

BDC

DESCRIPTION D'UN SCHEMA DE BASE DE DONNEES

**SQL : Un langage de
description des données**

SQL (Structured Query Language)

- SQL :
 - Conçu en 1976 dans les laboratoires de recherche d'IBM
 - Initialement appelé SEQUEL (Structured English QUery Language)
 - Est dérivé de l'algèbre relationnelle
 - Existe en plusieurs versions normalisées : SQL 86, SQL 89, SQL 92 ou SQL 2 (la version minimale existe dans ~tous les SGBDs), SQL 99 ou SQL 3
 - Est un langage déclaratif
 - Est utilisable en interactif ou avec un langage hôte (C, JAVA, COBOL, etc..)
- SQL comprend :
 - **Le langage de définition de schéma (LDD)**
 - Le langage de manipulation de schéma (LMD)

DESCRIPTION DU SCHEMA RELATIONNEL A L'AIDE DE SQL

- On peut spécifier :
 - Au niveau conceptuel
 - Les schémas des tables
 - Les domaines des attributs
 - Contraintes d'intégrité de la base (clé primaire, clé étrangère, etc.)
 - Au niveau externe
 - Les vues ou sous-schémas
 - Les privilèges d'accès aux tables
 - Au niveau interne
 - Les index
 - Les clusters
 -

TYPE DE DONNEES

- Type caractère

- Chaîne de caractères de **longueur fixe**

- CHAR (longueur)**

- longueur par défaut = 1 car.

- longueur max = 2000 car.

- Chaîne de caractères de **longueur variable**

- VARCHAR2 (longueur)**

- longueur par défaut = 1 car.

- longueur max = 4000 car.

- La chaîne de caractères n'occupe que l'emplacement correspondant à sa taille réelle

TYPE DE DONNEES : exemple

- Pour une déclaration CHAR (4) :

La chaîne 'bases de données' ne pourra être stockée

La chaîne 'ABCD' sera stockée 'ABCD'

La chaîne 'UE' sera stockée 'UE '

- Pour une déclaration VARCHAR2(4) :

La chaîne 'bases de données' ne pourra être stockée

La chaîne 'UE' sera stockée 'UE'

La chaîne 'ABCD' sera stockée 'ABCD'

TYPE DE DONNEES

- Type numérique

- Les nombres entiers

NUMBER (nombre_chiffres)

nombre_chiffres par défaut = 38

INTEGER

- Les nombres décimaux

NUMBER (nombre_chiffres, échelle)

DECIMAL (nombre_chiffres, échelle)

nombre_chiffres par défaut = 38

échelle = nombre_chiffres à droite de la marque décimale

échelle comprise entre -84 et +127

échelle négative signifie un arrondissement;

FLOAT (precision)

TYPE DE DONNEES :exemple

- Pour une déclaration NUMBER(3)
 - Un nombre de 3 chiffres stocké a la valeur $\in [0, 999]$
 - Le nombre 2456 ne pourra être stocké
 - Le nombre 245.78 sera stocké 246
 - Le nombre 245.4 sera stocké 245
- Pour une déclaration INTEGER
- Pour une déclaration NUMBER(7, 2)
 - Un nombre de 7 chiffres dont 2 après la virgule.
 - Ce nombre stocké a la valeur $\in [-99999.99, +99999.99]$
 - Le nombre 2456.78 sera stocké 2456.78
 - Les nombres 2456.783 et 124569.78 ne pourront être stocké
 - Le nombre 2456 sera stocké 2456
- Pour une déclaration NUMBER(4, -2)
 - Le nombre réel est arrondi à la centaine (à gauche de la virgule)
 - Le nombre 345.6 sera stocké 300
 - Le nombre 395.6 sera stocké 400

TYPE DE DONNEES

- Type booléen

- **BIT(n)**

vecteur de n bits

- Type date

- **DATE**

- Gestion des dates selon un format interne du système

- exemple de format :

DeuxChiffrePourJour-TroisPremièreLettrePourMois-DeuxChiffrePourAnnée

QuatreChiffrePourAnnée-DeuxChiffrePourMois-DeuxChiffrePourJour

TYPE DE DONNEES

■ Type heure

- **TIME**

- Gestion des heures selon un format interne du système

- Exemple de format

DeuxChiffrePourHeure: DeuxChiffrePourMinute: DeuxChiffrePourSeconde

■ Type date-heure

- **TIMESTAMP**

- Combine la date avec l'heure.

