**Mémo SQL**

**ATTENTION, IL EST EXTREMEMENT COMPLEXE DE MODIFIER UNE BDD EN-COURS DE PROJET.**

**PRENDRE LE TEMPS DE CONCEVOIR SA BDD EN PRENANT EN COMPTE LES EVOLUTIONS FUTURES, UNIQUEMENT EN FONCTION DES DONNEES A STOCKER, L’APPLICATIF S’ADAPTERA.**

**Commandes principales**

SELECT

INSERT

UPDATE

DELETE

**Ordre des commandes de SELECT**

SELECT \*

expression AS label

FROM table1 label preferez commencer par la plus petite table pour une meilleur performance

INNER JOIN table2 label ON expression condition expression retourne les enregistrements

communs à table1 et table2

LEFT JOIN table2 label ON expression condition expression tous les table1 + communs

RIGHT JOIN table3 …… table3 + communs

WHERE condition sur des expressions

GROUP BY expression

HAVING condition sur des champs calculés

{ UNION | INTERSECT | EXCEPT }

ORDER BY expression {ASC(tri croissant), DESC(tri décroissant)}

LIMIT count

OFFSET start

**Fonctions vues**

CURDATE() date du jour

NOW() date du jour

DATEDIFF(date1,date2) renvoie le nombre de jours date1 – date2

SUM(expresion opérateur expression) somme

COUNT(expresion) compteur

CHAR(valeur) converti tvaleur ASCII en caractère

CONCAT(col1,col2 SEPARATOR char) Concatène col1 et col2 avec char comme séparateur

requete 1

UNION ALL pour ajouter le résultat d’une requete 1 à la requete 2

requete 2

LIKE ‘%toto %’ équivaut à : égal chaine qui contient toto

LIKE ‘toto %’ équivaut à : égal chaine qui débute par toto

**Requetes réalisées le 11.01.2023 après les exercices query-1à11.sql**

Créer une commande de 3 articles différents (avec ses lignes de commande associées)

INSERT INTO

orders (number, customer\_id, date)

VALUES ('0000000006',2,'2023-01-11 14-57-00');

INSERT INTO

order\_product(order\_id,product\_id,quantity)

VALUES

(last\_insert\_id(),2,11),

(last\_insert\_id(),8,12),

(last\_insert\_id(),15,13);

Ajouter un produit avec sa catégorie et sa quantité

INSERT INTO

products(category\_id,name, price, weight, available, quantity)

VALUES (1,"D – Carottes",23,90,1,56);

Ajouter 100 à la quantité en stock d‘un produit

UPDATE products

SET quantity = quantity + 100

WHERE id=8

Augmenter de 5% le prix des produits d’une catégorie donnée

UPDATE products

SET price = price \* 1.05

WHERE category\_id=2;

Supprimer un article

DELETE

FROM products

WHERE id=4;

Supprimer les clients qui n’ont pas de commande

DELETE

FROM customers

WHERE NOT EXISTS(

SELECT customer\_id

FROM orders

WHERE customers.id = orders.customer\_id);

**Exemple réalisé le 11.01.2023**

SELECT number,

SUM(op.quantity\*p.price) AS prix\_Total

FROM orders

INNER JOIN order\_product op on orders.id = op.order\_id

INNER JOIN products p on op.product\_id = p.id

GROUP BY number

HAVING SUM(op.quantity\*p.price)>=100 AND SUM(op.quantity\*p.price)<=550

SELECT

first\_name,

last\_name,

COUNT(o.number) AS NbreCommande

FROM customers

LEFT JOIN orders o on customers.id = o.customer\_id

GROUP BY first\_name, last\_name

ORDER BY last\_name

**Exemple réalisé le 12.01.2023**

-- Exercice 1 Exemple d’une condition en utilisant le MAX d’un champ

SELECT \* avec un SELECT imbriqué

FROM test\_1.users

WHERE salary = (SELECT MAX(salary) FROM test\_1.users )

-- Exercice 6 Exemple UNION ALL

SELECT \*,'Groningen' Plaats FROM test\_6.bills\_groningen

UNION ALL

SELECT \*,'Amsterdam' Plaats FROM test\_6.bills\_amsterdam

UNION ALL

SELECT \*,'Antwerpen' Plaats FROM test\_6.bills\_antwerpen

UNION ALL

SELECT \*,'Haarlem' Plaats FROM test\_6.bills\_haarlem

UNION ALL

SELECT \*,'Leiden' Plaats FROM test\_6.bills\_leiden

ORDER BY bill\_id

-- Exercice 7 Exemple GROUP\_CONCAT et CHAR

SELECT GROUP\_CONCAT(CHAR(test\_7.message.char\_message) SEPARATOR '') FROM test\_7.message

