

## 3. LE LANGAGE SQL DDL

Hend AFFES  
Jean-Yves Tigli

BAT3 – Polytech Nice – Sophia

2013-2014

# 1. LE LANGAGE SQL DDL

## Contenu

1. Introduction
2. Création d'une table
3. Modification d'une table
4. Les structures physiques
5. Exemple

## 1.1 INTRODUCTION

- Les SGBD qui présentent les données sous la forme de tables proposent un langage de requête dénommé SQL (Structured Query Language).
- Présenté pour la première fois en 1973 par une équipe de chercheurs d'IBM: ce langage a été rapidement adopté comme standard potentiel et pris en charge par les organismes de normalisation ANSI et ISO.
- SQL permet de créer des structures de données et de les modifier. Il est principalement constitué de:
  - Le sous-langage DDL (Data Definition Language)
  - Le sous-langage DML (Data Manipulation Language)
- Etude de SQL92 ou SQL2
- Langage standardisé mais les éditeurs de SGBD ont pris beaucoup de libertés
- une instruction SQL constitue une requête: la description d'une opération que le SGBD doit exécuter.

## 1.1 Introduction

### **Les opérations du DDL :**

Création d'une table

colonnes (obligatoire)

domaine

identifiants primaire et secondaire

clé étrangère

Suppression d'une table

Ajout, suppression, modification d'une colonne

Ajout, suppression d'une contrainte

Ajout, suppression d'un index

Création d'un espace de stockage

## 1.2 CREATION D'UNE TABLE

### Contenu

- a) **Tables, colonnes et domaines**
- b) **Les identifiants**
- c) **Les clés étrangères**
- d) **Suppression d'une table**

## 1.2 Création d'une table - Tables, colonnes et domaines

### Création d'une table et de ses colonnes

CLIENT
NCLI: char (10)
NOM: char (32)
ADRESSE: char (60)
LOCALITE: char (30)
CAT[0-1]: char (2)
COMPTE: num (9,2)

```
create table CLIENT ( NCLI      char(10) ,  
                     NOM       char(32) ,  
                     ADRESSE   char(60) ,  
                     LOCALITE  char(30) ,  
                     CAT       char(2) ,  
                     COMPTE    decimal(9,2) );
```

-+ Production d'une table vide dont le schéma est conforme aux indications données par la requête create table

## 1.2 Création d'une table - Tables, colonnes et domaines

### Les domaines

Définition d'un domaine :

```
create domain MONTANT decimal(9,2);  
create domain MATRICULE char(10);  
create domain LIBELLE char(32);
```

Utilisation d'un domaine :

```
create table CLIENT ( NCLI      MATRICULE,  
                     NOM       LIBELLE,  
                     ADRESSE   char(60),  
                     LOCALITE  LIBELLE,  
                     CAT       char(2),  
                     COMPTE    MONTANT );
```

## 1.2 Création d'une table - Tables, colonnes et domaines

### Les colonnes obligatoires

Une colonne est **facultative** par défaut.  
Il faut déclarer explicitement les colonnes **obligatoires**

```
create table CLIENT ( NCLI      char(10) not null,  
                     NOM       char(32) not null,  
                     ADRESSE   char(60) not null,  
                     LOCALITE  char(30) not null,  
                     CAT       char(2) ,  
                     COMPTE    decimal(9,2) not null  
                     );
```

## 1.2 Création d'une table - Tables, colonnes et domaines

### Valeur par défaut d'une colonne

Sera assignée à la colonne si on ne spécifie pas de valeur lors de la création d'une ligne

```
create table CLIENT (  
    NCLI      char(10) not null,  
    NOM       char(32) not null,  
    ADRESSE   char(60) not null,  
    LOCALITE  char(30) not null default 'Paris',  
    CAT       char(2)  default 'B1',  
    COMPTE    decimal(9,2) not null default 0.0  
);
```

```
create domain MONTANT decimal(9,2) default 0.0;
```

## 1.2 Création d'une table - Les identifiants

### Les identifiants primaires (*primary key*)

CLIENT
<u>NCLI</u>
NOM
ADRESSE
LOCALITE
CAT[0-1]
COMPTE
id: NCLI

```

create table CLIENT ( NCLI      char(10) not null,
                     NOM       char(32) not null,
                     ADRESSE   char(60) not null,
                     LOCALITE  char(30) not null,
                     CAT       char(2) ,
                     COMPTE    decimal(9,2) not null,
                     primary key (NCLI) );
  
