

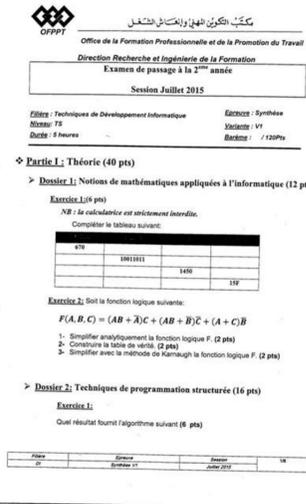
I'm not robot  reCAPTCHA

I am not robot!

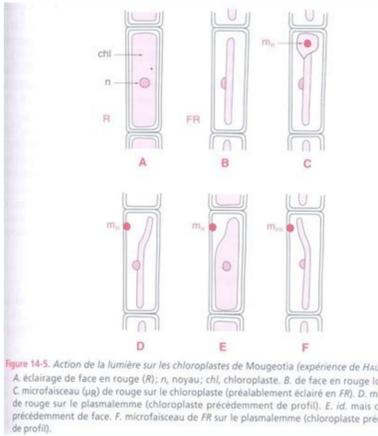
Examen corrigé developpement web pdf

Page 1 sur 5 DébutPrécédent12345SuivantFin Cours gratuits » Cours informatique » Cours développement web Participez au vote ☆☆☆☆☆ Microsoft Word - Université Nice Sophia Antipolis 2007-2008 LPSIL - UE SIL-TC3-P1 Module: OSI, INTERNET ET PROGRAMMATION WEB TP 5 – Programmation Web avec PHP

