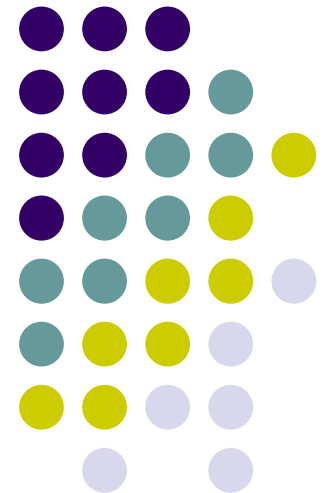


Le Langage de Définition des Données

- Ordre CREATE
 - Ordre DROP
 - Ordre ALTER





Présentation

- **CREATE**
 - Création du **schéma physique** des **tables**
- **ALTER**
 - Modification du **schéma physique** d'une table
- **DROP**
 - Suppression **physique** et **logique** de la table



CREATE : syntaxe

- Création de la structure des tables

```
CREATE TABLE <nom_table>
  (<colonne1>    type1[(longueur1)],
   <colonne2>    type2[(longueur2)],
   ... ,
   [contrainte_table1], [contrainte_table2], ...
);
```



Les types de données

| | |
|---------------------------------------|--|
| VARCHAR2 (<i>size</i>) | Données caractères de longueur variable |
| CHAR (<i>size</i>) | Données caractères de longueur fixe |
| NUMBER (<i>p</i> , <i>s</i>) | Numérique de longueur variable |
| DATE | Valeurs de date et d'heure |
| LONG | Données caractères de longueur variable (2Go) |
| CLOB | Données caractères (4 Go) |
| RAW | Binaire |
| BLOB | Binaire, jusqu' à 4 Go |
| BFILE | Binaire, stocké dans un fichier externe (4 Go) |



Intégrité

- Les données de la base respectent un ensemble de contraintes prédéfinies. Ces conditions doivent toujours être respectées.
- Il existe trois types d'intégrité :
 - des règles générales (type de la donnée, largeur maximum)
 - l'intégrité au niveau attributs (`primary key`, `check`, ...)
 - l'intégrité référentielle (contraintes d'inclusion)



Contraintes

- Il existe 2 façons d'exprimer des contraintes :
 - Sous forme de contrainte de colonne
 - Sous forme de contrainte de table
- Cette année, on ne vous parle que des contraintes de table !
 - ... et des contraintes les plus 'usuelles'



Contraintes

- Contraintes de table
 - Définies après la création de toutes les colonnes
 - **Peuvent porter sur une ou plusieurs colonnes**

```
[CONSTRAINT <nom_contrainte>]  
      type_contrainte (<colonne1> [<colonne2>,...])
```



Contraintes

Il est utile de donner un nom aux contraintes pour :

- documenter le message d'erreur s'il y a transgression de la contrainte d'intégrité
- documenter le Dictionnaire des Données
- faciliter l'activation, l'inhibition ou la suppression de la contrainte.



Types de contraintes (1)

- **CHECK (condition)**

```
[CONSTRAINT <nom_contrainte>] CHECK (condition)
```

- S'applique à toutes les lignes de la table
- Condition :
 - expression conditionnelle valide évaluant un résultat booléen



Types de contraintes (2)

- **PRIMARY KEY**

```
[CONSTRAINT <nom_contrainte>]  
    PRIMARY KEY (<nom_colonne1>, [<nom_colonne2>...])
```

- NOT NULL + UNIQUE
- Clé primaire d'une table
- Peut concerner plusieurs colonnes
- Peut être référencée par une clé étrangère
- **Il est obligatoire de déclarer une et une seule contrainte PRIMARY KEY sur chaque table**



Types de contraintes (3)

● FOREIGN KEY

```
[CONSTRAINT <nom_contrainte>]  
FOREIGN KEY <nom_colonne>  
REFERENCES <nom_table> (<nom_colonne>)
```

- Relation de clé étrangère avec une ou plusieurs colonnes déclarées comme clé primaires ou UNIQUE.
- Valeurs des colonnes
 - Soit nulles
 - Soit valeurs existant dans une table référencée.
- La contrainte peut également référencer une colonne dans la même table.