```

## 1.2 Création d'une table - Les identifiants

### Les identifiants secondaires (*candidate key*)

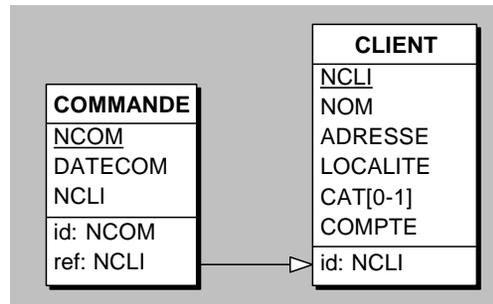
CLIENT
<u>NCLI</u>
NOM
ADRESSE
LOCALITE
CAT[0-1]
COMPTE
id: NCLI
id': NOM

```

create table CLIENT ( NCLI      char(10) not null,
                     NOM       char(32) not null,
                     ADRESSE   char(60) not null,
                     LOCALITE  char(30) not null,
                     CAT       char(2) ,
                     COMPTE    decimal(9,2) not null,
                     primary key (NCLI) ,
                     unique (NOM) );
  
```

## 1.2 Création d'une table - Les clés étrangères

### Les clés étrangères (*foreign key*)



```

create table COMMANDE (NCOM      char(12) not null,
                      NCLI      char(10) not null,
                      DATECOM   date not null,
                      primary key (NCOM),
                      foreign key (NCLI) references CLIENT);
  
```

Si l'identifiant cible est secondaire (*possible mais déconseillé*) :

```

foreign key (NOM) references CLIENT (NOM)
  
```

## 1.2 Création d'une table - Les clés étrangères

### Variante : contraintes de colonne

```
create table CLIENT (  
    NCLI char(10) not null primary key,  
    . . .  
);
```

```
create table CLIENT (  
    NCLI char(10) not null,  
    NOM char(32) not null unique,  
    . . .  
);
```

```
create table COMMANDE (  
    NCOM char(12) not null,  
    NCLI char(10) not null references CLIENT,  
    . . . );
```

## 1.2 Création d'une table - Les clés étrangères

### Variante : contraintes nommées

```
create table COMMANDE (NCOM      char(12) not null,  
                       NCLI      char(10) not null,  
                       DATECOM   date not null,  
                       constraint COMPK primary key (NCOM),  
                       constraint COMFK foreign key (NCLI)  
                       references CLIENT);
```

```
create table COMMANDE (  
    NCOM char(12) constraint COMPK not null,  
    NCLI char(10) not null constraint COMFK  
    references CLIENT,  
    . . . );
```

## 1.2 Création d'une table - Suppression d'une table

### Suppression d'une table

```
drop table COMMANDE;
```

#### Attention, opération sous haute surveillance !

- au préalable, l'opération supprime les lignes de la table; encore faut-il que ce soit possible (attentions aux lignes de DETAIL dépendantes)
- la table ne doit plus être référencée par une clé étrangère;

## 1.3 MODIFICATION D'UNE TABLE

### Contenu

- a) Ajout, retrait et modification d'une colonne
- b) Ajout et retrait d'une contrainte

## 1.3 Modification d'une table - Colonnes

### Ajout, retrait et modification d'une colonne

```
alter table PRODUIT add column POIDS smallint;
```

ajouter

```
alter table PRODUIT drop column PRIX;
```

supprimer

```
alter table CLIENT modify column CAT set '00';
```

modifier valeur  
par défaut

```
alter table CLIENT modify column CAT drop default;
```

supprimer valeur  
par défaut

## 1.3 Modification d'une table - Contraintes

### Ajout et retrait d'une contrainte

```
alter table PROSPECT add primary key (NCLI);
```

```
alter table CLIENT add unique (NOM,ADRESSE,LOCALITE);
```

```
alter table CLIENT modify CAT not null;
```

```
alter table CLIENT modify ADRESSE null;
```

```
alter table CLIENT  
add foreign key (CAT) references CATEGORIE ;
```

## 1.3 Modification d'une table - Contraintes

### Variante : contraintes nommées

```
alter table PROSPECT  
add constraint PROPK primary key (NCLI);
```

```
alter table CLIENT  
add constraint CLIUN unique (NOM,ADRESSE,LOCALITE);
```

```
alter table CLIENT  
add constraint CLIFK1 foreign key (CAT)  
references CATEGORIE ;
```

