

Stage d'Initiation

Elève Ingénieur En 1^{ère} Année Génie Informatique

Réalisé au sein de : YaaS



SUJET

- Initiation au Progiciel SAP, et pratique du langage de programmation SAP/ABAP.

Période de stage : Du 07/07/2020 au 08/08/2020.

Réalisé par :

MOUHIM Ghita

Encadré par :

Mr. OUZAYD Oussama

Présenté le **

/**/2020 Devant le Jury composé de :

- Pr :
- Pr :

Année Universitaire : 2019/ 2020

Remerciement :

Je tiens à remercier

ALLAH,

Puis mes parents, mes professeurs

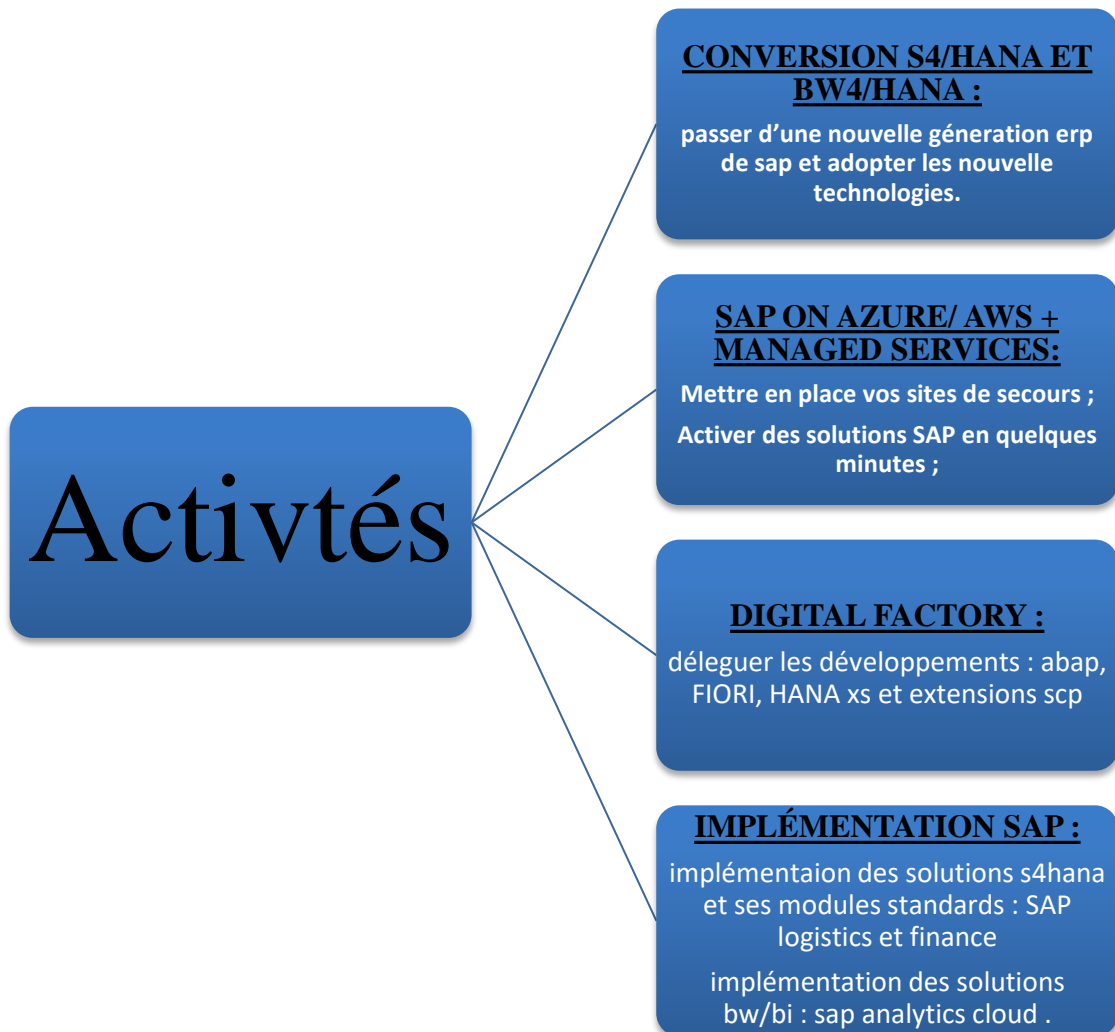
et toute personne qui m'a aidé

avec le minimum ou le maximum.

Présentation de la société :

Yaas est une société Franco-Marocaine, fondée par des experts en système d'informations avec une expérience technique et fonctionnelles.

Parmi ses activités :



Plan :

- Introduction.
- Formation.
- Ateliers.
- Conclusion.

