مكتب التكوين المهني وإنعاش الشغل

OFPPT

Office de la Formation Professionnelle
et de la Promotion du Travail

Direction Recherche et Ingénierie de Formation

Examen de fin formation, session juin 2006

Epreuve pratique

Filière : TSDI ***Barème :40 Pts.***

Niveau : TS ***Durée : 4H30***

Variante n°2

Important :
Il sera pris en considération la convivialité des interfaces et la clarté du code. La gestion des erreurs et la robustesse de l'application seront bien appréciées .
*La totalité de votre travail doit être enregistrée dans un même répertoire pour faciliter la sauvegarde par la suite. Pour cela, il faut créer sur **votre bureau** un dossier qui porte **votre nom & Prénom**, et c'est dans ce dernier que vous allez enregistrer vos réalisations par la suite.*

Partie 1 : DotNet (22 Points)

On se propose de développer une application qui permet de gérer les activités journalières d'une station de service. Il est question dans un premier temps de créer la partie permettant au propriétaire de la station de service d'entrer les ventes journalières et d'afficher un certain nombre d'états.

La station est équipée de sept citernes souterraines auxquelles sont connectés sept volucompteurs, à raison d'un volucompteur par citerne. (Voir contenu des tables ci-dessous)

Le volucompteur est l'appareil qui permet de pomper le carburant d'une citerne pour le verser dans le réservoir d'un véhicule. Il est équipé d'afficheurs qui indiquent la quantité livrée, le prix de vente ainsi que le montant à payer par le client. Il est également équipé d'un compteur général qui indique le nombre total de litres pompés de la citerne. C'est cette valeur qu'on doit relever à la fin de chaque journée pour la saisir dans notre base de données.

On souhaite enregistrer pour chaque article le code, désignation, prix d'achat, prix de vente, stock et stock minimum (exprimé en nombre de litres). La quantité en stock représente le total des stocks des citernes correspondantes au produit.

Une citerne est identifiée par un code, nom, capacité (en nombre de litre) et produit stocké.

Chaque volucompteur possède un code, nom et code de la citerne a laquelle il est rattaché.

A la fin de chaque journée, le pompiste note et saisi les valeurs des volucompteurs. Il saisi deux valeurs pour chaque volucompteur : le compteur du jour j (cpt2) et celui du jour j-1 (cpt1). Ainsi la différence cpt2 - cpt1 nous permet de connaître la quantité de carburant vendu pour chaque citerne.

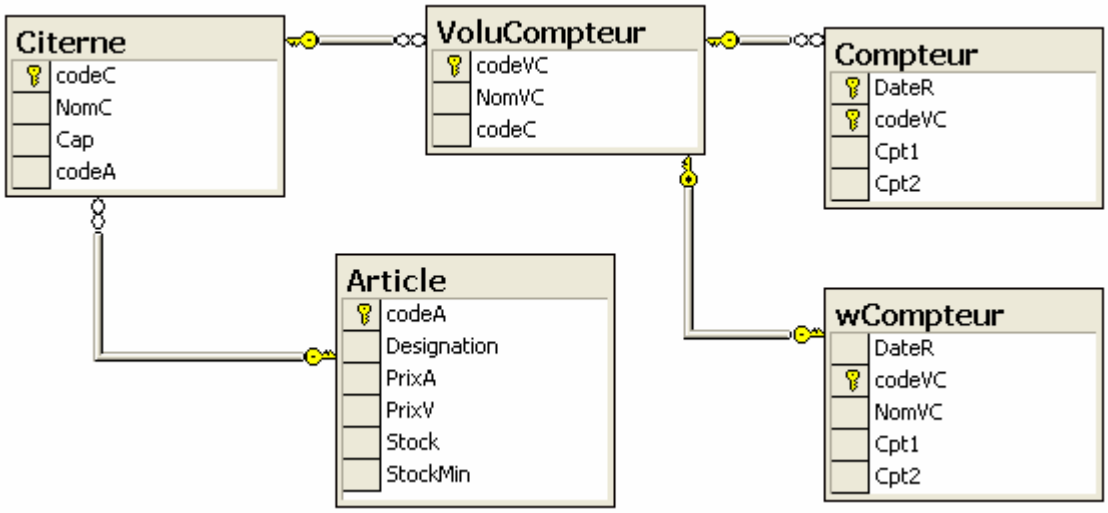
L'analyse a permit de construire le modèle de base de données suivant :

Article (**codeA** : int, designation : varchar(40), prixA : real, prixV : real, stock : real, stockMin : real)

Citerne (**codeC** : int, nomC : varcha(10), cap : int, ***codeA*** : int)