Types de contraintes (4)

● Clause ON DELETE CASCADE

- Permet de supprimer automatiquement les tuples d'une table Fille lorsque les références n'existent plus dans la table Mère
 - A indiquer à la création de la table fille lors de la déclaration de clé étrangère.
- Attention à son utilisation !

| IdService | NomService | Etage |
|--------------|-------------------------|--------------|
| 2 | Comptabilité | 2 |
| 3 | Informatique | 3 |

Services

Employés

| IdEmploye | NomEmploye | Salaire | DateEmbauche | #IdService |
|---------------|-------------------|------------------|-----------------------|--------------|
| 10 | Martin | 30000 | 01/02/1999 | 2 |
| 12 | Cartaux | 25500 | 30/03/2000 | 3 |
| 15 | Jean | 45000 | 01/01/2001 | 1 |
| 23 | Parot | 23000 | 23/06/2001 | 3 |



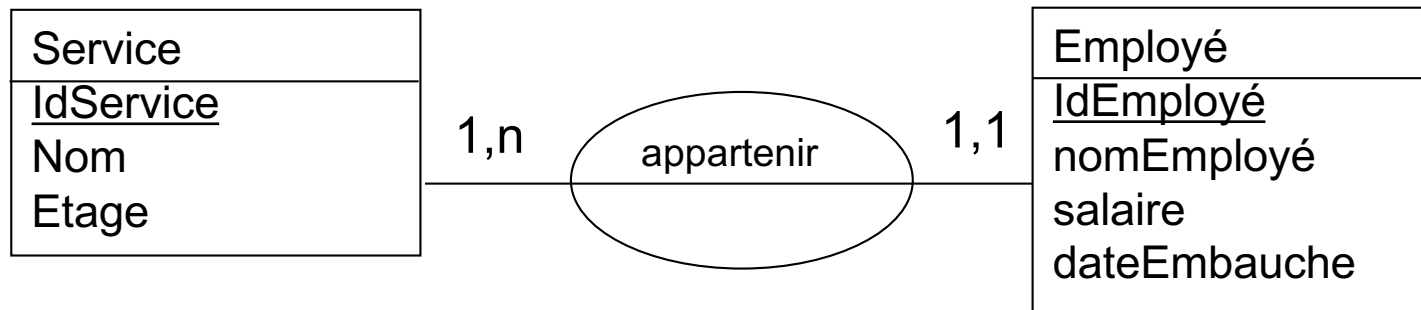
Contraintes de colonnes et de tables



- Conventions pour le nommage des contraintes (recommandation)
 - Clé primaire : `pk_nomTable`
 - Clé étrangère :
`fk_nomTableSource_nomTableCible`
 - Check: `ck_nomCol`



Exemple 1



- Service (IdService, nom, etage)
- Employé (IdEmployé, nomEmployé, salaire, dateEmbauche, #id_Service)



Exemple 1

--Table Mère

```
CREATE TABLE Service
  (IdService NUMBER(3) ,
   nomService VARCHAR2(30) ,
   Etage NUMBER(1) ,
   CONSTRAINT ck_nomService
               CHECK (nomService IS NOT NULL) ,
   CONSTRAINT ck_etage
               CHECK (Etage BETWEEN 1 AND 5) ,
   CONSTRAINT pk_service PRIMARY KEY(IdService)
) ;
```



Exemple 1

--Table Fille

```
CREATE TABLE Employe
  (IdEmploye NUMBER(5),
   nomEmploye VARCHAR(30),
   salaire NUMBER(5),
   dateEmbauche DATE,
   IdService NUMBER(3),
   CONSTRAINT pk_employe PRIMARY KEY(IdEmploye),
   CONSTRAINT ck_nomEmploye CHECK (nomEmploye IS NOT NULL),
   CONSTRAINT ck_IdService CHECK (idService IS NOT NULL),
   CONSTRAINT fk_emp_ser FOREIGN KEY (idService)
                        REFERENCES Service(IdService)
                        on delete cascade
  );
```

