

ORACLE®

E-BUSINESS SUITE

TD3 : Conception

TABLE DES MATIERES

Table des matières	2
Introduction	3
DESCRIPTION DE LA DEMANDE.....	4
INTERFACE UTILISATEUR	4
LISTE DES COMMANDES ACTIVES	4
PARAMETRAGE DU TRAITEMENT.....	5
CREATION DU L'EXECUTABLE ET DU TRAITEMENT	5
LISTE DES PARAMETRES	5
CREATION DU JEU DE VALEURS	7
CREATION OPTION DE PROFIL.....	8
MESSAGES	8
TABLES ET CHAMPS.....	9
OUTILS POUR LE DEVELOPPEMENT	10
RECUPERATION DU MONTANT	10
PREMIER CURSEUR.....	10
DEUXIEME CURSEUR	11
TROISIEME CURSEUR	11
APPEL DE NEW TEST.....	11
MODELE DE DONNEES	12
CONCLUSION.....	13

INTRODUCTION

Dans le cadre du projet avec notre nouveau client, ces derniers nous ont demandé de réaliser un nouveau composant spécifique à l'ERP Oracle E-Business Suite.

Nous devons impérativement mettre ces développements pour la fin du mois de Janvier. Au vu de la période actuelle, nous avons décidé de rédiger une documentation permettant de vous guider sur les développements demandés.

Cette documentation liste les différents éléments techniques qui devront être développés. La liste des tables et champs à utiliser., les différents éléments à paramétrer ainsi que toutes les informations nécessaires afin de développer cette solution.

DESCRIPTION DE LA DEMANDE

Cette partie permet de résumer le besoin du client. Les parties suivantes évoqueront les outils qui vont permettre de transformer ces demandes fonctionnelles en solution technique.

INTERFACE UTILISATEUR

Un écran permettant de lister l'ensemble des agences de l'entreprise. On peut y sélectionner une agence avec une case à cocher, choisir une date et lancer un traitement en cliquant sur un bouton.

LISTE DES COMMANDES ACTIVES

- Notre traitement lancé par l'écran devra :
 - Lister les commandes créées le jour choisi en paramètre et sur l'organisation choisie en paramètre. Les commandes récupérées devront être à un certain statut d'approbation défini au niveau d'Oracle Applications.
 - Pour chaque commande, le nom de l'acheteur devra être affiché
 - Pour chaque commande, le traitement donnera le nom du fournisseur associé et tous ces numéros de téléphone connus (appartient aux « contacts » des fournisseurs).
 - Les supérieurs hiérarchiques doivent contrôler certaines commandes. Il s'agit donc des commandes étant dans un certain état. Dans la sortie, l'état devra être affiché ainsi qu'une mention précisant si la commande est « à valider ». La liste de ces états est commune à toute l'entreprise (par exemple, une agence ne peut pas choisir un statut en plus ou en moins pour ces validations). De plus, les commandes dépassant un certain montant total TTC devront être également notées « A valider ». Chaque utilisateur lançant le traitement aura un montant de paramétré, qu'il pourra consulter mais non modifier (choisi par son responsable donc ; en général, tous les subordonnés d'un manager auront le même montant).
 - Pour chaque commande, on affichera le nom (= la description) de chaque article commandé.
 - Enfin, vous indiquerez si une facture AP a été générée pour cette commande (existe-elle ?) et, si oui, vous afficherez son numéro.
- La sortie sera au format PDF, la maquette d'affichage de sortie sera fournie par le client sous forme de fichier Word.
- Le traitement NEWTEST sera lancé automatiquement à la fin de cette extraction. Ce traitement ne fait pour l'instant rien, son contenu sera rempli par la suite (à ne pas faire dans ce projet) par de l'affichage de logs. L'utilisateur pourra choisir ou non, grâce à une coche que l'on ajoutera dans l'écran, s'il souhaite lancer ce sous traitement.
 - La première version de l'écran ne contiendra pas cette coche

- Néanmoins, la première version du traitement devra avoir implémenté cette fonctionnalité entièrement. Le code lançant le sous traitement sera donc désactivé, il devra pouvoir être réactivé facilement, sans modification du code.
- L'objectif des phases suivantes est de limiter le coût de l'ajout de NEWTEST en : développant le contenu de NEWTEST (incontournable), activant tous les éléments nécessaires au traitement d'extraction afin qu'il soit pris en charge, ne redéveloppant que l'écran (ajout de la coche).

PARAMETRAGE DU TRAITEMENT

Cette partie va permettre de lister les paramétrages nécessaires à réaliser au sein de l'ERP :

CREATION DU L'EXECUTABLE ET DU TRAITEMENT

Le traitement utilisé va consister à exécuter un script PL/SQL que vous aurez créé. La première tâche à réaliser est donc :

- Créer l'exécutable qui va pointer vers votre procédure PL/SQL
- Créer le traitement lié à cet exécutable

Nous n'avons pour le moment pas de véritables contraintes en ce qui concerne le nom de ces éléments.

