

SGBD(R)

Système de Gestion de Bases de Données (Relationnels)

OLIVARÈS Georges

25/05/2016

© Groupe CGI inc. CONFIDENTIEL

CGI

La force de l'engagement[MD]

SGBDr
Règles de conception



CHAPITRE 5

CGI

La force de l'engagement[MD]

Concepts

Une Base de donnée est définie par un schéma

Un schéma se compose d'un ensemble de schémas de relations

create_relational_table ::=

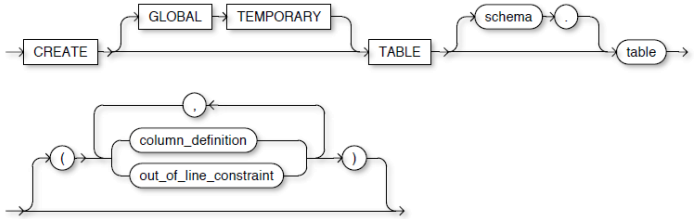
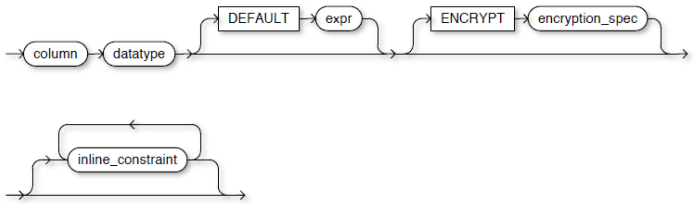


Schéma de relation:

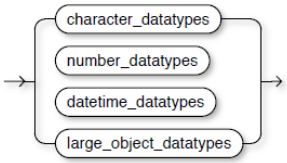
- Nom de la relation
- Liste de nom d'attributs, types et contraintes

column_definition ::=



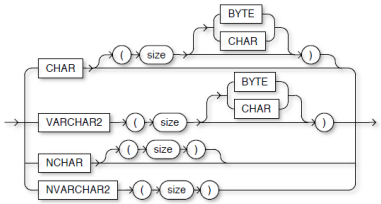
- Types: *character, numeric, decimal, integer, float, real, …*

datatypes ::=

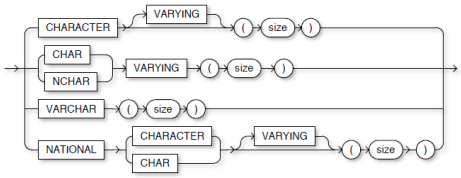


Types

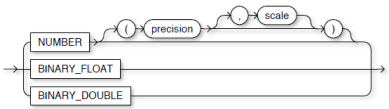
character_datatypes ::=



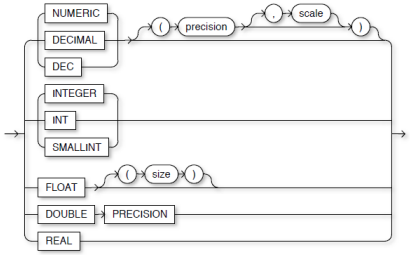
ansi_character_datatypes ::=



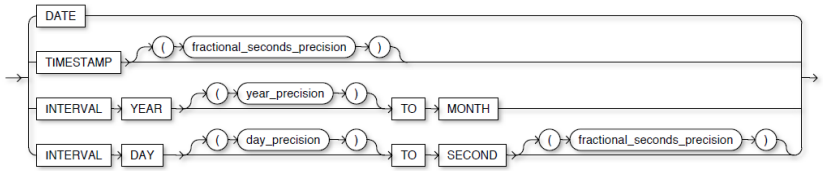
number_datatypes ::=



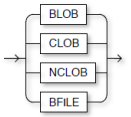
ansi_number_datatypes ::=



- *datetime_datatypes ::=*



large_object_datatypes ::=



Création d'une relation/table

```
CREATE TABLE IF NOT EXISTS `musiciens` (
  `nm` int(11) NOT NULL AUTO_INCREMENT,
  `nom` varchar(50) NOT NULL,
  `prenom` varchar(50) NOT NULL,
  `qualite` varchar(20) NOT NULL,
  PRIMARY KEY (`nm`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=121 ;
```



CGI

Contraintes d'intégrité

Contrainte NOT NULL

- Interdit la valeur de base d'être NULL

Contrainte Unique

- Interdit à plusieurs tuples d'avoir dans la même colonne ou la même combinaison de colonnes, mais autorise certaines valeurs à NULL

