



SQL – DDL

SQL DDL : Data Definition Language

Fabien Coelho, Claire Medrala

Mines Paris – PSL

Décembre 2023

SQL DDL

- Syntaxe
- Types
- TABLE
- SEQUENCE
- VIEW
- Conseils

commandes `CREATE ALTER DROP`

objets `TABLE VIEW SEQUENCE`
 aussi `FUNCTION OPERATOR TRIGGER AGGREGATE CAST`
 encore `INDEX DATABASE SCHEMA USER GROUP ROLE POLICY`
`CONVERSION RULE COLLATION SERVER TABLESPACE TYPE...`

options `OR REPLACE, IF [NOT] EXISTS, CASCADE/RESTRICT`



Différents types de bases, installés par défaut

Relations `CREATE / ALTER / DROP TABLE`

SQL DDL

- Syntaxe
- Types
- TABLE
- SEQUENCE
- VIEW
- Conseils

booléens `BOOL BOOLEAN`

entiers `INT INTEGER INT2 INT4 INT8`
`SMALLINT BITINT SERIAL SERIAL8 NUMERIC`

flottants `FLOAT4 FLOAT8 REAL FLOAT DOUBLE PRECISION DECIMAL`

textes `CHAR(8) VARCHAR(12) TEXT`

temps `DATE TIME TIMETZ TIMESTAMP INTERVAL`
 fonctions spéciales `NOW() CURRENT_DATE CURRENT_TIME`
 arithmétique de date et d'intervalles...

binaires `BIT(12) VARBIT(24) BYTEA`

extensions géométrie, argent, réseau, bibliothèque, JSON

SQL DDL

- Syntaxe
- Types
- TABLE
- SEQUENCE
- VIEW
- Conseils

- définition initiale d'une relation
 nommage, attributs, types, contraintes...
`CREATE TABLE`
`Personnage(-- nom de la relation`
`id INTEGER, -- attribut et type...`
`nom TEXT,`
`CONSTRAINT clef_personnage UNIQUE(id));`
- modification ultérieure
`ALTER TABLE Personnage ADD COLUMN age INTEGER;`
- suppression définitive !
`DROP TABLE Personnage;`



Position des contraintes

3 choix



Types de contraintes 1/4

SQL DDL

colonne concernée après sa déclaration

```
CREATE TABLE info
(id INTEGER PRIMARY KEY, nom TEXT NOT NULL);
```

table peut impliquer plusieurs colonnes

```
CREATE TABLE info
(id INTEGER, nom TEXT,
PRIMARY KEY(id),
CHECK (nom IS NOT NULL));
```

nommée manipulation explicite possible (*ajout, retrait*)

```
CREATE TABLE info
(id INTEGER, nom TEXT,
CONSTRAINT info_key PRIMARY KEY(id),
CONSTRAINT info_nom CHECK (nom IS NOT NULL));
```

5 / 18

SQL DDL

type de donnée de l'attribut !

texte avec taille maximale ou fixe, valeurs numériques...

```
CREATE TABLE SecuriteSociale(
nom VARCHAR(32), prenom VARCHAR(32)
secu CHAR(13), clef CHAR(2),
remboursement NUMERIC(10,2));
```

clef(s) attribut ou groupe d'attribut UNIQUE

utilisation d'un index pour garantir l'unicité

```
CREATE TABLE Film
(id INTEGER UNIQUE, titre TEXT, auteur TEXT,
-- un auteur ne fait un film qu'une fois ?
UNIQUE(titre, auteur));
```

6 / 18



Types de contraintes 2/4



Types de contraintes 3/4

SQL DDL

LA clef primaire PRIMARY KEY, implique UNIQUE NOT NULL

```
CREATE TABLE Voiture
(plaque VARCHAR(10) PRIMARY KEY,
marque TEXT, type TEXT);
CREATE TABLE Individu
(prenom TEXT, nom TEXT, ne DATE, addr TEXT,
PRIMARY KEY(prenom, nom));
```

clef étrangère FOREIGN KEY (...) REFERENCES tab(...)

```
CREATE TABLE CarteGrise
(numero TEXT PRIMARY KEY,
plaque VARCHAR(10) REFERENCES Voiture,
prenom TEXT, nom TEXT, delivree DATE,
FOREIGN KEY (prenom, nom) REFERENCES Individu);
```

conséquence si disparition de la clef référencée !?

7 / 18

SQL DDL

assertion CHECK(condition)

- expression booléenne simple (voir aussi TRIGGER)
- attention, pas de sous requête !

```
CREATE TABLE Elève(
prenom TEXT, nom TEXT, ne DATE,
-- nom bien défini
CHECK (LENGTH(prenom)>1 AND LENGTH(nom)>1),
-- né dans le passé !
CONSTRAINT eleve_né CHECK (CURRENT_DATE >= ne)
);
```

8 / 18



Types de contraintes 4/4

SQL DDL

facultatif **NOT NULL** ou **NULL** (par défaut peut être nulle)
la plupart des colonnes sont normalement **NOT NULL** !

```
CREATE TABLE Voiture
(marque TEXT NOT NULL,
conducteur TEXT NOT NULL,
passager TEXT NULL);
```

valeur par défaut **DEFAULT expr**
utilisé lors d'une insertion si valeur non spécifiée

```
CREATE TABLE compte
(insertion TIMESTAMP DEFAULT NOW(),
combien MONEY DEFAULT MONEY '0',
libel TEXT DEFAULT 'virement');
```

