

Rapport de Travaux Pratiques

Base de matériaux pour la thermique



Florine TEULIERES-QUILLET

18 Novembre 2023

Dans le cadre des cours d'option données, nous avons eu l'occasion de manipuler une base données via des commande SQL. Ce travail à permis de mettre en application les notions vues en cours notamment la gestion des hierarchies et la structuration des bases de données.

Ce TP fut aussi l'occasion de découvrir le logiciel **Access** du pack Microsoft qui est une interface permettant la manipulation facile de base de données notamment via les commande SQL.

1 Introduction

Durant cette manipulation, nous allons exploiter une véritable base de données matériau.mdb dont l'objectif est l'aide à la conception et la production de lanceurs basaltique. Cette base a été constituée par le département de Calculs Thermiques et recense les matériaux ou systèmes participant au contrôle thermique de ces lanceurs. Elle est constituée de deux tables: « **Caracteristiques** » listant les caractéristiques de chaque matériaux et « **Matériau** » présentant les différents matériaux dont « Nom » est une clé primaire.

Matériau			
<u>Nom</u>	Categorie	Type	Famille

Caracteristiques			
NomMat	NomConst	Valeur	Unite

2 Les premières requêtes SQL

2.1 Lister le nom et la famille de chaque matériau de la catégorie Structure et de type Métallique

La requête SQL permettant de créer cette liste est la suivante:

```
SELECT Nom, Famille
FROM Matériau
WHERE Catégorie = « Structure »
AND Type = « Métallique »
```

Cette requête permet ainsi de créer la table suivante. A noter que chaque requête est enregistré dans le logiciel comme une "VIEW" de la base de données et peut ensuite être manipuler aisément.

Nom	Famille
15CDV6	Aciers
2024	Aluminiums
2024 T 42	Aluminiums
2195 T 8	Aluminiums
25 CD 4	Aciers

2.2 Lister les masses volumiques connues de la famille des Aciers

Pour créer cette view là, nous allons avoir recours à une jointure entre les deux table, ce qui nécessite de préciser la clé commune entre les deux tables (le liens entre les deux).

```
SELECT Nom, Famille, Valeur as Masse Volumique, Unite
FROM Matériau as M, Caractéristique as C
WHERE M.Nom = C.NomMat
AND Famille = « Acier »
AND NomConst = « Rho »
```

On obtient alors la liste demandée:

Nom	famille	valeur	unite
15CDV6	Aciers	6,5	Kg/m3
15CDV6	Aciers	0,12	Kcal/kg.C
25 CD 4	Aciers	5,4	Kg/m3

2.3 Lister le nom des catégories et le nombre de matériaux recensés

Afin de réaliser cette liste, nous allons utiliser la fonction **COUNT** qui permettra de compter le nombre de matériau par catégorie, l'étoile signifie que le fonction comptera tout les éléments qui ont cette caractéristique commune sans autre distinction.

```
SELECT Nom, COUNT(*) as NombreParCate
FROM Matériau
GROUP BY Catégorie
```

On obtient donc:

Nom	famille	valeur	unite
15CDV6	Aciers	6,5	Kg/m3
15CDV6	Aciers	0,12	Kcal/kg.C
25 CD 4	Aciers	5,4	Kg/m3

Ces trois premières commande on permis de prendre en main le logiciel en créant, exécutant et enregistrant de nouvelles requêtes. Cette partie aura aussi permis l'apprentissage du langage SQL dont la théorie avait été vu en cours.

3 Restructuration de la table « Matériau »

3.1 Examen de la table « Matériau »

Dans le cadre de la restructuration de la base, nous devons exécuter la requête suivante :

```
INSERT INTO Matériau (Categorie, Type, Famille)  
VALUES (« Structure », « Métallique », « Titane »)
```

Cette requête a pour but d'ajouter une nouvelle ligne dans la table « Matériau » dont les caractéristiques sont décrites dans la requête. Cependant, lors de l'exécution de celle-ci, un message d'erreur s'affiche signifiant qu'il y a une violation de clé.

En effet, comme décrit lors de l'introduction cette table possède une clé primaire « Nom ». Or, cette clé n'a pas été renseignée dans la requête ce qui rend impossible son exécution.

En examinant de plus près la table, on peut constater que les attributs sont dépendants les uns des autres, l'ensemble de la hiérarchie est dans la même table.

