

# SQL Déclaratif

## Du standard à la pratique

### Manuel d'exercices

## MODULE I :

DDL (Data Definition Language - Langage de définition de données)

### Exercice 1.1

La syntaxe des ordres suivants est-elle correcte ? Si non, pourquoi ?

Attention, les tables sont peut-être liées... !

N'hésitez pas à tester les requêtes directement !

```
1 CREATE TABLE T_office
2 ( office_id INTEGER,
3 office_address VARCHAR(30),
4 CONSTRAINT PK_office PRIMARY KEY (office_id))
5
6 CREATE TABLE T_course
7 ( crs_code CHAR(8) NOT NULL PRIMARY KEY,
8 crs_name VARCHAR(30)
9 CONSTRAINT UK_crs UNIQUE (crs_name))
10
11 CREATE TABLE T_professor
12 ( prf_id INTEGER NOT NULL PRIMARY KEY,
13 prf_name VARCHAR(30),
14 prf_course CHAR(8)
15 CONSTRAINT PK_course REFERENCES T_course (crs_code),
16 ON DELETE SET NULL,
17 office_id CHAR(2) REFERENCES T_office,
18 CONSTRAINT prf_name UNIQUE (prf_name))
```

### Exercice 1.2

A partir des données présentées dans le tableau suivant, proposer le code de la table T\_MAINTENANCE\_MTN.

Cette table devra contenir les **4 contraintes** suivantes : **contrainte de clé primaire**, **contrainte d'unicité**, **contrainte check** et **contrainte NOT NULL**. Ces contraintes porteront sur 4 colonnes ou combinaisons de colonnes distinctes.

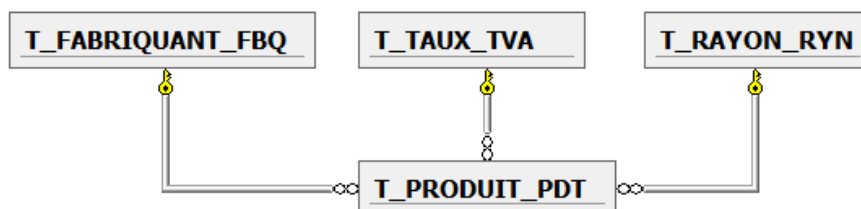
| Jour | Machine  | Numéro | Vitesse | Température | Heure  | Événement              |
|------|----------|--------|---------|-------------|--------|------------------------|
| Ven  | Massicot | 147    |         |             | 21 :18 | Défaut de lame         |
| Sam  | Relieuse | 63     | 16      |             | 16 :15 | Arrêt pour maintenance |
| Jeu  | Presse   | 87     | 6       | 62          | 11 :40 | Bavure encre           |
| Sam  | Relieuse | 79     | 16      |             | 17 :11 | Reprise                |
| Mer  | Presse   | 89     | 6       | 55          | 08 :28 | Recadrage              |
| Mar  | Presse   | 132    | 8       | 68          | 09 :58 | Changement encre       |
| Mer  | Massicot | 111    |         |             | 10 :17 | Graissage coulisseau   |

### Exercice 1.3

Créer une table pour y stocker les produits à vendre, avec les rubriques suivantes : identifiant, référence magasin, référence fabricant, code EAN13, prix de vente.

Cette table fera en outre référence aux tables T\_TAUX\_TVA, T\_RAYON\_RYN, T\_FABRICANT\_FBQ.

Mettez en place toutes les contraintes nécessaires. La table produit contiendra au minimum les colonnes proposées, mais peut en contenir d'autres au besoin.



## Exercice 1.4

Soit le code de création de table repris ci-après et pour lequel les annotations suivantes concernant les fonctions utilisées pourront être utiles (sous Oracle, demander le script et des explications au formateur) :

