

## Secteur Tertiaire Informatique Filière étude - développement

### Développer des composants d'interface

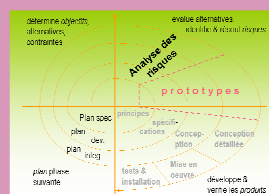
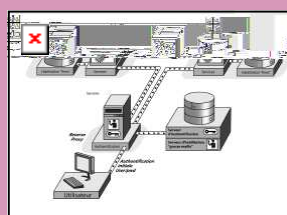
#### Accès aux données : Exercices

Accueil

Apprentissage

Période en  
entreprise

Evaluation



Code barre

# SOMMAIRE

<b>I</b>	<b>PRESENTATION .....</b>	<b>4</b>
I.1	Le modèle physique.....	4
<b>II</b>	<b>LE MODE CONNECTE .....</b>	<b>5</b>
II.1	Exercice 1 : Connexion à la base .....	5
II.1.1	Objectif .....	5
II.1.2	Les traitements .....	5
II.2	Exercice 2 : Recherche d'une ligne donnée dans une table .....	6
II.2.1	Objectif .....	6
II.2.2	Les traitements .....	6
II.3	Exercice 3 : Recherche d'un jeu de lignes dans une table .....	7
II.3.1	Objectif .....	7
II.3.2	Les Traitements .....	7
II.4	Exercice 4 : Mise à jour de table.....	8
II.4.1	Objectif .....	8
II.4.2	Les traitements .....	8
<b>III</b>	<b>LE MODE DECONNECTE .....</b>	<b>9</b>
III.1	Exercice 1 : Affichage des données a partir d'un DataSet.....	9
III.1.1	Objectif .....	9
III.1.2	Les traitements .....	9
III.2	Exercice 2 : Mise a jour des données d'une DataTable.....	10
III.2.1	Objectif .....	10
III.2.2	Les traitements .....	10
III.3	Exercice 3 : Liaison de données.....	11
III.3.1	Objectif .....	11
III.3.2	Les traitements .....	11

## I PRESENTATION

### I.1 LE MODELE PHYSIQUE

La base de données relationnelle PAPYRUS est constituée des relations suivantes :

**PRODUIT** (CODART, LIBART, STKLE, STKPHY, QTEANN, UNIMES)

**ENTCOM** (NUMCOM, OBSCOM, DATCOM, NUMFOU)

**LIGCOM** (NUMCOM, NUMLIG, CODART, QTECDE, PRIUNI, QTELIV, DERLIV)

**FOURNIS** (NUMFOU, NOMFOU, RUEFOU, POSFOU, VILFOU, CONFOU, SATISF)

**VENTE** (CODART, NUMFOU, DELLIV, QTE1, PRIX1, QTE2, PRIX2, QTE3, PRIX3)

Tous les exercices seront effectués sur cette base de données.

## II LE MODE CONNECTE

### II.1 EXERCICE 1 : CONNEXION A LA BASE

#### II.1.1 Objectif

Etablir une connexion sur une base données SqlServer avec l'objet SqlConnection, et afficher l'état de la connexion.

#### II.1.2 Les traitements

Le bouton **Connexion** permet d'ouvrir la connexion, le bouton **Déconnexion** de la fermer.

Le bouton **Fin** termine l'application

Inclure la connexion dans un traitement d'erreur (try...catch)

Dans la zone de texte, afficher l'état de la connexion avec la propriété State de l'objet connexion

Afficher les messages interceptés avec le bloc de gestion des erreurs

Après connexion bonne

The screenshot shows a window titled "Connexion à la base de données". It contains two text input fields: "Serveur :" with the value "AFPA-4715639C23" and "Base de données :" with the value "pp". Below these are three buttons: "Connexion", "Déconnexion", and "Quitter". The "Etat de la connexion" is displayed as "Closed". A text area at the bottom shows an error message: "Message : Impossible d'ouvrir la base de données 'pp' demandée par la connexion. La connexion a échoué. Échec de l'ouverture de session de l'utilisateur 'AFPA-4715639C23\cfpa\_afpa'. Error code : -2146232060 Numero : 4060".

Après connexion en erreur

The screenshot shows the same window as the previous one, but with the "Base de données" field set to "papyrus". The "Etat de la connexion" is displayed as "Open". The error message text area is empty.

## II.2 EXERCICE 2 : RECHERCHE D'UNE LIGNE DONNEE DANS UNE TABLE

### II.2.1 Objectif

Accéder à une table et afficher une occurrence avec les objets **SqlCommand** et **SqlDataReader**, et afficher les données issues de l'exécution de la requête dans des **TextBox**.

### II.2.2 Les traitements

Afficher les coordonnées d'un fournisseur à partir de la saisie de son code.  
Le code doit être saisi obligatoirement, et ne doit pas excéder sa longueur définie dans la base.

En cas d'absence, ou en cas d'inexistence dans la base, l'utilisateur sera averti.

Les données d'un fournisseur existant seront alors affichées dans la feuille 2.

#### Feuille 1



The screenshot shows a Windows-style window titled "Affichage Fournisseur". Inside, there is a label "Code fournisseur" followed by a text input box. Below the input box are two buttons: "Valider" and "Quitter".

#### Feuille 2



The screenshot shows the same "Affichage Fournisseur" window, but now it displays the details of a selected supplier. The fields are filled with the following data: "Nom : GROBRIGAN", "Adresse : 20 rue du papier", "CP / Ville : 92200 Papercity", "Contact : Georges", and "Satisfaction : 8". A "Retour" button is located at the bottom right of the window.

Vous utiliserez une requête SQL codée dans la feuille.

## II.3 EXERCICE 3 : RECHERCHE D'UN JEU DE LIGNES DANS UNE TABLE

### II.3.1 Objectif

Afficher la liste des commandes dans une **ListBox** d'un fournisseur sélectionné par son nom dans une **ComboBox**.

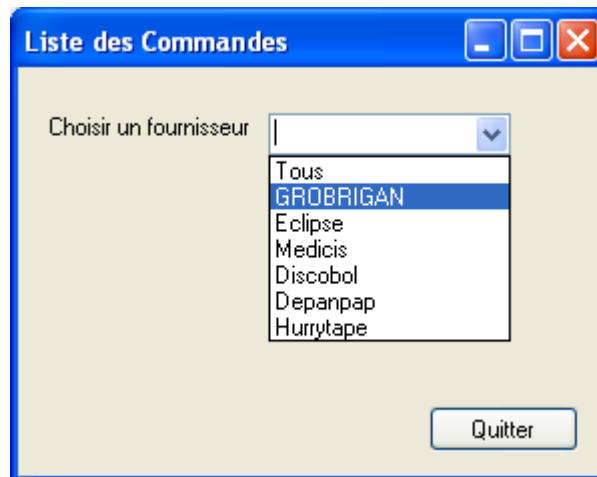
Le numéro de la commande, le texte commentaire et la date de commande seront affichés dans une **ListBox**.

### II.3.2 Les Traitements

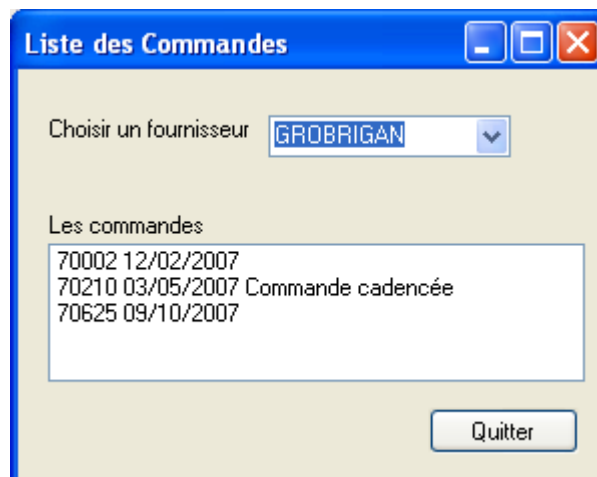
Construire la liste des fournisseurs (le nom) dans la comboBox. Ajouter en début de liste le mot « Tous » pour sélectionner tous les fournisseurs.

Sur la sélection du fournisseur, accéder à ses commandes par son code, et lister les informations dans la Listbox.

La feuille



La feuille (suite)



La recherche des commandes du fournisseur s'effectuera par une procédure stockée de nom GetEntCom que vous coderez.

Cet exercice peut être prolongé en remplaçant la ListBox des commandes par un ListView qui permet le colonage.

## II.4 EXERCICE 4 : MISE A JOUR DE TABLE

### II.4.1 Objectif

Mettre à jour (Insert, Update et Delete) la table FOURNIS avec l'objet **SqlCommand** et sa méthode **ExecuteNonQuery** .

### II.4.2 Les traitements

Afficher les coordonnées d'un fournisseur à partir de la saisie de son code.

Le code doit être saisi obligatoirement. L'utilisateur sera averti de sa non saisie.

Un click sur le bouton **Rechercher** permettra d'afficher dans les TextBox les informations du fournisseur.

Un bouton **Suppression** sera alors accessible pour permettre la suppression de cette ligne.

Un click sur le bouton **Créer** permettra d'afficher à vide les TextBox pour permettre la création.

On contrôlera la validité de la saisie pour chaque champ, par rapport aux contraintes induites dans la base de données :

La ligne sera alors ajoutée, modifiée ou supprimée selon le cas. Une confirmation de suppression est demandée, ainsi qu'un message de résultat, réussite ou échec pour chaque opération effectuée.

**Attention** : Plusieurs cas d'erreurs sont à gérer :

- En création d'un nouveau fournisseur, le code existe déjà
- La suppression d'un fournisseur pour lequel il existe des commandes ...
- La suppression d'un fournisseur pour lequel une vente existe.

L'interface graphique se limite à une seule feuille.

### III LE MODE DECONNECTE

#### III.1 EXERCICE 1 : AFFICHAGE DES DONNEES A PARTIR D'UN DATASET

##### III.1.1 Objectif

Savoir utiliser un DataSet typé en lecture seule.

