

# DON2 - Lucky Summary

Sm!le42

7 avril 2021

## Table des matières

<b>1</b>	<b>Dépendance fonctionnelle <math>X \rightarrow Y</math></b>	<b>1</b>
1.1	Théorie	1
1.2	Exercices	2
1.2.1	Trouver les dépendances fonctionnelles	2
1.2.2	Donner les implications des DF	2
<b>2</b>	<b>Schéma conceptuel</b>	<b>2</b>
2.1	Théorie	2
2.1.1	Identifiant (key)	2
2.1.2	Clé étrangère	3
2.1.3	$\langle null \rangle$	3
2.2	Exercices	3
2.2.1	En fonction de l'image, remplir le tableau suivant en y ajoutant les noms des attributs adéquats. (Certaines cellules peuvent rester vides)	3
2.2.2	Répondre aux questions	3
<b>3</b>	<b>Projection et sélection</b>	<b>4</b>
3.1	Théorie	4
3.1.1	$SELECT \pi_{\{attributs\}}(Relation)$	4
3.1.2	DISTINCT	4
3.1.3	$WHERE \sigma_{\{condition\}}$	4
3.1.4	IN	4
3.1.5	BETWEEN	4
3.1.6	LIKE ( _ et % )	4
3.1.7	AS	5
3.2	Exercices	5
3.2.1	Répondre aux questions	5
3.2.2	Donner la sémantique	5
<b>4</b>	<b>Jointure (A) <math>\bowtie_{\langle condition \rangle}</math> (B)</b>	<b>6</b>
4.1	Théorie	6
4.1.1	Produit cartésien $A \times B$	6
4.2	Exercices	6
4.2.1	Donner la liste des tuples de A	6
4.2.2	Calculer	6
4.2.3	Donner la sémantique (ATTENTION! RÉPONSES NON VÉRIFIÉE PAR UN ENSEIGNANT)	7

## 1 Dépendance fonctionnelle $X \rightarrow Y$

### 1.1 Théorie

Soit  $X$  et  $Y$  des ensembles d'attributs dans une relation  $R$ . Il existe une dépendance fonctionnelle  $X \rightarrow Y$  si, à tout instant, deux lignes de  $R$  qui ont les mêmes valeurs pour  $X$  ont aussi les mêmes valeurs pour  $Y$ .

#### Remarque :

On omettra les accolades liées à la notation ensembliste pour un unique attribut.

Ainsi,  $\{PRODUIT\} \rightarrow \{PRIX\}$  deviendra  $PRODUIT \rightarrow PRIX$ .

Exemple :

$PRODUIT \rightarrow PRIX$

- *PRODUIT* détermine (fonctionnellement) *PRIX*
- *PRIX* dépend (fonctionnellement) de *PRODUIT*
- *PRODUIT* est le déterminant et *PRIX* est le déterminé de la dépendance fonctionnelle

## 1.2 Exercices

### 1.2.1 Trouver les dépendances fonctionnelles

1. Un cours n'est donné que par un seul enseignant : **COURS → ENSEIGNANT**
2. Un enseignant ne peut pas donner deux cours en même temps : **ENSEIGNANT, HEURE → COURS**
3. Un cours porte sur une seule matière : **COURS → MATIÈRE**
4. Un numéro de plaque d'immatriculation est lié au propriétaire de la voiture : **PLAQUE → PROPRIO**
5. Un modèle LEGO est réalisé avec un certain nombre de pièces : **MODÈLE → NOMBRE DE PIÈCES**
6. Suivant la moyenne obtenue par un étudiant pour son année, celui-ci recevra un grade. (12 = satisfaisant, 14 = distinction. . .) : **MOYENNE → GRADE**
7. Un 0code *bic* est associé à chacune des banques : **BIC → BANQUE** et **BANQUE → BIC**