Contrainte Primary key

- Unique + NULL

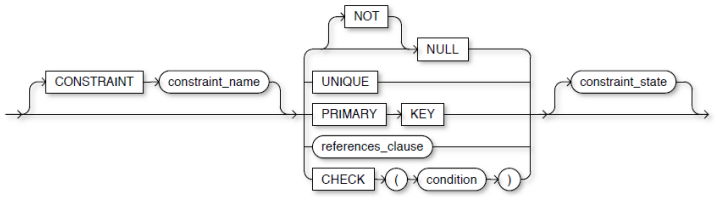
Contrainte Foreign key

- Requiert que les valeurs d'une table soit semblables à des valeurs dans une autre table

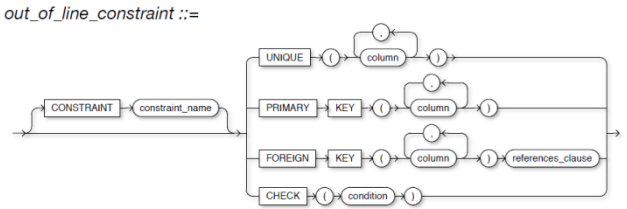
Contrainte Check

- Requiert que la valeur soit confirme à la condition spécifiée

inline_constraint ::=



out_of_line_constraint ::=



Contraintes – Clé primaire

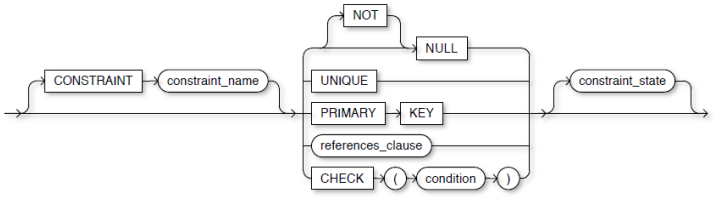
- Désigne un attribut ou un ensemble d'attributs comme étant la clé primaire d'une table/vue
- Pas de contrainte d'unicité
- Ne peut pas contenir de valeur à NULL
- Il ne peut y avoir qu'une seule clé primaire par table/vue
- Ne peut pas être appliqué à tous les types

```
«  PRIMARY KEY (`col1`, `col2`)  »

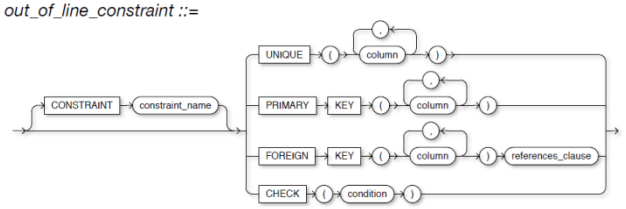
CREATE TABLE musicien (nm INTEGER NOT NULL UNIQUE, …)

CREATE TABLE musicien (nm INTEGER PRIMARY KEY, …)
```

inline_constraint ::=



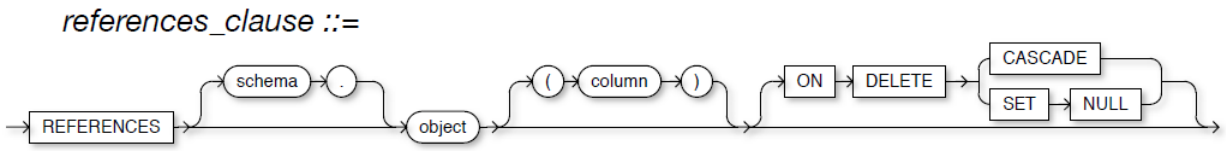
out_of_line_constraint ::=



Contraintes – Clé étrangère

- Désigne un attribut ou un ensemble d'attributs comme étant une clé étrangère et établit une relation entre cet attribut(s) et la clé primaire appelée référencée
- Par défaut, il utilise la clé primaire
- Une table peut comporter plusieurs clés étrangères

```
CREATE TABLE joue (
  nm INTEGER NOT NULL REFERENCES musicien(nm),
  no INTEGER NOT NULL REFERENCES oeuvre(no),
  lieu VARCHAR2(30) NOT NULL,
  "date" DATE NOT NULL,
  constraint MaClePrimaire PRIMARY KEY (nm, no, "date")
)
```