- « RTRIM(...) » et « LTRIM(...) » enlèves les espaces blancs respectivement à droite et à gauche de l'élément entre parenthèses
- « SUBSTRING(...,x,y) » renvoie la chaîne de caractère commençant à « x » et se terminant « y » caractères après « x », à partir de la chaîne de caractères donnée entre parenthèses
- « CONVERT(TYPE,...) » renvoie la valeur fournie dans le « TYPE » demandé

```

1 CREATE TABLE T_VOITURE_VTR (
2     VTR_ID                INTEGER    NOT NULL PRIMARY KEY,
3     VTR_IMMATRICUL        CHAR(10)   NOT NULL UNIQUE,
4     VTR_CARBURANT          CHAR(2)    NOT NULL DEFAULT 'ES' CHECK(VTR_CARBURANT IN ('ES','GO','PL')),
5     VTR_PUISSANCE_FISC    INTEGER    NOT NULL CHECK(VTR_PUISSANCE_FISC BETWEEN 1 AND 20),
6     VTR_NB_PLACES          INTEGER    NOT NULL CHECK(VTR_NB_PLACES BETWEEN 1 AND 7),
7     VTR_MODELE             VARCHAR(20) CHECK(RTRIM(LTRIM(VTR_MODELE)) NOT LIKE ''),
8     VTR_CONSTRUCTEUR       VARCHAR(16) CHECK(RTRIM(LTRIM(VTR_CONSTRUCTEUR)) NOT LIKE ''),
9     VTR_NUMERO_SERIE       VARCHAR(25) NOT NULL CHECK(RTRIM(LTRIM(VTR_NUMERO_SERIE)) NOT LIKE ''),
10    CONSTRAINT CK_IMMATRICULATION CHECK(((CONVERT(INTEGER,SUBSTRING(VTR_IMMATRICUL,9,1)) BETWEEN 0 AND 9)
11        AND (SUBSTRING(VTR_IMMATRICUL,10,1) BETWEEN '0' AND '9')
12        AND (SUBSTRING(VTR_IMMATRICUL,9,2) < '96'))
13        OR ((CONVERT(INTEGER,SUBSTRING(VTR_IMMATRICUL,9,1)) = 2)
14        AND (SUBSTRING(VTR_IMMATRICUL,10,1) IN ('A','B')))),
15    CONSTRAINT CK_PUISS_PLACE CHECK(VTR_NB_PLACES - 1 < VTR_PUISSANCE_FISC),
16    CONSTRAINT UK_MDL_CTR_NSR UNIQUE (VTR_MODELE,VTR_CONSTRUCTEUR,VTR_NUMERO_SERIE)
17 -)

```

Parmi les lignes suivantes, lesquelles seront refusées et pourquoi ?

| ID | IMMA          | CARB  | PUISS | PLC | MDL          | CONST      | NUM_SERIE    |
|----|---------------|-------|-------|-----|--------------|------------|--------------|
| 14 | '478 XDA 78'  | 'ES'  | 9     | 5   | '305'        | 'PEUGEOT'  | '00014578'   |
| 31 | '1447 MD 44'  | 'ES'  | '7'   | 5   |              | 'CITROEN'  | '0001578'    |
| 7  | '5475 MRT 91' | 'GO'  | 5     | 4   | '204'        | 'PEUGEOT'  | '0001474578' |
| 11 | '1744 BC 76'  | 'GO'  | 7     | 5   |              |            | '00025678'   |
| 15 | '4412 LR 75'  | 'GO'  | 7     | 4   | '305'        | 'PEUGEOT'  | '00014578'   |
| 17 | '971 VTR 96'  |       | 7     | 5   | '306'        | 'PEUGEOT'  | '00017548'   |
| 19 | '991 SDT 75'  | 'ES'  | 8     | 5   | 'MEGANE'     | 'RENAULT'  | '00014578'   |
| 20 | '991 SDT 75'  | 'ES'  | 5     | 4   | 'MEGANE'     | 'RENAULT'  | '00014578'   |
| 14 | '4875 ZT 94'  |       | 7     | 5   |              | 'RENAULT'  | '005784'     |
| 7  | '5474 MRT 91' | 'GPL' | 5     | 4   | 'PT CRUISER' | 'CHRYSLER' | '0000050214' |

### **Exercice 1.5**

Deux scripts vous sont fournis : « DBSlide\_LoadDB.sql » et « DBSlide\_LoadData.sql ». Créer une base de données que l'on appellera « DBSlide ». Tenter d'exécuter les scripts fournis... Cela ne devrait pas fonctionner. A vous de les corriger !

### **Exercice 1.6**

Une fois les scripts de l'exercice précédent corrigés, les tables créées et remplies, réaliser les modifications suivantes :

- Autoriser la table « SECTION » à accepter des valeurs NULL pour la colonne « delegate\_id »
- Ajouter à la table « SECTION » une clé étrangère faisant pointer la colonne « delegate\_id » vers la colonne « student\_id » de la table « STUDENT »
- Supprimer la colonne « course\_id » de la table « STUDENT »
- Faire en sorte que les données de la colonne « student\_id » de la table « STUDENT » soient auto-incrémentées
- En ne supprimant aucune donnée, modifier le type de la colonne « section\_id » de la table « section » afin qu'il soit en CHAR(4). Cela impliquera peut-être d'autres modifications...