9 / 18



Vérification des contraintes 1/4

Automatique

SQL DDL

```
CREATE TABLE Groupe
(gid INTEGER PRIMARY KEY,
nom TEXT NOT NULL,
CONSTRAINT nom_long CHECK(LENGTH(nom)>2));
```

```
INSERT INTO Groupe VALUES(1, 'Beatles'); -- OK
INSERT INTO Groupe VALUES(1, 'Pink Floyd');
-- ERROR: duplicate key violates
-- unique constraint "groupe_pkey"
INSERT INTO Groupe VALUES(2, 'U2');
-- ERROR: new row for relation "groupe" violates
-- check constraint "nom_long"
INSERT INTO Groupe VALUES(3, NULL);
-- ERROR: null value in column "nom" violates
-- not-null constraint
```

10 / 18



Vérification des contraintes 2/4

Automatique

SQL DDL

```
CREATE TABLE Artiste
(aid INTEGER NOT NULL,
nom TEXT NOT NULL,
gid INTEGER NOT NULL REFERENCES Groupe);
```

```
INSERT INTO Artiste VALUES(1, 'John', 1); -- OK
INSERT INTO Artiste VALUES(2, 'Paul', 1); -- OK
INSERT INTO Artiste VALUES(3, 'Syd', 7);
-- ERROR: insert or update on table "Groupe"
-- violates foreign key constraint "f1"
-- key gid=7 is not present in table "Groupe".
```

```
DELETE FROM Group WHERE gid=1;
-- ERROR: update or delete on "Group"
-- violates foreign key constraint "f1" on "Artiste"
-- key gid=1 is still referenced from table "Artiste".
```

11 / 18



Vérification des contraintes 3/4

compléments

SQL DDL

Comportement des références

ON DELETE/ON UPDATE

NO ACTION génère une erreur (défaut)
CASCADE modifie la référence ou efface le tuple
SET NULL retire la référence
SET DEFAULT valeur par défaut de la référence

Options

TABLE

TEMPORARY table détruite à la fin de la session
UNLOGGED rapide mais pas de transactions (pas de WAL)
FILLFACTOR taux de remplissage 100 – n% (**UPDATE** en place)

12 / 18



Vérification des contraintes 4/4

décalée



Séquence

CREATE / ALTER / DROP SEQUENCE

SQL DDL

- décalable : DEFERRABLE vs NOT DEFERRABLE
- si décalable : INITIALLY IMMEDIATE vs INITIALLY DEFERRED

```
CREATE TABLE Auteur
  (aid INTEGER PRIMARY KEY, nom TEXT);
CREATE TABLE Poeme
  (aid INTEGER NOT NULL REFERENCES Auteur
   DEFERRABLE INITIALLY DEFERRED,
   titre TEXT);
BEGIN;
INSERT INTO Poeme VALUES(1, 'La prose du Transibérien...');
-- attend le commit pour vérifier l'auteur
INSERT INTO Auteur VALUES(1, 'Blaise Cendrар');
COMMIT;
```

13 / 18

SQL DDL

- entier INT8 géré par la base de données
incréméntation automatique avec fonction NEXTVAL
- valeurs de départ (1), incrément (1), cyclique ou non. . .
- utilisé pour des numérotations automatiques

```
CREATE SEQUENCE compteur
  START WITH 123
  INCREMENT BY 1
  NO CYCLE;

SELECT NEXTVAL('compteur');
```

14 / 18



Numérotation automatique

SERIAL SERIAL8



Vues

CREATE / ALTER / DROP VIEW

SQL DDL

explicit CREATE SEQUENCE + DEFAULT

```
CREATE SEQUENCE info_sq;
CREATE TABLE info2
  (id INT4 PRIMARY KEY DEFAULT NEXTVAL('info_sq'),
   contenu TEXT);
```

implicit SERIAL

très utile pour clef primaire : PRIMARY KEY

```
CREATE TABLE info1
  (id SERIAL PRIMARY KEY, contenu TEXT);
```

15 / 18

SQL DDL

- **relation virtuelle** créée à partir d'un SELECT
- compatibilité ascendante si évolution : anciennes tables en vues
- unité de gestion des droits d'accès
- simplification de requêtes, d'attributs calculés. . .
- vues matérialisées, modifiables (si simples). . .

```
-- uniquement les élèves
CREATE VIEW les_élèves AS
  SELECT nom, prenom, né
  FROM Personnes
  WHERE isEleve;

GRANT SELECT ON les_élèves TO PUBLIC;
```

16 / 18



Quelques conseils

SQL DDL

Syntaxe
Types
TABLE
SEQUENCE
VIEW
Conseils

- identificateurs en minuscules id nom adresse
éventuellement tables capitalisées ? Artiste Groupe
- prévoir une clef primaire entière pid cid gid
de préférence **SERIAL + PRIMARY KEY**
simple, courte, non-sémantique donc pas de modification ?
- nommer de la même façon les clefs primaires et étrangères ?
plus facile à comprendre et à suivre !
- savoir mettre (transactionnel)
ou ne pas mettre (décisionnel, copie)
ou mettre plus tard (chargement)

17 / 18



Autres

SQL DDL

Syntaxe
Types
TABLE
SEQUENCE
VIEW
Conseils

Vues matérialisées

- précalcul persistant, réactualisé...
CREATE/REFRESH/ALTER/DROP MATERIALIZED VIEW

Collations – tri selon une langue

- requêtes, indexes, support du système
CREATE/ALTER/DROP COLLATION

Rules – ré-écriture au vol des opérations

- modification d'une vue, compatibilité...
CREATE/ALTER/DROP RULE

18 / 18