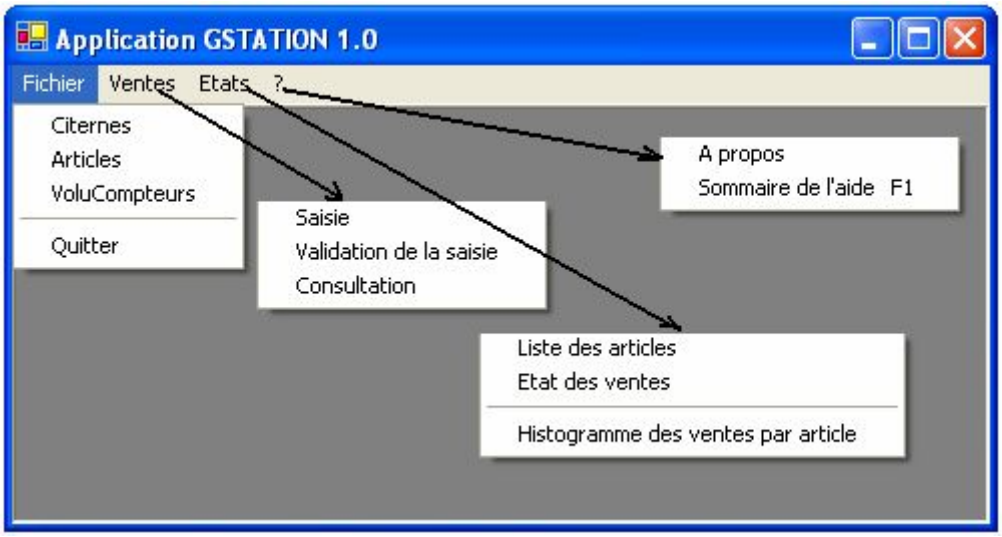
Volucompteur (**codeVC** : int, nomVC : varchar(10), ***codeC*** : int)

Compteur (**dateR** : int, ***codeVC*** : int , cpt1 : int, cpt2 : int)



Travail à faire :

1) Créer la base de données SQL server sur votre dossier personnel **(2pts)**
2) Créer la feuille MDI suivante : **(1pt)**



3) Développer les formulaires de mise à jour des articles, citernes et volucompteurs. **(3pts)**
4) Les ventes sont saisies dans la table **wcompteur** qui est considérée comme une table de travail. Avant de commencer, le pompiste entre la date des ventes (qui doit être initialisée par la date système) et click sur le bouton générer pour exécuter les taches suivantes :
   - Suppression de tout le contenu de la table **wcompteur**. Afficher un message d'avertissement lorsque la table n'est pas vide avec une possibilité d'abandon de l'opération.
   - Génération des enregistrements de la table **wcompteur** à partir de la table volucompteur. La date de vente est initialisée par celle que l'utilisateur a saisie sur le formulaire. Pour faciliter cette opération, créer une procédure stockée.

Utiliser une grille pour la saisie, verrouiller toutes les colonnes sauf celles qui correspondent aux deux compteurs. **(4pts)**

5) Après la saisie des ventes, l'utilisateur peut procéder à leur validation (option « **validation de la saisie** » du menu « **Ventes** »), ce qui aura pour conséquence :
   - Le transfert de tous les enregistrements de la table **wcompteur** vers la table **compteur.** Utiliser une procédure stockée pour cette opération.
   - Suppression de tout le contenu de la table **wcompteur. (2pts)**
6) L'option « **consultation** » du menu « **Ventes** » permet d'afficher le contenu de la table « **compteur**». **(2pts)**
7) Développez avec Crystal Report les états suivants
   - liste des articles **(1.5pt)**
   - Etat des ventes. **(1.5pt)**
   - Histogramme des ventes : affiche un histogramme par article et ventes cumulées. Utiliser une procédure stockée. **(2pts)**
8) Développer une fenêtre « A propos » qui affiche le nom, numéro de version de l'application, votre nom et prénom. **(1.5pts)**
9) Développer une page HTML qui contient le sommaire de l'aide de votre application**. (1.5pts)**
10) Détacher la base de données du serveur SQL

Partie 2 : Application Web (18 Points)

On se propose de développer une version Web de la même application.

L'application comportera la gestion des citernes (Ajout, suppression et consultation), une partie gestion des ventes qui constitue le cœur de l'application et un état des ventes.

