INF 725 – TP n°2 Bases de données

Introduction a SQLite

Table des matières

1 – Conception du schéma relationnel

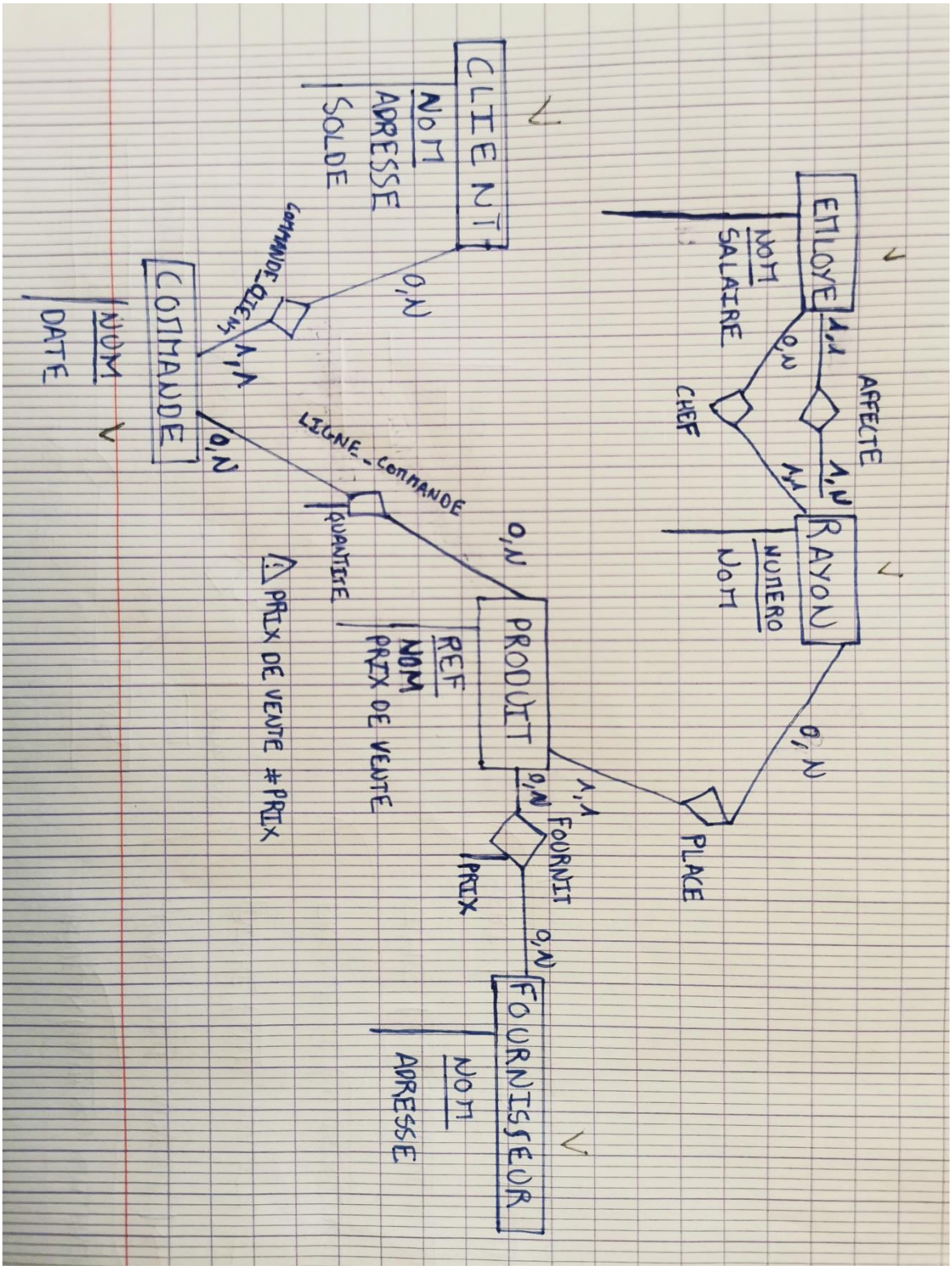


Schéma entité-association de la base de données Gestion_Magasin.db

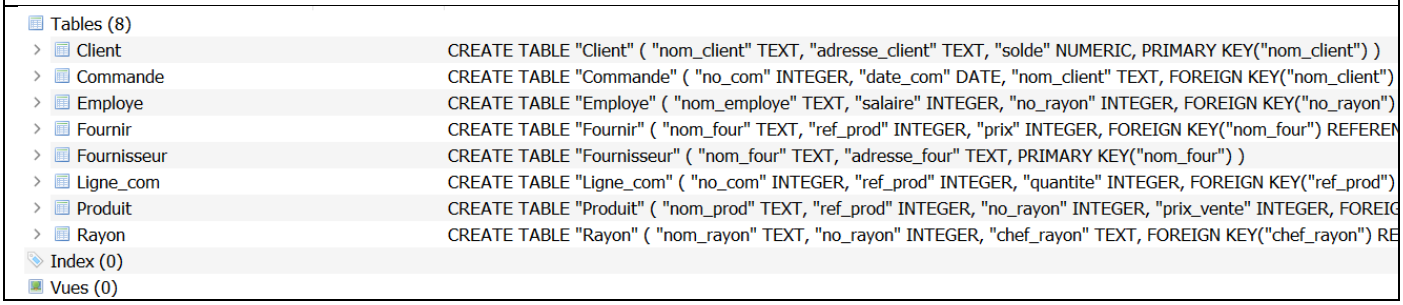
2 – Création du schéma relationnel sous SQLite

Ordre SQL pour le schéma de la table « Gestion_Magasin.db »

```
CREATE TABLE IF NOT EXISTS "Commande" (
	"no_com"	INTEGER,
	"date_com" DATE,
	"nom_client"	TEXT,
	FOREIGN KEY("nom_client") REFERENCES "Client"("nom_client"),
	PRIMARY KEY("no_com")
);
CREATE TABLE IF NOT EXISTS "Ligne_com" (
	"no_com"	INTEGER,
	"ref_prod" INTEGER,
	"quantite" INTEGER,
	FOREIGN KEY("ref_prod") REFERENCES "Produit"("ref_prod"),
	FOREIGN KEY("no_com") REFERENCES "Commande"("no_com"),
	PRIMARY KEY("no_com","ref_prod")
);
CREATE TABLE IF NOT EXISTS "Fournisseur" (
	"nom_four" TEXT,
	"adresse_four"	TEXT,
	PRIMARY KEY("nom_four")
);
CREATE TABLE IF NOT EXISTS "Fournir" (