### 1.2.2 Donner les implications des DF

1. **ENSEIGNANT → COURS**
  - (a) Un enseignant donne qu'un seul cours : **Vrai**
  - (b) Un enseignant donne toujours le même cours : **Faux**
  - (c) Le cours est donné par un seul enseignant : **Faux**
2. **CODEPOSTAL → VILLE**
  - (a) Une ville est associée à un et un seul code postal : **Faux**
  - (b) Un code postal est associé à une et une seule ville : **Vrai**
  - (c) Autre : **Rien**
  - (d) La dépendance fonctionnelle **CODEPOSTAL → VILLE** représente-t-elle la situation des codes postaux en Belgique ? : **Non**
3. **VERSIONUNICODE → TAILLECHARSET**
  - (a) Le nombre de caractères Unicode est différent d'une version à l'autre : **Pas forcément**
  - (b) Je connais le nombre de caractères admis dans mon système, je connais donc la version Unicode prise en charge par celui-ci : **Faux**
  - (c) Autre : **Si je connais la version Unicode, je connais le nombre de caractères admis**
4. **{PUissance, ÂGE} → TAXECIRCULATION**
  - (a) Si je connais le montant de la taxe de circulation, je connais la puissance et l'âge d'une voiture : **Faux**
  - (b) Si je connais l'âge d'une voiture, je connais le montant de la taxe : **Faux**
  - (c) Si je connais la puissance et l'âge d'une voiture, je connais le montant de la taxe : **Vrai**
  - (d) Autre : **Rien**

## 2 Schéma conceptuel

### 2.1 Théorie

#### 2.1.1 Identifiant (key)

Un identifiant est un ensemble d'un ou plusieurs attributs tel que toutes valeurs de ceux-ci nous permette de retrouver, à tout moment, au plus un tuple.

Une table peut posséder plusieurs identifiants. On choisit l'un d'eux, le plus souvent minimal, qu'on déclare **primaire**. Les autres sont dès lors, **secondaires**.

L'identifiant primaire est constitué de **colonnes obligatoires**.

#### 1. Identifiant minimal

Un identifiant est minimal si chacune de ses colonnes est nécessaire pour garantir la contrainte d'unicité.

#### 2. Identifiant simple

Identifiant composé d'un seul attribut.

#### 3. Identifiant composé

Identifié composé de plusieurs attributs.

### 2.1.2 Clé étrangère

Une clé étrangère définit une contrainte référentielle. Elle référence en principe l'identifiant primaire de la table cible. (Elle peut référencer un identifiant secondaire mais ce n'est pas recommandé).

Il se peut qu'une clé étrangère soit également un identifiant.

### 2.1.3 <null>

L'absence de valeur est généralement indiquée par <null> ou par rien.

**Remarque :** WHERE cat **IS** null ; (Et pas WHERE cat = null)

## 2.2 Exercices

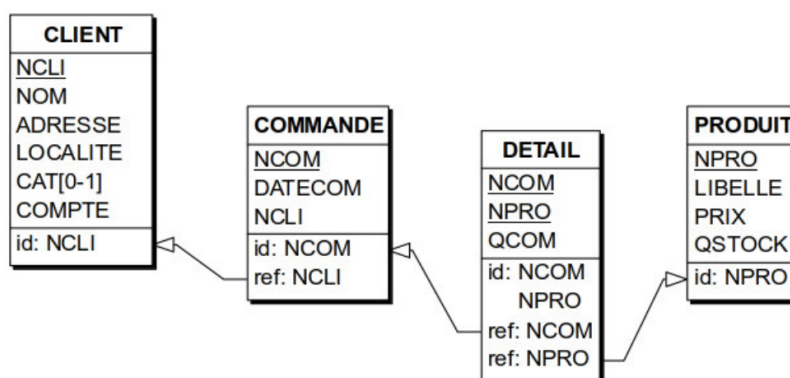


FIGURE 1 – Schéma Table

### 2.2.1 En fonction de l'image, remplir le tableau suivant en y ajoutant les noms des attributs adéquats. (Certaines cellules peuvent rester vides)

(Voir Figure 1)