TYPE DE DONNEES

- Les constantes
 - Constantes alphabétiques
 - exemple : ' COURS ' ou ' L ' 'ETUDIANT '
 - Constantes numériques
 - exemple : 10 ou 10,80
 - Constantes dates
 - exemple : DATE '01-FEB-00 '
 - Constantes heures
 - exemple : TIME '14:04:32'

DESCRIPTION D'UN ATTRIBUT

nomAttribut type [DEFAULT expression]

[Contrainte]



CHAR(n)
VARCHAR(n)
NUMBER(n)
NUMBER(n,m)
DATE
TIME
.....



constante
SYSDATE
LN(valeur)
.....



NOT NULL
UNIQUE
PRIMARY KEY
REFERENCES nomtable
.....

CREATION DE SCHEMA DE BD ET DE TABLES

- Création d'un schéma de base de données

- CREATE SCHEMA nomSchéma;

- Création d'une table

```
CREATE TABLE nomTable
```

```
( descriptionAttribut1 ,
```

```
  descriptionAttribut2 ,
```

```
  .....
```

```
  descriptionAttributn ,
```

```
  [description contraintes portant sur +ieurs attributs]
```

```
);
```

CONTRAINTES PORTANT SUR PLUSIEURS ATTRIBUTS D 'UNE RELATION

UNIQUE (attribut1, attribut2, ...)

Liste des attributs constituant la clé candidate



Les attributs constituant une clé sont forcément non nuls

PRIMARY KEY (attribut1, attribut2, ...)

Liste des attributs constituant la clé primaire

FOREIGN KEY (attribut1, attribut2, ...) **REFERENCES** nomtable (attribut1', attribut2', ...)

Liste des attributs constituant la clé étrangère

Liste des attributs constituant la référence (clé primaire ou candidate)
Si pas de liste , la référence est la clé primaire

CONTRAINTES PORTANT SUR PLUSIEURS ATTRIBUTS D 'UNE RELATION

CONSTRAINT NomContrainte **UNIQUE** (attribut1, attribut2, ...)

CONSTRAINT NomContrainte **PRIMARY KEY** (attribut1, attribut2, ...)

CONSTRAINT NomContrainte **FOREIGN KEY** (attribut1, attribut2, ...)

REFERENCES nomtable(attribut1', attribut2', ...)

EXAMPLE

CHAMBRE (NumChambre, Prix, NbrLits, NbrPers, Confort, Équipement)

CREATE TABLE CHAMBRE

(NumChambre NUMBER (3) PRIMARY KEY,

Prix NUMBER (5, 2) ,

NbrLits NUMBER (1) DEFAULT 1,

NbrPers NUMBER (1) DEFAULT 1,

Confort VARCHAR2 (10),

Équipement BIT(1)

);

EXAMPLE

CLIENT (NumClient, Nom, Prénom, Adresse,..., Tel)

{Nom, Prénom} et Tel clés candidates

CREATE TABLE CLIENT

(NumClient NUMBER (3) PRIMARY KEY,

Nom VARCHAR2 (15) ,

Prenom VARCHAR2 (15),

.....,

Tel NUMBER (10) UNIQUE,

UNIQUE (Nom, Prenom)

);

EXAMPLE

RESERVATION (NumClient, NumChambre, DateArrivée, DateDépart)

CREATE TABLE RESERVATION

(NumClient NUMBER (3),

NumChambre NUMBER (3),

DateArrivee DATE DEFAULT SYSDATE,

DateDepart DATE,

PRIMARY KEY (NumClient, NumChambre),

FOREIGN KEY (NumClient) REFERENCES CLIENT,

CONSTRAINT FK FOREIGN KEY (NumCHAMBRE)

REFERENCES CHAMBRE

);

