

## TP1 : Créer et peupler des tables

### BUTS PÉDAGOGIQUES

- Création d'une base de données MySQL
- Création de tables MySQL et de relations entre les tables
- Alimenter les tables
- Utilisation de l'interface d'administration phpMyAdmin

Dans ce TP nous avons utilisé la base de donnée nommée bureau\_etude pour vous la base de donnée aura un nom différent !!!

Pour le contexte du cas traité dans ce TP, nous nous projetons entre le **bureau d'étude**<sup>W</sup>, l'**approvisionnement**<sup>W</sup> et la **chaîne de production**<sup>W</sup> d'une entreprise. Le besoin est, pour fabriquer les nombreuses gammes de produits de l'entreprise, de s'approvisionner du bon nombre de composants pour faire fonctionner la chaîne de production. Pour ce faire, une base de données sera renseignée automatiquement par les logiciels de CAO du bureau d'étude. En voici le schéma relationnel :

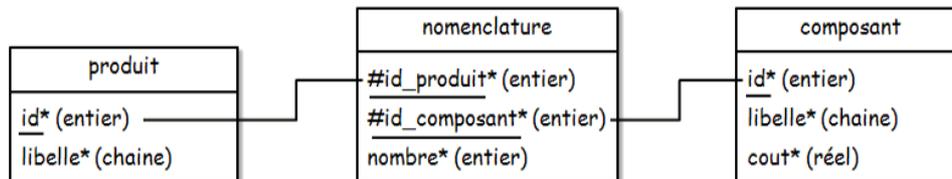


Figure 0.1 : Schéma relationnel de la base de données Bureau d'Étude.

Cette base de données avec trois tables reste très succincte comparée aux 30.000 tables de l'**ERP**<sup>W</sup> de SAP ! (L'éditeur de logiciels SAP est l'acteur qui occupe la plus grande part du marché des logiciels de gestion depuis plus de 30 ans). Mais, le but du cas présentement traité, que nous avons minimalisé à trois tables, est avant tout de vous permettre de découvrir les outils de gestion de bases de données.

## Exercice 1 • Première table

### Question 1.4 : Créer une nouvelle table

1. Toujours dans l'arborescence des bases (à gauche), sélectionnez la base de données `bureau_etude`
2. Donnez le nom de cette nouvelle table **produit**, le nombre de **2** colonnes et cliquez sur le bouton **Créer**
3. Choisissez le moteur de stockage **InnoDB**
4. Saisissez les paramètres des deux colonnes :
  - Colonne 1 : **Nom** = id, **type** = INT, **index** = PRIMARY, **A\_I (AUTO\_INCREMENT)** =
  - Colonne 2 : **Nom** = libelle, **type** = VARCHAR, **Taille/Valeurs**=50
5. Cliquez sur le bouton **Sauvegarder**
6. Vérifiez dans l'arborescence des bases que la nouvelle table `produit` apparaît au-dessous de la base de données `bureau_etude`
7. Toujours dans l'arborescence, cliquez sur la base de données `bureau_etude`, puis sur l'onglet **Concepteur** et visualisez graphiquement la table que vous venez de créer :



Figure 1.4.1 : Accès immédiat à l'onglet Concepteur, ou en déroulant le menu « plus ».

8. Organisez visuellement cette vue en déplaçant les tables, puis conservez le placement en enregistrant le placement avec le bouton  et en lui donnant un nom :



Figure 1.4.2 : La boîte de dialogue pour donner un nom à la vue.



**ATTENTION (ON NE LE RÉPÉTERA JAMAIS ASSEZ)** Pour les noms des **tables**, des **colonnes**, comme pour les noms des **bases** de données, NE JAMAIS utiliser de caractères accentués, de caractères spéciaux ou des espaces. Utilisez uniquement les caractères alphabétiques a-z, les chiffres 0-9, ou le caractère souligné **\_** (*underscore*, sur la touche 8 du clavier). Ainsi, tout se passera bien. Sinon, bon courage !

En fait, cette remarque est valable pour tous les noms que les programmeurs donnent aux variables, fichiers, répertoires, serveurs, programmes, .... Vous devrez donc prolonger cette bonne habitude tout au long de vos enseignements en informatique à l'ESTIA.



**L'onglet concepteur est absent ?** Le concepteur est rarement installé par défaut. Mais il a déjà été installé sur vos configurations ESTIA. Si tel n'est pas le cas, il est possible de l'installer sur votre serveur WAMP en suivant cette [procédure](#) <sup>Q</sup>.