LISTE DES PARAMETRES

Notre traitement va contenir 3 paramètres :

- Un premier paramètre, nommé p_date contiendra la date de la commande. Ce paramètre aura les spécificités suivantes :

Seq	Parameter	Description	Enabled
1	p_date	Date commande	<input checked="" type="checkbox"/>
2	p_agence	Liste des agences	<input checked="" type="checkbox"/>
3	p_newtest	lancement du newtest	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Validation

Value Set **DATE_LIST_STANDARD**
Default Type
☐ Required ☐ Enable Security

Description
Default Value
Range

Display

Display Size **11**
Concatenated Description Size **25**

Description Size **50**
Prompt **p_date**

Le format de la date sera le suivant : DD-MMM-AAAA. Grâce à la Value Set « DATE_LIST_STANDARD », même si l'utilisateur rentre un mauvais format de date, il en sera notifié par l'ERP. Il n'y aura donc aucun paramétrage à faire sur cette partie.

- Un deuxième paramètre « p_agence » va nous permettre de pouvoir cibler l'agence dont nous souhaitons afficher les commandes.

Concurrent Program Parameters

Program: TD4: LEMERET THIRET
Application: Public Sector Financials

Conflicts Domain: Security Group:

Seq	Parameter	Description	Enabled
1	p_date	Date commande	<input checked="" type="checkbox"/>
2	p_agence	Liste des agences	<input checked="" type="checkbox"/>
3	p_newtest	lancement du newtest	<input checked="" type="checkbox"/>

Validation

Value Set: GRP3_LOV_AGENCE Description: Liste des agences de l'orgar
Default Type: Default Value:
☐ Required ☐ Enable Security Range:

Display

Display Size: 100 Description Size: 50
Concatenated Description Size: 25 Prompt: p_agence

Token:

Ce paramètre utilisera le jeu de valeur que nous allons définir (nous l'évoquerons par la suite). L'utilisateur pourra donc cibler l'agence qui l'intéresse

- Un troisième paramètre « p_newtest » permettra à l'utilisateur de décider s'il doit lancer le traitement « NEW TEST » ou non. Nous utiliserons donc la Value Set « YES_NO ». L'utilisateur aura juste à rentrer les valeurs Y ou N. En cas de saisie non valide, notre ERP informera l'utilisateur. Si ce paramètre n'est pas renseigné, sa valeur sera négative par défaut.

Concurrent Program Parameters

Program: TD4: LEMERET THIRET
Application: Public Sector Financials

Conflicts Domain: Security Group:

Seq	Parameter	Description	Enabled
1	p_date	Date commande	<input checked="" type="checkbox"/>
2	p_agence	Liste des agences	<input checked="" type="checkbox"/>
3	p_newtest	lancement du newtest	<input checked="" type="checkbox"/>

Validation

Value Set: Yes_No
Default Type:
☐ Required ☐ Enable Security

Description: This value set only contains
Default Value:
Range:

Display

Display Size: 50
Concatenated Description Size: 25
Description Size: 50
Prompt: p_newtest

Token:

CREATION DU JEU DE VALEURS

Comme évoqué lors de la présentation du paramètre « p_agence_ ». Le jeu de valeur pointera vers la table « HR_ALL_ORGANIZATION_UNITS » qui contient la liste des différentes agences.

Ce jeu de valeurs pourra être défini comme suit :

Value Sets

Value Set Name: GRP3_LOV_AGENCE [Usages](#)

Description: Liste Agences Organisations

List Type: List of Values Security Type: No Security

Format Validation

Format Type: Char
☐ Numbers Only (0-9)
☐ Uppercase Only (A-Z)
☐ Right-justify and Zero-fill Numbers (0001)

Maximum Size: 200 Precision:

Min Value: Max Value:

Value Validation

Validation Type: Table [Edit Information](#)

En éditant les informations, on doit définir les colonnes comme suit :

Validation Table Information

Table Application: Table Name:

☐ Allow Parent Values

Table Columns

	Name	Type	Size
Value	<input type="text" value="NAME"/>	<input type="text" value="Varchar2"/>	<input type="text" value="240"/>
Meaning	<input type="text"/>	<input type="text" value="Char"/>	<input type="text"/>
ID	<input type="text" value="ORGANIZATION_ID"/>	<input type="text" value="Number"/>	<input type="text" value="22"/>

Where/Order By:

CREATION OPTION DE PROFIL

Comme précisé dans le besoin du client, nous devons pouvoir paramétrer entrer un montant maximal. Ce montant n'est pas modifiable par l'utilisateur en lui-même.