REMARQUE



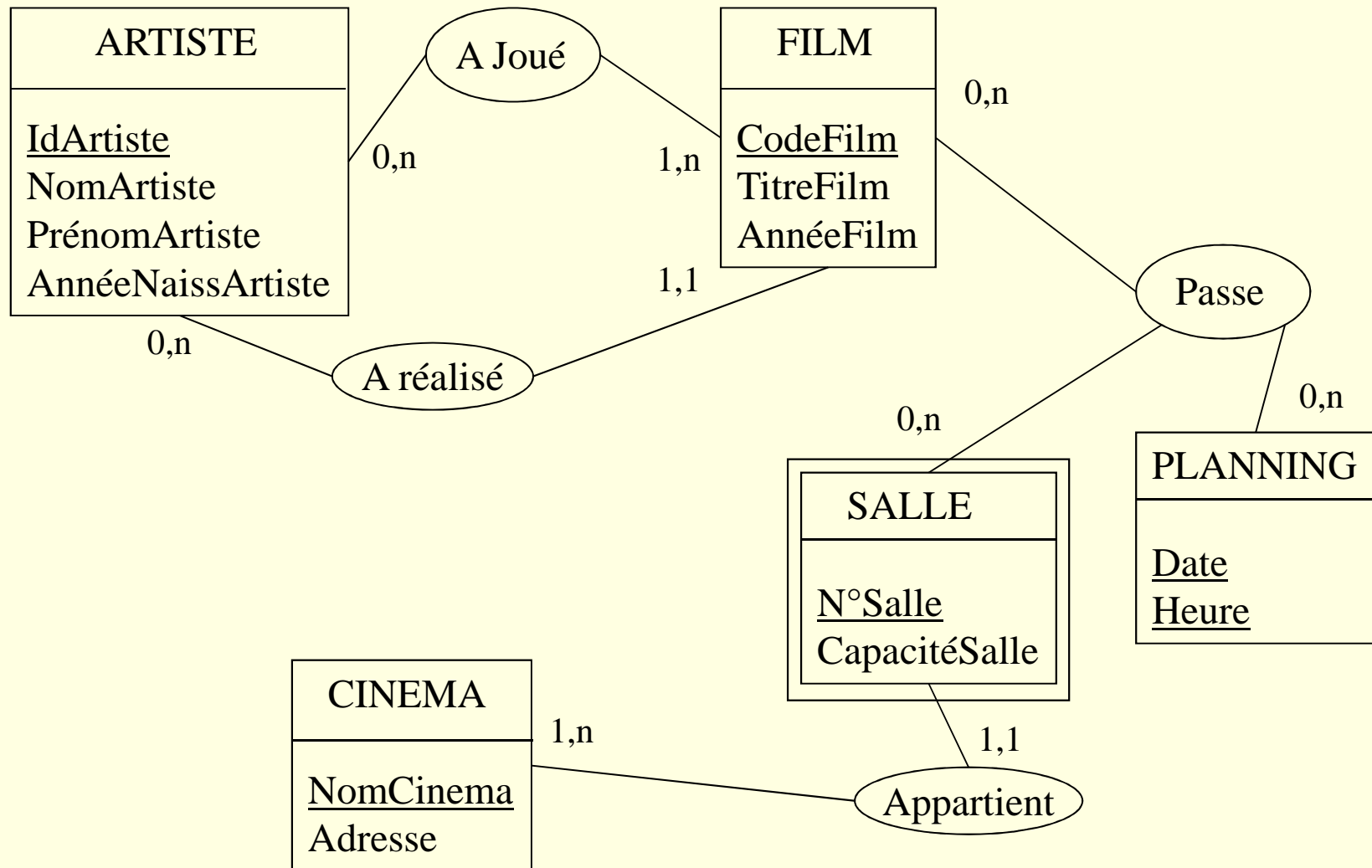
**TOUTE CONTRAINTE EST VERIFIEE A
L'INSERTION OU A LA MODIFICATION
D'UN TUPLE**

GESTION DES CLES ETRANGERES

■ REGLE N° 1

- A l'insertion ou à la modification d'un tuple la contrainte suivante est vérifiée : la valeur de la clé étrangère (si non nulle) est dans l'ensemble des valeurs de la clé à laquelle elle fait référence.