## Exercice 2 • Deuxième table



L'objectif est d'ajouter une nouvelle table `composant` dans la base de données `bureau_etude`

1. Toujours dans l'arborescence des bases (à gauche), sélectionnez la base de données `bureau_etude`
2. En dessous du nom de la base `bureau_etude` cliquez sur Nouvelle table
3. Choisissez le moteur de stockage **InnoDB**
4. Donnez le nom de cette nouvelle table **composant** et saisissez les paramètres des trois colonnes :
  - Colonne 1 : **Nom** = id, **type** = **INT**, index = PRIMARY, **A\_I (AUTO\_INCREMENT)** =
  - Colonne 2 : **Nom** = libelle, **type** = **VARCHAR**, **Taille/Valeurs**= 50
  - Colonne 3 : **Nom** = cout, **type** = **DECIMAL**, **Taille/Valeurs**=10,3
  - *Remarque* : Inutile de saisir la quatrième colonne, laissez-la en l'état, elles seront ignorées
5. Cliquez sur le bouton **Sauvegarder**
6. Vérifiez dans l'arborescence des bases que la nouvelle table `composant` apparaît au-dessous de la base de données `bureau_etude`
  - Vérifiez également la structure de cette nouvelle table `composant` :
  - Si jamais le DECIMAL a mal fonctionné (et que vous obtenez par exemple 10,0), exécutez la requête suivante (onglet SQL) pour rectifier :  

```
ALTER TABLE `composant` CHANGE `cout` `cout` DECIMAL(10,3) NOT NULL;
```
7. Toujours dans l'arborescence, cliquez sur la base de données `bureau_etude`, puis sur l'onglet **Concepteur** et visualisez graphiquement les tables que vous venez de créer.

## Exercice 3 • Troisième table



L'objectif est d'ajouter une nouvelle table `nomenclature` dans la base de données `bureau_etude`

1. Créer une nouvelle table **nomenclature** puis, en n'oubliant pas de choisir le moteur de stockage **InnoDB**, indiquez les trois colonnes suivantes :
  - Colonne 1 : **Nom** = id\_produit, **type** = INT, **index** = PRIMARY
  - Colonne 2 : **Nom** = id\_composant, **type** = INT, **index** = PRIMARY
  - Colonne 3 : **Nom** = nombre, **type** = INT
  - *Remarque* : Inutile de saisir la quatrième colonne, laissez-la en l'état, elles seront ignorées
2. Cliquez sur la base de données `bureau_etude`, puis sur l'onglet **Concepteur** et vérifiez la présence des trois tables.

## Exercice 4 • Peupler la base



Nous allons alimenter la base de données `bureau_etude` avec des produits, avec des composants, renseigner la nomenclature, puis faire quelques observations.

### Question 4.1 : Les produits

1. Dans l'arborescence, cliquez sur la table `produit` puis ouvrez l'onglet **Insérer**
2. Donnez les libellés des deux produits (inutile de saisir le champ `id`) :
  - `libelle` = Chaise
  - `libelle` = Banc
3. Cliquez sur le dernier bouton **Exécuter** tout en bas du formulaire
4. Ouvrez maintenant l'onglet **Parcourir**
5. Observez les lignes de la table. Quelles sont les valeurs des `id` des deux produits ?

### Question 4.2 : Les composants

1. Dans l'arborescence, cliquez sur la table `composant` puis ouvrez l'onglet **Insérer**
2. Donnez les libellés des six composants (inutile de saisir le champ `id`) :
  - `libelle` = Vis, `cout` = 0.05
  - `libelle` = Pied court, `cout` = 3.5
  - `libelle` = Dossier court, `cout` = 15
  - `libelle` = Dossier long, `cout` = 45
  - `libelle` = Assise courte, `cout` = 20
  - `libelle` = Assise longue, `cout` = 50
3. Cliquez sur le dernier bouton **Exécuter** tout en bas du formulaire
4. Ouvrez maintenant l'onglet **Parcourir**
5. Vérifiez que les six composants sont bien présents et observez la valeur des `id`