### **Exercice 1.7**

Améliorer le script « DBSlide\_LoadDB.sql » afin qu'il commence par supprimer les tables, pour ensuite les recréer sans leurs clés étrangères. Une fois chaque table créée, leur rajouter les clés étrangères

### **Exercice 1.8**

Afin de partir sur des bases communes pour les exercices à venir, exécuter les scripts « DBSlide\_LoadDB\_OK.sql » et « DBSlide\_loadData\_OK.sql »  
(ou « DBSlide\_LoadDB\_Oracle.sql » et « DBSlide\_LoadData\_Oracle.sql », sous Oracle)

```

1  -- SOLUTIONS Module 1
2  -- Exercice 1.1
3  -- CORRECT
4  CREATE TABLE T_office (
5      office_id INTEGER,
6      office_adresse VARCHAR(30),
7      CONSTRAINT PK_office PRIMARY KEY (office_id)
8  );
9
10 -- crs_name
11 create table T_course (
12     crs_code CHAR(8) PRIMARY KEY,
13     crs_name VARCHAR(30),
14     CONSTRAINT UK_crs UNIQUE (crs_name)
15 );
16
17 -- "virgule" / type office_id
18 CREATE TABLE T_professor(
19     prf_id INTEGER IDENTITY PRIMARY KEY,
20     prf_name VARCHAR(30),
21     prf_course CHAR(8)
22     CONSTRAINT PK_course REFERENCES T_course (crs_code) ON DELETE SET NULL,
23     office_id int REFERENCES T_office,
24     CONSTRAINT UK_name UNIQUE (prf_name)
25 );
26
27 -- Exercice 1.2
28 Create table T_maintenance_mtn (
29     jour char(3) not null,
30     machine varchar(20) not null,
31     numero integer not null,
32     vitesse integer,
33     temperature integer,
34     heure time not null,
35     evenement varchar(50) not null,
36     constraint PK_maintenance_mtn primary key (jour, heure, machine, numero),
37     constraint U_maintenance_mtn unique (machine, numero, heure),
38     constraint C_vitesse check (vitesse >= 0),
39     constraint C_Maintenance CHECK (jour in ('Lun', 'Mar', 'Mer', 'Jeu', 'Ven', 'Sam', 'Dim'))
40 );
41
42 -- Exercie 1.3
43 CREATE TABLE T_Fabricant_FBQ(
44     FBQ_ID int PRIMARY KEY
45 );
46
47 CREATE TABLE T_Taux_TVA (
48     TVA_ID int constraint PK_Taux Primary Key
49 );
50
51 CREATE TABLE T_Rayon_RYN (
52     RYN_ID int,
53     CONSTRAINT PK_Rayon PRIMARY Key (RYN_ID)
54 );
55
56 CREATE TABLE T_produit_PDT(
57     PDT_ID int Primary key ,
58     Ref_magasin varchar(50),
59     ref_fabriquant varchar(50),
60     code_ean13 char(13),
61     prix decimal (6,2),
62     TVA_ID int,
63     FBQ_ID int,
64     RYN_ID int,
65     constraint FK_produit_Fabricant foreign Key (FBQ_ID) REFERENCES T_Fabricant_FBQ (FBQ_ID),
66     constraint FK_produit_Taux foreign key (TVA_ID) REFERENCES T_Taux_TVA (TVA_ID),
67     constraint FK_produit_rayon foreign key (RYN_ID) REFERENCES T_Rayon_RYN (RYN_ID)
68 );

