






164 – LEGO Project- Documentation



Explication du fichier zip

Après avoir extrait notre fichier zip, vous allez trouver les éléments suivants :

Name

 Backup	Tous les backups de notre base de données
 Requetes	Toutes les requêtes ainsi que les screens de preuve
 Documentation.pdf	La documentation
 Module164_exam_final.pdf	Le cahier des charges
 Module164_exam_final.xlsx	Le fichier Excel de base, mais modifié selon nos besoins

Backup

-  INSERT - individuels
-  Module164_exam_Regamey_Vienet.sql

Dans le dossier backup, il y a deux éléments. Le deuxième est le plus important étant donné que c'est le backup **ENTIER** de notre base de données. Pour l'utiliser, il suffit juste de lancer

le script. À noter qu'il y aura sûrement une erreur en vous disant que la base de données n'est pas reconnue (Unknown database). Pour régler ce problème, il suffit d'aller dans les propriétés de connexion de local host et d'écrire pour le nom de la base : « Module164_exam_Regamey_Vienet ».

S'il y a n'importe quel problème, il y a tous les backups nécessaires individuels de chaque table et de chaque entrée dans le dossier « INSERT – individuels ».

À noter que le fichier .sql que l'on a reçu de base, a été totalement modifié. Il a été refait de A à Z vu les incohérences qu'il contenait. Il était plus simple de recommencer de 0 que d'essayer de le modifier à la volée.

Il faut cependant noter une particularité supplémentaire dans la table « LEGO_sets » ci-dessous :

```
CREATE TABLE IF NOT EXISTS `mydb`.`LEGO_Sets` (
  `id_LEGO_Set` INT NOT NULL,
  `id_LEGO_Piece` INT NOT NULL,
  `quantite_Piece` INT NOT NULL,
  `id_LEGO_Theme` INT NOT NULL,
  `id_Couleur` INT NOT NULL,
  PRIMARY KEY (`id_LEGO_Set`,`id_LEGO_Piece`),
  FOREIGN KEY (`id_LEGO_Piece`)
    REFERENCES `mydb`.`LEGO_Pieces` (`id_LEGO_Piece`),
  FOREIGN KEY (`id_LEGO_Theme`)
    REFERENCES `mydb`.`LEGO_Themes` (`id_LEGO_Theme`),
  FOREIGN KEY (`id_Couleur`)
    REFERENCES `mydb`.`LEGO_Couleurs` (`id_Couleur`));
```

Étant donné que nous ne pouvions pas insérer les sets avec le fichier .sql de base qui nous a été fourni, car une clé primaire ne peut qu'être unique. Nous avons dû modifier cette partie et faire une clé primaire composite, ce qui nous a permis de pouvoir insérer nos données.

L'autre possibilité était de considérer une relation N-N entre la table « LEGO_sets » et « LEGO_pieces » mais il aurait fallu changer le

schéma MLD de base qui nous a été fourni. Nous avons donc décidé de partir sur la première solution même si celle-ci était « moins juste ».

Requêtes



Screenshots



requetes_all.sql

Il y a ici toutes les requêtes effectuées dans un seul fichier .sql comme convenu lors de notre échange par email. Il y a deux requêtes qui nous posent problème. La #7 et la #11.

Comme expliqué, nous avons dû faire une clé primaire composite pour pouvoir respecter le MLD, cela nous pose problème lors de ces deux requêtes puisqu'il y a des doublons malgré le mot-clé « DISTINCT » utilisé lors de la requête #11. Après discussion avec notre formateur et d'autres camarades de classe, il n'y a pas moyen de changer ce problème sauf si l'on change le MLD en créant une table intermédiaire.

Fichier Excel

Nous avons légèrement modifié le tableau Excel pour pouvoir faire nos insertions correctement. Il nous a plus ou moins servi de NotePad++ secondaire.

Note de fin

Nous avons trouvé particulièrement dommage que la majorité de notre projet a été la reconstruction de la base de données et son adaptation (CREATE TABLE et INSERT) alors que nous nous attendions à faire plutôt des requêtes. Nous avons dû passer environ 30h sur les insertions et uniquement 2h sur toutes les différentes requêtes.

Malgré cela, le projet était très intéressant !

Si vous avez des questions, n'hésitez pas à nous contacter lors de la correction !