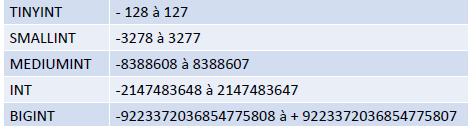
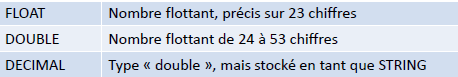
# SQL

## Différents types

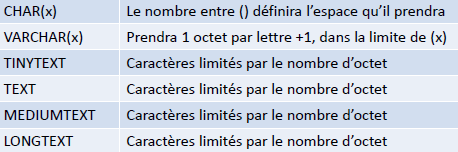
### Entiers



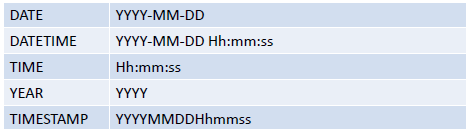
### Flottants



### Chaines de caractères



### Dates



## Conventions SQL

### Nom de tables et colonnes

• Pas d’espace (ils sont remplacés par des underscores)

• Pas de caractères spéciaux

• Pas de mots réservés (Ex : DATE, TEXT, TYPE, TABLE, etc.)

• Noms représentatifs, sans abréviations

• Privilégier le singulier

### Syntaxe des commandes

• Mots-clés en majuscules

• Noms des bases, des tables et des colonnes en minuscules

• Options facultatives des commandes entre crochets

## Connexion à MariaDB / PostGreSQL

• **Se connecter à MariaDB en local :** mariadb -u root -p

• **Se connecter au serveur MariaDB distant :** mariadb -h serveur\_distant -u user -p pwd

• **Se connecter à PostGreSQL :** psql -U postgres

## Manipulations de base

• **Effacer l’écran du terminal** (sur Windows) **:** \! cls

• **Écrire un commentaire :** --commentaire **OU :** # commentaire

### Commandes exclusives à MariaDB

• **Connaitre la base de données courante :** SELECT DATABASE();

• **Afficher les bases de données existantes :** SHOW DATABASES;

• **Afficher les bases de données existantes contenant une certaine chaine de caractères :** SHOW DATABASES like '%chaine%’;

• **Sélectionner une des bases de données** ou **changer la base de données courante :** USE bdd;

• **Afficher les tables d’une base de données :** SHOW TABLES;

• **Afficher les caractéristiques d’une table (structure) :** DESCRIBE table; **OU :** DESC table;

• **Afficher le nom de l’utilisateur connecté :** SELECT USER();

• **Afficher la date courante** (du serveur) **:** SELECT CURDATE();

• **Quitter / stopper la connexion :** EXIT; **OU :** QUIT;

• **Vérifier si le cache est activité ou désactivé :** SHOW VARIABLES LIKE '%query\_cache%';

### Commandes exclusives à PostGreSQL

• **Afficher la liste de toutes les commandes :** \?

• **Afficher la liste des bases de données :** \l

• **Changer la base de données courante :** \connect bdd, \c bdd

• **Afficher des informations sur la connexion en cours :** \conninfo

• **Afficher la version de PostGreSQL :** SELECT VERSION();

• **Afficher la date courante** (du serveur) **:** SELECT CURRENT\_DATE;

• **Afficher l’heure courante** (du serveur) **:** SELECT CURRENT\_TIME;

• **Afficher les tablespaces existants :** \db

• **Afficher plus d’informations sur les tablespaces existants :** \db+

• **Afficher les tables d’une base de données :** \d;

• **Afficher les caractéristiques d’une table (structure) :** \d table;

• **Retarder l’exécution du processus serveur de 5 secondes :** PG\_SLEEP(5);

• **Quitter :** \q

## Utilisation de fichiers

### MariaDB

• **Sauvegarder une copie des commandes que l’on va écrire dans un fichier :** TEE chemin/fichier.txt;

• **Arrêter l’enregistrement des commandes dans le fichier :** NOTEE;

• **Exécuter un script SQL sur la console :** SOURCE chemin/fichier.sql;

### PostGreSQL

• **Sauvegarder une copie des sorties que l’on va afficher dans un fichier :** \o chemin/fichier.txt;

• **Arrêter l’enregistrement des sorties dans le fichier :** \o;

• **Exécuter un script SQL sur la console (Linux) :** psql -U postgres -f chemin/fichier.sql;

• **Exécuter un script SQL sur la console (Windows) :** psql -U postgres \i chemin/fichier.sql;

## SELECT … FROM … WHERE …

### SELECT

• **Sélectionner tous les champs :** SELECT \*

• **Sélectionner un champ en supprimant ses doublons :** SELECT DISTINCT champ

• **Renommer un champ :** SELECT champ AS "champ\_renommé"

• **Préfixer un champ** (pour prévenir le risque d’ambigüité) **:** SELECT table.champ

Remarque : dès que l’on utilise une opération ou une fonction, il faut utiliser un alias de colonne.