## 1.3 Modification d'une table - Contraintes

### Suppression d'une contrainte nommée

```
alter table PROSPECT  
drop constraint PROPK;
```

## 1.4 LES STRUCTURES PHYSIQUES

Contenu

a) Les espaces de stockage

## 1.4 Les structures physiques - Les espaces de stockage

### Les espaces de stockage



CLI\_PRO\_DAT  
CLIENT  
PRODUIT

```
create dbspace CLI_PRO_DAT;
```

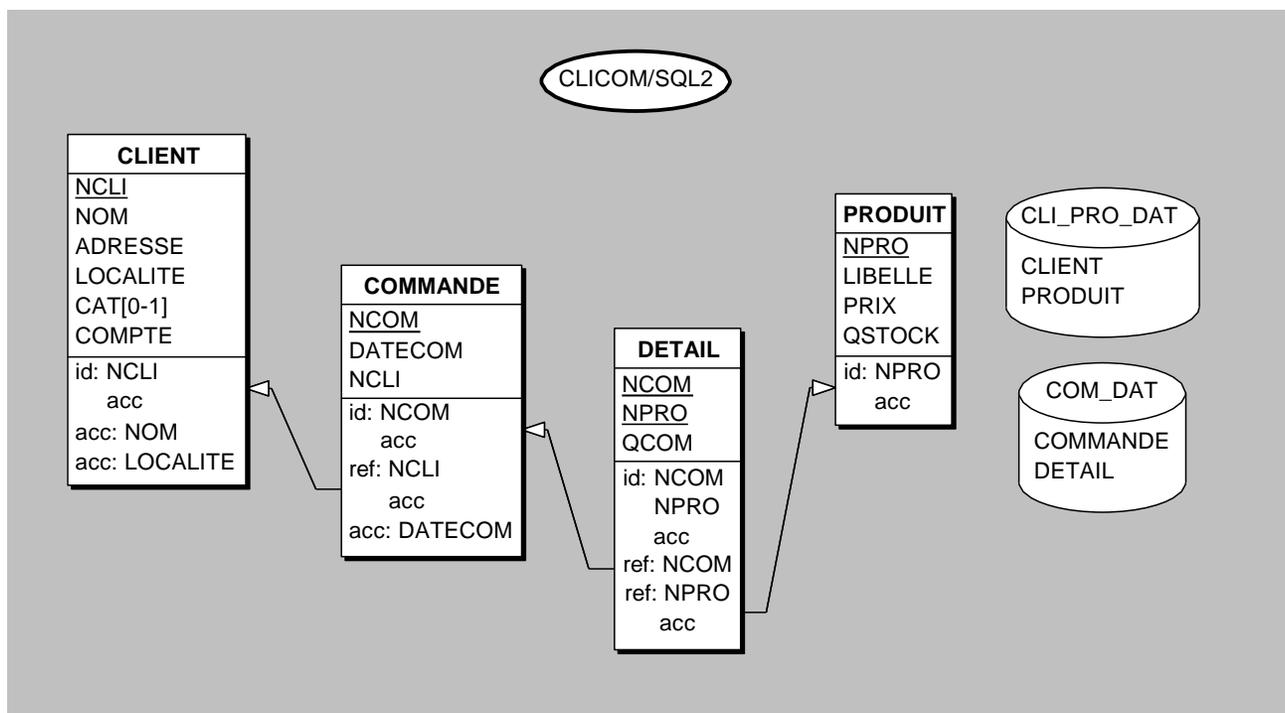
```
create table CLIENT ( . . . ) in CLI_PRO_DAT;
```

```
drop dbspace CLI_PRO_DAT;
```

si espace vide

## 1.5 Exemple - Le schéma

### Traduire ce schéma en SQL



## 1.5 Exemple - Le schéma et les espaces de stockage

```
create schema CLICOM;  
  
create dbspace CLI_PRO_DAT;  
  
create dbspace COM_DAT;
```

## 1.5 Exemple - Les tables

```

create table CLIENT (  NCLI      char(10) not null,
                      NOM       char(32) not null,
                      ADRESSE   char(60) not null,
                      LOCALITE  char(30) not null,
                      CAT       char(2),
                      COMPTE    decimal(9,2) not null,
                      primary key (NCLI) ) in CLI_PRO_DAT;

create table PRODUIT (  NPRO      char(15) not null,
                      LIBELLE   char(60) not null,
                      PRIX      decimal(6) not null,
                      QSTOCK    decimal(8) not null,
                      primary key (NPRO) ) in CLI_PRO_DAT;

create table COMMANDE (NCOM      char(12) not null,
                      NCLI      char(10) not null,
                      DATECOM   date not null,
                      primary key (NCOM),
                      foreign key (NCLI) references CLIENT) in COM_DAT;

create table DETAIL (  NCOM      char(12) not null,
                      NPRO      char(15) not null,
                      QCOM      decimal(8) not null,
                      primary key (NCOM,NPRO),
                      foreign key (NCOM) references COMMANDE,
                      foreign key (NPRO) references PRODUIT) in COM_DAT;

```