Table	id primaire	un id minimal	un id simple	un id composé	les clés étrangères
CLIENT	NCLI	NCLI	NCLI	NCLI, NOM	<null>
COMMANDE	NCOM	NCOM	NCOM	NCOM, NCLI	NCLI
DETAIL	NCOM, NPRO	NCOM, NPRO	<null>	NCOM, NPRO	NCOM et NPRO
PRODUIT	NPRO	NPRO	NPRO	NPRO, PRIX	<null>

### 2.2.2 Répondre aux questions

- {NCLI, NOM} constitue-t-il un identifiant ?  
Oui, NCLI étant l'identifiant primaire de CLIENT.
- DETAIL.NCOM est-il un identifiant de DETAIL ?  
Non, car l'identifiant minimal de DETAIL est un identifiant composé {NCOM, NPRO}.
- Comment retrouver de façon sûre un client ? Et un produit ?  
Avec l'identifiant NCLI pour CLIENT, et NPRO pour PRODUIT.
- Peut-il y avoir deux clients avec le même nom ?  
Oui, mais ils auront un identifiant NCLI différent.
- Un client a-t-il toujours au moins une commande ?  
Non, dans la table CLIENT rien ne fait référence à une COMMANDE.
- Une commande doit-elle toujours avoir au moins une ligne de détail ?  
Non, dans la table COMMANDE rien ne fait référence à un DETAIL.
- Une commande peut-elle exister sans connaître le client qui l'a demandée ?  
Non, dans la table COMMANDE l'attribut NCLI n'est pas facultatif, et il référence vers CLIENT.NCLI.
- Un produit doit-il toujours avoir une quantité en stock connue ?  
Oui car QSTOCK n'est pas facultatif, mais il peut éventuellement être à 0.
- Peut-il y avoir plusieurs fois le même produit dans une commande ?  
Non, l'identifiant primaire de DETAIL l'interdit.

10. Comment retrouver de façon sûre une ligne DETAIL ?  
Avec l'identifiant {NCOM, NPRO}
11. La catégorie du client doit-elle toujours avoir une valeur connue ?  
Non car CLIENT.CAT est facultatif ([0-1]).
12. Comment retrouver les clients qui ont commandé au moins une fois ?  
Avec l'identifiant COMMANDE.NCLI.
13. Comment retrouver les produits qui n'ont jamais été commandés ?  
En retirant les DETAIL.NPRO des PRODUIT.NPRO.

## 3 Projection et sélection

### 3.1 Théorie

#### 3.1.1 SELECT $\pi_{\{attributs\text{ de la projection}\}}$ (Relation)

Toute requête **SELECT** renvoie un résultat sous la forme d'une table.

**Notation :**  $\pi_{\{attributs\text{ de la projection}\}}$  (Relation) (SELECT attributs FROM Relation)

Exemple :

NUMERONOMCLIENT =  $\pi_{\{NCLI, NOM\}}$  (CLIENT)

signifie :

SELECT NCLI, NOM FROM CLIENT;

#### 3.1.2 DISTINCT

La commande **DISTINCT** permet de récupérer les informations en ignorant les doublons.

Exemple : SELECT DISTINCT LOCALITE FROM CLIENT;

#### 3.1.3 WHERE $\sigma_{\{condition\text{ de la sélection}\}}$

La commande **WHERE** permet d'ajouter une condition à la requête.

**Notation :**  $\sigma_{\{condition\text{ de la sélection}\}}$  (Relation) (SELECT \* FROM RELATION WHERE condition)

Exemple :

CLIENTCATB2 =  $\sigma_{\{CAT="B2"\}}$  (CLIENT)

signifie :

SELECT \* FROM CLIENT WHERE CAT="B2";

#### 3.1.4 IN

SELECT NCLI FROM CLIENT WHERE CAT IN ("C1", "C2", "C3");

CAT = "C1" ou "C2" ou "C3"

#### 3.1.5 BETWEEN

SELECT NCLI FROM CLIENT WHERE COMPTE BETWEEN 1000 AND 4000;

COMPTE >= 1000 et <= 4000

#### 3.1.6 LIKE (\_ et %)

SELECT NCLI FROM CLIENT WHERE CAT LIKE "B\_";

Le '\_' remplace un caractère quelconque. (Par exemple "B2" ou "B4" ou "BX")

C'est l'équivalent du '?' dans Linux. (B?)

