Informatique - ECG 2a - mathématiques 2 au 13 novembre 2026

Bases de données - TP 1 (2 séances)

Code de partage avec Capytale : a192-7980124

Voir en annexe pour les commandes initiales permettant de créer la base de données (création et remplissage des tables).

Travail préparatoire

Sur une feuille, représenter le schéma relationnel de cette base de données (on indiquera les tables, les noms des différents attributs, on identifiera les clés primaires et étrangères et on mettra en évidence les liens entre les tables).

Exercice 1- compléter la base de données

Nous allons créer des requêtes pour ajouter des informations relatives à une nouvelle commande :

Nouveau client :

▷ Nathan DUCHEMIN
▷ 2 square de la Raquette 49500 Nyoiseau 02 44 27 13 12

Nouvelle commande passée le 20 novembre à 20h30

▷ 2 Pizzas Régina
▷ 1 Pizza Calzonne

La commande sera livrée par Tudal

1. Pour déterminer la valeur maximale pour l'`Id` de la table `Clients`, on exécute :

   ```
   SELECT MAX(Id) FROM Clients
   ```

2. Pour ajouter un client dans la table `Clients`, on exécute :

   ```
   INSERT INTO Clients ( ‘Id‘, ‘Nom‘, ‘Prenom‘, ‘Adresse‘, ‘CodePoste‘, ‘Ville‘, ‘Tel‘)
   VALUES ( (SELECT MAX(Id)+1 FROM Clients), "DUCHEMIN", "Nathan", "2 square de la
   Raquette", "49500", "Nyoiseau", "02 44 27 13 12")
   ```

3. Sur le même modèle, ajouter la commande dans la table `Commandes`
4. Sur le même modèle, ajouter la commande dans la table `Preparer`

Exercice 2

Donner les requêtes qui permettront d'obtenir les informations recherchées :

1. afficher l'intégralité de la table `Commandes`
2. afficher tous les Id des différentes pizzas commandées dans la commande n°2
3. à l'aide d'une jointure, afficher l'`Id`, le nom, la quantité et le prix de chaque pizza commandée dans la commande n°2

   Les opérations arithmétiques usuelles pour les nombres : `+`, `-`, `*`, `/`, `MIN`, `MAX`,... sont naturelles avec SQL.

4. en déduire la requête qui permettra de calculer le montant de la commande n°2

Exercice 3

1. Afficher toutes les dates de toutes les commandes
2. Afficher combien de pizza Regina ont été commandées entre deux dates définie. On pourra utiliser dans le 'WHERE' la commande 'BETWEEN' ... 'AND' ...
3. Afficher toutes les pizzas commandées par Paul BAS, puis le nombre total de chacune des pizzas qu'il a commandées.
   On pourra utiliser la commande 'GROUP BY' en fin de requête qui permet de regouper selon un champ (ici Pizzas.Nom, à défaut on peut faire avec le IdPizza de la table `Preparer`).
4. Afficher le nom de la pizza la plus commandée par les clients
5. Paul DUCHEMIN a déménagé, il faut mettre à jour sa fiche client :
   ▷ Paul DUCHEMIN
   ▷ 18 rue de la Marquetterie 49520 Noyant-La-Gravoyère
6. Afficher le nombre de pizzas « Regina » qui ont été commandées
7. La pizzeria a un problème d'intoxication alimentaire avec la pizza « Jaurais Pas Du ». Afficher le nom et le numéro de téléphone de tous les clients qui ont commandé une pizza « Jaurais Pas Du ».

Exercice 4

La pizzeria souhaite mettre en place une carte de fidélité avec l'acquisition d'un point par tranche de 10 euros pour chaque commande de pizza.

1. Modifier le modèle relationnel pour insérer une table `Points` qui permette de mémoriser le nombre de points pour chaque client.
2. Afficher le montant total des commandes effectuées par chaque client au mois de novembre.
3. En déduire chaque requête pour ajouter le nombre de points à chaque client pour le mois de novembre.
4. Afficher le client ou la cliente qui a le plus de points.
5. Afficher le nombres de points au total pour toute la clientèle.
6. Afficher le nombre de pizzas offertes si tous les clients et clientes utilisent leur carte de fidélité fin novembre.

