

Les contraintes

Les contraintes permettent de limiter le type de données à insérer dans une table. Ces contraintes peuvent être spécifiées lors de la création de la table par l'intermédiaire de l'instruction CREATE TABLE, ou après la création de la table au moyen de l'instruction ALTER TABLE.

Les contraintes les plus communes sont :

- NOT NULL
- UNIQUE
- CHECK
- Clé primaire
- Clé étrangère

NOT NULL

Une colonne accepte par défaut une valeur NULL. Pour rejeter une valeur NULL dans une colonne, vous devez définir une contrainte sur cette colonne en indiquant que celle-ci n'accepte pas de valeur NULL.

Par exemple, dans l'instruction suivante

```
CREATE TABLE Customer (  
    SID integer NOT NULL,  
    Last_Name varchar (30) NOT NULL,  
    First_Name varchar(30)  
);
```

Les colonnes "SID" et "Last_Name" n'accepte pas de valeur NULL, alors que "First_Name" peut en accepter.

UNIQUE

La contrainte UNIQUE garantit que toutes les valeurs d'une colonne ne peuvent comporter de doublons.

Par exemple, dans l'instruction suivante

```
CREATE TABLE Customer (  
    SID integer Unique,  
    Last_Name varchar (30),  
    First_Name varchar(30)  
);
```

La colonne "SID" n'accepte pas de doublons, alors qu'une telle contrainte ne s'applique pas aux colonnes "Last_Name" et "First_Name".

Notez qu'une colonne qui est spécifiée en tant que clé primaire ne peut accepter de doublons. En même temps, une colonne qui n'accepte pas de doublons peut ou ne pas être une clé primaire.

CHECK

La contrainte CHECK permet que toutes les valeurs d'une colonne satisfassent des conditions données.

Par exemple, dans l'instruction suivante :

```
CREATE TABLE Customer (  
    SID integer CHECK (SID > 0),  
    Last_Name varchar (30),  
    First_Name varchar(30)  
);
```

La colonne "SID" ne peut accepter que des entiers supérieurs à 0.

Clé primaire

La clé primaire concourt à identifier uniquement chaque ligne d'une table. Elle peut représenter une partie d'un enregistrement concret, ou être un champ artificiel (un champ qui n'a rien à voir avec l'enregistrement réel). La clé primaire peut représenter un ou plusieurs champs d'une table. Lorsque la clé primaire représente plusieurs champs, elle est appelée « clé composite ».

Il est possible de spécifier les clés primaires au moment de la création de la table (à l'aide de CREATE TABLE) ou de la modification de la structure de la table existante (par le biais de ALTER TABLE).

```
CREATE TABLE Customer (  
    SID integer PRIMARY KEY,  
    Last_Name varchar(30),  
    First_Name varchar(30)  
);
```

Ou encore

```
ALTER TABLE Customer ADD PRIMARY KEY (SID);
```

Remarque : avant d'utiliser la commande ALTER TABLE pour ajouter une clé primaire, vous devez vous assurer que le champ est défini en tant que 'NOT NULL' ; en d'autres mots, ce champ n'accepte pas de valeur NULL.

Clé étrangère

La clé étrangère représente un champ (ou des champs) qui pointe vers la clé primaire d'une autre table. L'objectif de la clé étrangère est d'assurer l'intégrité référentielle des données. En d'autres mots, seules les valeurs devant apparaître dans la base de données sont permises.

Par exemple, nous avons deux tables, l'une appelée CUSTOMER qui inclut toutes les données du client, et l'autre ORDERS qui comprend ses commandes. La contrainte ici est que toutes les commandes doivent être associées à un client qui se trouve déjà référencé dans la table CUSTOMER. Dans ce cas, une clé étrangère devra être placée sur la table ORDERS et mise en relation avec la clé primaire de la table CUSTOMER. De cette façon, il sera possible d'assurer que toutes les commandes de la table ORDERS sont mises en relation avec un client de la table CUSTOMER. En d'autres mots, la table ORDERS ne peut contenir d'informations sur un client qui ne se trouve pas dans la table CUSTOMER.

La structure de ces deux tables sera comme suit :

Table **CUSTOMER**

nom de colonne	caractéristique
SID	Clé primaire
Last_Name	
First_Name	

Table **ORDERS**

nom de colonne	caractéristique
Order_ID	Clé primaire
Order_Date	
Customer_SID	Clé étrangère
Amount	

Dans l'exemple ci-dessus, la colonne Customer_SID de la table ORDERS représente une clé étrangère pointant vers la colonne SID de la table CUSTOMER.

Dans l'exemple ci-dessus, la colonne Customer_SID de la table ORDERS représente une clé étrangère pointant vers la colonne SID de la table CUSTOMER.

```
CREATE TABLE ORDERS (  
    Order_ID integer primary key,  
    Order_Date date,  
    Customer_SID integer references CUSTOMER(SID),  
    Amount double  
);
```

ou en modifiant

```
ALTER TABLE ORDERS ADD (CONSTRAINT fk_orders1) FOREIGN KEY  
(customer_sid) REFERENCES CUSTOMER(SID);
```