SELECT NPRO FROM PRODUIT WHERE LIBELLE LIKE "%SAPIN%";

Le % remplace 0 ou plusieurs caractères quelconques. (Par exemple "1SAPIN2" ou "SAPINXYZ" ou "SAPIN")

C'est l'équivalent du '\*' dans Linux. (\*SAPIN\*)

### 3.1.7 AS

SELECT NPRO AS Produits, 0.21\*prix\*qstock AS ValeurTVA FROM PRODUIT WHERE QSTOCK > 500;

On donne un alias aux colonnes. (Leur nom sera Produits et ValeurTVA)

## 3.2 Exercices

Etudiant					
EtuNo	EtuNom	EtuPnom	EtuSec	EtuAn	etuTel
32345	Dupont	Marc	R	2	0455334455
33568	Durant	Pierre	R	1	047857456
38514	Dupont	François	G	1	34578901

FIGURE 2 – Projection Selection 1

### 3.2.1 Répondre aux questions

(Voir Figure 2)

1. Fournissez une expression relationnelle donnant les identifications de sections organisées dans lesquelles au moins un étudiant est inscrit

$\pi_{\{etuSec\}}(\text{Etudiant})$

2. Fournissez une expression relationnelle donnant les identifications de sections organisées dans lesquelles au moins un étudiant de troisième bloc est inscrit

$\pi_{\{etuSec\}}(\sigma_{\{etuAn=3\}}(\text{Etudiant}))$

### 3.2.2 Donner la sémantique

(Voir Figure 2)

1.  $\sigma_{\{(etuAn=2 \text{ OU } etuAn=3) \text{ ET } etuSec='R'\}}(\text{Etudiant})$   
Donne toutes les infos des étudiants de 2ème ou 3ème de la section R.
2.  $\pi_{\{etuSec\}}(\sigma_{\{etuNom='Durant' \text{ ET } etuAn=1\}}(\text{Etudiant}))$   
Donne la section des étudiants qui s'appellent Durant et qui sont en 1ère.
3.  $\pi_{\{etuSec, etuTel\}}(\sigma_{\{etuNom \neq 'Durant' \text{ OU } etuAn \neq 1\}}(\text{Etudiant}))$   
Donne la section et le numéro de téléphone des étudiant ne s'appelant pas Durant ou n'étant pas en 1ère.
4. SELECT \*  
FROM CLIENT;  
Donne toutes les informations de la table CLIENT.
5. SELECT DISTINCT NCLI  
FROM COMMANDE  
WHERE NCOM BETWEEN 30178 AND 30188;  
Donne la liste sans doublon des numéros de clients ayant passé les commandes dont le numéro est compris entre 30179 et 30188 inclus.
6. SELECT QSTOCK\*PRIX  
FROM PRODUIT  
WHERE LIBELLE LIKE "CHE%";  
Donne la valeur du stock des produits dont le libellé commence par "CHE".
7. SELECT NOM, NCLI FROM CLIENT  
WHERE CAT NOT IN ("B1", "C1") AND CAT IS NOT null;  
Donne le nom et le numéro des clients qui ont une catégorie non nulle et autre que B1 et C1.
8. SELECT DISTINCT NPRO  
FROM PRODUIT  
WHERE LIBELLE LIKE "%SAPIN%" AND PRIX BETWEEN 100 AND 150;  
Donne les différents numéros de produit ayant SAPIN dans leur libellé et un prix entre 100 et 150 inclus.

## 4 Jointure (A) $\bowtie_{<condition>}$ (B)

### 4.1 Théorie

La jointure permet de produire une table constituée de données extraites de plusieurs tables. C'est donc l'ensemble des lignes du produit cartésien qui vérifient la condition.

