

# Structured Query Language

## SQL DDL

L2A

Semestre 4

Mehdi Benzine

# Langage de définition de données

Le langage de définition de données comprend les instructions suivantes:

- L'instruction CREATE, pour créer de nouveaux objets du schéma (table, type, domain, index, trigger ...).
- L'instruction ALTER pour modifier des objets existants.
- L'instruction DROP pour supprimer des objets existants.



# Instruction CREATE

Permet de créer de nouveaux objets du schéma (tables, index ...).

A ne pas confondre avec l'instruction INSERT.

Pour créer une table:

```
CREATE TABLE nom_table (colonne1 type1 [contrainte d'attribut1], colonne2 type2  
[contrainte d'attribut2] ..., [[CONSTRAINT nom_contrainte1] contrainte1 de table]  
[, [CONSTRAINT nom_contrainte2] contrainte2 de table])
```

Exemple:

```
CREATE TABLE Projet(Num_Projet INT, Description CHAR (200), Date_Début DATE,  
Date_Fin DATE, Budget FIXED(8,2))
```

FIXED: Réel à virgule fixe FIXED(a, b): nombre à a chiffres au maximum dont b chiffres après la virgule.

# Contraintes d'intégrité

- Contrainte de clé primaire (PRIMARY KEY)  
La clé primaire de chaque table doit être spécifiée explicitement
- Contrainte de non vacuité (NOT NULL)  
Les attributs devant avoir une valeur doivent être spécifiés
- Contrainte d'unicité (UNIQUE)  
Les attributs devant être uniques doivent également être spécifiés.
- Contrainte de domaine (CHECK, DEFAULT)  
La valeur d'un attribut doit être comprise dans un domaine de valeurs.
- Contrainte référentielle (FOREIGN KEY)  
La clé étrangère d'une relation doit obligatoirement référencer une valeur existante.

La contrainte PRIMARY KEY regroupe les contraintes UNIQUE et NOT NULL.

Quand une contrainte doit s'appliquer à plus d'un attribut, elle doit être exprimée comme contrainte de table.



# Instruction CREATE

```
CREATE TABLE Projet(Num_Projet INT PRIMARY KEY,  
                    Description CHAR (200) NOT NULL,  
                    Date_Début DATE,  
                    Date_Fin DATE,  
                    Budget FIXED(8,2) DEFAULT 20000.00
```

```
)  
CREATE TABLE Employé(Num_Employé INT PRIMARY KEY,  
                    Nom CHAR(30) NOT NULL,  
                    Prénom CHAR(30),  
                    Date_Naissance DATE CHECK (Date_Naissance <= '11/04/1996'),  
                    FONCTION CHAR(15),  
                    Est_Cadre CHAR(1) DEFAULT 'o'
```

) Bien que présent sur certains SGBDR, le type booléen n'est pas spécifié par la norme SQL2

# Instruction CREATE

```
CREATE TABLE Affectation(  
  Num_Employé INT REFERENCES Employé(Num_Employé),  
  Num_Projet INT REFERENCES Projet(Num_Projet),  
  Debut_Affect DATE NOT NULL,  
  Fin_Affect DATE,  
  Supérieur INT REFERENCES Employé(Num_Employé),  
  PRIMARY KEY (Num_Employé, Num_Projet)  
)
```



# Instruction CREATE

```
CREATE TABLE Affectation(  
  Num_Employé INT REFERENCES Employé(Num_Employé),  
  Num_Projet INT REFERENCES Projet(Num_Projet),  
  Debut_Affect DATE NOT NULL,  
  Fin_Affect DATE,  
  Supérieur INT REFERENCES Employé(Num_Employé),  
  CONSTRAINT cle_affectation PRIMARY KEY (Num_Employé,  
  Num_Projet)  
)
```

Toutes les contraintes sur attributs peuvent être définies comme contraintes sur tables.

# Contraintes sur suppression de tuples

Il est en principe interdit de supprimer un tuple référencé par un autre tuple dans le schéma relationnel.

Avant de supprimer le tuple référencé, je dois d'abord supprimer tous les tuple le référençant.

Exemple: Avant de supprimer l'employé 1023, il faut d'abord supprimer toutes ses affectations.



