

SQL : STRUCTURE DE LA BD

⇒ **RELATIONS**

➤ **Création**

```
CREATE TABLE nomRelation  
    (nomAtt1 domaine [contrainte]  
    [ ..., nomAttN domaine [contrainte] ]  
    [ , contrainte ] )
```

Domaine : Ensemble des valeurs possibles

Types courants : CHAR(n), VARCHAR(n), NUMERIC(n[+m,m]), DATE, LONG

Contraintes : Règles que doivent suivre les données. Nommer les contraintes.
CONSTRAINT nomC lacontrainte

clé primaire : nomAtt_j PRIMARY KEY ou
PRIMARY KEY (listeAtt) en fin de relation si clé composée

clé étrangère : [FOREIGN KEY (nomAtt_j)] REFERENCES nomRelation_h [(nomAtt_h)]

valeur obligatoire : NOT NULL

domaine : CHECK (condition)

unicité : UNIQUE

Des contraintes contrôlant les changements de valeur peuvent être définies à l'aide de déclencheurs

➤ **Suppression**

```
DROP TABLE nomRelation
```

➤ **Modification**

```
ALTER TABLE nomRelation  
    ADD (nomAttj domaine [contrainte])  
    ADD / DROP CONSTRAINT nomC contrainte  
    MODIFY (nomAttj domaine et/ou contrainte)  
    DROP COLUMN nomAttj
```

⇒ **DES VUES & DES INDEX**

Vues : CREATE VIEW nomVue_j [(liste_att)]
AS SELECT ...

Index : CREATE [UNIQUE] INDEX nomIndex_j
ON nomRelation (nomAtt_j | liste_att)

SQL : AFFECTATION / REVOCATION DE DROITS

⇒ Des privilèges sur des objets de la base de données

GRANT SELECT | DELETE | INSERT [(liste_att)] | UPDATE [(liste_att)] ON objet TO user
REVOKE ... ON objet FROM user

⇒ Des privilèges système

GRANT / REVOKE CONNECT | RESOURCE | DBA TO / FROM user

SQL : CONSULTATION ET MISE A JOUR

➤ Insertion d'un n-uplet dans une relation

INSERT INTO nomRelation [(nomAtt_i..., nomAtt_N)]
VALUES (.. , .. , ..)

Autre forme syntaxique : INSERT INTO ... SELECT ...

➤ Suppression de n-uplets

DELETE FROM nomRelation
[WHERE condition]

➤ Modification de n-uplets

UPDATE nomRelation
SET nomAtt_i = val_i [, ..., nomAtt_N = val_N]
[WHERE condition]

➤ Consultation de n-uplets

SELECT [DISTINCT] expr | liste_expr
FROM nomRelation [alias] | liste_rel [listealias]
[WHERE condition]
[GROUP BY expr
[HAVING condition]]
[UNION | MINUS | INTERSECT SELECT ...]
[ORDER BY expr | liste_expr [ASC | DESC]]

expr désigne un attribut ou un calcul sur cet attribut voire une constante

agrégats : COUNT(), SUM(), MIN(), MAX(), AVG()

Opérateurs utilisés dans les conditions :

[NOT] LIKE avec % et _ , [NOT] IN, BETWEEN .. AND .., <> , = , < , > , >= , <=
ANY / ALL un_opérateur (sous-requête) [NOT] EXISTS (sous-requête)

Syntaxe des jointures (SQL2)

Jointure interne :
SELECT ... FROM nomRel₁
JOIN nomRel₂
ON condition_de_jointure
[WHERE condition] ...

Jointure externe :
SELECT ... FROM nomRel₁
LEFT | RIGTH | FULL OUTER JOIN nomRel₂
ON condition_de_jointure
[WHERE condition]