Contraintes – ON DELETE

- Détermine la façon dont le SGBD maintien l'intégrité référentielle si une clé référencée est supprimée
- ON DELETE CASCADE: supprime également les clés étrangères dépendantes
- ON DELETE SET NULL: affecte NULL

```
CREATE TABLE joue (
   nm INTEGER NOT NULL REFERENCES musicien(nm)
          ON DELETE CASCADE,
   no INTEGER NOT NULL REFERENCES oeuvre(no)
          ON DELETE CASCADE,
   lieu VARCHAR2(30) NOT NULL,
   "date" DATE NOT NULL,
   constraint MaClePrimaire PRIMARY KEY (nm, no, "date")
)
```

- ***ON DELETE CASCADE***

OEUVRE	NOM	PERIODE	NC
	Concerto n°2	Romantique	120

COMPOSITEUR	NC	NOM	NATIONALITE
	120	Chopin	Polonaise

- ***ON DELETE SET NULL***

OEUVRE	NOM	PERIODE	NC
	Concerto n°2	Romantique	120
	6e Symphonie	Classique	Null

COMPOSITEUR	NC	NOM	NATIONALITE
	120	Chopin	Polonaise

OEUVRE	NOM	PERIODE	NC
	Concerto n°2	Romantique	120
	6e Symphonie	Classique	100

COMPOSITEUR	NC	NOM	NATIONALITE
	100	Beethoven	Allemande
	120	Chopin	Polonaise

Contraintes – Check

- Définit un prédicat qui doit être vérifié pour chaque tuple de la relation

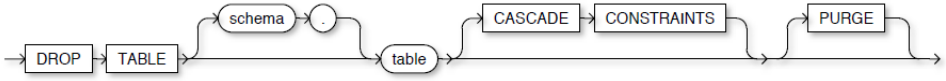
```
CREATE TABLE oeuvre (
  no INTEGER PRIMARY KEY,
  nom VARCHAR2(30) NOT NULL,
  type VARCHAR2(15) NOT NULL DEFAULT 'Concerto',
  duree INTEGER CONSTRAINT Positive CHECK(duree >= 0),
  …
)
```

CRUD d'une relation

Supprimer – DROP TABLE

- Supprime les données, les index, les déclencheurs, l'espace allouée (LOBs), …
- CASCADE CONSTRAINTES obligatoire pour supprimer également les contraintes
- PURGE pour supprimer sans mettre dans la « corbeille »

drop_table ::=

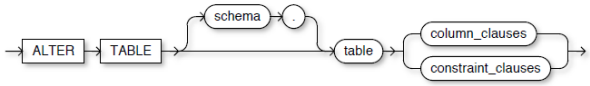


CRUD d'une relation

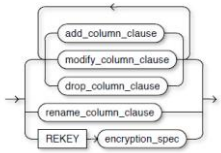
Modifier – ALTER TABLE

- Supprime les données, les index, les déclencheurs, l'espace allouée (LOBs), …
- CASCADE CONSTRAINTES obligatoire pour supprimer également les contraintes
- PURGE pour supprimer dans mettre dans la « corbeille »