L'administrateur peut donc définir ce montant max, comme vu ci-dessous dans notre exemple il est à 4000 :

System Profile Values

Profile Option Name	Site
<input type="text" value="GRP7_LEMERET_THIRET_TD4_MON"/>	<input type="text" value="4000"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>

MESSAGES

Certains messages devront être configurés. Nous aurons des messages sur la sortie dans les cas :

- Si la date entrée en paramètre est supérieure à la date du jour

Nous aurons également des messages dans le journal :

- Indiquant le début de la procédure

- Indiquant la date de début de la procédure
- Indiquant la date de fin de la procédure
- Indiquant la fin de la procédure
- Indiquant si le traitement NEW TEST a été appelé

TABLES ET CHAMPS

Lors de vos développements, vous aurez besoin d'utiliser certaines tables de l'application. Je vais répertorier ces dernières avec la liste des champs qui pourraient vous être utile :

- PO_HEADERS_ALL : contient les commandes
 - PO_HEADERS_ID
 - VENDOR_ID : Clé étrangère vers la table PO_VENDORS
 - VENDOR_CONTACT_ID : Clé étrangère vers la table PO_VENDOR_CONTACT
 - VENDOR_SITE_ID : Clé étrangère vers la table PO_VENDOR_SITE
 - AGENT_ID : Clé étrangère vers la table HR_EMPLOYEES
 - ORGANIZATION_ID : Clé étrangère vers la table HR_ORGANIZATION_UNITS
 - AMOUT_LIMIT
- HR_ALL_ORGANIZATION_UNITS : Table qui contient les agences de l'entreprise
 - ORGANIZATION_ID
 - NAME
- PO_LINES_ALL : contient les informations sur chaque ligne de commandes :
 - ITEM_DESCRIPTION
 - UNIT_PRICE
 - QUANTITY
 - PO_HEADER_ID : Clé étrangère vers la table PO_HEADERS_ALL
- PO_VENDORS : contient les informations concernant les fournisseurs :
 - VENDOR_ID
 - VENDOR_NAME
- PO_VENDOR_CONTACTS : contient les informations de contacts des fournisseurs :

- PHONE :
- PO_VENDOR_SITE_ID : Clé étrangère vers la table PO_VENDOR_SITES_ALL
- PO_VENDORS_SITES_ALL : contient les sites des fournisseurs
 - VENDOR_SITE_ID
 - PHONE
 - VENDOR_ID : Clé étrangère vers la table PO_VENDOR
- PO_DISTRIBUTIONS_ALL : contient les distributions des bon de commandes
 - PO_DISTRIBUTION_ID
 - PO_HEADER_ID : Clé étrangère vers la table PO_HEADERS_ALL
- AP_INVOICE_DISTRIBUTIONS_ALL : contient des informations sur la distribution de la facture
 - PO_DISTRIBUTIONS_ID : Clé étrangère vers la table PO_DISTRIBUTIONS_ALL
 - INVOICE_ID : Clé étrangère vers la table AP_INVOICE_ALL
- AP_INVOICES_ALL : contient des informations sur les factures
 - INVOICE_ID
 - INVOICE_NUM
- HR_EMPLOYEES : contient les informations sur les acheteurs
 - EMPLOYEE_ID
 - FULL_NAME

OUTILS POUR LE DEVELOPPEMENT

Comme vous en avez l'habitude, un script PL/SQL devra être développé afin de réaliser ce dont nous avons besoin.

RECUPERATION DU MONTANT

Comme vu précédemment, nous avons défini une option de profil avec le montant maximal. Il nous sera possible de le récupérer de la manière suivante :

```
--On recupere le montant definit dans l'option de profile
p_montant varchar2(1000) := FND_PROFILE.VALUE('GRP3_LEMERET_THIRET_TD4_MONTANT');
```

PREMIER CURSEUR

Comme vous l'avez vu précédemment, il y a un bon nombre de liens entre nos différentes tables. Ainsi avec les différents liens, il sera facile d'établir un curseur ou une requête va nous permettre de récupérer les informations de notre commande.

En partant de PO_HEADERS_ALL, nous pourrions donc, récupérer le nom des fournisseurs, le numéro de la commande, etc...

Il nous sera également possible de calculer le montant en utilisant la table PO_LINES_ALL qui contient toutes les lignes de notre commande. Une simple Somme devra être mise en place afin de récupérer le montant de chaque ligne et de calculer finalement le montant total de notre commande. Cependant, si le montant excède la valeur indiquée dans l'option de profile créée précédemment, la commande sera au statut : « A VALIDER ».

DEUXIEME CURSEUR

Nous devons également faire en sorte de pouvoir récupérer la description de chaque article comamndé.

Ainsi, nous avons pensé, que développer un 2ème curseur sera bénéfique, afin d'avoir un code qui est le plus lisible possible.

