

Travaux Pratiques n°1 : LDD

I. Base « Entreprise »

I.1- Contexte

Pour ce premier exercice, vous allez travailler avec une société utilisant une base suivant le schéma suivant :

```
SOCIETE (code_societe, raison_sociale, adresse, ville, code_postal)
CONTRAT (no_contrat, date_expiration, duree, montant, societe)
MATERIEL (no_materiel, no_serie, designation, no_contrat, type)
TYPE (no_type, designation, caracteristiques)
```

L'entreprise dont il est question a pour client des sociétés.

Chaque société est identifiée par son code. La raison sociale d'une société est son nom.

Les sociétés passent des contrats avec l'entreprise. Ces contrats sont identifiés par leur numéro.

Ces contrats sont relatifs à du matériel (par exemple une société achète du matériel auprès de l'entreprise et passe un contrat pour cela).

I.2- Manipulations

Création de la base

Créez la base de données sous Postgresql. Comme vous allez réutiliser cette base dans d'autres TP, nommez cette base de données « Entreprise », et non « TP1 » car sinon vous ne vous souviendrez plus de quelle base il s'agit alors que dans les énoncés elle sera toujours désignée par le nom « Entreprise ».

Création des tables

Créez un script appelé `Entreprise_LDD.sql` par exemple. Dans ce script, créez à l'aide du langage de définition SQL les 4 tables.

Remarque : Pensez à sauvegarder ce script, vous pourrez être amenés à le réutiliser

- **Analysez le sens de chacun des attributs.**
- **Choisissez avec soin les types des attributs**
- **Ajoutez des contraintes d'existence et d'unicité.**

Remarque : Attention, si vous utilisez le type SERIAL, ce n'est pas un standard SQL. Moodle n'est pas capable de le gérer (remplacer alors par Integer)

Destruction des tables

Testez également la destruction de ces tables. Mettez les lignes de destruction des tables en début du script :

`DROP table if exists <nomtable> ;`

1.3- Préparation du CM2

Les transparents relatifs au CM2 sont disponibles sur Moodle. Consultez les (notion de clé primaire et de clé étrangère).

Indications :

Une clé primaire n'est pas toujours constituée d'un seul attribut

Une clé étrangère référence généralement une clé primaire

Deux attributs ayant le même nom dans des tables différentes n'ont pas forcément de rapport entre eux.

Dans l'autre sens, deux attributs ayant des noms différents dans deux tables peuvent être liés, par exemple par une relation de clé étrangère.

- Définissez une clé primaire par table (éventuellement constituée de plusieurs attributs)
- Définissez également les clés étrangères
- Réorganisez l'ordre de destruction de vos tables. Le mot-clé CASCADE est interdit.
- Copiez-collez votre code SQL sur Moodle (Attention : deux tentatives, 5 minutes seulement. Moodle ne doit pas être utilisé comme outil de test)

II. Base « Gestion »

II.1- Contexte

Repchauf SA est une entreprise spécialisée dans la maintenance des systèmes de chauffage pour particuliers. L'entreprise offre des interventions de maintenance ou de réparation dans les appartements et les maisons individuelles de ses clients.

Il s'agit d'une PME qui utilise un système automatisé pour la gestion de ses interventions et éventuellement des remplacements de pièces effectués.

Voici les informations structurées que vous avez réussi à réunir.

Table **CLIENTS**

No_client	Nom	prenom	adresse
105	Dallalon	Jean	5 rue Jean Moulin
101	Rivoire		18 Rue Ronde
102	Favero	André	43 Rue Beaujolais
103	Provent	Catherine	38 Rue du stade
104	Labric		35 Rue des fleurs

Table **PRODUITS**

Reference	Designation	PrixHT
DT010	Disjoncteur 10A	7.21
DT180	Bloc Huger	6.12
DT802	Boîte contrôle	68.35
DT711	Cellule	25.36
DT125	Bloc soc	6.89
DT015	Disjoncteur 15A	14.94
DT205	Brûleur Huger	153.37
DT310	Brûleur soc	200.20
DT120	Connecteur	20.35

sur les **FACTURES**

no_facture	date_facture	Etat
1000	01/01/2009	R
1001	12/02/2009	R
1002	17/03/2009	R
1003	24/04/2009	R
1004	16/05/2009	R
1005	08/07/2009	R
1006	08/07/2009	R
1007	15/07/2009	R
1008	15/07/2009	R
1009	22/07/2009	C
1010	22/07/2009	C
1011	29/07/2009	C
1012	30/08/2009	R
1013	19/10/2009	R

R= réglée C=en cours

sur les **REPLACEMENTS** (pièces remplacées).