### FROM

• **Renommer une table :** table AS table\_renommée

• **Combiner les données de plusieurs tables :**

SELECT table1.colonne1, table2.colonne2

FROM table1 JOIN table2 ON table1.colonne\_clé1 = table2.colonne\_clé2

-INNER JOIN : renvoie l’info quand elle est vraie dans les deux tables ABCDE

-LEFT JOIN : renvoie toutes les colonnes de la table de gauche et dans les colonnes de la table droite affiche NULL si pas d’info

WHERE table1.colonne1 > valeur;

### WHERE

• **Définir un intervalle :** champ BETWEEN borne1 AND borne2

• **Comparer un champ à une liste de valeurs :** champ (NOT) IN(valeur1, valeur2, valeur3)

• **Comparer partiellement une chaine :** champ (NOT) LIKE '%chaine\_' ; champ (NOT) LIKE 'ch\_\_\_aine'

Remarques :

— le symbole % remplace zéro, un ou plusieurs caractères

— le symbole \_ remplace un caractère

— utiliser le caractère d’échappement « \ » pour utiliser les caractères % ou \_ dans la requête

• **Combiner plusieurs conditions :**

— **et retourner vrai si les deux conditions sont vraies :** (condition1) AND (condition2)

— **et retourner vrai si l’une au moins des conditions est vraie :** (condition1) OR (condition2)

— **et** **retourner vrai si la condition qui suit l’opérateur est fausse :** (condition1) NOT (condition2)

Ordre de priorité : parenthèses > NOT > AND > OR.

• **Vérifier si une valeur est nulle ou pas :** valeur IS (NOT) NULL

• **Évaluer une sous-requête :**

— **comparer un champ aux résultats d’une sous-requête :** champ IN (sous-requête)

— **comparer un champ à chaque résultat d’une sous-requête :** champ ANY (sous-requête)

— **comparer un champ à tous les résultats d’une sous-requête :** champ ALL (sous-requête)

### GROUP BY

• **Organiser des données identiques en groupes** (« en fonction de champ1 et de champ2 », « pour chaque champ1 et chaque champ2 »)**:** GROUP BY champ1, champ2

• Remarque : Dès que l’on souhaite afficher plusieurs champs dont l’un est composé d’une fonction de calcul, il faut effectuer un regroupement.

Toute colonne ou expression de la liste SELECT autre qu’une fonction de groupe doit être incluse dans la clause GROUP BY.

### HAVING

• **Filtrer des agrégations :** HAVING condition

Ex : SELECT MAX(champ)

FROM table

HAVING MAX(champ) > valeur/(sous-requête)

### ORDER BY

• **Afficher le résultat dans un ordre croissant, alphabétique, chronologique :** ORDER BY champ1 ASC, champ2 ASC

• **Afficher le résultat dans l’ordre inverse :** ORDER BY champ DESC

### LIMIT

• **Limiter le nombre de lignes à afficher :** LIMIT nombre\_de\_ligne\_à\_afficher OFFSET numéro\_première\_ligne\_à\_afficher

## INSERT, UPDATE, DELETE

• **Insérer de nouvelles lignes dans une table :**

INSERT INTO table(champ1, champ2, …, champN)

VALUES(valeur1, valeur2, …, valeurN), (valeurA, valeurB, …, valeurX);

Remarques :

— On peut utiliser les valeurs spéciales comme « SYSDATE » qui renvoie la date et l’heure courantes.

— On peut utiliser des variables de substitution en écrivant : « &valeur ».

• **Copier des lignes d’une autre table :**

INSERT INTO table1(champ1, champ2)

SELECT champ1, champ2

FROM table2

WHERE condition;

• **Modifier des données d’une table :**

UPDATE table

SET champ1 = valeur1, champ2 = valeur2, …, champN = valeurN

WHERE condition;

• **Modifier des lignes d’une table en fonction d’une autre table :**

UPDATE table

SET champX = (sous-requête1)

WHERE champY = (sous-requête2);

• **Supprimer des lignes d’une table :**

DELETE FROM table

WHERE condition;

• **Supprimer des lignes en faisant référence à une autre table :**

DELETE FROM table

WHERE champ = (sous-requête);

Remarque : Une sous-requête ressemble à : SELECT champ FROM table WHERE condition

## CREATE, ALTER, DROP

### Bases de données

• **Créer une base de données :** CREATE DATABASE bdd;

