1-Clients (code\_client, nom, prénom, catégorie) Produits (code\_produit, désignation, prix) Factures (num\_facture, code\_client, date\_facture) Lignes\_Factures (num\_facture, code\_produit, quantité, prix\_unitaire)

1. Scripts de création de la base de don

CRÉER UNE BASE DE DONNÉES SeneMarket ; ALLER

UTILISER SeneMarket ; ALLER

CREATE TABLE Clients ( code\_client INT PRIMARY KEY, nom VARCHAR(50), prénom VARCHAR(50), catégorie VARCHAR(50) );

CREATE TABLE Produits ( code\_produit INT PRIMARY KEY, désignation VARCHAR(50), prix DECIMAL(10, 2) );

CREATE TABLE Factures ( num\_facture INT PRIMARY KEY, code\_client INT, date\_facture DATE, FOREIGN KEY (code\_client) REFERENCES Clients(code\_client) );

CREATE TABLE Lignes\_Factures ( num\_facture INT, code\_produit INT, quantité INT, prix\_unitaire DECIMAL(10, 2), FOREIGN KEY (num\_facture) REFERENCES Factures(num\_facture), FOREIGN KEY (code\_produit) REFERENCES Produits(code\_produit) );

1. Insertion de données sur les tables :

INSÉRER DANS LES VALEURS DES CLIENTS (1, 'Doe', 'John', 'VIP'); INSÉRER DANS LES VALEURS DES CLIENTS (2, 'Smith', 'Mary', 'Ordinaire'); INSÉRER DANS LES VALEURS DES CLIENTS (3, 'Jones', 'David', 'Potentiel');

INSÉRER DANS LES VALEURS Produits (1, 'Ordinateur', 500); INSÉRER DANS LES VALEURS Produits (2, 'Imprimante', 100); INSÉRER DANS LES VALEURS Produits (3, 'Souris', 15);

INSÉRER DANS LES VALEURS Factures (1001, 1, '2023-01-01'); INSÉRER DANS LES VALEURS Factures (1002, 2, '2023-02-01'); INSÉRER DANS LES VALEURS Factures (1003, 3, '2023-03-01');

INSÉRER DANS LES VALEURS Lignes\_Factures (1001, 1, 1, 500, 1) ; INSÉRER DANS LES VALEURS Lignes\_Factures (1001, 2, 2, 100, 2) ; INSÉRER DANS LES VALEURS Lignes\_Factures (1002, 2, 1, 500, 1) ; INSÉRER DANS LES VALEURS Lignes\_Factures (1002, 3, 5, 15, 1) ; INSÉRER DANS LES VALEURS Lignes\_Factures (1003, 3, 10, 15, 2) ;

1. Requête permettant d'afficher toutes les factures, avec pour chacune : le numéro de la facture, les nom et prénom du client, la désignation et le prix du produit, et la quant

SELECT f.num\_facture, c.nom, c.prénom, p.désignation, p.prix, lf.quantité FROM Factures f JOIN Clients c ON f.code\_client = c.code\_client JOIN Lignes\_Factures lf ON f.num\_facture = lf.num\_facture JOIN Produits p ON lf.code\_produit = p.code\_produit;

1. Requête permettant de calculer le chiffre d'affaires, c'est-à-dire le nombre de produits achetés multipliés par leur prix, pour chaque fait

SELECT f.num\_facture, SUM(p.prix \* lf

1. Demandes pour la catégorisation des clients et la création de la

un. Requête permettant de renvoyer la liste des clients avec leur chiffre d'affaire et leur catégorie :

SELECT c.code\_client, c.nom, c.prénom,

SUM(p.prix \* lf.quantité) comme chiffre\_affaire,

CASE WHEN SUM(p.prix \* lf.quantité) > 500 ALORS 'VIP' QUAND SUM(p.prix \* lf.quantité) ENTRE 50 ET 500 PUIS 'Ordinaire' AUTRE 'Potentiel' FIN comme catégorie FROM Clients c JOIN Factures f ON c.code\_client = f.code\_client JOIN Lignes\_Factures lf ON f.num\_facture = lf.num\_facture JOIN Produits p ON lf .code\_produit = p.code\_produit GROUPE PAR c.code\_client, c.nom, c.prénom;

b. Création de la vue :

CREATE VIEW v\_chiffre\_affaire AS SELECT c.code\_client, c.nom, c.prénom, SUM(p.prix \* lf.quantité) as chiffre\_affaire, CASE WHEN SUM(p.prix \* lf.quantité) > 500 ALORS 'VIP' QUAND SUM (p.prix \* lf.quantité) ENTRE 50 ET 500 PUIS 'Ordinaire' AUTRE 'Potentiel' FIN comme catégorie FROM Clients c JOIN Factures f ON c.code\_client = f.code\_client JOIN Lignes\_Factures lf ON f.num\_facture = lf.num\_facture JOIN Produits p ON lf.code\_produit = p.code\_produit GROUP BY c.code\_client, c.nom, c.prénom;

c. Demande pour afficher la catégorie d'un client ayant pour nom Morin :

SELECT catégorie FROM v\_chiffre\_affaire WHERE nom = 'Morin';

1. Demandes pour l'inventaire des produits :

un. Fonction stockée pour retourner la demande en fonction du numéro de produit :

CRÉER UNE FONCTION dbo.get\_demande (@code\_produit INT) RETOURNE VARCHAR(10) AS BEGIN DECLARE @demande VARCHAR(10) SELECT @demande = CASE WHEN SUM(quantité) > 15 ALORS 'forte' QUAND SUM(quantité) ENTRE 11 ET 15 ALORS 'moyenne' ELSE 'faible' FIN FROM Lignes\_Factures lf JOIN Produits p ON lf.code\_produit = p.code\_produit OÙ p.code\_produit = @code\_produit RETOUR @demande FIN;

b. Demande pour présenter la liste des produits avec la demande :

SELECT code\_produit, désignation, prix, dbo.get\_demande(code\_produit) selon la demande FROM Produits;