reference	No_interv	Qte_remplacee
DT802	1043	1
DT711	1043	2
DT180	1043	1
DT205	1044	1
DT125	1045	2
DT010	1045	1
DT310	1046	1
DT711	1047	3
DT120	1047	2
DT015	1048	1
DT180	1049	4
DT711	1049	2
DT205	1050	1
DT711	1051	2
DT120	1051	1
DT120	1052	3

sur les INTERVENTIONS

no_interv	Date_interv	Nom_resp	Nom_interv	Temps	No_client	no_facture
1039	10/02/2009	Mauras	Saultier	1	101	1001
1040	10/03/2009	Foucher	Saultier	1	103	1002
1041	15/03/2009	Foucher	Saultier	2	103	1002
1042	06/04/2009	Foucher	Saultier	1	101	1003
1043	03/07/2009	Mauras	Saultier	2	105	1005
1044	04/07/2009	Mauras	Saultier	0.5	101	1006
1045	08/07/2009	Mauras	Bonnaz	1.5	102	1007
1046	10/07/2009	Foucher	Crespin	1	102	1007
1047	11/07/2009	Mauras	Crespin	2	103	1008
1048	15/07/2009	Foucher	Bonnaz	1	105	1009
1049	18/07/2009	Foucher	Saultier	1.5	101	1010
1050	22/07/2009	Foucher	Saultier	0.5	104	1011
1051	23/07/2009	Mauras	Bonnaz	2.5	104	1011
1052	29/07/2009	Mauras	Saultier	1.5	104	1011

II.2- Manipulations

Création de la base

Créez la base de données sous Postgresql. Nommez cette base de données « Gestion ».

Création des tables

Créez un script appelé Gestion_LDD.sql. Dans ce script, créez à l'aide du langage de définition SQL les 5 tables.

Remarque : Pensez à sauvegarder ce script, vous pourrez être amenés à le réutiliser.

Dans un premier temps, vous prendrez en compte, uniquement les contraintes suivantes : tous les attributs de ces 5 tables sont obligatoires (non nuls), sauf l'attribut « prenom » de la table CLIENTS

Gestion des dates

Afin de fixer le style de date au format européen (où le jour est avant le mois) rajoutez la commande suivante au tout début du script : set datestyle to 'european';

Destruction des tables

Afin de pouvoir ré-exécuter à volonté le script de création de la base (en cas de mauvaise manipulation dans la suite du TP par exemple), rajoutez au tout début du fichier la liste des commandes : DROP table if exists <nomtable> ;

Pour vérifier si tout fonctionne, essayez d'exécuter plusieurs fois ce script et vérifiez qu'il n'y a pas d'erreur.

Ajout de contraintes « simples »

En vous aidant de la documentation de PostGres faites en sorte que :

- le numéro de client dans la table CLIENTS soit supérieur strictement à 100
- la référence dans la table PRODUITS et REMPLACEMENTS soit composée de 5 caractères exactement, et les deux premiers caractères sont 'DT'
- l'attribut 'état' dans FACTURES soit ou bien 'C' (en cours), ou bien 'R' (réglée), et 'C' est sa valeur par défaut,
- le temps d'intervention ne puisse être égal à 0 et est ≤ 8 heures

Remarque : Il n'est pas demandé d'utiliser des expressions régulières.

Modification de table

- On désire maintenant ajouter la ville, le code postal et le téléphone pour chaque client. Ajoutez ces trois attributs dans la table CLIENTS, sachant que le numéro de téléphone est composé d'exactly 14 caractères de format XX-XX-XX-XX-XX (ne créez pas de domaine pour l'instant, simplement une contrainte statique de format de données basique). Vous pouvez utiliser la syntaxe « alter table ».
- Rajoutez la contrainte suivante sur la table : aucun des trois attributs supplémentaires ville, code postal ni téléphone ne peuvent être nuls. **Utilisez un alter table, ne changez pas la partie de votre code correspondant à la création de la table CLIENTS**

Ajout de table

Selon le niveau de complexité de l'intervention (réparation ou simple vérification par exemple), le prix de l'heure de main d'œuvre (taux horaire) est différent. Rajoutez donc dans la base de données la table TAUXHORAIRE qui précise le taux horaire de chaque type d'intervention. Plus précisément, cette table sera constituée d'un attribut entier codetaux servant de clé primaire, et du tauxhoraire fixant le taux en question. **La table TAUXHORAIRE doit être créée après INTERVENTION**

Puis, ajoutez également dans la table INTERVENTIONS un attribut non nul appelé codetaux. Utilisez un ALTER TABLE.

Ajout d'attributs (Relation PRODUITS)

Vous ajouterez dans la relation PRODUITS deux nouveaux attributs, qte_en_stock et qte_securite (seuil en-dessous duquel il ne faut pas que le stock passe).

Donc, rajoutez la contrainte que la quantite en stock doit toujours être supérieure à la quantité de sécurité.

Par défaut, vous placerez la valeur 0 à qte_securite et 1 à qte_stock.

Créez les deux colonnes et les contraintes.

Peuplement

Vous trouverez sur Moodle un fichier « Peuplement_Gestion.sql ».

- Vous ne devez pas modifier ce fichier.
- En modifiant éventuellement votre code, faites en sorte qu'en jouant votre code puis ce fichier de peuplement, il n'y ait pas d'erreur.

II.3- Préparation du CM2

- Comme dans le premier exercice, ajouter une clé primaire par table (qui peut être composée de plusieurs attributs)
- Ajoutez également les clés étrangères.
- Ré-organisez l'ordre de destruction des tables. Le mot-clé CASCADE est interdit.
- **Faites en sorte que l'attribut codetaux de la table INTERVENTIONS référence l'attribut code_taux de la table taux_horaire (clé étrangère).** Utilisez un ALTER TABLE.

II.4- Pour aller plus loin

- Créez des séquences pour no_facture, no_client
- Créez un domaine utilisant des expressions régulières (SIMILAR TO) pour l'attribut REFERENCE.
- Faites des tests unitaires pour éprouver vos contraintes sur ce domaine.