• **Supprimer une base de données :** DROP DATABASE bdd;

### Tables

• **Créer une table :** CREATE TABLE table(colonne1 type\_données, colonne2 type\_données);

• **Renommer une table :** ALTER TABLE table RENAME TO nouveau\_nom;

• **Supprimer une(des) table(s) :** DROP TABLE table1, table2;

### Colonnes

• **Ajouter une colonne :** ALTER TABLE table ADD colonne type\_données;

• **Modifier une colonne :** ALTER TABLE table MODIFY colonne type\_données;

• **Renommer une colonne :** ALTER TABLE table RENAME COLUMN colonne TO nouveau\_nom;

• **Supprimer une colonne :** ALTER TABLE table DROP colonne;

### Index

• **Créer un index sur une(des) colonne(s) :** CREATE INDEX idx\_colonne ON table(colonne1, colonne2);

• **Supprimer un index :** DROP INDEX idx\_colonneS ON table(colonne);

Remarque : les deux index par défaut sont « UNIQUE » et « PRIMARY KEY ».

### Vues

• **Créer une vue :** CREATE VIEW v\_vue AS

SELECT colonne1, colonne2

FROM table

…

• **Modifier une vue :** CREATE OR REPLACE VIEW v\_vue AS (pareil que pour la création)

• **Extraire les données d’une vue :** SELECT \* FROM v\_vue;

• **Supprimer une vue :** DROP VIEW v\_vue;

### Séquences

• **Séquence :**  permet de générer automatiquement des numéros uniques.

• **Créer une séquence :**

CREATE SEQUENCE séquence

START valeur\_départ

INCREMENT valeur\_incrémentale

MINVALUE valeur\_minimum

MAXVALUE valeur\_maximum

OWNED BY table.colonne (associe une séquence à la colonne d’une table)

• **Insérer de nouvelles lignes dans une table avec une valeur incrémentée par une séquence** (MariaDB) **:**

INSERT INTO table(champ1, champ2, …, champN)

VALUES(valeur1, NEXT VALUE FOR séquence, …, valeurN), (valeurA, NEXT VALUE FOR séquence, …, valeurX);

• **Insérer de nouvelles lignes dans une table avec une valeur incrémentée par une séquence** (PostGreSQL) **:**

INSERT INTO table(champ1, champ2, …, champN)

VALUES(valeur1, nextval(séquence), …, valeurN), (valeurA, nextval(séquence), …, valeurX);

• **Afficher la valeur actuelle d’une séquence** (MariaDB)**:** SELECT PREVIOUS VALUE FOR séquence;

• **Afficher la valeur actuelle d’une séquence** (PostGreSQL) **:** SELECT CURRVAL(séquence);

• **Supprimer une séquence :** DROP SEQUENCE séquence;

## Contraintes

• **Imposer à un champ qu’à tout enregistrement de la table, il soit rempli :** NOT NULL

Syntaxe : CREATE TABLE table (colonne type\_données NOT NULL);

ALTER TABLE table ALTER COLUMN colonne SET NOT NULL;

• **S’assurer que toutes les valeurs d’un champ sont uniques et distinctes** **:** UNIQUE

Syntaxe :

CREATE TABLE table ( … CONSTRAINT UQ\_colonne NOT NULL (colonne), … );

ALTER TABLE table ADD COLUMN colonne type\_données UNIQUE;

ALTER TABLE table ADD CONSTRAINT UQ\_colonne UNIQUE (colonne);

ALTER TABLE table DROP INDEX UQ\_colonne;

• **Cumuler les contraintes de NOT NULL et UNIQUE et créer le moyen de référence à un et un seul enregistrement dans une table :** PRIMARY KEY

Syntaxe :

CREATE TABLE table ( … colonne type\_données PRIMARY KEY, … );

CREATE TABLE table ( … CONSTRAINT PK\_colonne PRIMARY KEY (colonne1, colonne2) );

ALTER TABLE table ADD CONSTRAINT PK\_colonne PRIMARY KEY (colonne1, colonne2);

ALTER TABLE table DROP CONSTRAINT PK\_colonne;

• **Mettre en place des liaison entre les tables**(la clé étrangère est un champ faisant référence à un champ d’une autre table dont il est clé primaire)**:** FOREIGN KEY

Syntaxe :

CREATE TABLE table1 ( … FOREIGN KEY (colonne1) REFERENCES table2(colonne2) );

CREATE TABLE table ( … CONSTRAINT FK\_table1table2 FOREIGN KEY (colonne1) REFERENCES table2(colonne2) );