```



```

69 -- Exercice 1.4
70 -- les lignes refusées?
71 -- Ligne 1 (ID 14): OK 1ere qui passe
72 -- Ligne 2 (ID 31): OK - MODELE pas de problème sauf si ' ' / PUISSANCE conversion automatique ca marche
73 -- Ligne 3 (ID 7): NOT OK immatriculation > 10 car l'espace compte comme caractère
74 -- Ligne 4 (ID 11): OK !Null ce n'est pas un vide!
75 -- Ligne 5 (ID 15): NOT OK car problème unicité avec la 1ere ligne avec CONST et Num_serie sont identiques
76 -- Ligne 6 (ID 17) : NOT OK - Immatriculation 95 max - par contre le default agira sur la colonne Carburant
77 -- Ligne 7 (ID 19): OK -peut avoir 2 Num serie les même pour 2 constructeurs différents
78 -- Ligne 8 (ID20): NOT OK - meme num serie et meme nume plaque
79 -- Ligne 9 (ID 14): NOT OK - ID 14 déjà pris par la 1er ligne
80 -- Ligne 10 (ID 7): NOT OK - ID primary key pas respecté + carburant n'ets pas dans les choix
81 -- + carburant 3 caractères + Immatriculation est trop longue
82
83 -- Exercice 1.5
84 -- Ordre Cle primaire -> cle étrangère
85 -- SECTION - PROF - COURSE - STUDENT - GRADE
86 -- reference clé int et bon de colonne
87
88 CREATE TABLE section (
89     section_id int NOT NULL,
90     section_name varchar(50),
91     delegate_id int NOT NULL
92     CONSTRAINT PK_section PRIMARY KEY (section_id)
93 )
94
95 CREATE TABLE professor (
96     professor_id int NOT NULL,
97     professor_name varchar(30) NOT NULL,
98     professor_surname varchar(30) NOT NULL,
99     section_id int NOT NULL,
100     professor_office int NOT NULL,
101     professor_email varchar(30) NOT NULL,
102     professor_hire_date datetime NOT NULL,
103     professor_wage int NOT NULL,
104     CONSTRAINT PK_professor PRIMARY KEY (professor_id),
105     constraint FK_professor_section foreign key (section_id) references section (section_id)
106 )
107
108 CREATE TABLE course (
109     course_id varchar(8) NOT NULL ,
110     course_name varchar(200) NOT NULL ,
111     course_ects decimal(3,1) NOT NULL,
112     professor_id int NOT NULL,
113     CONSTRAINT PK_course PRIMARY KEY (course_id),
114     constraint FK_course_professor foreign key (professor_id) references professor (professor_id)
115 )
116
117 CREATE TABLE student (
118     student_id int NOT NULL,
119     first_name varchar(50),
120     last_name varchar(50),
121     birth_date datetime,
122     login varchar(50),
123     section_id int,
124     year_result int,
125     course_id varchar(6) NOT NULL,
126     CONSTRAINT PK_student PRIMARY KEY (student_id),
127     constraint FK_student_section foreign key (section_id) references section (section_id)
128 )
129
130 CREATE TABLE grade (
131     grade char(2) NOT NULL check (grade in ('E','TB','B','S','F','I','IG')),
132     lower_bound int NOT NULL,
133     upper_bound int NOT NULL,
134     CONSTRAINT PK_grade PRIMARY KEY (grade)
135 )
136
137
138 -- Insert into
139 -- order d'execution SECTION - PROF - COURSE - STUDENT - GRADE
140 -- ID section 1110 existait pas (referencer prof et student)
141 -- Id professor 8 dans table course n'existait pas -> 6
142 -- le format date
143 -- grade 'N/A' trop long et pas possible
144
145

```



```
145
146 -- Exercise 1.6
147 Alter table section Alter column delegate_id int NULL
148
149 ALTER TABLE section add constraint FK_section_student foreign key (delegate_id)
150     references student (student_id);
151
152 ALTER TABLE student DROP COLUMN cours_id;
153
154 ALTER TABLE section drop constraint Fk_section_student
155 ALTER TABLE student drop constraint PK_student
156 ALTER TABLE student ADD student_id2 int identity(1,1) constraint PK_student Primary Key
157 ALTER TABLE student DROP column student_id
158 ALTER TABLE section ADD constraint FK_section_student foreign key (delegate_id)
159     references student (student_id2);
160
161
162 ALTER TABLE student DROP constraint FK_section_student
163 ALTER TABLE professor DROP constraint FK_section_professor
164 ALTER TABLE section DROP CONSTRAINT PK_section
165 ALTER TABLE section ALTER COLUMN section_id char(4)
166 ALTER TABLE student ALTER COLUMN section_id char(4)
167 ALTER TABLE professor ALTER COLUMN section_id char(4)
168 ALTER TABLE section ADD CONSTRAINT PK_section PRIMARY KEY (section_id)
169 ALTER TABLE student add constraint FK_section_student foreign key (section_id) references section (section_id)
170 ALTER TABLE student add constraint FK_section_professor foreign key (section_id) references section (section_id)
171
172 -- Find snippet of all this code @ https://snipit.io/lists/15914/50880
```