

IUT Paris Descartes

Département STID Info

Nom : VAPOSTOLET

Groupe : 112

Matière : SGBD

Prénom : Aurélie

Date : 21/03/18

Notes :

16,5  
20

Appréciation :

Et avec :

1)

a) DELETE PLANTATION  
WHERE DatePlantation IS LIKE ('\*17');

1,5

b) SELECT PLANTE.LibellePlante,  
COUNT(PLANTATION.QuantitéDuJeu)  
FROM PLANTE INNER JOIN PLANTATION  
ON PLANTE.IdPlante = PLANTATION.IdPlante  
ORDER BY PLANTE.LibellePlante;  
group by ?

0,5

2) UPDATE PLANTATION SET QuantitéDuJeu

SELECT (QuantitéDuJeu FROM  
TEMP

2

WHERE PLANTATION.IdJardinier =  
TEMP.IdJardinier AND PLANTATION.IdPlante

1/3

= TEMP. IdPlante AND PLANTATION. DatePlantation  
= SYSDATE ; )

Exercice 2

1)

Série 1 :

ERREUR : Instruction 1 : Date Naiss Eleve  
n'est pas spécifiée. Validation de contrainte NOT NULL.  
CONTRAINTE : NN Date Naiss Eleve

1

Série 2 :

ERREUR : Instruction 3 : Un id Instr  
n 1 existe déjà (Instruction ?).  
CONTRAINTE : PK INSTRUMENT

1

8/2

Série 3:

ERREUR: Instruction 6: Il n'existe pas d'instrument avec l'Idinstr à 7.

1

CONTRAINTE: FK\_PROFESSEUR\_INSTRUMENT

Série 4:

ERREUR: Instruction 9: Violation de contrainte d'intégrité référentielle, la catégorie avec Idcat à 40 est référencée dans la table instrument.

2

CONTRAINTE: FK\_INSTRUMENT\_CATEGORIE

Série 5:

ERREUR: Instruction 17: Violation de contrainte NOT NULL. IdinstrPratique ne peut pas être NULL.

2

CONTRAINTE: NN\_IdinstrPratique.

2) Cette exécution rend la base incohérente car elle crée deux catégories différentes mais qui possèdent le même libellé, ce qui est illogique.

Pour empêcher cela, on peut ajouter la requête :

```
ALTER TABLE CATEGORIE
ADD CONSTRAINT U_LibelleCateq
UNIQUE (LibelleCateq);
```

```
3) ALTER TABLE ELEVE
MODIFY PK_ELEVE
ON DELETE CASCADE;
```

IUT Paris Descartes

Département STID info

Nom : CARO STOLFI

Groupe : M2

Matière : SGBD

Prénom : Arsenie

Date : 21/03/18

Note :

Appréciation :

2)

a) schéma relationnel

CATEGORIE (IdCat, LibelleCat)

INSTRUMENT (IdInstr, \* IdCat, NomInstr)

ELEVE (IdEleve, NomEleve, PrenomEleve, DateNaissEleve)