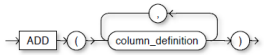
alter_table ::=



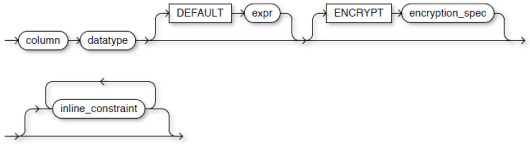
column_clauses ::=



add_column_clause ::=



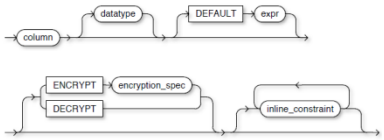
column_definition ::=



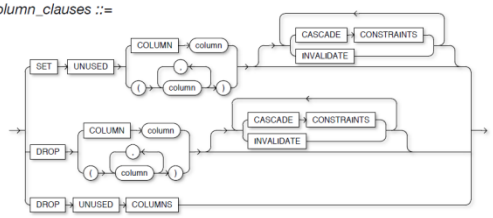
modify_column_clause ::=



modify_col_properties ::=



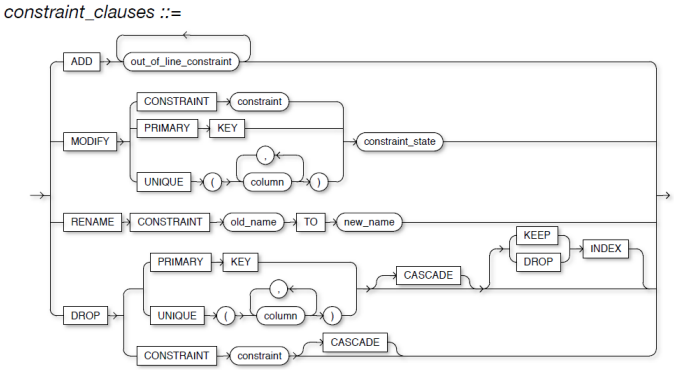
drop_column_clauses ::=



rename_column_clauses ::=



CRUD d'une relation



Ajout de l'age du musicien

```
ALTER TABLE musicien ADD( age INTEGER DEFAULT 20 CONSTRAINT Age CHECK(age > 0))
```

Modification contrainte d'intégrité

```
ALTER TABLE musicien DROP CONSTRAINT age
ALTER TABLE musicien MODIFY( age CONSTRAINT Age CHECK(age >= 0))
```

Suppression de l'age

```
ALTER TABLE musicien DROP COLUMN age
```

Création d'index

Ajout de l'age du musicien

- Comporte une entrée pour chaque valeur apparaissant dans une ou plusieurs colonnes indexées d'une table
- Fournit un accès direct et rapide aux tuples

Création d'un index sur le nom du musicien

- ***CREATE INDEX*** *nom_ix* ***ON*** *MUSICIEN****(****NOM****);***

Création d'un index sur la date

- ***CREATE UNIQUE INDEX*** *date_ix* ***ON*** *OEUVRE****(****"DATE"****) DESC;***

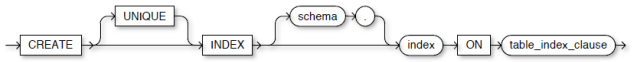
Création d'un index basé sur une fonction

- ***CREATE INDEX*** *nom_ix* ***ON*** *MUSICIEN****(UPPER(****NOM****));***

et usage dans une requête

- ***SELECT * FROM*** *MUSICIEN* ***WHERE UPPER(****NOM****) IS NOT NULL AND UPPER (****NOM****) = 'DUPONT';***

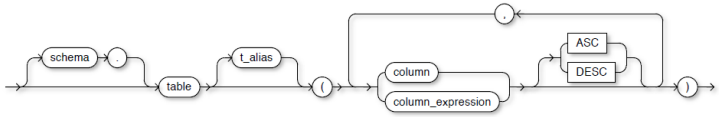
create_index ::=



drop_index ::=



table_index_clause ::=



Exemple

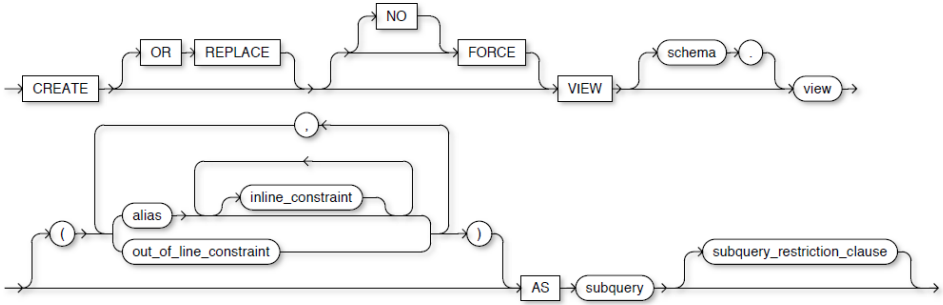
- ***DROP INDEX*** *nom_ix****;***

Création d'une vue

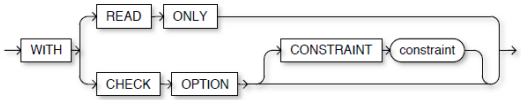
- Donne un point de vue particulier d'une BDD
- C'est une table virtuelle, produite dynamiquement

```
CREATE VIEW excellent_musicien AS
   SELECT nom, prenom, qualite FROM musicien
   WHERE qualite = 'Experimenté' OR qualite = 'Virtuose'
   WITH CHECK OPTION;
```

create_view ::=



subquery_restriction_clause ::=



drop_view ::=

