**Lexique SQL**

\*CREATE : Permet de créer une table, database, view

Exemple : CREATE TABLE DEPT(

--code

);

\*DROP : Permet de supprimer une table, database

Exemple : DROP TABLE IF EXISTS EMP;

\*ALTER : Permet de modifier une table

Exemple : ALTER TABLE EMP

ADD PROJETNO SMALLINT;

\*ADD : Ajout d’un élément dans une table

Exemple : ALTER TABLE EMP

ADD PROJETNO SMALLINT;

\*IF NOT EXISTS : Si il n’existe pas voici quelques exemples :

CREATE TABLE IF NOT EXISTS test (

--code ) ;

\*SELECT FROM : Permet de choisir plusieurs attributs depuis une entité

Exemple : SELECT EMP.PROJETNO, JOB, EMP.DEPTNO FROM EMP;

\*WHERE : Définit une condition

Exemple : WHERE DEPTNO IN(30)

Dans ce cas on veut les personnes dans le département 30

\*LIKE : Utilisé dans la clause WHERE il est utilisé dans certains cas pour voir si certaines occurrences contiennent certaine lettres

Exemple : SELECT \*

FROM EMP

WHERE ENAME LIKE '%e';

Dans ce cas on cherche les noms terminant par e

\*HAVING : Est presque similaire a WHERE sauf qu’il permet de mieux filtrer les SUM(), MIN(), MAX()…

Exemple : HAVING COUNT(ENAME)>=2

Dans ce pas on a un conteur de nom supérieur à 2

\*ORDER BY : Permet de créer un ordre dans l’ordre croissant (ASC) ou décroissant (DESC)

Exemple : ORDER BY JOB DESC, SAL DESC;

\*GROUP BY : Permet de regrouper plusieurs instances d’un attribut

Exemple : ORDER BY EMP.DEPTNO;

\*PRIMARY KEY : Permet d’avoir une clé primaire dans une table

Exemple : city\_zipcode VARCHAR (5) PRIMARY KEY,

\*CONSTRAINT : Permet de faire une contrainte sur une clé primaire ou étrangère

Exemple : CONSTRAINT PK\_EMP\_EMPNO\_ID PRIMARY KEY (EMPNO),

CONSTRAINT FK\_EMP\_DEPTNO FOREIGN KEY (DEPTNO) REFERENCES DEPT (DEPTNO),

Dans ce cas cela permet de renommer la clé primaire (1ere ligne) et la clé étrangère (2ème ligne)

\*IDENTITY : Permet d’avoir une auto-incrémentation sur en nombre

Exemple : EMPNO SMALLINT NOT NULL IDENTITY (1,1),

Ici on a une incrémentation Qui commence à 1 avec un pas de 1

\*FOREIGN KEY : Permet d’avoir une clé étrangère dans une table qui est souvent un attribut d’une autre table

Exemple : CONSTRAINT FK\_EMP\_DEPTNO FOREIGN KEY (DEPTNO) REFERENCES DEPT (DEPTNO),

\*REFERENCES : Lié à la FOREIGN KEY qui permet de faire la référence à l’entité de l’attribut utilisé

Exemple : CONSTRAINT FK\_EMP\_DEPTNO FOREIGN KEY (DEPTNO) REFERENCES DEPT (DEPTNO),

\*INSERT INTO : Permet d’insérer des données dans une table déjà existante

Exemple : INSERT INTO PROJETPROJETNO, PROJETNAME, BUDGET)

VALUES

(101, 'ALPHA', 96000),

(102, 'BETA', 82000),

(103, 'GAMMA', 15000);

Dans la table projet on ajoute des données dans PROJETNO, PROJETNAME, BUDGET

\*DISTINCT : Permet de lister une seule occurrence d’un attribut

Exemple : SELECT DISTINCT JOB FROM EMP;

\*COUNT(…) : Permet de faire un compteur d’un attribut, le fonctionnement est le même pour AVG(), SUM(), MIN(), MAX()

Exemple : SELECT EMP.DEPTNO, JOB, SUM(SAL) AS SUMSAL, COUNT(ENAME) AS NUMBEMP, AVG(SAL) AS AVERAGE, SAL FROM EMP

CHECK :

FORMAT : Permet de définir un format de la date

Exemple : FORMAT (HIREDATE, 'ddd dd MMM yyyy','FR-fr')

Dans ce cas le format est : ven 28 fev 1997

CAST : Permet de transformer le type de donnée en un autre type

Entité CLIENTS

* id : BIGINT
* prenom : VARCHAR
* nom : VARCHAR
* date\_ajout : DATETIME
* budget : FLOAT
* num\_tel :VARCHAR(10)

Exemple : SELECT id, prenom, nom, CAST( date\_ajout AS DATE ) AS date\_ajout\_cast, budget, num\_tel

FROM client

Dans ce cas on change le type DATETIME en DATE on garde plus l’heure

CONVERT : Semblable au CAST() il permet de changer le type de donnée ( INT en FLOAT ou un DATE en DATETIME)

Exemple : SELECT CONVERT(int, 52.79);

-- résultat : 52

\*NOT : Le NOT seul est une négation comme le NOT NULL

Exemple : DEPTNO TINYINT NOT NULL,

Ici l’attribut est obligatoire

\*IN : Lié à WHERE qui permet de vérifier une valeur est égale à une autre et l’afficher

Exemple : SELECT \*

FROM adresse

WHERE addr\_ville IN ( 'Paris', 'Graimbouville');

Ici on affiche les villes cité dans le IN

DELETE : commande associé à WHERE il permet de supprimer 1 ou plusieurs lignes

Exemple : DELETE FROM `table`

WHERE condition

Ici on veut supprimer dans la table via le FROM et dans le WHERE on choisit les lignes (attributs)

\*UPDATE : Mise à jour des données

Exemple : SELECT EMP.PROJETNO, JOB, EMP.DEPTNO

FROM EMP

INNER JOIN PROJET

ON EMP.PROJETNO = PROJET.PROJETNO

UPDATE EMP

SET PROJETNO = 101

WHERE DEPTNO = 30 AND JOB = 'SALESMAN'

UPDATE EMP

SET PROJETNO = 102

WHERE DEPTNO <> 30 AND JOB <> 'SALESMAN';

Ici on ajoute des données dans l’attribut PROJETNO dans l’entité EMP

UPDATE CASCADE :