**Notation :** (A)  $\bowtie_{<condition>}$  (B) (SELECT \* FROM A JOIN B ON condition)

#### 4.1.1 Produit cartésien A $\times$ B

Chaque ligne d'une table (A) est couplée avec chaque ligne d'une autre table (B).

### 4.2 Exercices

A			B		
a1	a2	a3	b1	b2	b3
10	aaaa	15	33	xxxx	10
20	bbbb	15	15	bbbb	40
30	cccc	22	41	tttt	40
40	dddd	33	49	dddd	75

FIGURE 3 – Tables Exercices Jointures

#### 4.2.1 Donner la liste des tuples de A

(Voir Figure 3)

1. La valeur de a2 apparaît dans b2

(A)  $\bowtie_{a2=b2}$  (B)

a1	a2	a3	b1	b2	b3
20	bbbb	15	15	bbbb	40
40	dddd	33	49	dddd	75

2. La valeur de a2 n'apparaît pas dans b2

(A)  $\bowtie_{a2 \neq b2}$  (B)

a1	a2	a3	b1	b2	b3
10	aaaa	15	33	xxxx	10
10	aaaa	15	15	bbbb	40
10	aaaa	15	41	tttt	40
10	aaaa	15	49	dddd	75
20	bbbb	15	33	xxxx	10
<del>20</del>	<del>bbbb</del>	<del>15</del>	<del>15</del>	<del>bbbb</del>	<del>40</del>
20	bbbb	15	41	tttt	40
20	bbbb	15	49	dddd	75
30	cccc	22	33	xxxx	10
30	cccc	22	15	bbbb	40
30	cccc	22	41	tttt	40
30	cccc	22	49	dddd	75
40	dddd	33	33	xxxx	10
40	dddd	33	15	bbbb	40
40	dddd	33	41	tttt	40
<del>40</del>	<del>dddd</del>	<del>33</del>	<del>49</del>	<del>dddd</del>	<del>75</del>

#### 4.2.2 Calculer

(Voir Figure 3)

1. (A)  $\bowtie_{a1=b3}$  ( $\sigma_{b3=75}$  (B))

=  $\emptyset$

2.  $\pi_{\{a1, b1, b2, b3\}}(\sigma_{b1=15}((A) \bowtie_{a1=b3} (B)))$

(a) D'abord (A)  $\bowtie_{a1=b3}$  (B)

a1	a2	a3	b1	b2	b3
10	aaaa	15	33	xxxx	10
40	dddd	33	15	bbbb	40
40	dddd	33	41	tttt	40

(b) Ensuite  $\sigma_{\{b1=15\}} ((A) \bowtie_a 1 = b3 (B))$

a1	a2	a3	b1	b2	b3
40	dddd	33	15	bbbb	40

(c) Enfin  $\pi_{\{a1, b1, b2, b3\}} (\sigma_{\{b1=15\}} ((A) \bowtie_{a1=b3} (B)))$

a1	b1	b2	b3
40	15	bbbb	40

3.  $\pi_{\{a1, a2, b1, b2\}} ((A) \bowtie_{a2=b2} (B))$

(a) D'abord  $(A) \bowtie_a 2 = b2 (B)$

a1	a2	a3	b1	b2	b3
20	bbbb	15	15	bbbb	40
40	dddd	33	49	dddd	75

(b) Enfin  $\pi_{\{a1, a2, b1, b2\}} ((A) \bowtie_{a2=b2} (B))$

a1	a2	b1	b2
20	bbbb	15	bbbb
40	dddd	49	dddd

4.  $(A) \bowtie_{A.a1=C.b3} ((A) \bowtie_{a3=b1} (B))C$

(a) D'abord  $(A) \bowtie_{a3=b1} (B)$  ce qui nous donne C

a1	a2	a3	b1	b2	b3
10	aaaa	15	15	bbbb	40
20	bbbb	15	15	bbbb	40
40	dddd	33	33	xxxx	10