	"nom_four" TEXT,
	"ref_prod" INTEGER,
	"prix"	INTEGER,
	FOREIGN KEY("nom_four") REFERENCES "Fournisseur"("nom_four"),
	FOREIGN KEY("ref_prod") REFERENCES "Produit"("ref_prod"),
	PRIMARY KEY("nom_four","ref_prod")
);
CREATE TABLE IF NOT EXISTS "Produit" (
	"nom_prod" TEXT,
	"ref_prod" INTEGER,
	"no_rayon" INTEGER,
	"prix_vente"	INTEGER,
	FOREIGN KEY("no_rayon") REFERENCES "Rayon"("no_rayon"),
	PRIMARY KEY("ref_prod")
);
CREATE TABLE IF NOT EXISTS "Employe" (
	"nom_employe"	TEXT,
	"salaire"	INTEGER,
	"no_rayon" INTEGER,
	FOREIGN KEY("no_rayon") REFERENCES "Rayon"("no_rayon"),
	PRIMARY KEY("nom_employe")
);
CREATE TABLE IF NOT EXISTS "Rayon" (
	"nom_rayon"	TEXT,
	"no_rayon" INTEGER,
	"chef_rayon"	TEXT,
```

```
      FOREIGN KEY("chef_rayon") REFERENCES "Employe"("nom_employe"),
      PRIMARY KEY("no_rayon")
);
CREATE TABLE IF NOT EXISTS "Client" (
      "nom_client"    TEXT,
      "adresse_client"TEXT,
      "solde"    NUMERIC,
      PRIMARY KEY("nom_client")
);
```