**mySQL, gestion des utilisateurs**

**On peut le faire dans datagrip en allant directement dans la table mysql/tables/users**

**On peut le faire aussi dans une console mysql**

Une fois dans le serveur SQL : mysql>

Création d’un utilisateur avec tous les droits sur toutes les bases

CREATE USER ‘nom\_utilisateur’@’localhost’ IDENTIFIED BY ‘mdp’ ;

GRANT ALL PRIVILEGE \*.\* TO ‘nom\_utilisateur’@’localhost’ ;

FLUSH PRIVILEGES ;

Accorder à un utilisateur avec tous les droits sur la base toto et la table lulu

GRANT ALL PRIVILEGE ON toto.lulu TO ‘nom\_utilisateur’@’localhost’ ;

Accorder à un utilisateur les droits de lecture et d’écriture (pas delete et update) sur toutes les bases

GRANT SELECT,INSERT \*,\* TO ‘nom\_utilisateur’@’localhost’ ;

Création d’un utilisateur avec tous les droits sur toutes les bases et le droit de créer des utilisateurs

CREATE USER ‘nom\_utilisateur’@’localhost’ IDENTIFIED BY ‘mdp’  WITH GRANT OPTION

Renomer un utilisateur

RENAME USER ‘nom\_utilisateur’@’localhost’

Supprimer un utilisateur

DROP USER ‘nom\_utilisateur’@’localhost’

Modifier mdp utilisateur

SET PASSWORD FOR 'user'@'localhost' = PASSWORD('newpassword');

Révoquer les privilèges d’un utilisateur

REVOKE ALL PRIVILEGES, GRANT OPTION FROM 'user'@'localhost';

**Script de création de table - Exemple réalisé le 13.1.2023**

-- TABLE POSTS  
-- Suppression de la table si elle existe  
DROP TABLE IF EXISTS iteration3.posts; pour supprimer la table  
-- Création de la table  
CREATE TABLE iteration3.posts ( indiquer le chemin de la table  
 `id` int PRIMARY KEY NOT NULL AUTO\_INCREMENT,  
 `user\_id` int,  
 FOREIGN KEY (user\_id) REFERENCES users(id) ON DELETE CASCADE , clef etrangere  
 `title` varchar(255), effacement en cascade  
 `created\_at` date,  
 `updates\_ad` date DEFAULT NULL  
);  
-- Initialisation de la table  
LOCK TABLES iteration3.posts WRITE;  
INSERT INTO iteration3.posts (user\_id, title, created\_at, updates\_ad)  
VALUES  
 (1,'Monsieur','2023-01-12','2023-01-13'),  
 (2,'Monsieur','2023-01-12','2023-01-13');  
UNLOCK TABLES;

**Script de création d’une base – Exemple réalisé le 17.1.2023**

CREATE DATABASE IF NOT EXISTS square\_games;

**Script de création de 2 tables avec chacune ayant une clef étrangère la liant à l’autre**

-- creation boards  
DROP TABLE IF EXISTS square\_games.boards;  
CREATE TABLE square\_games.boards  
(  
 id INTEGER(12) PRIMARY KEY AUTO\_INCREMENT NOT NULL,  
 size INT(2),  
 display\_size INT(4),  
 dark\_color CHAR(6) DEFAULT '333333',  
 light\_color CHAR(6) DEFAULT 'DDDDDD',  
 game\_id INT(10) NOT NULL  
 -- ,FOREIGN KEY (game\_id) REFERENCES games(id) ON DELETE CASCADE  
);  
-- creation games  
DROP TABLE IF EXISTS square\_games.games;  
CREATE TABLE square\_games.games (  
 id INTEGER(10) PRIMARY KEY AUTO\_INCREMENT NOT NULL,  
 name VARCHAR(32) NOT NULL ,  
 age\_mini\_authorization INT(3) NOT NULL,  
 age\_maxi\_authorization INT(3) NOT NULL,  
 age\_mini\_against\_human INT(3) NOT NULL,  
 age\_maxi\_against\_human INT(3) NOT NULL,  
 default\_board\_id INT(10),  
 FOREIGN KEY ( default\_board\_id ) REFERENCES boards(id) ON DELETE CASCADE  
  
);  
ALTER TABLE square\_games.boards ADD CONSTRAINT FOREIGN KEY (game\_id) REFERENCES games(id) ON DELETE CASCADE;