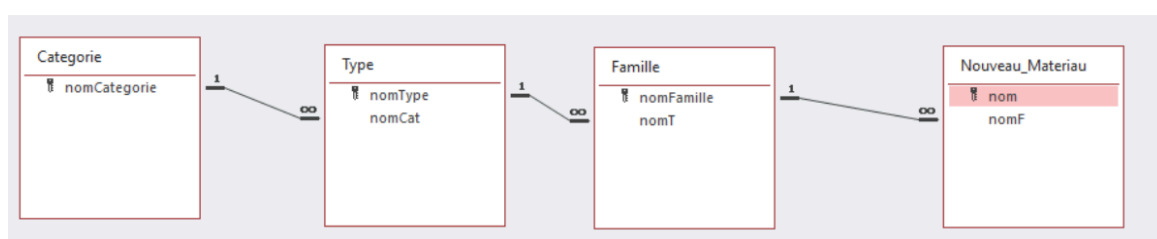
3.2 Restructuration de la base

Pour éviter les problèmes de dépendances entre les attributs, il faut restructurer la table en suivant la théorie de flocons de neige.

Quatre nouvelles tables vont donc être créées : Famille, Type, Catégorie et Nouveau_Matériau.

```
Categorie( nomCategorie );  
Type( nomType, #nomCat );  
Famille( nomFamille, #nomT );  
Nouveau_Matériau( nom, #nomF );
```

Sur le logiciel Access, il est possible d'accéder aux relations entre les tables en cliquant sur « Base de données » puis « relations » (Chaque clé primaire sera accompagnée d'un symbole clé à côté).



4 Création des nouvelles tables

4.1 Requêtes SQL pour créer les nouvelles tables

Maintenant que nous avons restructurer la base de données, il faut créer ces nouvelles tables via des requêtes SQL. Attention, dans le cadre de ce logiciel, chaque requête de création de table doit être faite séparément des autres, dans notre cas nous aurons donc 4 requêtes différentes pour créer les 4 tables. Dans ces requêtes, le mot clé « **PRIMARY KEY** » permettra de considérer l'attribut comme une clé primaire et « **FOREIGN KEY** » pour distinguer les clé étrangères en ajoutant « **REFERENCES** » pour renseigner la table dont provient la clé.

```
CREATE TABLE Categorie (nomCategorie VARCHAR(20) PRIMARY KEY);
```

```
CREATE TABLE Type (nomType VARCHAR(20) PRIMARY KEY, nomCat VARCHAR(20),  
FOREIGN KEY (nomCat) REFERENCES Categorie(nomCategorie));
```

```
CREATE TABLE Famille (nomFamille VARCHAR(20) PRIMARY KEY, nomT  
VARCHAR(20),  
FOREIGN KEY (nomT) REFERENCES Type(nomType));
```

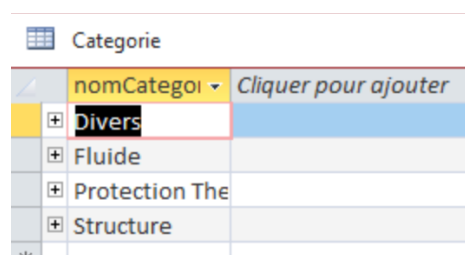
```
CREATE TABLE Nouveau_Materiau (nom VARCHAR(20) PRIMARY KEY, nomF  
VARCHAR(20), FOREIGN KEY (nomF) REFERENCES Famille(nomFamille));
```

4.2 Peuplage des tables

Une fois ces tables créées, il faut les remplir en utilisant les tables précédentes:

- Pour la table Catégorie:

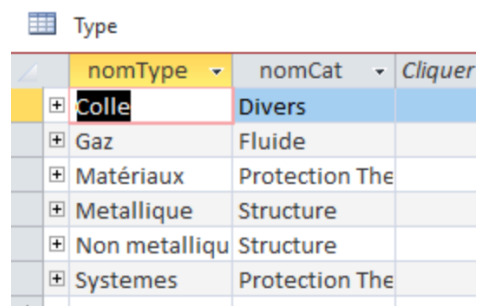
```
INSERT INTO Categorie (nomCategorie)  
SELECT DISTINCT Categorie  
FROM Materiau
```



nomCategori	Cliquer pour ajouter
Divers	
Fluide	
Protection The	
Structure	

- Pour la table Type:


```
INSERT INTO Type (nomType, nomCat)  
SELECT DISTINCT Type, Categorie  
FROM Materiau
```



nomType	nomCat	Cliquer
Colle	Divers	
Gaz	Fluide	
Matériaux	Protection The	
Metallique	Structure	
Non metalliqu	Structure	
Systemes	Protection The	

- Pour la table Famille:

INSERT INTO Famille (nomFamille, nomT)
SELECT DISTINCT Famille, Type
FROM Materiau

 Famille

	nomFamille ▾	nomT ▾	Cliquer
+	Ablatifs	Matériaux	
+	Aciers	Metallique	
+	Aluminiums	Metallique	
+	Colles	Colle	
+	Composites	Non metalliqu	
+	Gaz	Gaz	
+	Pavés	Systemes	

- Pour la table Nouveau_Materiau:


INSERT INTO Nouveau_Materiau (nom, nomF)
SELECT DISTINCT Nom, Famille
FROM Materiau

5 Reformulation des questions

5.1 Reformuler l'ajout de ligne

Il faut maintenant, ajouter la nouvelle ligne que nous souhaitons ajouter en partie 3.1. Pour cela, il suffira d'ajouter une nouvelle ligne dans la table Famille, par relation extérieures les autres tables se rempliront automatiquement.