- Correction Objectif : apprentissage PHP, Base de Données et PHP Correction : la correction sera diffusée la semaine qui suit : refaire les exercices avec la correction Note : vous créerez sur npx un répertoire ~/web/TP5 dans lequel vous placerez vos fichiers html ainsi qu'une page Objectif : créer des scripts du côté serveur avec PHP, utiliser les variables de session Note : vous placerez vos fichiers PHP sur linserv1 : 134.59.22.1 ou linserv2 : 134.59.22.24 port 80 et 443, dans un répertoire ~/web/TP5 Pour plus d'informations sur ces serveurs : pour plus d'infos sur ces serveurs: 1.Scripts du côté serveur avec PHP Exemple d'introduction éez une page qui affiche simplement la chaîne de caractères "Hello PHP, nous sommes le "suive de la date du jour sur le serveur. Hello PHP, nous sommes le éez une deuxième version permettant d'afficher à la suite de la date, le message "Bon matin" ou "Bonne après midi" en fonction de l'heure sur le serveur. Nous sommes leVariables d'environnement 1.3.Afficher dans un tableau la signification et la valeur des variables d'environnement suivantes \$SERVER_ADDR \$HTTP_HOST \$REMOTE_ADDR gethostbyAddr(\$REMOTE_ADDR) \$HTTP_USER_AGENT Quelques variables d'environnement Variable Valeur Adresse IP du serveur web Nom du serveur web Adresse IP du client Nom de la machine cliente Navigateur du client 1.4.Afficher toutes les variables d'environnement en appelant la fonction phpinfo() Créer simplement une page, avec le contenu suivant : Formulaire pour créer un formulaire comportant 2 champs de texte nom, prénom, un menu d'options sexe (M ou F), et un menu de choix multiples vins (bordeaux, beaujolais, Loire). Vous ferez appel à un script pour traiter les données du formulaire. Elles seront envoyées par la méthode GET. Nom: Prénom: Sexe:M F Vins:bordeauxbeaujolaisLoire de provenance Affichage des données saisies Essayez à nouveauez une nouvelle version qui permettra à la fois de créer et de traiter le formulaire de l'exercice précédent - si le script est appelé sans paramètre il créera le formulaire - s'il est appelé par l'envoi de données (bouton submit du formulaire), il traitera les données. Vous utiliserez un champ caché permettant de tester si le script est appelé par l'envoi de données ou pas (ce champ caché est nécessaire dans le cas où le script serait appelé par envoi du formulaire avec tous les champs laissés vide) Nom: Prénom: Sexe:M F Vins:bordeauxbeaujolaisLoire de provenance éez un formulaire comportant une zone de texte, et ensuite créez le script qui renvoie le texte saisi en remplaçant les caractères nouvelle ligne par des balises. (utiliser la fonction nl2br de PHP), vos commentaires Affichage des données saisies Fonctions et classes 1.8.Définir une classe Personne possédant les propriétés nom, prénom, et une méthode presenter() qui renvoie la chaîne de caractères "je m'appelle " suivi du nom et prénom. // fichier éer un programme de test qui instancie 2 personnes, puis affiche leurs descriptions. 1.10.Ajoutez dans la classe Personne une propriété date de naissance et une méthode age() renvoyant l'âge. 1.11.Tester la classe Personne. 2.Sessions PHP de tester les variables de session, commencez par créer la page ci-dessous : Test de variables de session Ouvrir une sessionAfficher variables de sessionDétruire variables de sessionéer le script qui : ? crée une nouvelle session si aucune session n'existe (un SID est engendré et transmis dans un cookie) ? ou bien restaure la session en cours (connue par son identifiant de session SID) Pour cela il faudra appeler en début de script la fonction PHP session_start(). Afficher ensuite un formulaire permettant de saisir le nom et le prénom de l'utilisateur. Ces deux informations seront conservées dans des variables de session lorsque le formulaire est envoyé. Ouvrir une session Nom: Prénom: éer le script créé qui enregistre les données du formulaire en variables de session Ce script renverra une page html affichant les variables de session, un lien en bas de page permettra de retourner au menu principal. Variables de session créées Retour menuéer le script affichant les variables de session, vous ajouterez un compteur en variable de session afin d'afficher le nombre de fois que la page a été vue. Affichage des variables de session Retour menuéer le script, qui détruit les variables de session, utiliser la fonction PHP session_destroy(). Détruire les variables de session Retour menu3.Créer une table sous ORACLE une fenêtre telnet sur npx, ouvrir une session SQL*Plus sous Oracle avec la commande : \$ sqlplus saisi votre nom d'utilisateur et votre mot de passe oracle SQL*Plus changer votre mot de passe oracle avec la commande : SQL> GRANT CONNECT TO login IDENTIFIED BY passwd ; un bloc note rédiger la requête SQL permettant de créer une table ANNUAIRE donnant les numéros de poste téléphonique du personnel en fonction de leurs noms et prénoms. - La table comportera trois champs : nom, prénom, numéro de poste, déclarés en VARCHAR2 (taille) - Les numéros de poste sont de la forme 99.99 - La clé primaire sera composée des deux champs nom et prénom - Les champs nom et prénom seront déclarés NOT NULL CREATE TABLE annuaire (nom VARCHAR2(30) NOT NULL , prénom VARCHAR2(30) NOT NULL , no_poste VARCHAR2(5) , PRIMARY KEY (nom, prénom)) ; 3.4.A l'aide d'un Copier/Collier, introduire la requête SQL sous SQL*Plus et l'exécuter. 3.5.Vérifier la structure de votre table ANNUAIRE sous SQL*Plus SQL> describe annuaire éer quelques tuples dans votre table ANNUAIRE, tester: SQL> INSERT INTO annuaire VALUES ('X', 'x', '82.19') ; SQL> INSERT INTO annuaire VALUES ('Y', 'y', '82.20') ; SQL> INSERT INTO annuaire VALUES ('Z', 'z', '82.21') ; SQL> SELECT * FROM annuaire ; 3.7. Donner le droit de consultation publique sur votre table ANNUAIRE SQL> GRANT select ON annuaire TO PUBLIC ; 4.Créer les pages PHP pour consulter et mettre à jour la table annuaire éer un script php permettant d'effectuer à partir de pages web, les opérations suivantes : 1. Rechercher une entrée dans la table annuaire à partir des premières lettres du nom et/ou du prénom 2. Ajouter une nouvelle entrée dans l'annuaire 3. Modifier un numéro de poste 4. Supprimer une entrée de l'annuaire Vous utiliserez une feuille de style pour la présentation des pages web. Feuille de style Script Annuaire Téléphonique 5.Créer des menus déroulants à partir de données puisées dans des tables Nous considérons les deux relations suivantes : - CATEGORIE (id cat, designation, taux TVA) une ligne représente une catégorie de produit (clé id cat) - PRODUIT (id pro, id cat, designation, marque, prix uht, qstock) une ligne représente un produit (clé id pro) un produit appartient à une seule catégorie (clé étrangère id cat référence une catégorie) Un jeu d'essai est créé sous oracle dans le compte ET (le script ayant permis de créer les tables et le jeu d'essai est donné en Annexe2). Les tables sont accessibles en lecture pour tous les utilisateurs d'oracle. Nous voulons créer une page comportant trois cadres comme dans l'exemple ci-dessous : - le cadre en haut à gauche contient un menu permettant de sélectionner une catégorie - le cadre en haut à droite contient un deuxième menu permettant de sélectionner un produit de la catégorie qui a été sélectionnée à gauche - le cadre en bas affiche le produit sélectionné éer le script qui découpe la fenêtre en trois cadres. éer le script qui affiche dans le cadre en haut à gauche le menu permettant de sélectionner une catégorie. Lorsque l'utilisateur sélectionne une catégorie, une fonction javascript permettra d'envoyer au serveur une requête avec en paramètre l'identifiant de la catégorie sélectionnée. Ce script renverra dans le cadre à droite le menu permettant de sélectionner un produit de la catégorie qui a été passée en paramètre. Lorsque l'utilisateur sélectionne un produit, une fonction javascript permettra d'envoyer au serveur une requête avec en paramètre l'identifiant du produit sélectionné. Ce script renverra dans le cadre en bas la fiche du produit sélectionné. Choix produit éer le script qui renvoie dans le cadre en bas la fiche du produit sélectionné. 6. Annexe1 - Fonctions OCI (Oracle Call Interface) de PHP Etablit une connexion à un serveur Oracleresource ociologon (string username, string password, string db) ociologon() retourne un identifiant de connexion, nécessaire à la plus part des fonctions OCI. Si l'option ORACLE_SID n'est pas précisée, PHP utilisera la variable d'environnement ORACLE_SID pour déterminer le serveur de connexion. Les connexions sont partagées, à l'intérieur d'une même page avec ociologon().



