gardons trace de la cause de sa réparation, de sa date et de sa durée. Une voiture peut être réparée plusieurs fois par les mêmes techniciens.

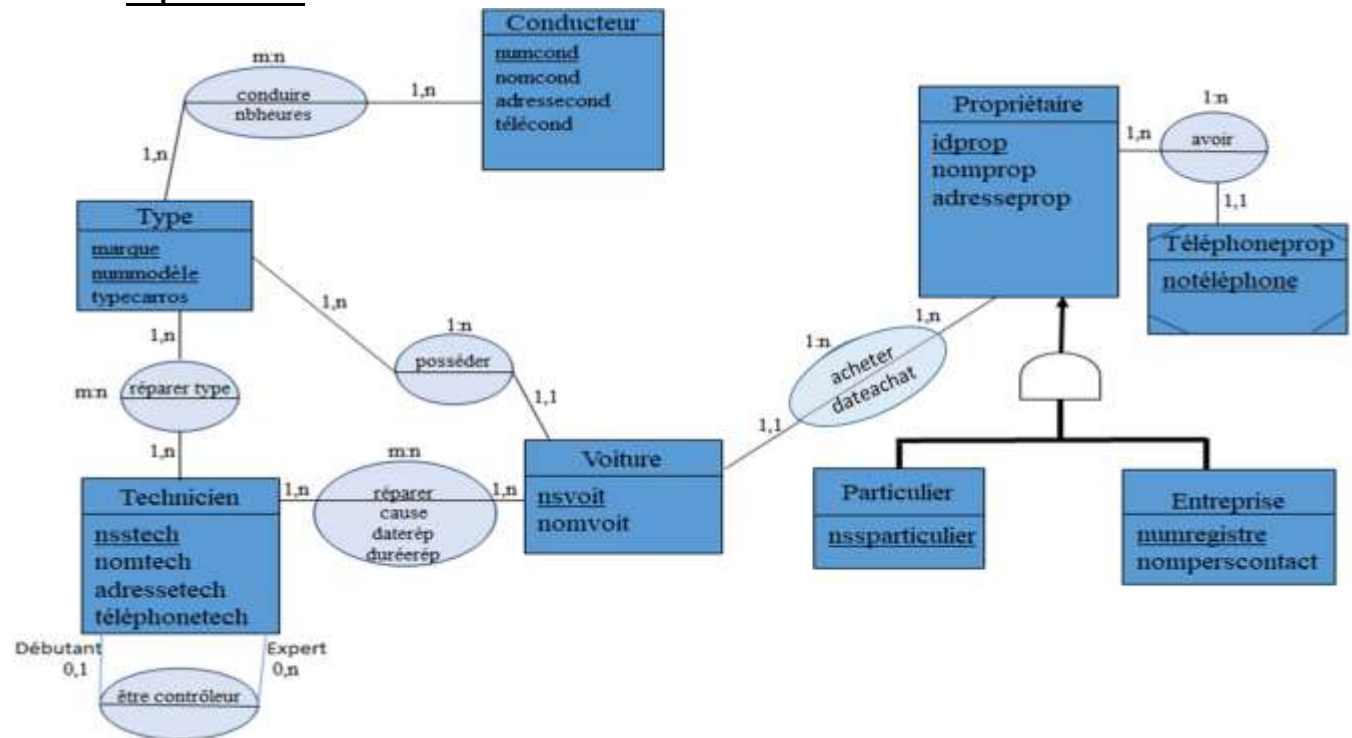
Chaque technicien est identifié par un NSS et caractérisé par un nom, une adresse, un numéro de téléphone et les types de voitures qu’il peut réparer.

Un technicien expert peut contrôler un ou plusieurs techniciens débutants. Un technicien débutant doit être contrôlé par un seul expert.

#### 3.1- En utilisant la méthode Merise pour la modélisation, proposez un diagramme Entité-Association (7 pts)

- a) Identifiez les types d’entités avec leur(s) identifiant(s) et leur(s) attribut(s); **(2 pts)**
- b) Identifiez les types (fort ou faible) des types d’entités; **(0.5 pt)**
- c) Identifiez les types d’associations avec leurs attributs s’ils existent; **(2 pts)**
- d) Identifiez les cardinalités; **(1.5 pt)**
- e) Identifiez les types (binaire 1:n, etc.) des types d’associations. **(1 pt)**