INSERT INTO Famille (nomFamille, nomT)
VALUES (« Titane », « Metallique »)

 Famille

	nomFamille ▾	nomT ▾	Cliquer
+	Ablatifs	Matériaux	
+	Aciers	Metallique	
+	Aluminiums	Metallique	
+	Colles	Colle	
+	Composites	Non metalliqu	
+	Gaz	Gaz	
+	Pavés	Systemes	
+	Titane	Metallique	

5.2 Reformuler la première requête

On vient maintenant, reformuler la toute première requête (partie 2.1):

```
SELECT Nom, f.nomFamille
FROM Nouveau_Materiau as M, Famille as F, Type as T
WHERE M.nomF = f.nomFamille
AND F.nomT = T.nomType
AND T.nomCat = « Structure »
AND T.nomType = « Metallique »
```

Pour vérification, cette requête donne bien le même résultat qu'en première partie:

nom	nomFamille
15CDV6	Aciers
2024	Aluminiums
2024 T 42	Aluminiums
2195 T 8	Aluminiums
25 CD 4	Aciers

6 Restructuration de la table « Caractéristiques »

Comme dans la partie 3.2, on vient restructurer la table « Caractéristique », de la façon suivante:

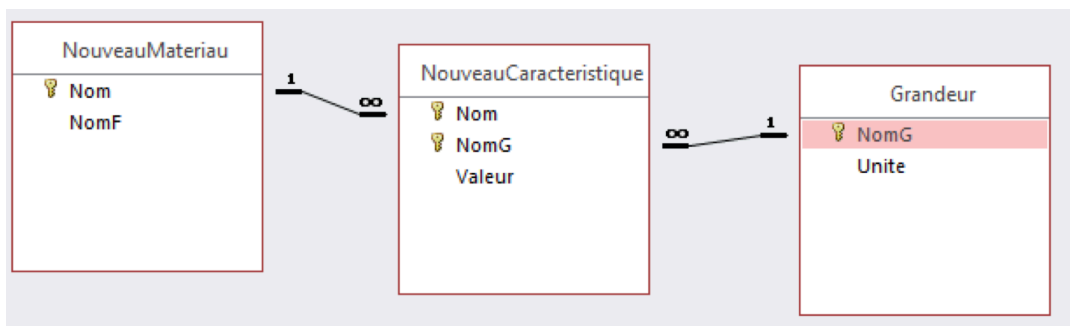
```
CREATE TABLE Grandeur (NomG VARCHAR(20) PRIMARY KEY, Unite VARCHAR(15));
```

```
CREATE TABLE Nouveau_Caracteristique (nom VARCHAR(20), NomG VARCHAR(20),  
Valeur REAL,
```

```
FOREIGN KEY (nom) REFERENCES Nouveau_Materiau(nom),
```

```
FOREIGN KEY (NomG) REFERENCES Grandeur(NomG),
```

Voici, le nouveau diagramme de relation:



7 Importation sous Excel

7.1 Peupler les nouvelles tables

```
INSERT INTO Gandeur (NomG, Unite)
SELECT DISTINCT NomConst, Unite
FROM Caractéristique
```

```
INSERT INTO NouveauCaracteristique (Nom, NomG, Valeur)
SELECT DISTINCT NomMat, NomConst, Valeur
FROM Caractéristique
```

On obtient donc les tables suivantes:

Nom	NomG	Valeur
15CDV6	Cp	0,12
15CDV6	Rho	6,5
2024	Cp	0,217
2024	CT	237
25 CD 4	Rho	5,4
3DCC	CT	12,7
3DCC	Rho	1,56
3DCC	Tfusion	2546
AETB-12	CT	0,03
AETB-12	Rho	2,3
AETB-12	Tfusion	3780

NomG	Unite
Cp	Kcal/kg.C
CT	W.m-1.K-1
Rho	Kg/m3
Tfusion	K

7.2 Réaliser un graphique de type « secteur » permettant de visualiser les données

En exportant les données sous excel, on peut obtenir des graphiques descriptifs:

