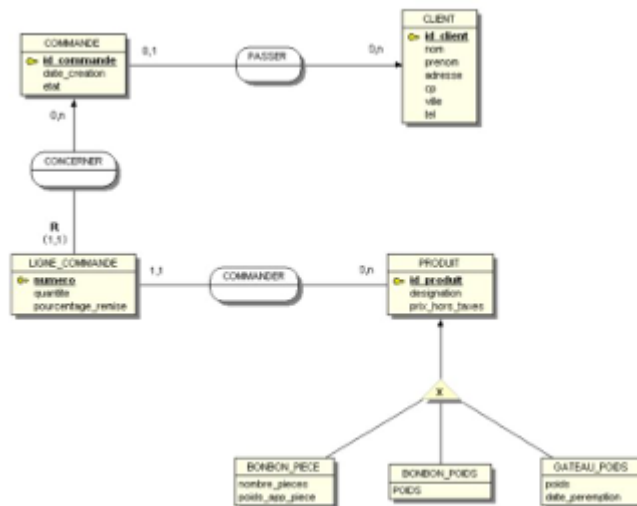


## TP Sucrerie

### Objectif du Script

Le script définit une base de données appelée *Sucrerie*, destinée à gérer une boutique de bonbons et de pâtisseries. Il permet de stocker des informations sur les clients, les commandes, les produits et applique des règles pour différencier les types de produits.



1. **Base de données Sucrerie** La base contient des tables pour gérer les clients, les commandes et les produits.

### 2. Tables

#### ○ Client :

- **id\_client**: Identifiant unique (clé primaire)
- **nom, prenom**: Nom et prénom du client
- **adresse, cp, ville, tel**: Informations de contact

```
create table Client(  
    id_client int ,  
    nom varchar(250) ,  
    prenom varchar(250) ,  
    adresse varchar(250) ,  
    cp int ,  
    ville varchar(250) ,  
    tel int,  
    constraint pk_client primary key (id_client)  
)engine = innodb ;
```

■

#### ○ Commande :

- **id\_commande**: Identifiant unique (clé primaire)
- **date\_creation, etat**: Informations sur la commande

- **id\_client**: Référence au client ayant passé la commande

```
create table commande(
    id_commande int ,
    date_creation date,
    etat int ,
    id_client int ,
    constraint pk_commande primary key (id_commande),
    constraint fk_commande_client FOREIGN KEY (id_client) REFERENCES Client(id_client)
);
```

○ **Ligne\_Commande :**

- **id\_commande**: Référence à la commande
- **id\_produit**: Référence au produit
- **quantite, pourcentage\_remise**: Détails de la commande

```
create table ligne_commande (
    numero int ,
    id_commande int ,
    id_produit int ,
    quantite int,
    pourcentage_remise float,
    constraint pk_ligne_commande primary key(numero , id_commande) ,
    constraint fk_id_commande_ligne foreign key (id_commande) references commande(id_commande),
    constraint fk_ligne_produit foreign key (id_produit) references produit(id_produit)
);
```

○ **Produit :**

- **id\_produit**: Identifiant unique (clé primaire)
- **designation**: Nom ou description
- **prix\_hors\_taxe**: Prix du produit

```
create table produit (
    id_produit int ,
    designation varchar(250) ,
    prix_hors_taxe float,
    constraint pk_produit primary key (id_produit)
);
```

■

○ **Bonbon\_Piece :**

- **id\_produit**: Référence au produit
- **nombre\_pieces, poids\_app**: Détails du produit vendu à l'unité

○ **Bonbon\_Poids :**

- Détails des bonbons vendus au poids.

○ **Gâteau\_Poids :**

- **id\_produit**: Référence au produit
- **poids, date\_peremption**: Détails des gâteaux vendus au poids

3. **Triggers** Les triggers assurent qu'un produit ne peut appartenir qu'à une seule catégorie (bonbon\_piece, bonbon\_poids, gateau\_poids).

- **partitionBonbonPiece** : Vérifie qu'un produit n'existe pas déjà dans **bonbon\_poids** ou **gateau\_poids**.

- **partitionGâteau** : Vérifie qu'un produit n'existe pas déjà dans **bonbon\_piece** ou **bonbon\_poids**
- **partitionPoids** : Vérifie qu'un produit n'existe pas déjà dans **bonbon\_piece** ou **gâteau\_poids**.

```
create table bonbon_piece (  
  id_produit int ,  
  nombre_pieces int ,  
  poids_app float ,  
  constraint pk_bonbon_piece primary key (id_produit) ,  
  constraint fk_bonbon_produit foreign key (id_produit) references produit(id_produit)  
);  
  
create table bonbon_poids (  
  id_produit int ,  
  poids float,  
  constraint pk_bonbon_piece primary key (id_produit) ,  
  constraint fk_poids_produit foreign key (id_produit) references produit(id_produit)  
);  
  
create table gâteau_poids (  
  id_produit int ,  
  poids float ,  
  date_peremption date ,  
  constraint pk_bonbon_piece primary key (id_produit) ,  
  constraint fk_gâteau_produit foreign key (id_produit) references produit(id_produit)  
);
```