Introduction :

A cause du confinement il y avait un grand problème de trouver stage spécialement en présentiel, pour cela la plupart des sociétés acceptaient des stagiaires à distance à l'aide de LINKEDIN c'était le cas pour YaaS.

Je les ai contacté et m'ont accepté autant que stagiaire débutante dans le domaine d'informatique, ils m'ont proposé de le faire dans SAP/ ABAP.

Au début j'étais surprise car je n'ai jamais entendu de telle chose, mais la chose qui m'a encouragé le faire c'est de me faire une formation et puis des ateliers j'ai accepté pour avoir cette expérience et découvrir un nouvel langage et enrichir mes compétences dans le domaine d'informatique.

I. Formation :

La formation consiste en premier dans introduction de programmer en ABAP, celle-ci était en 5 parties : (la formation était en anglais)

- 1- Les bases de programmation en ABAP ;
- 2- Le codage et le débogage en ABAP ;
- 3- Les conditions et les loops ;
- 4- Les erreurs ;
- 5- Des techniques additionnelles pour programmer en ABAP.

1- Les bases de programmation en ABAP :

a- Développer un simple ABAP program :

Tout d'abord on lance soit SAP logon soit à partir de l'installation d'ABAP editor dans Eclipse,

Connexion_variable...

Vue de l'espace de t \

Connexions

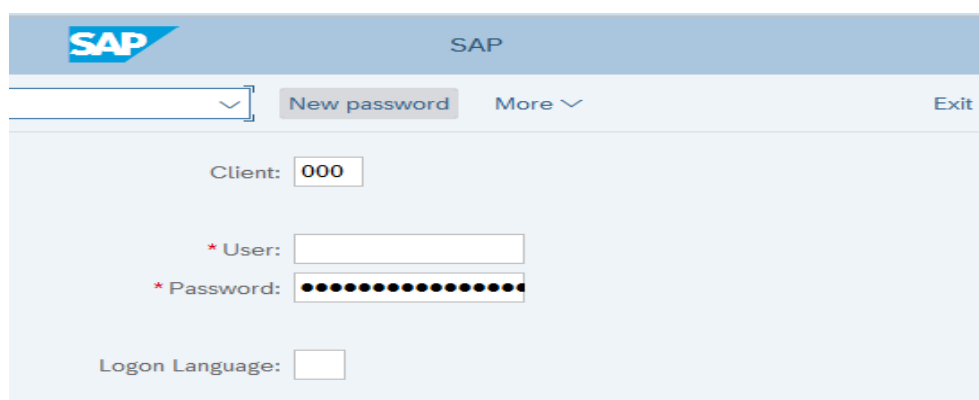
Filtrer objets

Workspaces

Local

Nom	Description système	SID	Groupe/Serveur	N° ...	Serveur de messages	Routeur
<div><div></div>S4D [51...</div>		S4D	51.91.82.152	00		

On fait entrer les données concernant la connexion au domaine de SAP, puis on se connecte dans cette fenêtre :



SAP

Client: 000

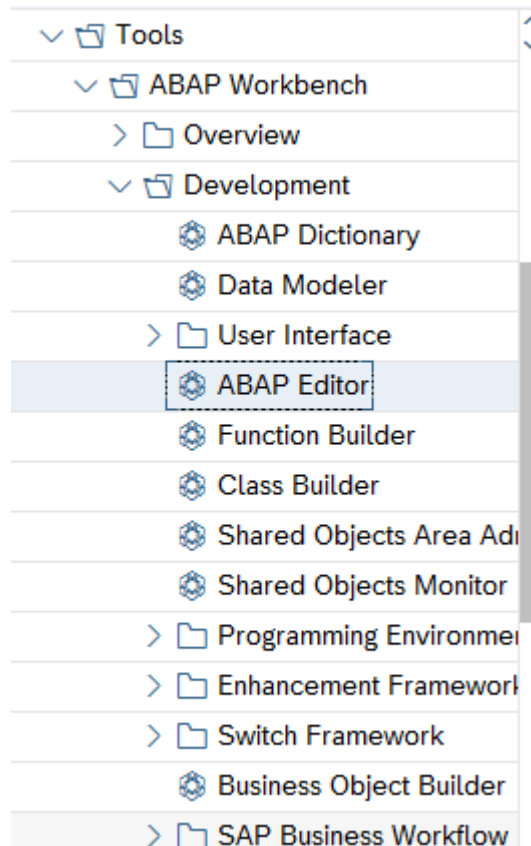
* User:

* Password:

Logon Language:

New password More Exit

En suivant ces étapes on obtient la fenêtre du programme qu'on va utiliser au fur et au mesure :



Les points suivants sont obligatoires :

- Le nom du programme doit commencer nécessairement par : « y » ou « z ».
- Pour les attributs : le titre est obligatoire et doit être nommé différemment du programme.
- Le type du programme doit être défini on s'intéresse aux programmes exécutables.
- Le programme peut être soit sous forme de package (dossier) ou local.

Les principes de la syntaxe lors de la programmation :

- L'espace est nécessaire entre deux mots et spécialement après « **write** ».
- On peut avoir une alignement au début de chaque déclaration.
- Il y a des mots « **ABAP KEYWORD** » qui débute les déclarations.
- Il faut noter bien qu'ABAP ne fait pas différence entre majuscule et minuscule ...

b- La syntaxe d'ABAP :

- **Définitions :**

ABAP: est abréviation d'Advanced Business Application Programming

Il s'agit d'un langage de programmation développé par SAP pour programmer des applications commerciales dans un environnement de SAP.

SAP: signifie « Systems, Applications & Products in data processing ».

Il s'agit d'un logiciel de **planification des ressources d'entreprise** auquel nous pouvons également nous référer comme étant un **système centralisé** et la mise en marché comme un système de comptabilité financière.

- **Caractéristiques d'ABAP :**

- Typique ;
- Active pour les applications multilingues ;
- Actif pour SQL ;
- Il est parmi les langages orientés objets ;
- Indépendant du système d'exploitation utilisé (la même syntaxe dans tous les OS) ;

- **Déclaration dans ABAP :**

- On utilise des mots clés.
- A la fin de chaque déclaration on doit mettre «.»
- Les mots sont séparés avec espace
- On peut avoir des opérations et plusieurs lignes ;
- Pour faire un ensemble de paramètres du même mot clés est de la forme suivante :

PARAMETERS: p type string, in type i.

PARAMETERS est un mot clés qui définit les entrées comme entrée de sélection, qui est une fois remplie elle est stockée dans une variable qui peut être utilisée après.

Pour traduire le texte en autre langue on suit les étapes suivantes :

Go to → Text Element → selection Texts

Go to → translation.

- **Commentaires :**

Si on veut que toute la ligne soit commentaire on utilise *, si on veut insérer dans n'importe quelle position on utilise ".

- **Les types :**

Type	Meaning	Use
String	Variable length character string	Any character has an arbitrary length
C	Fixed length character string	Any character has fixed length
D	Date	YYYY MM DD
T	Time	HH MM SS
I	Integer	Whole numbers (no decimal places)
P	Packed number	Decimal numbers

- **Les listes :**

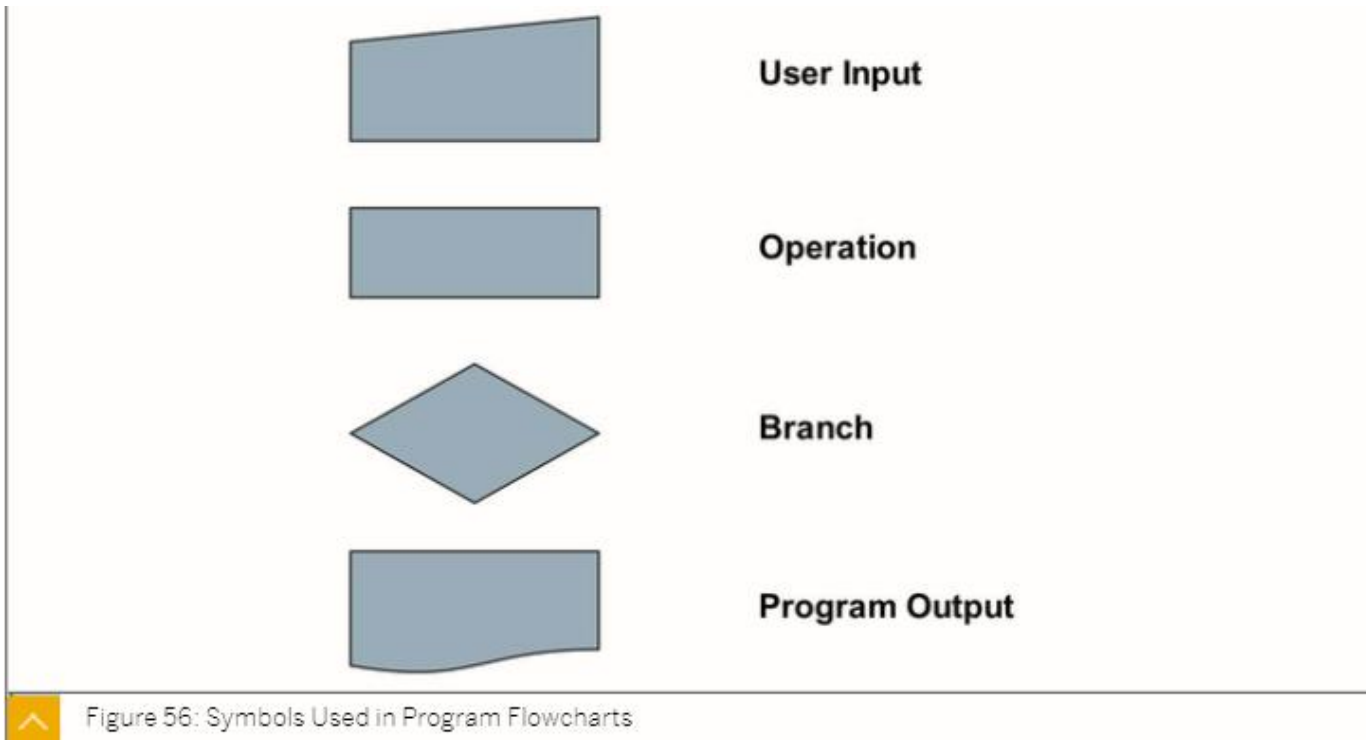
- Pour créer les listes on tape: More → Go to → text element → list heading.
- Concaténation : **concatenate** (le caractère) **into** (list) **separated by** (soit espace ou définir un séparateur).
- La recherche: **Find** (le caractère) **in** (list) **ignoring case**.
- Remplacer: **replace** (le caractère) **in** (list) **with** (le caractère) **ignoring case**.
- Séparer: **SPLIT** (on sépare le mot) **AT** (avec l'indice) **INTO** (on insère dans les paramètres).