RESULTAT



3 – Mise à jour de la base de données

Ordre SQL pour le peuplement de la table « Gestion_Magasin.db »

```
INSERT INTO "Client" VALUES ('dumont','paris',500);
INSERT INTO "Client" VALUES ('dupont','lyon',-200);
INSERT INTO "Client" VALUES ('dupond','marseille',-300);
INSERT INTO "Client" VALUES ('dulac','paris',800);
INSERT INTO "Client" VALUES ('dumas','lyon',-300);

INSERT INTO "Commande" VALUES (1,'2013-01-01','dupont');
INSERT INTO "Commande" VALUES (2,'2014-01-05','dupond');
INSERT INTO "Commande" VALUES (3,'2014-01-18','dupont');
INSERT INTO "Commande" VALUES (4,'2014-01-25','dumas');
INSERT INTO "Commande" VALUES (5,'2015-01-31','dumas');

INSERT INTO "Fournisseur" VALUES ('f1','paris');
INSERT INTO "Fournisseur" VALUES ('f2','lyon');
INSERT INTO "Fournisseur" VALUES ('f3','marseille');

INSERT INTO "Rayon" VALUES ('jouet',1,NULL);
INSERT INTO "Rayon" VALUES ('vetement',2,NULL);
INSERT INTO "Rayon" VALUES ('jardin',3,NULL);

INSERT INTO "Employe" VALUES ('durand',1000,1);
INSERT INTO "Employe" VALUES ('dubois',1500,1);
INSERT INTO "Employe" VALUES ('dupont',2000,1);
INSERT INTO "Employe" VALUES ('dumoulin',1200,2);
INSERT INTO "Employe" VALUES ('dutilleul',1000,2);
INSERT INTO "Employe" VALUES ('duchene',2000,2);
INSERT INTO "Employe" VALUES ('duguesclin',1500,3);
INSERT INTO "Employe" VALUES ('duduche',2000,3);
```

```
UPDATE "Rayon" SET "chef_rayon" = "dupont" where "nom_rayon" = "jouet";
UPDATE "Rayon" SET "chef_rayon" = "duchene" where "nom_rayon" = 'vetement';
UPDATE "Rayon" SET "chef_rayon" = "duduche" where "nom_rayon" = 'jardin';

INSERT INTO "Produit" VALUES ('train',1,1,100);
INSERT INTO "Produit" VALUES ('avion',2,1,75);
INSERT INTO "Produit" VALUES ('bateau',3,1,70);
INSERT INTO "Produit" VALUES ('pantalon',4,2,30);
INSERT INTO "Produit" VALUES ('veste',5,2,38);
INSERT INTO "Produit" VALUES ('robe',6,2,50);
INSERT INTO "Produit" VALUES ('rateau',7,3,5);
INSERT INTO "Produit" VALUES ('pioche',8,3,7);
INSERT INTO "Produit" VALUES ('brouette',9,3,38);

INSERT INTO "Fournir" VALUES ('f1',1,90);
INSERT INTO "Fournir" VALUES ('f1',4,25);
INSERT INTO "Fournir" VALUES ('f1',5,30);
INSERT INTO "Fournir" VALUES ('f1',7,4);
INSERT INTO "Fournir" VALUES ('f2',2,70);
INSERT INTO "Fournir" VALUES ('f2',3,60);
INSERT INTO "Fournir" VALUES ('f2',6,45);
INSERT INTO "Fournir" VALUES ('f3',8,5);
INSERT INTO "Fournir" VALUES ('f3',9,32);

INSERT INTO "Ligne_com" VALUES (1,1,1);
INSERT INTO "Ligne_com" VALUES (1,4,2);
INSERT INTO "Ligne_com" VALUES (1,5,5);
INSERT INTO "Ligne_com" VALUES (2,1,3);
INSERT INTO "Ligne_com" VALUES (2,2,3);
INSERT INTO "Ligne_com" VALUES (2,8,4);
INSERT INTO "Ligne_com" VALUES (3,2,2);
INSERT INTO "Ligne_com" VALUES (3,5,1);
INSERT INTO "Ligne_com" VALUES (3,6,1);
INSERT INTO "Ligne_com" VALUES (3,7,1);
INSERT INTO "Ligne_com" VALUES (3,8,5);
INSERT INTO "Ligne_com" VALUES (4,8,10);
INSERT INTO "Ligne_com" VALUES (4,9,4);
INSERT INTO "Ligne_com" VALUES (5,1,2);
INSERT INTO "Ligne_com" VALUES (5,9,3);
```