EXEMPLE



EXEMPLE

ARTISTE (IdArtiste, NomArtiste, PrénomArtiste, AnnéeNaissArtiste)

FILM (CodeFilm, TitreFilm, AnnéeFilm, Réalisateur)

SALLE (N°Salle, NomCinéma, CapacitéSalle)

JOUE (IdArtiste, CodeFilm)

PLANNING (CodeFilm, N°Salle, NomCinéma, Date, Heure)

CINEMA (NomCinéma, Adresse)

Un exemple de clé étrangère :
Réalisateur de FILM
fait référence à
ARTISTE.
Elle est obligatoire.

EXEMPLE

```
CREATE TABLE FILM
```

```
(
```

```
    CodeFilm NUMBER (3) PRIMARY KEY,
```

```
    TitreFilm CHAR(20),
```

```
    AnneeFilm  NUMBER (4) ,
```

```
    Realisateur  NUMBER (3) NOT NULL,
```

```
    FOREIGN KEY (Realisateur) REFERENCES ARTISTE
```

```
);
```

A l'ajout d'un nouveau film dans FILM , l'attribut « Realisateur » doit être renseigné (valeur non nulle) et doit être dans la liste des valeurs de « IdArtiste » de ARTISTE. Si l'une des deux conditions n'est pas satisfaite , le SGBD rejette l'insertion même si les autres contraintes sont satisfaites.

EXEMPLE

```
CREATE TABLE FILM
```

```
(
```

```
    CodeFilm NUMBER (3) PRIMARY KEY,
```

```
    TitreFilm CHAR(20),
```

```
    AnneeFilm  NUMBER (4) ,
```

```
    Realisateur  NUMBER (3) NOT NULL,
```

```
    FOREIGN KEY (Realisateur) REFERENCES ARTISTE
```

```
);
```

A la modification d'un film dans FILM , si cette modification concerne l'attribut « Realisateur», la nouvelle valeur doit être non nulle et doit être dans la liste des valeurs de « IdArtiste » de ARTISTE. Si l'une des deux conditions n'est pas satisfaite , le SGBD rejette la modification même si les autres contraintes sont satisfaites.

EXEMPLE

Hypothèse : supposons que l'on a oublié de mentionner la non nullité de Realisateur :

```
CREATE TABLE FILM
(
  CodeFilm NUMBER (3) PRIMARY KEY,
  .....
  Realisateur NUMBER (3) ,
  FOREIGN KEY (Realisateur) REFERENCES ARTISTE
);
```

A l'ajout d'un nouveau film dans FILM, si l'attribut « Realisateur » de ce film est renseigné (valeur non nulle) mais n'est pas dans la liste des valeurs de « IdArtiste » de ARTISTE , le SGBD rejette l'insertion.

A l'ajout d'un nouveau film dans FILM, si l'attribut « Realisateur » de ce film n'est pas renseigné (valeur nulle), le SGBD accepte l'insertion si les contraintes restantes sont vérifiées.

A la modification d'un film dans FILM , si cette modification concerne l'attribut « Realisateur», la nouvelle valeur si non nulle, doit être dans la liste des valeurs de « IdArtiste » de ARTISTE. Si cette condition n'est pas satisfaite , le SGBD rejette la modification même si les autres contraintes sont satisfaites.

EXEMPLE

FILM

IdFilm	Titre	Annee	Realisateur
103	« Cirano »	1992	23
304	« Le dernier métro »	1980	79

ARTISTE

IdArtiste	NomArtiste	PrenomArtiste	AnnéeNaissArtiste
23	-----	-----	-----
79	-----	-----	-----

Nouveaux films à insérer

Insertion possible → 298 « Les oiseaux » 1965 23

Insertion impossible → 500 « mon film » 2000 200

**Insertion possible si « Realisateur »
n'a pas été déclaré non null**

→ 500 « mon film » 2000 null

GESTION DES CLES ETRANGERES

■ REGLE N° 2

- La mise à jour d'une clé primaire à laquelle la clé étrangère fait référence dépend de la politique de gestion choisie pour la contrainte.

