

SQL dans un langage de programmation

Révisions SQL

Utilisation des fonctions

Etape 1 - Création des tables et données

Créer une base de données nommée 'mezabi3' en utf8, collate utf8 (ou supprimer les tables existantes si vous utilisez le serveur lampetu).

Exécuter le mezabi3.sql donné dans les ressources du TP dans la nouvelle base de données ou dans votre base si vous utilisez le serveur lampetu.

Etape 2

Créer une fonction permettant d'insérer une facture dans 'factures_entetes'.

Le numéro de facture est constitué de l'année, le mois et un numéro progressif sur 3 caractères (AAAAMMXXX). La fonction doit retourner le numéro de facture. Elle a en paramètre en entrée l'ID du client. La date insérée sera la date du jour. Générer plusieurs entêtes de factures à l'aide de votre fonction pour les clients dont les ID sont '1' et '2'.

Etape 3 - Suppression des codes barres existants

- Supprimer les codes barres de tous les articles par une requête (NULL dans la colonne).

Etape 4 - Fonction pour générer des codes barres

Structure d'un code barre :

La norme de structure des codes-barres s'intitule le GTIN (Global Trade Item Number) qui regroupe différents types de code-barres (GTIN 8, GTIN12, GTIN13, GTIN14). En France, le code GTIN 13 est le code-barres est le plus utilisé dans la distribution, 13 correspondant au nombre de chiffres associé au code.

Calcul de la clé :

Cette série de chiffre est organisé de manière suivante :

- Les **7 premiers chiffres** indiquent le pays d'origine du produit (300 à 379 pour la France) et le numéro du fabricant. (A récupérer dans la table parametres, ID = 'DEB_GENCODE', Valeur dans la colonne CONTENU_A).

- Du **8ème au 12ème** : numéro du produit du fabricant (Récupérer et incrémenter le dernier numéro utilisé dans la table parametres, ID = 'LAST_GENCODE', valeur dans la colonne CONTENU_N).

- Le **13ème chiffre** : clé de contrôle qui sert à valider le code-barres.

13 chiffres

← →

GCP + CIP													C	
------------------	--	--	--	--	--	--	--	--	--	--	--	--	----------	--

1	attribuer le rang : de 1 à 13 et de droite à gauche	13	12	11	10	9	8	7	6	5	4	3	2	1						
	code (exemple)	3	4	5	3	1	2	0	2	3	6	4	5	?						
2	additionner les chiffres du code de rang pair		4	+		3	+		2	+		2	+		6	+		5		=22
3	multiplier le résultat par 3														=22x3=66					
4	additionner les chiffres du code de rang impair	3	+		5	+		1	+		0	+		3	+		4		=16	
5	additionner le résultat des étapes 3 et 4														=66+16=82					
6	soustraire le résultat de l'étape 5 du multiple de 10 supérieur														=90-82=8					
	code complet	3	4	5	3	1	2	0	2	3	6	4	5	8						

Lorsque le résultat de la phase 5 est un multiple de 10 (exemple / 60, 70, 80), le caractère de contrôle est égal à 0.

SQL dans un langage de programmation

Révisions SQL

Créer une fonction **genereCodeBarre** sans paramètre qui génère un code barre en utilisant les valeurs contenues dans la table 'parametres' (incrémentation à chaque création de la valeur contenue dans LAST_GENCODE). La fonction retourne le code barre généré (VARCHAR(13)).

Pour vérifier si vos codes barres sont conformes à la norme :

- Calcul de la clé : <http://www.gomaro.ch/lecheck.htm>
- Vous pouvez vérifier votre calcul de clé sur : <https://www.gs1.fr/calculer-cle-contrôle>

Etape 5

Générer les codes barres pour toutes les lignes de stockprix.

Aide mémoire

CURDATE()	→ Date du jour AAAA-MM-JJ
DATE_FORMAT(maDate, '%d/%m/%Y')	→ 29/09/2022
Test variable nulle	→ IF variable IS NULL THEN
Division entière	→ DIV (83 DIV 10 = 8)
Reste de la division	→ MOD(9,2)=1
Extraction sous chaîne d'une chaîne	→ SUBSTRING(Chaîne, depart, longueur)
Concaténation	→ CONCAT(Chaîne1, Chaîne 2, ...)
Sous chaîne à gauche	→ LEFT(Chaîne, longueur)