**DELETE FROM Employé WHERE Num\_Employé = 1023**

Suppression impossible car le tuple employé est référencé par 2 tuples affectations

Num_Employé	Nom	Prénom	Date_Naissance	Fonction	Est_Cadre
1001	Belaid	Toufik	12/05/1965	Concepteur	1
1009	Touati	Rachid	13/09/1941	Chef de projet	1
1023	Kadri	Amine	23/11/1970	Développeur	1
1053	Djabi	Fatiha	04/06/1980	Analyste	0
1026	Bouras	Kamel	19/04/1968	Administrateur	1
1005	Djabi	Fatiha	22/08/1976	Développeur	0

Num_Employé	Num_Projet	Début_Affect	Fin_Affect	Supérieur
1009	122	07/03/2011	13/11/2011	NULL
1001	122	08/03/2011	28/06/2011	1009
1023	122	15/06/2011	04/10/2011	1009
1009	103	12/09/2010	01/11/2010	NULL
1001	208	15/06/2011	12/10/2011	1009
1009	208	15/06/2011	06/03/2012	NULL
1023	208	01/09/2011	17/12/2011	1009
1009	133	06/11/2011	19/02/2012	NULL
1053	208	01/09/2011	06/03/2012	1026
1026	208	19/08/2011	06/03/2012	1009

# Suppression en cascade

La clause ON DELETE CASCADE permet de supprimer en cascade tous les tuples référençant un tuple supprimé.

```
CREATE TABLE Affectation(  
  Num_Employé INT REFERENCES Employé(Num_Employé) ON DELETE  
  CASCADE, Num_Projet INT FOREIGN KEY REFERENCES  
  Projet(Num_Projet), Debut_Affect DATE NOT NULL,  
  Fin_Affect DATE,  
  Supérieur INT REFERENCES Employé(Num_Employé), PRIMARY KEY  
  (Num_Employé, Num_Projet))
```



**DELETE FROM Employé WHERE Num\_Employé = 1023**

Suppression du tuple employé et de tous les tuples qui le référencent dans la table affectation.

Num_Employé	Nom	Prénom	Date_Naissance	Fonction	Est_Cadre
1001	Belaid	Toufik	12/05/1965	Concepteur	1
1009	Touati	Rachid	13/09/1941	Chef de projet	1
1023	Kadri	Amine	23/11/1970	Développeur	1
1053	Djabi	Fatiha	04/06/1980	Analyste	0
1026	Bouras	Kamel	19/04/1968	Administrateur	1
1005	Djabi	Fatiha	22/08/1976	Développeur	0

Num_Employé	Num_Projet	Début_Affect	Fin_Affect	Supérieur
1009	122	07/03/2011	13/11/2011	NULL
1001	122	08/03/2011	28/06/2011	1009
1023	122	15/06/2011	04/10/2011	1009
1009	103	12/09/2010	01/11/2010	NULL
1001	208	15/06/2011	12/10/2011	1009
1009	208	15/06/2011	06/03/2012	NULL
1023	208	01/09/2011	17/12/2011	1009
1009	133	06/11/2011	19/02/2012	NULL
1053	208	01/09/2011	06/03/2012	1026
1026	208	19/08/2011	06/03/2012	1009

# Suppression et mise à jour

En plus de la clause ON DELETE CASCADE, la norme SQL2 prévoit les clauses suivantes:

- ON DELETE SET NULL
- ON DELETE SET DEFAULT
- ON UPDATE CASCADE
- ON UPDATE SET NULL
- ON UPDATE SET DEFAULT



# Types de données

La norme SQL2 prévoit un grand nombre de types de données:

- Numériques
- Caractères
- Temporels
- Binaires

Les types de données Oracle ne sont pas conformes à la norme SQL2.

# Les types numériques

## SQL- 2

- Entiers : TINYINT (sur 1 octet, de 0 à 225
- Décimaux avec un nombre fixe de décimales : NUMERIC, DECIMAL
- Numériques non exactes à virgule flottante :

REAL (le nombre de chiffres significatifs varie )

DOUBLE PRECISION ( avec au moins 15 chiffres significatifs)

*La définition du nombre de chiffres significatifs varie selon le SGBD*

## Oracle

- NUMBER : est un nombre à virgule flottante. La taille d'un attribut NUMBER peut aller de 1 à 22 octets. On peut préciser le nombre maximum de chiffres et de décimales

NUMBER

NUMBER (taille\_max)

NUMBER (taille\_max, décimales)

INTEGER = NUMBER(\*, 0)

Si aucun paramètre n'est pas spécifié, la capacité maximale est prise par défaut. La valeur absolue du nombre doit être inférieur à  $2^{126}$

**Exemple : NUMBER(6, 2) 6 chiffres dont 2 après la virgule**



# Les types chaînes de caractères

## SQL2 - Oracle

- Sont entourées par des apostrophes. Si la chaîne contient une apostrophe, celle-ci devra être doublée.