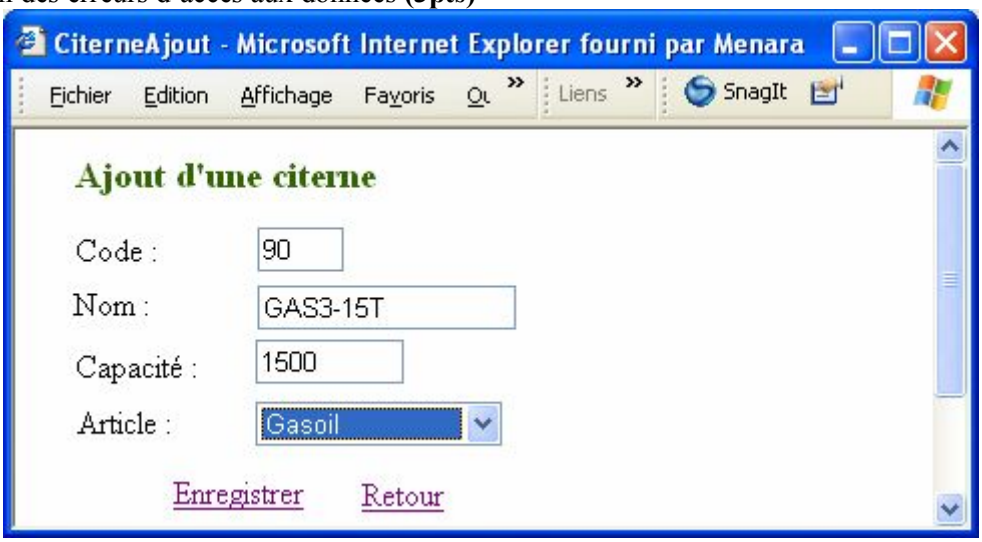
1) Pour contrôler l'accès à l'application, on vous demande de mettre en place la sécurité par formulaire et de permettre uniquement à l'utilisateur ayant le Login « OFPPT » et le mot de passe également « OFPPT » d'accéder aux pages de l'application. **(3pts)**

2) Créer la page de démarrage suivante. Elle comporte le menu de l'application **(2pts)**

3) Ajout d'une citerne: l'utilisateur saisi les informations d'une citerne et clique sur le bouton Enregistrer.
L'application commence par vérifier les données saisies :

- Présence des informations obligatoires, à savoir : le code, le nom citerne, la capacité et le code produit
- Vérification de la capacité qui doit être supérieure 1000

Afficher un message en cas d'erreur.
Prévoir la gestion des erreurs d'accès aux données **(3pts)**

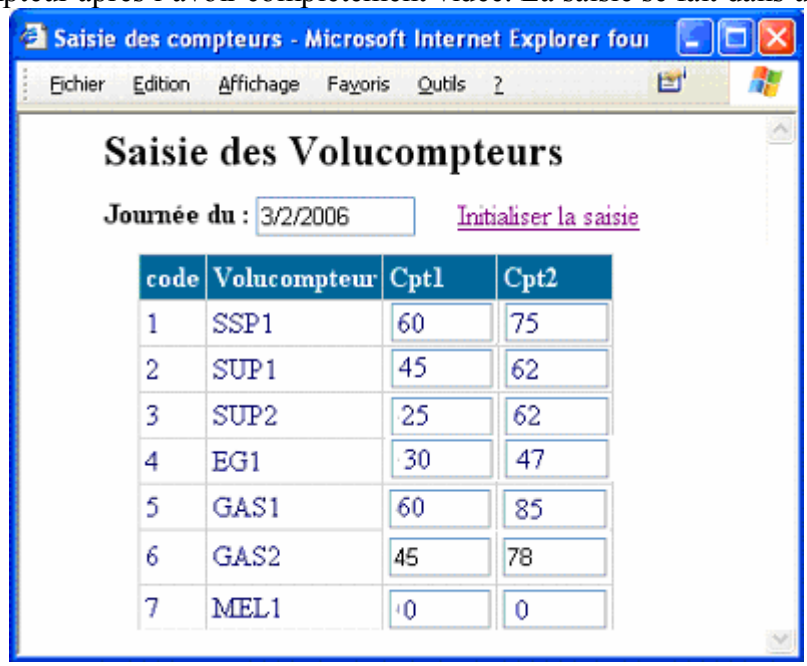


4) Suppression d'une citerne : l'utilisateur entre un code et clique sur le bouton Chercher. L'application affiche les autres informations de la citerne quand elles existent ou le message « Citerne non trouvée » en face du nom. Le bouton Supprimer doit être activé et la zone de texte verrouillée quand la citerne est trouvée. L'utilisateur peut alors confirmer la suppression.
Prévoir la gestion des erreurs d'accès aux données **(4pts)**

5) Afficher la liste de toutes les citernes **(2pts)**

codeC	NomC	Cap	Designation
10	SSP1-10T	10000	Super SP
20	SUP1-10T	10000	Super
30	SUP2-5T	5000	Super
40	EG1-15T	15000	Gasoil 350
50	GAS1-15T	15000	Gasoil
60	GAS2-15T	15000	Gasoil
70	MEL1-5T	5000	Melange

6) Saisie des compteurs : le bouton « Initialiser la saisie » fait appel à la procédure stockée qui permet de remplir la table wcompteur après l'avoir complètement vidée. La saisie se fait dans une grille. **(4pts)**



code	Volucompteur	Cpt1	Cpt2
1	SSP1	60	75
2	SUP1	45	62
3	SUP2	25	62
4	EG1	30	47
5	GAS1	60	85
6	GAS2	45	78
7	MEL1	0	0