Si l'option ORACLE_SID n'est pas précisée, PHP utilisera la variable d'environnement ORACLE_SID pour déterminer le serveur de connexion. Les connexions sont partagées, à l'intérieur d'une même page avec ociologon(). Cela signifie que COMMIT et ROLLBACK s'appliquent à toutes les transactions commencées à l'intérieur d'une même page, même si vous avez créé de multiples connexions. Déconnexion d'un serveur Oracle int ociologoff (resource connection) ociologoff() ferme la connexion Oracle. Analyse une requêteint ociparse (ocifreesc conn, strint query) ociparse() analyse la requête query sur la connexion conn, et retourne TRUE si la requête query est valide, et FALSE, si ce n'est pas le cas. query peut être n'importe quelle requête SQL. Exécute une commande int ociexecute (resource statement, int mode) ociexecute() exécute une commande déjà préparée (voir ociparse()). L'option mode vous permet de spécifier le mode d'exécution (par défaut, il est à OCI_COMMIT_ON_SUCCESS). Si vous ne voulez pas que la commande soit automatiquement validée, utilisez le mode OCI_DEFAULT. Retourne le nombre de lignes affectéesint ocirowcount (resource statement) ocirowcount() retourne le nombre de lignes affectées par une commande de modification. Cette fonction ne vous indiquera pas le nombre de lignes retournées par un SELECT : il faut que les lignes aient été modifiées. Retourne la valeur d'une colonne dans une ligne luemixed ocresult (resource statement, mixed column) ocresult() retourne les données de la colonne column dans la ligne courante (voir ocifetch()). ocifetch()retournera tout les types, sauf les types abstraits (ROWIDs, LOBs et FILES). Modifie la prochaine ligne dans le pointeur interne de résultat. int ocifetch (resource statement) ocifetch() place la prochaine ligne (d'une commande SELECT) dans le pointeur interne de résultat. Retourne la ligne suivante dans un tableautint ocifetchinto (resource stmt, array \$result, int mode) ocifetchinto() retourne la ligne suivante dans le tableauint ocifetchinto (resource stmt, array \$result, int mode) ocifetchinto() retourne la ligne suivante (pour une commande SELECT) dans le tableau result. ocifetchinto() écrasera le contenu de result. Par défaut, result sera un tableau à index numérique, commençant à 1, et qui contiendra toute les colonnes qui ne sont pas NULL. L'option mode vous permet de modifier le comportement par défaut de la fonction. Vous pouvez passer plusieurs modes simplement en les additionnant (i.e. OCI_ASSOC+OCI_RETURN_NULLS). Les modes valides sont : • OCI_ASSOC Retourne un tableau associatif.