##### III.1.2 Les traitements

Créer un DataSet typé comprenant la table PRODUIT, à l'aide de l'assistant **Configuration de sources de données**.

A l'initialisation de la feuille, peuplez le DataSet et chargez le libellé du produit dans une ListBox.

A chaque sélection d'un produit dans la ListBox, afficher son détail dans des TextBox non modifiables.

#### La feuille

The screenshot shows a Windows application window titled "Affichage des produits". Inside the window, there is a section labeled "Liste des produits" containing a list box with the following items: "Papier 1 ex continu", "Papier 2 ex continu", "Papier 3 ex continu", "Papier 4 ex continu", "Pré imprimé commande", "Pré imprimé facture", and "Pré imprimé bulletin paie". To the right of the list box, there are seven text boxes displaying details for the selected product, "Papier 1 ex continu": "Code" (I100), "Libellé" (Papier 1 ex continu), "Stock d'alerte" (100), "Stock physique" (557), "Quantité annuelle" (3500), and "Unité de mesure" (B1000). A "Quitter" button is located at the bottom right of the window.



## III.2 EXERCICE 2 : MISE A JOUR DES DONNEES D'UNE DATATABLE

### III.2.1 Objectif

Sur le DataSet précédemment créé, l'utilisateur aura la possibilité de modifier les données de l'élément sélectionné dans la liste.

### III.2.2 Les traitements

Reprendre le traitement de l'exercice précédent.

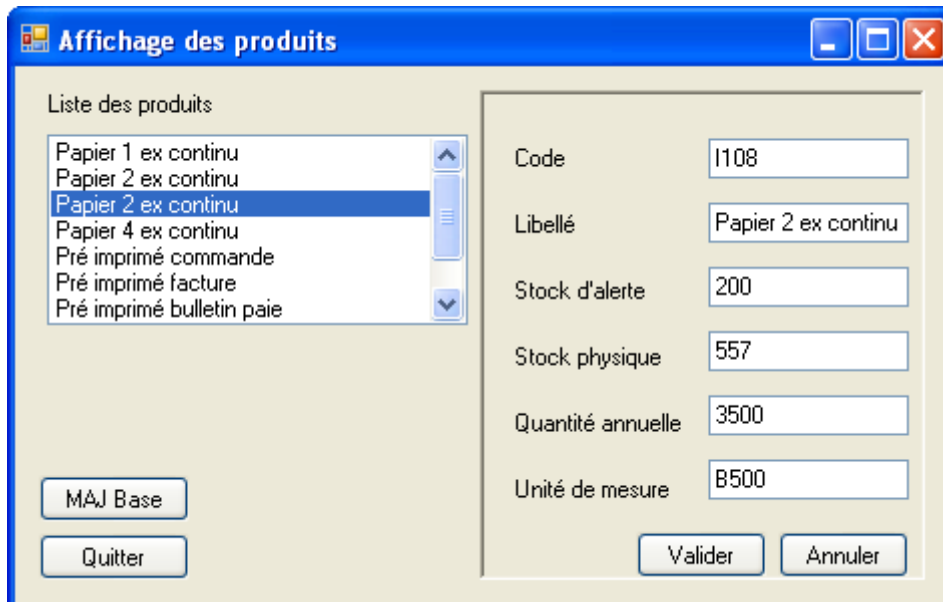
Au click sur le bouton Valider :

- Ajoutez le contrôle des champs saisi dans les TextBox
- Mettez à jour le DataSet.
- Si le libellé a été modifié, rechargez la liste

Au click sur le bouton Annuler, les données sélectionnées sont cachées.

Au click sur le bouton Maj Base, les modifications sont répercutées dans la base de données.

### La feuille



**Vous pouvez prolonger cet exercice en ajoutant l'insertion et la suppression.**

### **III.3 EXERCICE 3 : LIAISON DE DONNEES**

#### **III.3.1 Objectif**

Savoir développer une application RAD

#### **III.3.2 Les traitements**

En utilisant les outils proposés par Visual Studio, développer l'application de Mise à jour des produits.

- En utilisant un DataGridView et des contrôles de détail
- En utilisant la liaison de données



## **Etablissement référent**

*Marseille Saint Jérôme*

## **Equipe de conception**

*Elisabeth Cattaneo*

## **Remerciements :**

### **Reproduction interdite**

Article L 122-4 du code de la propriété intellectuelle.  
« toute représentation ou reproduction intégrale ou partielle faite sans le  
consentement de l'auteur ou de ses ayants droits ou ayants cause est  
illicite. Il en est de même pour la traduction, l'adaptation ou la reproduction  
par un art ou un procédé quelconques. »

Date de mise à jour 09/05/2008  
afpa © Date de dépôt légal mai 08



**afpa / Direction de l'Ingénierie 13 place du Générale de Gaulle / 93108 Montreuil  
Cedex  
association nationale pour la formation professionnelle des  
adultes  
Ministère des Affaires sociales du Travail et de la  
Solidarité**