- Il existe deux types :

- pour les chaînes de longueur constante (inférieur à 2000 caractères, pour ORACLE) CHAR (longueur) :

- pour les chaînes de longueur variable (Tous les SGBD ont une longueur maximale pour ces chaînes, 4000 sous ORACLE)

- VARCHAR (longueur) ou

- VARCHAR2(longueur) sous ORACLE

VARCHAR est maintenant obsolète

**Longueur est le nombre maximale de caractères qu'il sera possible de stocker dans le champ**

# Les types temporels

## SQL<sub>2</sub>

- **DATE** : réserve 2 chiffres pour le mois et le jour et 4 pour l'année
- **TIME** : pour les heures, minutes et secondes
- **TIMESTAMP** : indique un moment précis par une date avec heures, minutes et secondes (6 chiffres après la virgule)

## Oracle

**DATE** : chaîne de caractères entre apostrophes. Le format dépend des options que l'administrateur a choisies au moment de la création de la base.

Un type DATE inclut un temps en heures, minutes et secondes.



# Les types binaires

- **SQL2 n'a pas normalisé ce type de données**
- **Ce type permet d'enregistrer des données telles que les images et les sons de très grand taille avec divers formats**
- **Les différents SGBD fournissent un type pour ces données. Les noms varient :**
  - **LONG RAW pour Oracle**
  - **IMAGE pour Sybase**
  - **BYTE pour Informix**

# CREATE sous Oracle

```
CREATE TABLE Projet(Num_Projet INTEGER PRIMARY KEY,  
                    Description VARCHAR2 (200) NOT NULL,  
                    Date_Début DATE,  
                    Date_Fin DATE,  
                    Budget NUMBER(8,2) DEFAULT 20000.00
```

```
)  
CREATE TABLE Employé(Num_Employé INTEGER PRIMARY KEY,  
                    Nom VARCHAR2(30) NOT NULL,  
                    Prénom VARCHAR2(30),  
                    Date_Naissance DATE CHECK (Date_Naissance <= '11/04/1996'),  
                    FONCTION VARCHAR2(15),  
                    Est_Cadre CHAR(1) DEFAULT 'o'
```

) Bien que présent sur certains SGBDR, le type booléen n'est pas spécifié par la norme SQL2



# CREATE sous Oracle

```
CREATE TABLE Affectation(  
  Num_Employé INTEGER REFERENCES Employé(Num_Employé),  
  Num_Projet INTEGER REFERENCES Projet(Num_Projet),  
  Debut_Affect DATE NOT NULL,  
  Fin_Affect DATE,  
  Supérieur INTEGER REFERENCES Employé(Num_Employé),  
  PRIMARY KEY (Num_Employé, Num_Projet)  
)
```

# Instruction ALTER

L'instruction ALTER permet de modifier le schéma d'une table.

Modifier le schéma d'une table consiste à effectuer un des changements suivants:

- Ajouter une ou plusieurs colonnes (ADD (col1 type1[, col2 type2 ...]))
- Modifier une ou plusieurs colonnes (MODIFY (col1 type 1[, col2 type2...]))
- Supprimer une ou plusieurs colonnes (DROP (col1 [,col2 ...]))
- Ajouter une contrainte (ADD CONSTRAINT ...)
- Supprimer une contrainte (DROP CONSTRAINT nom\_contrainte)
- ...



# Instruction ALTER

Exemples:

Ajouter une colonne Salaire à la table Employé

```
ALTER TABLE Employé ADD Salaire NUMBER(7,2)
```

Modifier le degré de précision de la colonne Budget et augmenter la taille maximale de la colonne Description de projet

```
ALTER TABLE Projet MODIFY (Budget NUMBER(10,2),  
Description VARCHAR2(255))
```

Supprimer la colonne Date\_Fin

```
ALTER TABLE Projet DROP Date_Fin
```

# Instruction ALTER

Ajouter une contrainte sur le Budget

```
ALTER TABLE Projet ADD CONSTRAINT Budget_Min CHECK(Budget >= 20000)
```

On ne peut ajouter une contrainte que si toutes les données déjà existantes vérifient cette contrainte.

Num_Projet	Description	Date_Début	Date_Fin	Budget
103	Mise en place d'un réseau intranet	12/09/2010	05/12/2010	39000.00
122	Développement d'une application de gestion	07/01/2011	13/05/2011	86000.00
133	Réalisation d'un CD-ROM interactif de formation	04/11/2011	02/07/2012	15000.00
208	Développement d'un site internet	15/06/2011	06/03/2012	50000.00



# Instruction DROP

L'instruction DROP permet de supprimer un objet du schéma relationnel (table, contrainte, index...).

Supprime la table employé et tous les tuples qu'elle contient.  
A ne pas confondre avec l'instruction DELETE.

Pour supprimer une table:  
`DROP TABLE nom_table`

Exemple:  
`DROP TABLE Employé`

# Instruction DROP

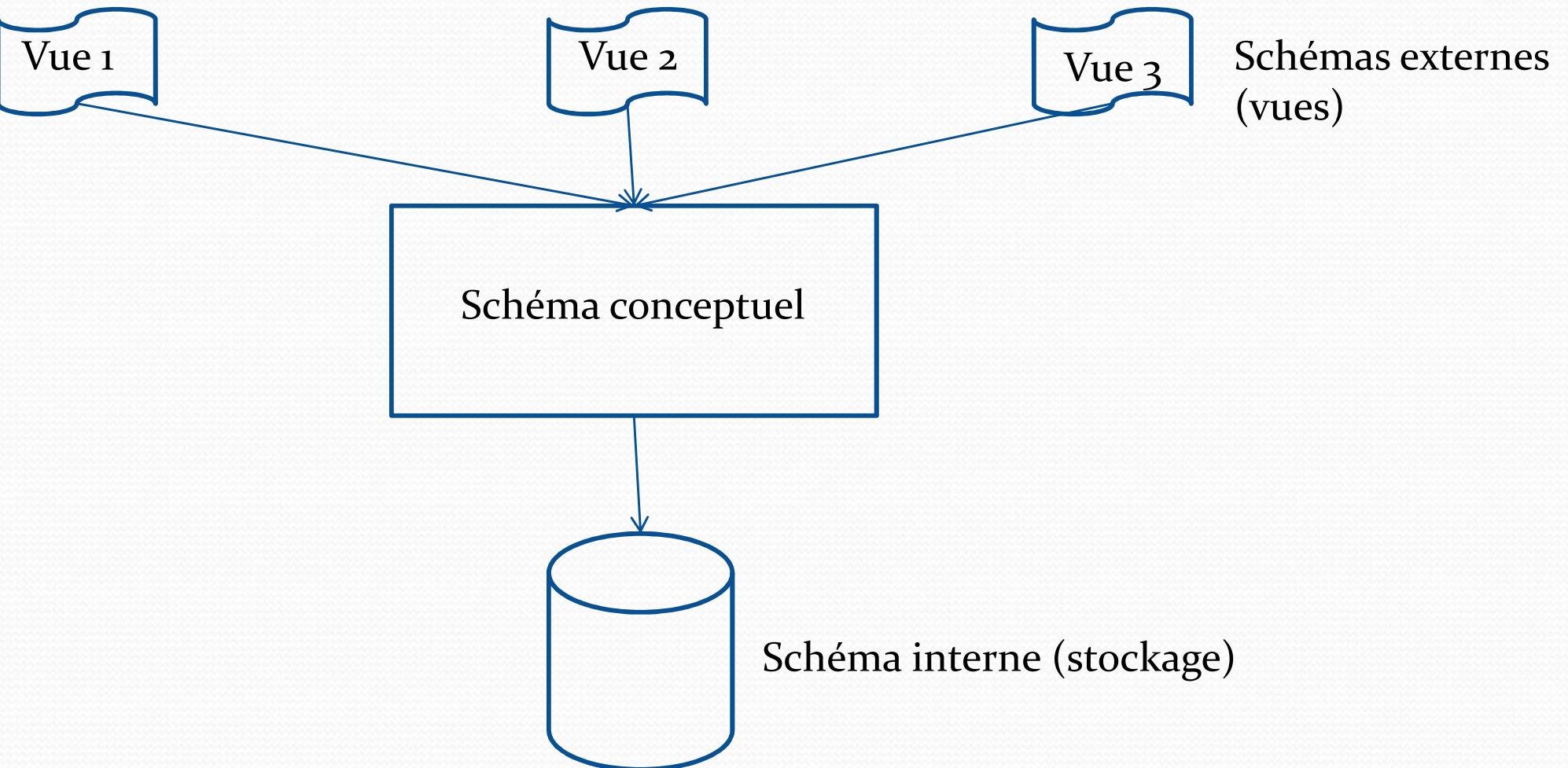
Il est interdit de supprimer une table référencée par une autre table.

