



École nationale supérieure d'informatique et de mathématiques appliquées

# Langage SQL (LDD)

**Ensimag 2A**  
*Equipe pédagogique BD*



# Création de relation

## Types de base :

number[(x)], integer, int : entiers

decimal(x,y), float : réels

char[(n)], varchar[(n)] : chaînes de caractères

date : date

```
CREATE TABLE <table> ((<colonne> <type>
                        [DEFAULT <expression>] <contrainte_c>+)*
                        [<contrainte_t>*]);
```

```
<contrainte_c> := [CONSTRAINT <nom_c>]
  (
    [NOT] NULL
    | UNIQUE
    | PRIMARY KEY
    | REFERENCES <table>[ (<colonne>)] [ON DELETE CASCADE]
    | CHECK (<condition>)
  )
```

```
<contrainte_t> := [CONSTRAINT <nom_c>]
  (
    (UNIQUE | PRIMARY KEY) (<colonne>*)
    | FOREIGN KEY (<colonne>*)
      REFERENCES <table>[ (<colonne>*)] [ON DELETE CASCADE]
    | CHECK (<condition>)
  )
```

# Exemple

```
CREATE TABLE Eleves (  
    prenom varchar(30) NOT NULL,  
    nom varchar (30) NOT NULL,  
    email varchar (30),  
    option varchar (5),  
    PRIMARY KEY (prenom, nom)  
);  
  
CREATE TABLE Notes (  
    cours varchar (30) NOT NULL,  
    prenom varchar (30) NOT NULL,  
    nom varchar (30) NOT NULL,  
    note integer,  
    PRIMARY KEY(cours, prenom, nom),  
    FOREIGN KEY (prenom, nom) REFERENCES Eleves(prenom, nom)  
);
```

# Autre exemple

**BUVEURS** (NB, Nom, Prénom, Adresse)

**VINS** (NV, Cru, Millesime, Degre, Qualite)

**A\_GOUTE** (NV,NB, DateA, Lieu, Quantite)

**PRODUCTEURS** (NP, Nom, Prenom, Region)

**RECOLTES** (NP, NV, Quantite)

# Contraintes (1)

## **CREATE TABLE BUVEURS**

**(nb INTEGER CONSTRAINT Knb PRIMARY KEY,  
nom VARCHAR (20),  
prenom VARCHAR (20), adresse VARCHAR(40))**

CREATE TABLE VINS (nv INTEGER CONSTRAINT Knv PRIMARY KEY,  
cru VARCHAR(20),  
millesime NUMBER(4) NOT NULL CHECK(millesime > 1900),  
degre FLOAT, qualite CHAR(10) DEFAULT ' inconnue' )

CREATE TABLE A\_GOUTE (nv INTEGER REFERENCES vins,  
nb INTEGER REFERENCES buveurs,  
dateA DATE, lieu VARCHAR(20),  
quantite INTEGER,  
CONSTRAINT abus\_pk PRIMARY KEY(nv, nb, dateA) )

# Contraintes (2)

CREATE TABLE PRODUCTEURS

(np INTEGER CONSTRAINT Knp PRIMARY KEY (np),  
nom CHAR (20) CONSTRAINT NOM-UNIQUE UNIQUE,  
prenom CHAR(20),  
region CHAR CONSTRAINT Ci CHECK (region IN ('E','O','S','N')) )

CREATE TABLE RECOLTES

(np INTEGER NOT NULL,  
nv INTEGER NOT NULL,  
quantite INTEGER,

CONSTRAINT Krecolte PRIMARY KEY (np,nv),  
CONSTRAINT Rnp FOREIGN KEY (np) REFERENCES PRODUCTEURS (np),  
CONSTRAINT Rnv FOREIGN KEY (nv) REFERENCES VINS(nv) ON DELETE CASCADE )

// La suppression d'un n-uplet de *vins* entraîne la suppression des n-uplets de *recoltes* correspondant.

# Suppression de relations

**DROP TABLE** <table> [**CASCADE**] ;

Supprime les données de la relation et son schéma

Si CASCADE :

supprime les relations qui référencent <table>

Exemple :

DROP TABLE Notes;

DROP TABLE Eleves;

# Modification de relation

```
ALTER TABLE <table> (  
    ADD (  
        (<colonne> <type> [DEFAULT <expression>] <contrainte_c>)*  
        | <contrainte_t>*  
    )  
    | MODIFY (  
        (<colonne> <type> [DEFAULT <expression>] <contrainte_c>)*  
        | <contrainte_t>*  
    )  
    | DROP (  
        COLUMN <colonne>  
        | PRIMARY KEY | UNIQUE(<colonne>*)  
        | CONSTRAINT <nom_c>  
    )  
    [CASCADE] ;
```

- **On ne peut que :**
  - ◆ Supprimer un attribut « vide »
  - ◆ Ajouter un attribut acceptant des valeurs nulles
  - ◆ Agrandir la taille d'une colonne
  - ◆ Modifier un attribut Null → Not Null



# Exemples

- ◆ ALTER TABLE **BUVEURS** ADD (**typebuveur** CHAR (10))
- ◆ ALTER TABLE **VINS** MODIFY (**degre** NUMBER (4, 2))
- ◆ ALTER TABLE **A\_GOUTE** MODIFY (**quantite** NUMBER NOT NULL)

*que si tous les  $n$ -uplets contiennent des valeurs non nulles*



École nationale supérieure d'informatique et de mathématiques appliquées

---

# Manipulation de schémas



# Définition de vues (schémas externes)

```
CREATE VIEW <vue> AS <requête> ;
```

# Gestion des utilisateurs

```
CREATE USER <login> IDENTIFIED BY <passwd>;
```

```
DROP USER <login> [CASCADE];
```

```
ALTER USER <login> IDENTIFIED BY <passwd>;
```

# Gestion de droits d' accès

**GRANT** <privilège> [**ON** <relation | vue>] **TO** <utilisateur>;

**REVOKE** <privilège> [**ON** <relation | vue>] **FROM** <utilisateur>;

## Privilège :

**Select** : droit de lire la relation/vue

**Update** : droit de lire et modifier les n-uplets

**Insert** : droit d' ajouter de nouveaux n-uplets

**Delete** : droit de supprimer des n-uplets

# Gestion de transactions

**BEGIN ;** : début de transaction (souvent inutile car implicite).

**COMMIT [WORK] ;** : validation des opérations effectuées depuis le début de la transaction.

**ROLLBACK [WORK] ;** : annulation des opérations effectuées depuis le début de la transaction.

**SAVEPOINT** <savepoint name>;

**ROLLBACK TO** <savepoint name>;