**Réponse 3.1 :****Pénalités :**

- (-0.25 pt au total) pour chacune des représentations qui ne respectent pas les représentations de Merise : TE, identifiant, attributs, TA & Attributs, cardinalités, type TA
- (-0.5 pt au total) pour la duplication des noms quelque soit le type
- (-0.15 pt) pour le manque d'un attribut au niveau de chaque TE
- (-0.5 pt) pour un manque de chaque TA
- (-0.25 pt) pour un manque de chaque TE
- (-0.25 pt) pour un manque d'un attribut au niveau du TA
- (-0.5 pt) pour un ajout au niveau du TE un attribut qui appartient au TA qui lie les TE en question
- (-0.1 pt) par cardinalité
- (-0.25 pt) pour chaque type de TA (1 :n, m :n,...)
- (-0.25 pt) pour chaque type de TE (faible ou fort)

## Exercice 4- Transactions (3 points)

Supposons que nous avons la table Produit suivante :

idproduit	descriptionproduit	quantitéproduit
1	Chocolat	50
2	Fromage	20
3	Lait	57

Six transactions seront exécutées simultanément sur cette table Produit par deux utilisateurs : Utilisateur 1 et Utilisateur 2 comme suit :

**1- Utilisateur 1:** Update Produit  
Set quantitéproduit = 60  
Where idproduit = 3;

**2- Utilisateur 2:** Commit;

**3- Utilisateur 2:** Select \*  
From Produit  
Where idproduit = 3;

**4- Utilisateur 2:** Delete from Produit  
Where idproduit = 3;

**5- Utilisateur 2:** Commit;

**6- Utilisateur 1:** Commit;

**7- Utilisateur 1:** Select \*  
From Produit  
Where idproduit = 3;

**4.1- Expliquez la fonctionnalité de chacune de ces transactions en précisant si elle sera mise en place au niveau physique (p) ou au niveau logique (L) de la table Produit. Affichez au niveau de chacune de ses sept transactions mentionnées ci-dessus le résultat obtenu (s'il y a lieu) selon l'utilisateur en question. (3 pts)**

**Réponse 4.1 :**

**1- Utilisateur1:** Update Produit **(L)** (0.25 pt)  
Set quantitéproduit = 60  
Where idproduit = 3;

**2- Utilisateur 2:** Commit; **(P), mais aucune transaction** (0.5 pt)

**3- Utilisateur 2:** Select \* (0.5 pt)  
From Produit  
Where idproduit = 3;

idproduit	descriptionproduit	quantitéproduit
3	Lait	57

**4- Utilisateur 2:** Delete from Produit **(L)**  
Where idproduit = 3; (0.25 pt)

**5- Utilisateur 2:** Commit; **(P), mise à jour de la suppression au niveau physique de l'enregistrement pour idproduit = 3** (0.5 pt)

**6- Utilisateur 1:** Commit; **(P), mise à jour de la valeur de quantitéproduit pour 60 au lieu de 57 au niveau physique de l'enregistrement pour idproduit = 3, mais cette transaction ne peut pas être appliquée, parce que cet enregistrement est déjà supprimé par Utilisateur 2.** (0.5 pt)

**7- Utilisateur 1:** Select \* (0.5 pt)  
From Produit  
Where idproduit = 3;

idproduit	descriptionproduit	quantitéproduit
-----------	--------------------	-----------------

## Exercice 5- Déclencheurs (4 points)

Supposons que nous avons les deux tables suivantes :

**Produit** (idproduit, descriptionpr, quantitepr, prixunitairepr)

**Commande** (nocommande, idproduit, qtycommande, prix, montant)

5.1- Écrivez un déclencheur qui supprime toutes les lignes de commandes dans la table « Commande » du produit après la suppression des informations de ce produit dans la table « Produit ». (4 pts)

Réponse 5.1 :

```
CREATE OR REPLACE FUNCTION deleteCommandeProduit() (0.25 pt) RETURNS TRIGGER AS
$$ (0.25 pt)
BEGIN (0.25 pt)
    DELETE FROM Commande (0.25 pt)
    WHERE Commande.idproduit = :Old.idproduit (0.5 pt)
    RETURN Old; (0.25 pt)
END;
$$ languageplpgsql; (0.25 pt)
```

```
CREATE TRIGGER af_del_produit (0.25 pt)
    AFTER DELETE ON Produit (0.5 pt)
FOR EACH ROW (0.25 pt)
WHEN (Produit.idroduit = Old.idproduit) (0.5 pt)
EXECUTE PROCEDURE deleteCommandeProduit(); (0.5 pt)
```

Ou bien

```
CREATE OR REPLACE TRIGGER af_del_produit (0.5 pt)
    AFTER DELETE ON Produit (0.5 pt)
FOR EACH ROW (0.5 pt)
WHEN (Produit.idroduit = Old.idproduit) (0.5 pt)
Begin
    DELETE FROM Commande (0.5 pt)
    WHERE Commande.idproduit = :Old.idproduit (1 pt)
End; (0.5 pt)
```

**Bon travail!**