Pour supprimer les tables Employé, Projet et Affectation, il faut d'abord supprimer la table Affectation et ensuite les tables Employé, Projet.

```
DROP TABLE Affectation;  
DROP TABLE Employé,  
DROP TABLE Projet;
```



# Architecture à 3 niveaux



# Vues

Une vue est une **relation virtuelle** qui peut être manipulée exactement comme une relation "classique" dans une requête SQL.

Les vues permettent:

- D'assurer le principe d'indépendance logique vis-à-vis du schéma conceptuel.
- De garantir la confidentialité des données.



# Vues

```
CREATE [OR REPLACE] VIEW vue (col1, col2)
AS SELECT att1, att2 FROM table WHERE att3 = 'valeur'
```

Une vue des employés chefs de projet:

```
CREATE VIEW Chef_Projet (Num_Employé, Nom, Prénom, Date_Naissance,
Est_Cadre)
AS SELECT Num_Employé, Nom, Prénom, Date_Naissance, Est_Cadre
FROM Employé
WHERE Fonction = 'Chef de projet'
```

Une vue du nombre d'affectations par employé:

```
CREATE VIEW Nombre_Affectations (Num_Employé, Nb_Affect)
AS SELECT Num_Employé, COUNT(*)
FROM Affectation
GROUP BY Num_Employé
```

# Vues

Afficher les noms, prénoms et nombre d'affectations de chaque employé:

```
SELECT Nom, Prénom, Nb_Affect  
FROM Employé e, Nombre_Affectations na  
WHERE e.Num_Employé = na.Num_Employé
```

Afficher les affectations des chefs de projet

```
SELECT *  
FROM Chef_Projet cp, Affectation a  
WHERE cp.Num_Employé = a.Num_Employé
```



# Mise à jour de vues

Il est parfois possible de mettre à jour les données d'une table à travers une vue.

```
INSERT INTO Chef_Projet VALUES (1099, 'Smati', 'Kamel', '12/12/1980', 'o')
```

Le nouveau tuple apparaîtra dans la table Employé mais pas dans le résultat de la vue Chef\_Projet.

Num_Employé	Nom	Prénom	Date_Naissance	Fonction	Est_Cadre
1001	Belaid	Toufik	12/05/1965	Concepteur	1
1009	Touati	Rachid	13/09/1941	Chef de projet	1
1023	Kadri	Amine	23/11/1970	Développeur	1
1053	Djabi	Fatiha	04/06/1980	Analyste	0
1026	Bouras	Kamel	19/04/1968	Administrateur	1
1005	Djabi	Fatiha	22/08/1976	Développeur	0
1099	Smati	Kamel	12/12/1980	NULL	0

Si la colonne Fonction devait obligatoirement être non nulle, cette insertion aurait été impossible.

# Mise à jour de vues

```
CREATE TABLE Employé(Num_Employé INTEGER PRIMARY KEY,  
    Nom VARCHAR2(30) NOT NULL,  
    Prénom VARCHAR2(30),  
    Date_Naissance DATE CHECK (Date_Naissance <= '11/04/1996'),  
    FONCTION VARCHAR2(15),  
    Est_Cadre CHAR(1) DEFAULT 'o'  
)
```

```
CREATE TABLE Employé(Num_Employé INTEGER PRIMARY KEY,  
    Nom VARCHAR2(30) NOT NULL,  
    Prénom VARCHAR2(30),  
    Date_Naissance DATE CHECK (Date_Naissance <= '11/04/1996'),  
    FONCTION VARCHAR2(15) NOT NULL,  
    Est_Cadre CHAR(1) DEFAULT 'o'  
)
```



# Mise à jour de vues

INSERT INTO Chef\_Projet VALUES (1099, 'Smati', 'Kamel', '12/12/1980', 'o')

Insertion impossible!!! La fonction d'un employé doit être connue (NOT NULL).

Num_Employé	Nom	Prénom	Date_Naissance	Fonction	Est_Cadre
1001	Belaid	Toufik	12/05/1965	Concepteur	1
1009	Touati	Rachid	13/09/1941	Chef de projet	1
1023	Kadri	Amine	23/11/1970	Développeur	1
1053	Djabi	Fatiha	04/06/1980	Analyste	0
1026	Bouras	Kamel	19/04/1968	Administrateur	1
1005	Djabi	Fatiha	22/08/1976	Développeur	0