(b) Enfin  $(A) \bowtie_{A.a1=C.b3} ((A) \bowtie_{a3=b1} (B))C$

a1	a2	a3	a1	a2	a3	b1	b2	b3
40	dddd	33	10	aaaa	15	15	bbbb	40
40	dddd	33	20	bbbb	15	15	bbbb	40
10	aaaa	15	40	dddd	33	33	xxxx	10

#### 4.2.3 Donner la sémantique (ATTENTION! RÉPONSES NON VÉRIFIÉE PAR UN ENSEIGNANT)

(Voir Figure 4)

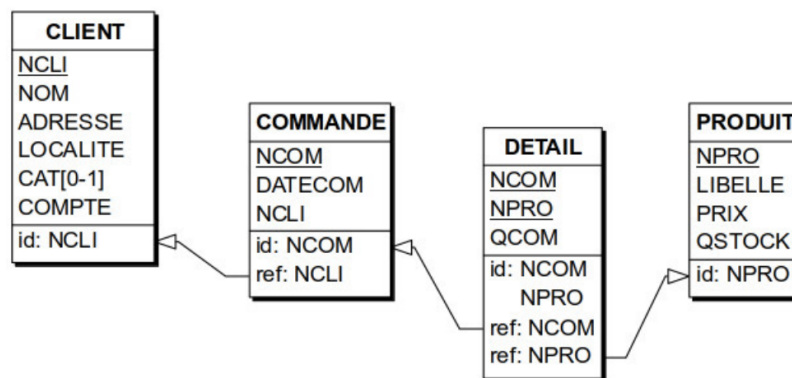


FIGURE 4 – Jointure Tables Exercices

1. SELECT DETAIL.NCOM, DETAIL.NPRO, QCOM, QSTOCK  
FROM DETAIL  
JOIN PRODUIT ON DETAIL.NPRO = PRODUIT.NPRO  
WHERE QCOM > QSTOCK;

Réponse non vérifiée pas un enseignant... Attention aux erreurs !

Retourne le numéro de commande, le numéro de produit, la quantité de commandes et la quantité en stock des produits où la quantité de commande dépasse la quantité en stock.

2. SELECT DISTINCT P.NPRO, LIBELLE, PRIX

FROM PRODUIT P

JOIN DETAIL D ON P.NPRO = D.NPRO

JOIN COMMANDE COM ON COM.NCOM = D.NCOM

JOIN CLIENT C ON C.NCLI = COM.NCLI

WHERE LOCALITE IN ("Bruxelles", "Liège", "Namur");

Réponse non vérifiée pas un enseignant... Attention aux erreurs !

Retourne le numéro de produit, le libellé et le prix des produits qui ont déjà été commandés par des client dont la localité est "Bruxelles", "Liège", ou "Namur" (sans doublon).

3. SELECT DISTINCT CLIENT.NCLI

FROM COMMANDE

JOIN CLIENT ON CLIENT.NCLI = COMMANDE.NCLI

WHERE COMPTE < 0;

Réponse non vérifiée pas un enseignant... Attention aux erreurs !

Retourne le numéro de client des clients qui ont effectué une commande et dont le compte est négatif (sans doublon).

4. Si la table CLIENT comporte 16 lignes et la table COMMANDE 7, combien de lignes seront retournées par la requête suivante : (Voir Figure 5)

SELECT \*

FROM COMMANDE

JOIN CLIENT ON CLIENT.NCLI != COMMANDE.NCLI;