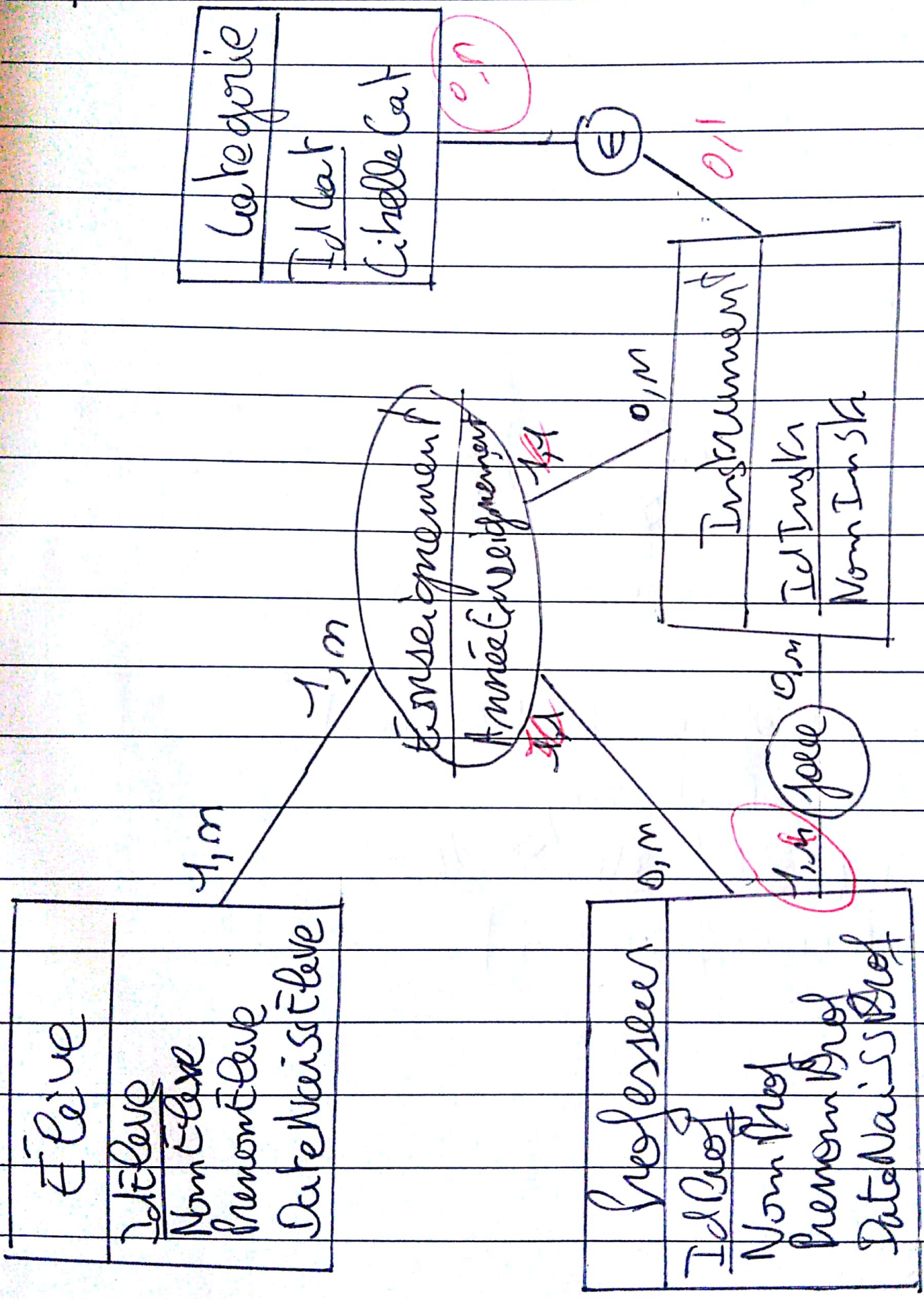
PROFESSEUR (IdProf, NomProf, PrenomProf, DateNaissProf, \* IdInstrumentique)

ENSEIGNEMENT (\* IdEleve, \* IdProf, \* IdInstr, AnnéeEnseignement)

2

5/8

b)



① On a fait l'hypothèse qu'un élève peut participer à plusieurs enseignements.

② On a fait l'hypothèse qu'un élève doit faire un enseignement pour être dans la base.

③ On a fait l'hypothèse que pour avoir lieu, un enseignement doit avoir au moins 1 élève et que plusieurs peuvent y participer.

④ On a fait l'hypothèse qu'un enseignement ne concerne qu'un instrument.

⑤ On a fait l'hypothèse qu'un instrument peut être joué par aucun ou plusieurs professeurs et être enseigné dans aucun ou plusieurs enseignements.

⑥ On a fait l'hypothèse qu'un professeur joue aucun ou plusieurs enseignements et qu'un enseignement est géré par un seul professeur.

7/8

⑤. On fait l'hypothèse qu'un professeur pose de un ou plusieurs instruments

## DST SGBD2

### Exercice 1 : EXPLOITATION D'UNE BASE

Soit l'extrait d'un schéma relationnel d'une base de données d'un maraîcher. Cette base recense les plantations faites par les différents jardiniers :

**JARDINIER** (IdJardinier, NomJardinier, PrenomJardinier)

**PLANTE** (IdPlante, LibellePlante)

**PLANTATION** (#IdJardinier, #IdPlante, DatePlantation, QuantiteDuJour)

*N.B : Les clés primaires sont soulignées. Les clés étrangères sont précédées d'un #.*

On considère que la base est en production : toutes les tables ont été créées et contiennent des données ; toutes les contraintes de clés primaires et d'intégrité référentielle sont en place.

#### 1. Il vous est demandé de traduire en SQL la requête suivante :

- Supprimer de la base les plantations de 2017.
- Afficher pour chaque plante, la quantité totale des plantations effectuées.