Annexe - création et remplissage des tables

Pour commencer, on exécute les deux séries de commandes ci-dessous, respectivement, pour créer les tables et les remplir.

On exécute les requêtes suivantes pour créer les tables.

```
CREATE TABLE 'Clients' ('Id' INTEGER PRIMARY KEY NOT NULL, 'Nom' TEXT, 'Prenom' TEXT,
'Adresse' TEXT, 'CodePoste' TEXT, 'Ville' TEXT, 'Tel' TEXT);

CREATE TABLE 'Livreurs' ('Id' INTEGER PRIMARY KEY NOT NULL, 'Nom' TEXT);
```

```
CREATE TABLE ‘Pizzas‘ (‘Id‘ INTEGER PRIMARY KEY NOT NULL, ‘Nom‘ TEXT, ‘PU‘ FLOAT);

CREATE TABLE ‘Commandes‘ (‘Num‘ INTEGER PRIMARY KEY NOT NULL, ‘Date‘ date, ‘Heure‘ time,
‘IdClient‘ INTEGER, ‘IdLivreur‘ INTEGER, FOREIGN KEY (‘IdClient‘) REFERENCES Clients(‘Id‘)
FOREIGN KEY (‘IdLivreur‘) REFERENCES Livreurs(‘Id‘) );

CREATE TABLE ‘Preparer‘ (‘IdCmde‘ INTEGER NOT NULL, ‘IdPizza‘ INTEGER, ‘Qte‘ INTEGER,
FOREIGN KEY (‘IdCmde‘) REFERENCES Commandes(‘Num‘), FOREIGN KEY (‘IdPizza‘) REFERENCES
Pizzas(‘Id‘));
```

On exécute les requêtes suivantes pour remplir les tables qui ont été créées.

```
INSERT INTO ‘Pizzas‘ (‘Id‘, ‘Nom‘, ‘PU‘) VALUES
( 0, "Regina", 7.50),
( 1, "Quatre fromages", 8.50),
( 2, "Margarita", 8.00),
( 3, "Arrabiata", 8.50),
( 4, "Jaurais Pas Du", 9.50),
( 5, "Calzonne", 10.00),
( 6, "Speciale chef", 15.00),
( 7, "Farsus", 12.00) ;

INSERT INTO ‘Livreurs‘ ( ‘Id‘, ‘Nom‘ ) VALUES
( 0, "Klervi"),
( 1, "Gwendal"),
( 2, "Tudal"),
( 3, "Mona") ;

INSERT INTO ‘Clients‘ ( ‘Id‘, ‘Nom‘, ‘Prenom‘, ‘Adresse‘, ‘CodePoste‘, ‘Ville‘, ‘Tel‘)
VALUES
(0, "BAS", "Paul", "2 rue des Fosses", "49475", "Setré", "02 41 25 12 37"),
(1, "HAUT", "Paul", "4 av. des Sommets", "49510", "Segre-En-Anjou-Bleu", "06 12 45 27
13"),
(2, "SALOMON", "Pierre", "4 chemin du Bois", "49630", "Tourlaville", "02 41 25 12 37"),
(3, "DUCHEMIN", "Paul", "4 av. des Sommets", "49500", "Segre-En-Anjou-Bleu", "06 12
45 27 13"),
(4, "BAS", "Mireille", "2 rue des Fosses", "49475", "Setré", "02 41 25 12 37"),
(5, "SALOMON", "Renée", "4 chemin du Bois", "49630", "Tourlaville", "06 12 45 27 13") ;

INSERT INTO Commandes (‘Num‘, ‘Date‘, ‘Heure‘, ‘IdClient‘, ‘IdLivreur‘) VALUES
( 0, "2023-09-05", "20:46", 2, 0 ), ( 1, "2023-09-05", "20:46", 0, 0 ),
....
- voir sur Capytale pour la suite des données
INSERT INTO Preparer (‘IdCmde‘, ‘IdPizza‘, ‘Qte‘ ) VALUES
( 0, 7, 1 ), ( 1, 0, 3 ), ( 2, 7, 1 ), ( 2, 5, 1 ),
....
- voir sur Capytale pour la suite des données
```