RESULTAT

Table : Rayon

	nom_rayon	no_rayon	chef_rayon
	Filtre	Filtre	Filtre
1	jouet	1	dupont
2	vetement	2	duchene
3	jardin	3	duduche

Table : Fournisseur

	nom_four	adresse_four
	Filtre	Filtre
1	f1	paris
2	f2	lyon
3	f3	marseille

4 – Ecriture de requêtes SQL

Augmenter de 1000 le solde du client 'dumas'.

```
UPDATE "Client" SET "solde" = "solde" + 1000 WHERE "nom_client"='dumas' ;
```

RESULTAT

3	dumas	lyon	700

Supprimer les clients n'ayant jamais passé de commande.

```
DELETE FROM "Client" WHERE "nom_client" not in (select "nom_client" from
"Commande");
```

Supprimer toutes les informations liées au client 'dupont'.

```
DELETE FROM "Ligne_com" WHERE "no_com" = 1 or "no_com" = 3;
DELETE FROM "Commande" WHERE "nom_client" = "dupont";
DELETE FROM "Client" WHERE "nom_client" = "dupont";
```

RESULTAT

	nom_client	adresse_client	solde
	Filtre	Filtre	Filtre
1	dupond	marseille	-300
2	dumas	lyon	1700

	no_com	date_com	nom_client
	Filtre	Filtre	Filtre
1	2	2014-01-05	dupond
2	4	2014-01-25	dumas
3	5	2015-01-31	dumas

	no_com	ref_prod	quantite
	Filtre	Filtre	Filtre
1	2	1	3
2	2	2	3
3	2	8	4
4	4	8	10
5	4	9	4
6	5	1	2
7	5	9	3

5 – Ecriture des Triggers

Q7 : Définition d'un Trigger

Un trigger est une fonctionnalité SQL permettant de lancer de manière automatique des opérations SQL à chaque modification d'une table (DELETE, UPDATE, INSERT). Les triggers permettent d'automatiser certaines tâches nécessaires à la gestion de la base de données.

Q8 : Trigger permettant de supprimer automatiquement toutes les lignes d'une commande lorsqu'on détruit un n-uplet de la table Commande

Avant le trigger si on exécuter une commande SQL DELETE on obtient le message d'erreur suivant :

```
DELETE FROM "Commande" WHERE "nom_client" = "dupond";
```

```
L'exécution s'est terminée avec des erreurs.
Résultat : FOREIGN KEY constraint failed
À la ligne 1 :
DELETE FROM "Commande" WHERE "nom_client" = "dupond";
```

On rédige le Trigger permettant d'automatiser la suppression d'une ligne de la table Commande

```
CREATE TRIGGER log_patron_delete BEFORE DELETE on Commande
FOR EACH ROW
BEGIN
DELETE FROM Ligne_com
    WHERE Ligne_com.no_com = OLD.no_com;
END;
```

On test le trigger

```
DELETE FROM "Commande" WHERE "nom_client" = "dupond";
```

RESULTAT

	no_com	date_com	nom_client
	Filtre	Filtre	Filtre
1	4	2014-01-25	dumas
2	5	2015-01-31	dumas

	no_com	ref_prod	quantite
	Filtre	Filtre	Filtre
1	4	8	10
2	4	9	4
3	5	1	2
4	5	9	3

Q9 : Trigger permettant d'empêcher l'insertion d'un produit avec un prix négatif.

On rédige le Trigger permettant d'automatiser la suppression d'une ligne de la table Commande

```
CREATE TRIGGER negative_price
   BEFORE INSERT ON Produit
BEGIN
   SELECT
      CASE
    WHEN NEW.prix_vente < 0 THEN
         RAISE (ABORT,'Invalid Price : You are trying to insert negative
price')
       END;
END;
```

On test le trigger

```
INSERT INTO "Produit" VALUES ('train',1,1,-100)
```

RESULTAT

```
L'exécution s'est terminée avec des erreurs.
Résultat : Invalid Price : You are trying to insert negative price
À la ligne 1 :
INSERT INTO "Produit" VALUES ('train',1,1,-100);
```