Clé étrangère + clause on delete cascade

Exemple 1

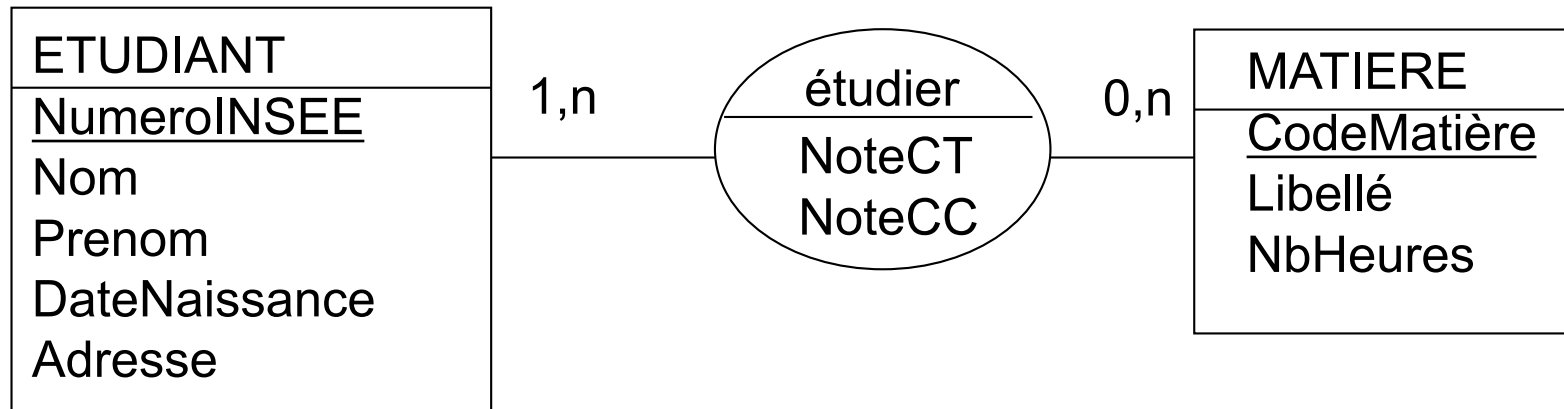
Quelques remarques



- Attention à l'ordre de création des tables
- La cardinalité 1,1 du côté d'Employé (CIF) se traduit par une contrainte `CHECK (... IS NOT NULL)` lors de la déclaration de l'attribut qui sera clé étrangère
- L'attribut clé étrangère est forcément du même type que l'attribut clé primaire référencé



Exemple 2



- Etudiant (NumeroINSEE, nom, prenom, dateNaissance, adresse)
- Matiere (CodeMatiere, libellé, nbHeures)
- Etudier (#NumeroINSEE, #CodeMatiere, noteCT, noteCC)



Exemple 2

```
CREATE TABLE Etudiant
  (numeroINSEE NUMBER,
   nom VARCHAR2(30),
   prenom VARCHAR2(30),
   dateNaissance date,
   adresse VARCHAR2(100),
   CONSTRAINT pk_etudiant PRIMARY KEY(numeroINSEE),
   CONSTRAINT ck_nom CHECK (nom IS NOT NULL)
  );
```



Exemple 2

```
CREATE TABLE Matiere
  (codeMatiere NUMBER,
   libelle VARCHAR2(30),
   nbHeures NUMBRER(3),
   CONSTRAINT pk_matiere PRIMARY KEY(codeMatiere),
   CONSTRAINT ck_libelle CHECK (libelle IS NOT NULL),
   CONSTRAINT ck_nbHeures CHECK (nbHeures>0)
  );
```



Exemple 2

```
CREATE TABLE Etudier
  (numeroINSEE NUMBER,
   codeMatiere NUMBER,
   noteCC NUMBER(2),
   noteCT NUMBER(2),
   CONSTRAINT fk_etudier_Etudiant FOREIGN KEY (numeroINSEE)
                                     REFERENCES Etudiant (numeroINSEE),
   CONSTRAINT fk_etudier_matiere FOREIGN KEY (codeMatiere)
                                     REFERENCES Matiere (codeMatiere),
   CONSTRAINT pk_etudier
             PRIMARY KEY (numeroINSEE, codeMatiere)
  );
```



DROP : Syntaxe

- Suppression complète de la table : définition et données

```
DROP TABLE <nom_table>;
```

- Suppression aussi des contraintes de référence filles

```
DROP TABLE <nom_table> CASCADE CONSTRAINTS;
```



L'ordre ALTER TABLE

- Modification de la structure des tables
- Ajout de colonnes

```
ALTER TABLE <nom_table>  
    ADD (<colonne1> type1 [,<colonne2> type2]);
```

- Modification de colonnes

```
ALTER TABLE <nom_table>  
    MODIFY (<colonne1> [type1]  
           [,<colonne2> type2]);
```

- Ajout de contraintes

```
ALTER TABLE <nom_table>  
    ADD CONSTRAINT nom_contrainte type_contrainte;
```

- Suppression de contraintes

```
ALTER TABLE <nom_table>  
    DROP CONSTRAINT nom_contrainte;
```



Exemples

```
ALTER TABLE Services  
    ADD Budget NUMBER (10,2);
```

```
ALTER TABLE Employes  
    MODIFY nomEmploye VARCHAR2(60);
```

```
ALTER TABLE Employes  
    ADD CONSTRAINT ck_salaire CHECK (salaire>3000);
```




Activation et désactivation de contraintes

- Désactivation de contraintes

```
ALTER TABLE <nom_table>  
    DISABLE CONSTRAINT nom_contrainte;
```

- Les contraintes existent toujours dans le dictionnaire de données mais ne sont pas actives
- Chargement de données volumineuses extérieures à la base

- Activation de contraintes

```
ALTER TABLE <nom_table>  
    ENABLE CONSTRAINT nom_contrainte;
```