### Question 4.3 : Les nomenclatures

1. Dans l'arborescence, cliquez sur la table `nomenclature` puis ouvrez l'onglet **Insérer**
2. En bas du formulaire, procédez avec Continuer l'insertion avec 10 lignes
3. Donnez les valeurs suivantes :
  - `id_produit` = 1, `id_composant` = 1, `nombre` = 6
  - `id_produit` = 1, `id_composant` = 2, `nombre` = 4
  - `id_produit` = 1, `id_composant` = 3, `nombre` = 1
  - `id_produit` = 1, `id_composant` = 5, `nombre` = 1
  - `id_produit` = 2, `id_composant` = 1, `nombre` = 8
  - `id_produit` = 2, `id_composant` = 2, `nombre` = 6
  - `id_produit` = 2, `id_composant` = 4, `nombre` = 1
  - `id_produit` = 2, `id_composant` = 6, `nombre` = 1
4. Inutile de saisir les autres lignes, laissez-les en l'état, elles seront ignorées
5. Cliquez sur le dernier bouton **Exécuter** tout en bas du formulaire
6. Ouvrez maintenant l'onglet **Parcourir**
  - Vérifiez que les huit lignes de nomenclatures sont présentes dans la table. Que cela représente-t-il ?
  - Nous savons ainsi que le produit chaise de l'entreprise a 4 pieds courts et que le produit banc en a 6. Combien de vis sont nécessaires pour assembler une chaise ? Et un banc ?

## Exercice 5 • Dépeupler

### Question 5.1 : Dépeuplement et incohérence

1. Dans l'arborescence, cliquez sur la table `composant` (l'onglet **Parcourir** apparaît par défaut du moment que des lignes existent)
2. Avec le bouton  Effacer, supprimez le composant Assise longue
3. Cliquez sur l'onglet **Parcourir** et vérifiez que les seuls composants restant sont Vis, Pied court, Dossier court, Dossier long et Assise courte.
4. Dans l'arborescence, cliquez sur la table `nomenclature` et observez que les nomenclatures du banc (`id_produit = 2`) sont toujours présentes ! Nous disons alors que la base de données est dans un état incohérent. En effet, la clé étrangère `id_composant` de deux lignes de la table `nomenclature` désigne le `id` d'un composant qui n'existe plus.

Comment remédier à cela ? Tout simplement, en définissant des relations ...

1. Tout d'abord, remettons en place le composant Assise longue (mais en faisant attention d'avoir son `id = 6`)
  1. Dans l'arborescence, cliquez sur la table `composant` puis ouvrez l'onglet **Insérer**
  2. Donnez `id = 6`, `libelle = Assise longue` et `cout = 50`
  3. Cliquez sur le bouton **Exécuter**
2. Ouvrez maintenant l'onglet **Parcourir**
3. Vérifiez que les six composants sont bien tous de nouveaux présents et observez la valeur des `id`
4. À votre avis, que ce serait-il passé si nous n'avions pas précisé que `id = 6` lors de cette nouvelle insertion ?



Pour se prémunir d'un état incohérent de la base de données :

- Soit supprimer d'abord toutes les lignes dans toutes les tables qui font référence à la clé primaire de la donnée à supprimer. Puis la supprimer. Historiquement c'est ce que faisaient les programmeurs MySQL;

- Soit nous pouvons gérer avec MySQL les relations entre les clés primaires et les clés étrangères. Cela est notamment possible avec le moteur InnoDB avec lequel vos tables ont été créées. Si jamais ce n'était pas le cas, vous pouvez le corriger avec les requêtes suivantes :

```
ALTER TABLE `composant` ENGINE=INNODB;  
ALTER TABLE `produit` ENGINE=INNODB;  
ALTER TABLE `nomenclature` ENGINE=INNODB;
```

Maintenant, définissons une relation entre les deux tables `nomenclature` et `composant`.

### Question 5.2 : Relations entre clés étrangères et clés primaires

#### Question 5.2 : Relations entre clés étrangères et clés primaires

1. Allez sur l'onglet **Structure** de la table `composant` et activez l'indexation de la colonne `id` en cliquant sur  Index. Faites de même pour les deux clés étrangères de la table `nomenclature`, ainsi que pour la clé primaire de la table `produit`.
2. Puis, allez sur l'onglet **Concepteur** de la base de données `bureau_etude` (Rappel : il faut d'abord cliquer sur la base de données `bureau_etude` dans l'arborescence)
3. Cliquez sur le bouton Nouvelle relation , puis :
  - Cliquez d'abord sur la clé primaire `id` de la table `composant`
  - Cliquez ensuite sur la clé étrangère `id_composant` de la table `nomenclature`
  - Cliquez sur OK dans la boîte de dialogue
  - Une nouvelle relation apparaît entre les deux tables `composant` et `nomenclature`

Nouvelle relation  
FOREIGN KEY

on delete --

on update --

OK Annuler



Comment supprimer une relation ? Vous pouvez le faire graphiquement en cliquant sur l'extrémité arrondie  de la relation.