■ 3 STRATEGIES POSSIBLES

■ **LA STRATEGIE : NO ACTION (par défaut)**

Lorsqu'une mise à jour de la clé primaire (suppression de la ligne contenant cette clé primaire ou tentative de modification de la valeur de la clé primaire) provoque une violation de contrainte alors le SGBD rejette cette mise à jour.

EXEMPLE

```
CREATE TABLE FILM
(
    CodeFilm NUMBER (3) PRIMARY KEY,
    TitreFilm CHAR(20),
    AnneeFilm NUMBER (4) ,
    Realisateur NUMBER (3) NOT NULL,
FOREIGN KEY (Realisateur) REFERENCES ARTISTE
ON DELETE NO ACTION
ON UPDATE NO ACTION
);
```

A la suppression d'un artiste de ARTISTE ou à la modification de la clé d'un artiste de ARTISTE, si cet artiste est réalisateur d'au moins un film dans FILM, la suppression est rejetée par le SGBD.



ON DELETE NO ACTION est facultatif (idem pour ON UPDATE NO ACTION)

GESTION DES CLES ETRANGERES

■ LA STRATEGIE : CASCADE

- Ce choix force la modification des clés étrangères lorsqu'une tentative de mise à jour de la clé primaire provoque une violation de contrainte.
- En cas de suppression d'une valeur de clé primaire, si l'on souhaite supprimer tous les tuples qui référencent cette valeur de clé, on utilise la clause : **ON DELETE CASCADE**
- En cas de modification d'une valeur de clé primaire, si l'on souhaite répercuter la modification sur tous les tuples qui référencent cette valeur de clé, on utilise la clause : **ON UPDATE CASCADE**



ON UPDATE CASCADE ne fonctionne pas sous ORACLE

EXEMPLE

```
CREATE TABLE FILM
(
    CodeFilm NUMBER (3) PRIMARY KEY,
    TitreFilm CHAR(20),
    AnneeFilm  NUMBER (4) ,
    Realisateur  NUMBER (3) NOT NULL,
FOREIGN KEY (Realisateur) REFERENCES ARTISTE
ON DELETE CASCADE
ON UPDATE CASCADE
);
```

A la suppression d'un artiste de ARTISTE, si cet artiste est réalisateur d'au moins un film dans FILM, le SGBD supprime aussi tous ses films.

A la modification de la clé d'un artiste de ARTISTE, si cet artiste est réalisateur d'au moins un film dans FILM, le SGBD répercute la modification de la valeur de la clé sur tous les films de ce réalisateur.

EXEMPLE

ARTISTE

IdArtiste	----
23	----
79	----
89	----
125	----

FILM

IdFilm	Réalisateur
103	23
304	79

**Artistes à
supprimer
dans ARTISTE**

23	----
----	------

125	----
-----	------

IdArtiste	----
79	----
89	----

IdFilm	Réalisateur
304	79

EXEMPLE

ARTISTE

IdArtiste	----
23	----
79	----
89	----
125	----

FILM

IdFilm	Réalisateur
103	23
304	79

Modification de la clé 23 par 24



IdArtiste	----
24	----
79	----
89	----
125	----

IdFilm	Réalisateur
103	24
304	79

GESTION DES CLES ETRANGERES

■ STRATEGIE : SET NULL

- Ce choix force à nul des clés étrangères lorsqu'une tentative de mise à jour de la clé primaire provoque une violation de contrainte.
- En cas de suppression d'un tuple d'une valeur de clé primaire, si l'on souhaite forcer à nul les clé étrangères de tous les tuples qui référencent ce tuple, on utilise la clause : **ON DELETE SET NULL**
- En cas de modification d'une valeur de clé primaire, si l'on souhaite forcer à null les clé étrangères de tous les tuples qui référencent cette valeur de clé, on utilise la clause : **ON UPDATE SET NULL**



Suppose que la clé étrangère est facultative

EXEMPLE

```
CREATE TABLE FILM
(
    CodeFilm NUMBER (3) PRIMARY KEY,
    TitreFilm CHAR(20),
    AnneeFilm  NUMBER (4) ,
    Realisateur  NUMBER (3),
    FOREIGN KEY (Realisateur) REFERENCES ARTISTE
    ON DELETE SET NULL
    ON UPDATE SET NULL
);
```

A la suppression d'un artiste de ARTISTE ou encore à la modification de la clé d'un artiste de ARTISTE, si cet artiste est réalisateur d'au moins un film dans FILM, le SGBD positionne à null l'attribut Réalisateur pour tous les films réalisés par l'artiste à supprimer ou à modifier.