En réutilisant la table PO_LINES_ALL, nous pourrions récupérer chaque article de notre commande.

Et ainsi afficher la description de chacun de nos articles. Ce curseur pourrait avoir la forme suivante :

```
CURSOR cursor_ligne (id_commande INTEGER) IS
  select poh.po_header_id,
         REPLACE(poh.item_description,',', ' ') AS Description_ligne,
         poh.item_id AS item_id
  from po_lines_all poh
 where po_header_id = id_commande;
```

TROISIEME CURSEUR

Nous voulons également pouvoir récupérer la ou les factures qui seraient liées à notre commande. Un troisième curseur permettra de faire cela.

La table distributions_all contient les distributions des bons de commandes. En plus de cela, des liens existe vers la table AP_INVOICE_DISTRIBUTION_ALL. Grâce à ce lien, il nous sera possible de récupérer facilement les numéros de factures liés à notre commande.

Ce curseur pourrait avoir la forme suivante :

```
CURSOR cursor_facture (id_commande INTEGER) IS
  select
    ai.invoice_num as numero_facture
  from po_distributions_all pda
  left join
    AP_INVOICE_DISTRIBUTIONS_ALL aid on pda.po_distribution_id = aid.po_distribution_id
  left join
    AP_INVOICES_ALL ai on aid.invoice_id = ai.invoice_id
  where pda.po_header_id = id_commande;
```

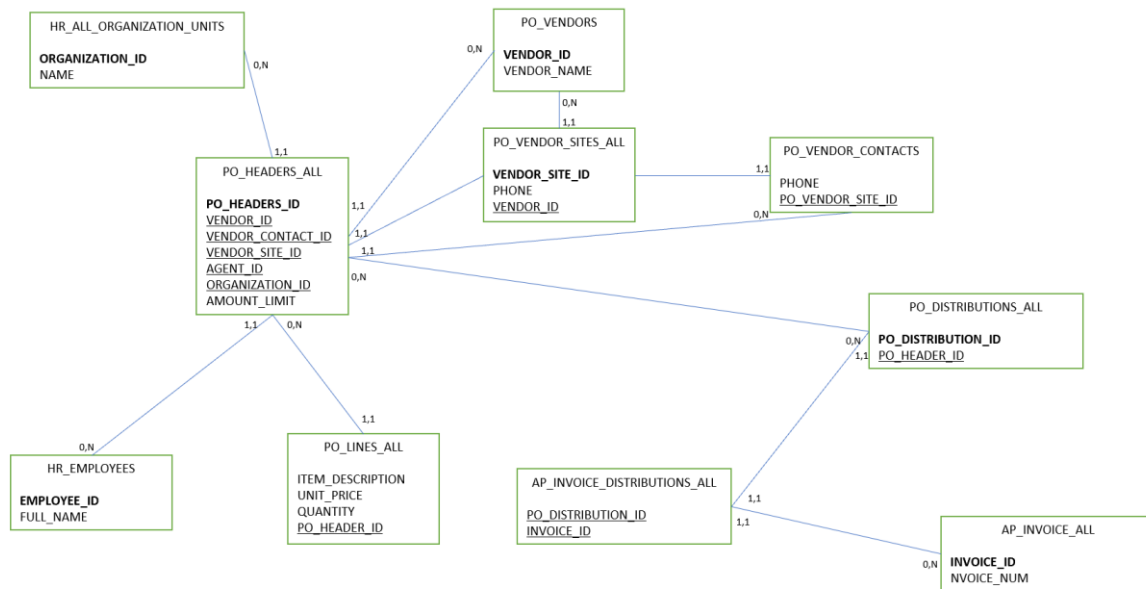
APPEL DE NEW TEST

Comme évoqué auparavant, nous aurions besoin d'appeler par la suite le traitement NEW TEST.

Il faudra donc récupérer ce paramètre. Si l'utilisateur a décidé de lancer ce traitement, nous devons donc appeler depuis notre code, le traitement NEW_TEST

MODELE DE DONNEES

Afin d'effectuer un récapitulatif de ces éléments, nous avons décidé de réaliser un Modèle Entité Association qui nous permet d'avoir une vision globale des tables et des champs qui nous seront utiles :



CONCLUSION

Cette documentation va vous permettre de comprendre le besoin du client et de pouvoir le retranscrire techniquement.

Notre objectif est donc de développer un traitement listant les commandes de l'agence qui aura été choisie en paramètre.

Nous avons via ce document lister les paramétrages et les différents champs et tables qui vous seront nécessaires. Nous avons par ailleurs indiqué quelques méthodes de développements qui pourraient facilement répondre à ce besoin.

Notre script devra donc appeler par la suite le traitement NEW_TEST, quand ce dernier aura été développé lors de la suite de notre projet.