#### 2. Mise à jour de la table « PLANTATION » :

La requête d'insertion suivante est exécutée tous les matins pour initialiser la journée de plantation :

```
INSERT INTO PLANTATION (IdJardinier, Idplante, DatePlantation, QuantiteDuJour)
SELECT J.IdJardinier, P.IdPlante, SYSDATE, 0 FROM PLANTE P, JARDINIER J ;
```

Ci-dessous, quelques lignes ajoutées ce matin :

#### PLANTATION

IdJardinier	IdPlante	DatePlantation	QuantiteDuJour
1	...		
1	10	21/03/18	0
1	11	21/03/18	0
1	12	21/03/18	0
2	1	21/03/18	0
...			

Chaque soir, les résultats des plantations de la journée sont mis dans une table temporaire nommée « TEMP ». Un extrait de cette table est donné pour exemple :

#### TEMP

IdJardinier	IdPlante	QuantiteJournee
1	12	2
1	13	45
2	10	12
...		

Il faut donc chaque soir lancer une requête de mise à jour qui va cumuler les quantités de la journée avec celles déjà présentes dans la table « PLANTATION ».

**Question** : On vous demande de donner le code SQL de cette requête de mise à jour sur la table « PLANTATION ».

## Exercice 2 : EXPERTISE d'un SCRIPT SQL

On vous fournit en annexe le script SQL nommé « ScriptCoursMusique.sql ». Ce script est syntaxiquement correct. Il a été exécuté sur le SGBD Oracle et a permis d'implanter les tables et contraintes du script.

### 1. Analyse d'erreurs

On considère que les tables du script sont implantées et que la base est vide c'est-à-dire sans données. On exécute sur cette base vide une série de commandes SQL. Notez bien que pour chaque série on repart d'une base vide et que toutes les syntaxes SQL proposées sont correctes.

**Question** : Pour chacune des séries ci-dessous, on vous demande de déterminer si son exécution sur ORACLE génère ou non une erreur. Si oui, vous préciserez le numéro de l'instruction qui engendre l'erreur, la raison de l'erreur et le nom de contrainte concernée.

- **Série 1** : (on part d'une base vide)

1) `INSERT INTO ELEVE (IdEleve, NomEleve, PrenomEleve) VALUES (100, 'MARINI', 'LOUISE');`

- **Série 2** : (on part d'une base vide)

2) `INSERT INTO INSTRUMENT (IdInstr, NomInstr) VALUES (1, 'VIOLONCELLE');`  
3) `INSERT INTO INSTRUMENT (IdInstr, NomInstr) VALUES (1, 'GUITARE ELECTRIQUE');`

- **Série 3** : (on part d'une base vide)

4) `INSERT INTO INSTRUMENT (IdInstr, NomInstr) VALUES (1, 'GUITARE ACCOUSTIQUE');`  
`INSERT INTO PROFESSEUR (IdProf, NomProf, PrenomProf, DateNaissProf, IdInstrPratique)`  
5) `VALUES (1, 'ERMIN', 'PAUL', 1970, 1);`  
`INSERT INTO PROFESSEUR (IdProf, NomProf, PrenomProf, DateNaissProf, IdInstrPratique)`  
6) `VALUES (2, 'DURANTO', 'MARIE', 1983, 2);`

- **Série 4** : (on part d'une base vide)

7) `INSERT INTO CATEGORIE (IdCat, LibelleCat) VALUES (10, 'INSTRUMENTS A VENT');`  
8) `INSERT INTO INSTRUMENT (IdInstr, NomInstr, IdCat) VALUES (9, 'PIANO', 10);`  
9) `DELETE CATEGORIE WHERE IdCat=10;`

- **Série 5** : (on part d'une base vide)

10) `INSERT INTO INSTRUMENT (IdInstr, NomInstr) VALUES (1, 'PICOLO');`  
11) `INSERT INTO PROFESSEUR (IdProf, NomProf, PrenomProf, DateNaissProf, IdInstrPratique)`  
`VALUES (1, 'LEONARDI', 'MARIE', 1943, 1);`  
12) `UPDATE PROFESSEUR SET IdInstrPratique=NULL WHERE IdInstrPratique IN (`  
`SELECT IdInstr FROM INSTRUMENT WHERE NomInstr='PICOLO')`

### 2. Résolution d'anomalie

L'exécution des deux requêtes suivantes ne génère pas d'erreur :

1) `INSERT INTO CATEGORIE (IdCat, LibelleCat) VALUES (SeqCategorie.nextval, 'CUIVRES');`  
2) `INSERT INTO CATEGORIE (IdCat, LibelleCat) VALUES (SeqCategorie.nextval, 'CUIVRES');`

**Question** : Cette exécution rend la base incohérente. Pourquoi ? Donner la requête SQL à ajouter au script pour empêcher que cette incohérence se produise.