EXEMPLE

ARTISTE

IdArtiste	-----
23	-----
79	-----
89	-----
125	-----

FILM

IdFilm	Réalisateur
103	23
304	79

Artistes à
supprimer
dans ARTISTE

23

125

IdArtiste	-----
79	-----
89	-----

IdFilm	Réalisateur
103	null
304	79

EXEMPLE

ARTISTE

IdArtiste	----
23	----
79	----
89	----
125	----

FILM

IdFilm	Réalisateur
103	23
304	79

Modification de la clé 23 par 24



IdArtiste	----
24	----
79	----
89	----
125	----

IdFilm	Réalisateur
103	null
304	79



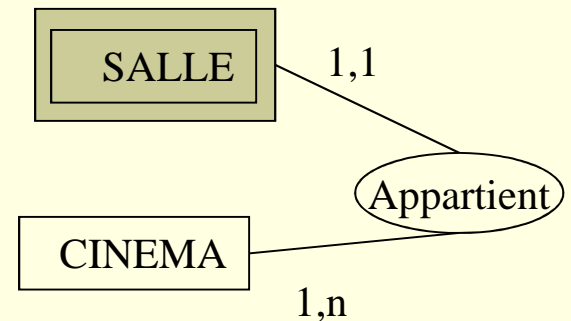
QUELQUES REMARQUES

- Le choix de la stratégie de gestion des contraintes référentielles dépend souvent du réel représenté par la base de données.
 - Exemple : Doit on ou non supprimer tous les films d'un artiste à supprimer?
- La stratégie CASCADE est intéressante pour décrire la sémantique d'un lien de composition

- Prenons l'exemple de SALLE et CINEMA :

SALLE (N°Salle, NomCinéma, CapacitéSalle)

CINEMA (NomCinéma, Adresse)



- Il est intéressant, pour cet exemple de choisir la stratégie ON DELETE CASCADE pour la clé étrangère NomCinéma de SALLE car une salle ne peut exister sans son cinéma.



QUELQUES REMARQUES

- On ne peut pas appliquer la stratégie SET NULL pour des clés étrangères constituant la clé primaire d'une relation
 - Prenons l'exemple de SALLE et CINEMA :

SALLE (N°Salle, NomCinéma, CapacitéSalle)

CINEMA (NomCinéma, Adresse)

- Cette stratégie n'est pas applicable pour NomCinéma de SALLE car cela violerait la contrainte de non nullité des constituants de la clé primaire.

MODIFICATION ET SUPPRESSION DE TABLES

ALTER TABLE nomtable

[ADD attribut],
[ADD contrainte],
[MODIFY attribut],
[MODIFY CONSTRAINT NomContrainte],
[DROP COLUMN attribut],
[DROP NomContrainte],
[DROP Contrainte];

DROP TABLE nomtable;



- Attention aux modifications de tables déjà peuplées
- La clause DROP supprime le schéma donc les tuples aussi.

EXEMPLE DE MODIFICATION DE TABLES

- ALTER TABLE RESERVATION

MODIFY CONSTRAINT C1-REFCLT FOREIGN KEY (NumCLIENT)
REFERENCES CLIENTPOTENTIEL;

- ALTER TABLE CLIENT

DROP UNIQUE(Nom, Prénom),
ADD ChiffreAffaire Number (3) NOT NULL;

} Pas possible si table déjà peuplée à cause de
la clause « NOT NULL »

- ALTER TABLE CLIENT

DROP UNIQUE(Nom, Prénom),
ADD ChiffreAffaire Number (3) DEFAULT 0;

- ALTER TABLE CLIENT

DROP UNIQUE(Nom, Prénom),
ADD ChiffreAffaire Number (3);