ALTER TABLE table1 ADD FOREIGN KEY (colonne1) REFERENCES table2(colonne2);

ALTER TABLE table DROP FOREIGN KEY FK\_colonne;

• **Attribuer une valeur par défaut à un champ lorsqu’elle n’est pas spécifiée pour ne pas avoir de champs vides :** DEFAULT

Syntaxe :

CREATE TABLE table ( … colonne type\_données DEFAULT valeur\_par\_défaut,… );

ALTER TABLE table ALTER COLUMN colonne SET DEFAULT valeur\_par\_défaut;

ALTER TABLE table ALTER COLUMN colonne DROP DEFAULT;

• **S’assurer que les données respectent un format bien précis, appartiennent à une liste de valeurs ou respectent une règle de gestion :** CHECK

Syntaxe :

CREATE TABLE table ( … colonne type\_données CHECK (condition) );

CREATE TABLE table ( … CONSTRAINT CHK\_contrainte CHECK (condition) );

ALTER TABLE table ADD CONSTRAINT CHK\_contrainte CHECK (condition);

ALTER TABLE table DROP CONSTRAINT CHK\_contrainte;

Exemples de condition : age>= 18 AND city='Sandnes' ; dateLivraison < datePeremption ; idPays ~ '[A-Z][A-Z][A-Z]' ; value>0

## Commit, rollback

• **Valider les modifications réalisées depuis le début de la transaction :** COMMIT;

• **Annuler** **les modifications réalisées sur la base de données jusqu’à obtenir l’état qu’elle avait avant le début de la transaction :** ROLLBACK;

• **Créer un point de sauvegarde :** SAVEPOINT point\_sauvegarde;

• **Annuler les modifications réalisées sur la base de données jusqu’à obtenir l’état qu’elle avait lors de la création du point de sauvegarde :** ROLLBACK TO SAVEPOINT point\_sauvegarde;

## Fonctions sur les chiffres

• **Compter le nombre de lignes dans une table :** COUNT(\*)

• **Compter le nombre de lignes non vides d’une colonne :** COUNT(colonne)

• **Compter le nombre de lignes différentes d’une colonne :** COUNT(DISTINCT colonne)

• **Obtenir la plus petite valeur de toutes les valeurs sélectionnées d’une colonne :** MIN(colonne)

• **Obtenir la plus grande valeur de toutes les valeurs sélectionnées d’une colonne :** MAX(colonne)

• **Obtenir la moyenne des valeurs d’une colonne :** AVG(colonne)

• **Obtenir la somme de toutes les valeurs d’une colonne :** SUM(colonne)

• **Obtenir la somme des valeurs d’une colonne sans compter les valeurs dupliquées :** SUM(SELECT colonne)

• **Obtenir la variance des valeurs d’une colonne :** VARIANCE(colonne)

• **Obtenir l’écart-type des valeurs d’une colonne :** STDDEV(colonne)

• **Arrondir un résultat numérique :** ROUND(colonne, nombre\_de\_décimal)

abcde

## Fonctions sur les chaines

• **Transformer tous les caractères d’une chaine de caractères en minuscules :** LOWER(chaine)

• **Transformer tous les caractères d’une chaine de caractères en majuscules :** UPPER(chaine)

## Fonctions temporelles

•

## Espace de stockage (tablespaces)

• **Tablespace :** espace de stockage, conteneur de données.

• **Créer un tablespace et l’associer à un répertoire :** CREATE TABLESPACE tablespace LOCATION 'chemin' ;

• **Changer le nom d’un tablespace :** ALTER TABLESPACE tablespace RENAME TO nouveau\_nom;

• **Changer le propriétaire d’un tablespace :** ALTER TABLESPACE tablespace OWNER nouveau\_propriétaire;

• **Supprimer un tablespace** (il doit être vide pour pouvoir être supprimé)**:** DROP TABLESPACE tablespace;

• **Créer une base de données dans un tablespace :** CREATE DATABASE bdd TABLESPACE = tablespace;

• **Créer une table en spécifiant un tablespace :** CREATE TABLE table ( … ) TABLESPACE tablespace;

## Espaces de noms (schémas)

• **Schéma :** espace de noms qui est utilisé pour organiser les noms des objets de base de données. Cela permet d’avoir plusieurs objets portant le même nom dans une même base de données.

• **Créer un schéma dont le propriétaire est un certain utilisateur :** CREATE SCHEMA schéma AUTHORIZATION utilisateur;

• **Supprimer un schéma vide :** DROP SCHEMA schéma;

• **Supprimer un schéma qui possède des objets :** DROP SCHEMA schéma CASCADE;

• **Accéder à un objet d’un schéma :** schéma.objet