Figure 5.2.1 : Boîte de dialogue pour les règles de clé étrangère.

Maintenant, retournez sur l'affichage des lignes de la table `composant` et essayez de nouveau de supprimer le composant Assise longue. Que se passe-t-il ?

Nous allons maintenant mettre en place un autre comportement possible ...

### Question 5.3 : Suppressions en cascade

1. Dans l'arborescence, cliquez sur la table `nomenclature` puis ouvrez l'onglet **Structure**
2. Cliquez sur  Vue relationnelle
3. Modifiez les deux paramètres de la relation qui porte sur la clé étrangère `id_composant`
  - ON DELETE : RESTRICT devient CASCADE
  - ON UPDATE : RESTRICT devient CASCADE

4. Cliquez sur le bouton **Sauvegarder**
5. Dans l'arborescence, cliquez sur la table `composant` puis effacez  le composant Assise longue
6. Dans l'arborescence, cliquez maintenant sur la table `nomenclature` et observez les lignes de nomenclatures. Combien de lignes de nomenclatures sont encore présentes ? À quels composants correspondent ces lignes de nomenclatures ?
7. Pourquoi les lignes qui comportaient `id_composant = 6` ont disparues ? La base de données `bureau_etude` est-elle dans un état cohérent ?

Depuis la vue **Concepteur**, créez de même une nouvelle relation entre la clé étrangère `id_produit` de la table `nomenclature` et la clé primaire `id` de la table `produit`. Sauf que cette fois, vous choisirez l'option **CASCADE** pour définir le comportement à adopter en cas de suppression (on delete) ou de mise à jour (on update) :



Figure 5.3.1 : Boîte de dialogue pour les règles de clé étrangère.

## Exercice 6 • Exporter les scripts de création

~~Vous allez maintenant pouvoir observer le code SQL permettant de créer ces trois tables et de les peupler. Ce code SQL pourrait par exemple être utilisé pour recréer à l'identique la base de données sur un autre serveur.~~

1. Dans l'arborescence, cliquez sur la base de données `bureau_etude` puis ouvrez l'onglet **Exporter**
2. Laissez en place les deux options Rapide et SQL. Cliquez sur **Exécuter** et enregistrez le fichier `bureau_etude.sql` sur votre disque dur.
3. Depuis l'explorateur de fichier, allez dans le répertoire où vous avez sauvegardé le fichier. Faites un clic droit sur le fichier et choisissez **Edit with Notepad++**
4. Retrouvez dans ce fichier les trois requêtes **CREATE** de création de tables. Retrouvez les trois requêtes **INSERT** de peuplement de la base de données. Retrouvez la valeur de l'auto-incrément de la table `composant`.

Nous allons maintenant supprimer la base de données `bureau_etude`, puis la remettre en place :

## Exercice 7 • Déjà fini ?

*Félicitation pour votre efficacité !*

Vous pouvez profiter des quelques minutes restantes pour copier quelques requêtes **SELECT** pour interroger la base de donnée `bureau_etude`.

Tout d'abord, peuplez la base avec un nouveau composant « Plateau table » (de coût 100) et trois produits (chaise, banc et table basse) et ajoutez les nomenclatures correspondantes.

Ouvrir la fenêtre de requêtes avec l'onglet **SQL** puis cocher « Conserver la boîte de requête ».

1. Copiez une première requête : **SELECT \* FROM `produit` ;**
2. Cliquez sur le bouton **Exécuter** en bas de la fenêtre
3. Observez le résultat dans la fenêtre principale

Ou encore, successivement, les requêtes SQL :

- **SELECT \* FROM `produit` WHERE `id` = 1 ;**
- **SELECT \* FROM `composant` WHERE `id` = 2 ;**
- **SELECT \* FROM `composant` WHERE `cout` > 10 ;**
- **SELECT \* FROM `produit` JOIN `nomenclature` ON `id` = `id\_produit` WHERE `nombre` > 1 ;**

C'est un rapide aperçu de ce que vous ferez lors des séances suivantes.

## Exercice 8 • Pour s'exercer

Prenez le temps d'implanter le schéma relationnel de l'exercice 2 du TD 1 : créez la base de données (nommée `approvisionnement`) et ses quatre tables avec `phpMyAdmin`.