CLIENT					
NCLI	NOM	ADRESSE	LOCALITE	(CAT)	COMPTE
B062	GOFFIN	72, r. de la Gare	Namur	B2	-3200
B112	HANSENNE	23, r. Dumont	Poitiers	C1	1250
B332	MONTI	112, r. Neuve	Genève	B2	0
B512	GILLET	14, r. de l'Eté	Toulouse	B1	-6700
C003	AVRON	8, r. de la Cure	Toulouse	B1	-1700
C123	MERCIER	25, r. Lemaître	Namur	C1	-2300
C400	FERRARD	65, r. du Terre	Poitiers	B2	350
D063	MERCIER	201, bd du Nord	Toulouse		-2250
F010	TOUSSAINT	5, r. Godefroid	Poitiers	C1	0
F011	PONCELET	17, Clos des Erables	Toulouse	B2	0
F400	JACOB	78, ch. du Moulin	Bruxelles	C2	0
K111	VANBIST	180, r. Florimont	Lille	B1	720
K729	NEUMAN	40, r. Bransart	Toulouse		0
L422	FRANCK	60, r. de Wépion	Namur	C1	0
S127	VANDERKA	3, av. des Roses	Namur	C1	-4590
S712	GUILLAUME	14a, ch. des Roses	Paris	B1	0

PRODUIT			
NPRO	LIBELLE	PRIX	OSTOCK
CS262	CHEV. SAPIN 200x6x2	75	45
CS264	CHEV. SAPIN 200x6x4	120	2690
CS464	CHEV. SAPIN 400x6x4	220	450
PA45	POINTE ACIER 45 (2K)	105	580
PA60	POINTE ACIER 60 (1K)	95	134
PH222	PL. HETRE 200x20x2	230	782
PS222	PL. SAPIN 200x20x2	185	1220

DETAIL		
NCOM	NPRO	QCOM
30178	CS464	25
30179	CS262	60
30179	PA60	20
30182	PA60	30
30184	CS464	120
30184	PA45	20
30185	CS464	250
30185	PA60	15
30185	PS222	600
30186	PA45	3
30188	CS464	180
30188	PA45	22
30188	PA60	70
30188	PH222	92

COMMANDE		
NCOM	NCLI	DATECOM
30178	K111	21/12/2008
30179	C400	22/12/2008
30182	S127	23/12/2008
30184	C400	23/12/2008
30185	F011	20/1/2009
30186	C400	20/1/2009
30188	B512	30/1/2009

FIGURE 5 – Tables Ex2

Réponse non vérifiée pas un enseignant... Attention aux erreurs !

105 lignes seront retournées car  $16 \times 7 - 7$ . Cette requête retourne toutes les infos des clients qui n'ont jamais passé de commandes.

5. SELECT SUP.DptNo, SUP.dptLib, DPT.dptLib

FROM DEPARTEMENT SUP

JOIN DEPARTEMENT DPT ON DPT.dptAdm=SUP.dptNo;

(Voir Figure 6)

DptNo	DptLib	DptMgr	DptAdm
A00	DEVELOPPEMENT	320	D21
B01	PRODUCTION	020	A00
C01	MAINTENANCE	030	A00
D11	SUPPORT	060	E11
D21	DIRECTION	070	null
E01	MARKETING	050	E11
E11	VENTES	340	D21
E21	FORMATION	100	E11

FIGURE 6 – Tables Ex3

Réponse non vérifiée pas un enseignant... Attention aux erreurs !

Retourne le numéro de département et le libellé de département du produit cartésien de la table DÉPARTEMENT par elle-même ainsi que le libellé de département de la table DÉPARTEMENT où l'Adm du



département de la table est égal au numéro de département du produit cartésien de la table DÉPARTEMENT par elle-même.

6. La requête suivante retournera combien de ligne ?

(Voir Figure 6)

```
SELECT SUP.dptNo, SUP.dptLib, DPT.dptLib
FROM DEPARTEMENT SUP
JOIN DEPARTEMENT DPT ON DPT.dptAdm != SUP.dptNo;
```

Réponse non vérifiée pas un enseignant... Attention aux erreurs !

8 lignes seront retournées car 64 – 56. Cette requête retourne le numéro de département et le libellé de département du produit cartésien de la table DÉPARTEMENT par elle-même ainsi que le libellé de département de la table DÉPARTEMENT où l'Adm du département de la table est différent du numéro de département du produit cartésien de la table DÉPARTEMENT par elle-même.