Retourne le nombre de lignes affectéesint ocirowcount (resource statement) ocirowcount() retourne le nombre de lignes affectées par une commande de modification. Cette fonction ne vous indiquera pas le nombre de lignes retournées par un SELECT : il faut que les lignes aient été modifiées. Retourne la valeur d'une colonne dans une ligne luemixed ocresult (resource statement, mixed column) ocresult() retourne les données de la colonne column dans la ligne courante (voir ocifetch()). ocifetch()retournera tout les types, sauf les types abstraits (ROWIDs, LOBs et FILES). Modifie la prochaine ligne dans le pointeur interne de résultat. int ocifetch (resource statement) ocifetch() place la prochaine ligne (d'une commande SELECT) dans le pointeur interne de résultat. Retourne la ligne suivante dans un tableautint ocifetchinto (resource stmt, array \$result, int mode) ocifetchinto() retourne la ligne suivante dans le tableauint ocifetchinto (resource stmt, array \$result, int mode) ocifetchinto() retourne la ligne suivante (pour une commande SELECT) dans le tableau result. ocifetchinto() écrasera le contenu de result. Par défaut, result sera un tableau à index numérique, commençant à 1, et qui contiendra toute les colonnes qui ne sont pas NULL. L'option mode vous permet de modifier le comportement par défaut de la fonction. Vous pouvez passer plusieurs modes simplement en les additionnant (i.e. OCI_ASSOC+OCI_RETURN_NULLS). Les modes valides sont : • OCI_ASSOC Retourne un tableau associatif. • OCI_NUM Retourne un tableau à index numérique (DEFAULT, valeur par défaut) • OCI_RETURN_NULLS Retourne les colonnes vides. • OCI_RETURN_LOBS Retourne la valeur des objets LOB plutôt que leur descripteur. Retourne toutes les lignes d'un résultatint ocifetchstatement (resource stmt, array \$variable) ocifetchstatement() retourne toutes les lignes d'un résultat dans le tableau variable. ocifetchstatement() retourne le nombre de lignes retournées. Teste si la valeur d'une colonne est NULL int ocicolumnisnull (resource stmt, mixed column) ocicolumnisnull() retourne TRUE si la colonne col du résultat stmt est NULL.

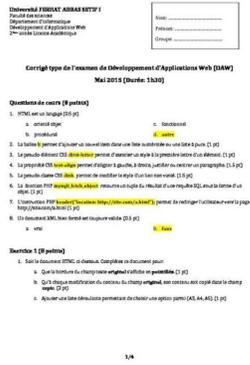


Ce script renverra dans le cadre en bas la fiche du produit sélectionné. Choix produit éer le script qui renvoie dans le cadre en bas la fiche du produit sélectionné. 6. Annexe1 - Fonctions OCI (Oracle Call Interface) de PHP Etablit une connexion à un serveur Oracleresource ociologon (string username, string password, string db) ociologon() retourne un identifiant de connexion, nécessaire à la plus part des fonctions OCI. Si l'option ORACLE_SID n'est pas précisée, PHP utilisera la variable d'environnement ORACLE_SID pour déterminer le serveur de connexion. Les connexions sont partagées, à l'intérieur d'une même page avec ociologon(). Cela signifie que COMMIT et ROLLBACK s'appliquent à toutes les transactions commencées à l'intérieur d'une même page, même si vous avez créé de multiples connexions. Déconnexion d'un serveur Oracle int ociologoff (resource connection) ociologoff() ferme la connexion Oracle. Analyse une requêteint ociparse (ocifreesc: conn, strint query) ociparse() analyse la requête query sur la connexion conn, et retourne TRUE si la requête query est valide, et FALSE, si ce n'est pas le cas. query peut être n'importe quelle requête SQL. Exécute une commande int ociexecute (resource statement, int mode) ociexecute() exécute une commande déjà préparée (voir ociparse()). L'option mode vous permet de spécifier le mode d'exécution (par défaut, il est à OCI_COMMIT_ON_SUCCESS). Si vous ne voulez pas que la commande soit automatiquement validée, utilisez le mode OCI_DEFAULT. Retourne le nombre de lignes affectéesint ocirowcount (resource statement) ocirowcount() retourne le nombre de lignes affectées par une commande de modification. Cette fonction ne vous indiquera pas le nombre de lignes retournées par un SELECT : il faut que les lignes aient été modifiées. Retourne la valeur d'une colonne dans une ligne luemixed ocresult (resource statement, mixed column) ocresult() retourne les données de la colonne column dans la ligne courante (voir ocifetch()). ocifetch()retournera tout les types, sauf les types abstraits (ROWIDs, LOBs et FILES). Modifie la prochaine ligne dans le pointeur interne de résultat. int ocifetch (resource statement) ocifetch() place la prochaine ligne (d'une commande SELECT) dans le pointeur interne de résultat. Retourne la ligne suivante dans un tableautint ocifetchinto (resource stmt, array \$result, int mode) ocifetchinto() retourne la ligne suivante (pour une commande SELECT) dans le tableau result. ocifetchinto() écrasera le contenu de result. Par défaut, result sera un tableau à index numérique, commençant à 1, et qui contiendra toute les colonnes qui ne sont pas NULL. L'option mode vous permet de modifier le comportement par défaut de la fonction. Vous pouvez passer plusieurs modes simplement en les additionnant (i.e. OCI_ASSOC+OCI_RETURN_NULLS).



Lorsque l'utilisateur sélectionne une catégorie, une fonction javascript permettra d'envoyer au serveur une requête avec en paramètre l'identifiant de la catégorie sélectionnée. Ce script renverra dans le cadre à droite le menu permettant de sélectionner un produit de la catégorie qui a été passée en paramètre. Lorsque l'utilisateur sélectionne un produit, une fonction javascript permettra d'envoyer au serveur une requête avec en paramètre l'identifiant du produit sélectionné. Ce script renverra dans le cadre en bas la fiche du produit sélectionné. Choix produit éer le script qui renvoie dans le cadre en bas la fiche du produit sélectionné. 6. Annexe1 - Fonctions OCI (Oracle Call Interface) de PHP Etablit une connexion à un serveur Oracleresource ociologon (string username, string password, string db) ociologon() retourne un identifiant de

connexion, nécessaire à la plus part des fonctions OCI. Si l'option ORACLE_SID n'est pas précisée, PHP utilisera la variable d'environnement ORACLE_SID pour déterminer le serveur de connexion. Les connexions sont partagées, à l'intérieur d'une même page avec ociagon(). Cela signifie que COMMIT et ROLLBACK s'appliquent à toutes les transactions commencées à l'intérieur d'une même page, même si vous avez créé de multiples connexions. Déconnexion d'un serveur Oracle int ociagoff (resource connection) ociagoff() ferme la connexion Oracle. Analyse une requêteint ociparse (ocifreedesc conn, strint query) ociparse() analyse la requête query sur la connexion conn, et retourne TRUE si la requête query est valide, et FALSE, si ce n'est pas le cas. query peut être n'importe quelle requête SQL. Exécute une commande int ociexecute (resource statement, int mode) ocieecute() exécute une commande déjà préparée (voir ociparse()). L'option mode vous permet de spécifier le mode d'exécution (par défaut, il est à OCI_COMMIT_ON_SUCCESS). Si vous ne voulez pas que la commande soit automatiquement validée, utilisez le mode OCI_DEFAULT. Retourne le nombre de lignes affectéesint ociorowcount (resource statement) ociorowcount() retourne le nombre de lignes affectées par une commande de modification. Cette fonction ne vous indiquera pas le nombre de lignes retournées par un SELECT : il faut que les lignes aient été modifiées. Retourne la valeur d'une colonne dans une ligne luemixed ociresult (resource statement, mixed column) ociresult() retourne les données de la colonne column dans la ligne courante (voir ocifetch()). ocifetch()retournera tout les types, sauf les types abstraits (ROWIDS, LOBs et FILEs). Modifie la prochaine ligne dans le pointeur interne de résultat. int ocifetch (resource statement) ocifetch() place la prochaine ligne (d'une commande SELECT) dans le pointeur interne de résultat. Retourne la ligne suivante dans un tableauint ocifetchinto (resource stmt, array &result, int mode) ocifetchinto() retourne la ligne suivante (pour une commande SELECT) dans le tableau result.



Cela signifie que COMMIT et ROLLBACK s'appliquent à toutes les transactions commencées à l'intérieur d'une même page, même si vous avez créé de multiples connexions. Déconnexion d'un serveur Oracle int ociagoff (resource connection) ociagoff() ferme la connexion Oracle. Analyse une requêteint ociparse (ocifreedesc conn, strint query) ociparse() analyse la requête query sur la connexion conn, et retourne TRUE si la requête query est valide, et FALSE, si ce n'est pas le cas. query peut être n'importe quelle requête SQL. Exécute une commande int ociexecute (resource statement, int mode) ocieexecute() exécute une commande déjà préparée (voir ociparse()). L'option mode vous permet de spécifier le mode d'exécution (par défaut, il est à OCI_COMMIT_ON_SUCCESS). Si vous ne voulez pas que la commande soit automatiquement validée, utilisez le mode OCI_DEFAULT. Retourne le nombre de lignes affectéesint ociorowcount (resource statement) ociorowcount() retourne le nombre de lignes affectées par une commande de modification. Cette fonction ne vous indiquera pas le nombre de lignes retournées par un SELECT : il faut que les lignes aient été modifiées. Retourne la valeur d'une colonne dans une ligne luemixed ociresult (resource statement, mixed column) ociresult() retourne les données de la colonne column dans la ligne courante (voir ocifetch()). ocifetch()retournera tout les types, sauf les types abstraits (ROWIDS, LOBs et FILEs). Modifie la prochaine ligne dans le pointeur interne de résultat. int ocifetch (resource statement) ocifetch() place la prochaine ligne (d'une commande SELECT) dans le pointeur interne de résultat. Retourne la ligne suivante dans un tableauint ocifetchinto (resource stmt, array &result, int mode) ocifetchinto() écrasera le contenu de result. Par défaut, result sera un tableau à index numérique, commençant à 1, et qui contiendra toute les

colonnes qui ne sont pas NULL. L'option mode vous permet de modifier le comportement par défaut de la fonction. Vous pouvez passer plusieurs modes simplement en les additionnant (i.e. OCI_ASSOC+OCI_RETURN_NULLS). Les modes valides sont :

- OCI_ASSOC Retourne un tableau associatif.
- OCI_NUM Retourne un tableau à index numérique (DEFAULT, valeur par défaut)
- OCI_RETURN_NULLS Retourne les colonnes vides.
- OCI_RETURN_LOBS Retourne la valeur des objets LOB plutôt que leur descripteur. Retourne toutes les lignes d'un résultatint ocifetchstatement (resource stmt, array &variable) ocifetchstatement() retourne toutes les lignes d'un résultat dans le tableau variable. ocifetchstatement() retourne le nombre de lignes retournées. Teste si la valeur d'une colonne est NULL int ocicolumnisnull (resource stmt, mixed column) ocicolumnisnull() retourne TRUE si la colonne col du résultat stmt est NULL. Vous pouvez utiliser le numéro de colonne (l'indexation des colonnes commence à 1) ou le nom de la colonne, pour le paramètre col. Validez les transactions en coursint ocicommit (resource connection) ocicommit() valide toutes les transactions en cours sur la connexion Oracle connection. Annule les transactions en cours int ociorollback (resource connection) ociorollback() annule les transactions en cours sur la connexion Oracle connection. Libère toutes les ressources occupées par une commande. int ocifreestatement (resource stmt) ocifreestatement() retourne TRUE en cas de succès, et FALSE en cas d'échec. Retourne la dernière erreur de stmt[conn]global. array ociorr (int stmt[conn]) ociorr() retourne la dernière erreur trouvée. Si l'option stmt[conn n'est pas fournie, la dernière erreur rencontrée est retournée. Si aucune erreur n'est trouvée, ociorr() retourne FALSE. 7. Annexe2 - jeu d'essai (compte ET sous oracle) DROP TABLE produit; DROP TABLE categorie; CREATE TABLE categorie (id_cat VARCHAR2 (2) PRIMARY KEY , designation VARCHAR2 (50) , taux_tva NUMBER (4,2)); CREATE TABLE produit (id_pro VARCHAR2 (8) PRIMARY KEY , id_cat VARCHAR (2) REFERENCES categorie , designation VARCHAR2 (50) , marque VARCHAR2 (50) , prix_uht NUMBER (9,2) , qstock NUMBER (2)); /* jeu d'essai */ INSERT INTO categorie VALUES ('C1', 'Ordinateurs', 10); INSERT INTO categorie VALUES ('C2', 'Logiciels', 20); INSERT INTO categorie VALUES ('C3', 'Imprimantes', 30); INSERT INTO produit VALUES ('10', 'C1', 'ps', 'ibm', 500.00, 10); INSERT INTO produit VALUES ('20', 'C1', 'mac', 'apple', 750.00, 10); INSERT INTO produit VALUES ('30', 'C1', 'aptiva', 'ibm', 1200.00, 10); INSERT INTO produit VALUES ('40', 'C1', 'power mac', 'apple', 2000.00, 10); INSERT INTO produit VALUES ('50', 'C2', 'word', 'microsoft', 100.00, 10); INSERT INTO produit VALUES ('60', 'C2', 'access', 'microsoft', 100.00, 10); INSERT INTO produit VALUES ('70', 'C2', 'paradox', 'borland', 80.00, 30); Tables sous MySql CREATE TABLE IF NOT EXISTS 'annuaire' (`id ann` int(11) NOT NULL auto increment, `nom` varchar(50) NOT NULL, `prenom` varchar(35) NOT NULL, `no_poste` varchar(5), PRIMARY KEY (`id ann`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO INCREMENT=3; -- Contenu de la table `annuaire` -- INSERT INTO `dept` VALUES (1, 'Dupont', 'Jean', '82.19'); INSERT INTO `dept` VALUES (2, 'Martin', 'Paul', '81.12'); -- ----- Structure de la table `categorie` -- CREATE TABLE IF NOT EXISTS 'categorie' (`id_cat` int(11) NOT NULL auto increment, `designation` varchar(30) NOT NULL, `tva` float NOT NULL, PRIMARY KEY (`id_cat`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO INCREMENT=11; -- Structure de la table `produit` -- CREATE TABLE IF NOT EXISTS 'produit' (`id_pro` int(11) NOT NULL auto increment, `id_cat` int(11) REFERENCES categorie, `designation` varchar(30), `marque` varchar(30), `prix_uht` float, `qstock` int, PRIMARY KEY (`id_pro`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO INCREMENT=11; Type de fichier : pdfTaille du fichier : 108.01 Kb Télécharger Type de fichier : pdfTaille du fichier : 1,130.17 Kb Télécharger Type de fichier : pdfTaille du fichier : 607.98 Kb Télécharger Type de fichier : pdfTaille du fichier : 869.60 Kb Télécharger Type de fichier : pdfTaille du fichier : 335.84 Kb Télécharger Type de fichier : pdfTaille du fichier : 185.77 Kb Télécharger