### 3. Modification du script

On souhaite modifier le script de la base pour prendre en compte le principe suivant : lors de la suppression d'un élève de la base, on doit supprimer tous ses enseignements associés.

Ecrire la ou les requêtes SQL à ajouter au script pour prendre en compte ce principe.

### 4. Reverse Engineering

On vous demande de fournir :

- Le schéma relationnel associé au script « ScriptCoursMusique ». Vous l'écrirez sous la forme **NOMTABLE (Colonne1, Colonne2, ...)** en soulignant la clé primaire et en préfixant par un # les clés étrangères.
- Le modèle conceptuel des données (MCD) qui a permis de générer ce schéma relationnel.

## ANNEXE : script ScriptCoursMusique.sql

### REM \*\*\*\*\* CREATION DES TABLES ET SEQUENCES

```
CREATE SEQUENCE SeqCategorie;
```

```
CREATE TABLE CATEGORIE(  
IdCat INTEGER,  
LibelleCat VARCHAR(20) CONSTRAINT NN_LibelleCat NOT NULL);
```

```
CREATE TABLE INSTRUMENT(  
IdInstr INTEGER,  
NomInstr VARCHAR(20) CONSTRAINT NN_NomInstr NOT NULL,  
IdCat INTEGER);
```

```
CREATE TABLE ELEVE(  
IdEleve INTEGER,  
NomEleve VARCHAR(20) CONSTRAINT NN_NomEleve NOT NULL,  
PrenomEleve VARCHAR(20) CONSTRAINT NN_PrenomEleve NOT NULL,  
DateNaissEleve DATE CONSTRAINT NN_DateNaissEleve NOT NULL);
```

```
CREATE TABLE PROFESSEUR(  
IdProf INTEGER,  
NomProf VARCHAR(20) CONSTRAINT NN_NomProf NOT NULL,  
PrenomProf VARCHAR(20) CONSTRAINT NN_PrenomProf NOT NULL,  
DateNaissProf INTEGER CONSTRAINT NN_DateNaissProf NOT NULL,  
IdInstrPratique INTEGER CONSTRAINT NN_IdInstrPratique NOT NULL);
```

```
CREATE TABLE ENSEIGNEMENT(  
IdEleve INTEGER,  
IdProf INTEGER,  
AnneEnseignement INTEGER);
```

*IdInstrument INTEGER*

Aucun document autorisé.  
Pas de calculatrice.  
Durée: 2h00

Année 2017-2018

**REM \*\*\*\*\* AJOUT DES CONTRAINTES**

```
ALTER TABLE INSTRUMENT
  ADD CONSTRAINT PK_INSTRUMENT PRIMARY KEY(IdInstr);
ALTER TABLE CATEGORIE
  ADD CONSTRAINT PK_CATEGORIE PRIMARY KEY(IdCat);
ALTER TABLE ENSEIGNEMENT
  ADD CONSTRAINT PK_ENSEIGNEMENT PRIMARY KEY(IdEleve, IdProf, IdInstr);
ALTER TABLE ELEVE
  ADD CONSTRAINT PK_ELEVE PRIMARY KEY(IdEleve);
ALTER TABLE PROFESSEUR
  ADD CONSTRAINT PK_PROFESSEUR PRIMARY KEY(IdProf);

ALTER TABLE INSTRUMENT ADD CONSTRAINT FK_INSTRUMENT_CATEGORIE
  FOREIGN KEY(IdCat) REFERENCES CATEGORIE(IdCat);
ALTER TABLE ENSEIGNEMENT ADD CONSTRAINT FK_ENSEIGNEMENT_ELEVE
  FOREIGN KEY(IdEleve) REFERENCES ELEVE(IdEleve);
ALTER TABLE ENSEIGNEMENT ADD CONSTRAINT FK_ENSEIGNEMENT_PROFESSEUR
  FOREIGN KEY(IdProf) REFERENCES PROFESSEUR(IdProf);
ALTER TABLE PROFESSEUR ADD CONSTRAINT FK_PROFESSEUR_INSTRUMENT
  FOREIGN KEY(IdInstrPratique) REFERENCES INSTRUMENT(IdInstr);

ALTER TABLE INSTRUMENT ADD CONSTRAINT U_NomInstr UNIQUE(NomInstr);
```

~~ALTER TABLE INSTRUMENT~~

**Barème indicatif :**

**Exercice 1 :** 6 pts

**Exercice 2 :** 14 pts