2- Le codage et le débogage :

- Dans le débogage il y a single step et continue afin de trouver solution du problème.
- Break point nous permet de déterminer la ou les lignes qu'on veut déboguer.
- Il y a aussi les Watch points qui est similaire au break point, mais lorsque le contenu de la variable le déboguer stop pour identifier les problèmes et il est utilisé pour les programmes complexes et la variable peut fréquemment changer.

3- Les conditions et les loops :

- Conditional logic :



Il y a deux types pour tester les conditions :

- If : qui suit la structure suivante :

If (condition) $sy-subrc = 0$ (la condition est bien verifler).

And (condition) (ou OR).

Statement (ce qui doit être fait).

Elseif (condition).

Statement.

Else.

Statement.

Endif.

- Case : dont la structure est comme suit :

Case var.

When ' '.

Statement .

When ' '.

Statement.

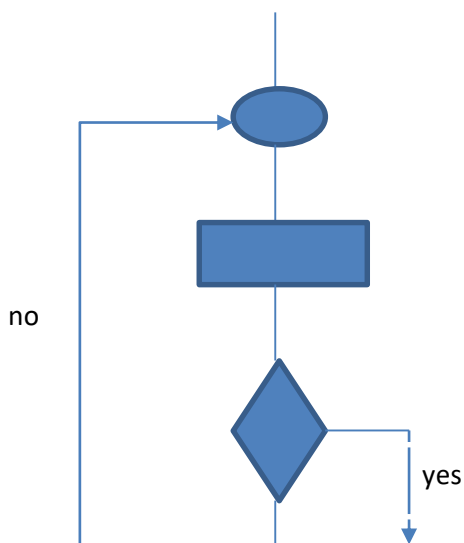
When others ' '.

Statement.

End case.

- Loops:

Les loops répètent une opération tant que la condition est vérifiée et rencontrée.



Les loops utilisées par ABAP sont DO et While dont la syntaxe de chacune est la suivante :

- Do :

Do statement.

If condition. Exit. Endif.

EndDo.

Ou:

Do n times.

Statement.

EndDo.

- While:

While condition.

Statement.

EndWhile.

- **Modularisation :**

Il s'agit de l'appel d'un code et le réutiliser dans un autre code, mais si ce code est modifié on doit le modifier dans tous les codes qui le réutilisent.

On fait appel à la fonction en utilisant la commande « **call function** ».

4- Les erreurs :

- **Erreurs d'exécution (runtime errors) :**

Lorsqu'il y a ce genre d'erreur, la ligne qui ne peut pas être exécuté est indiquée par >>>>>.

Afin d'éviter ce genre d'erreur on peut utiliser les conditions soit if ou case.

- **Gestion des erreurs :**

En travaillant dans le domaine de SAP, on peut rencontrer « dialog messages » qui nous informe soit en donnant des informations sur l'application ou nous montre les problèmes.

On peut définir ce dialogue message en écrivant un message dans l'éditeur et on y double clic.

5- Techniques additionnels pour programmer dans ABAP :

Parmi les techniques de plus dans ABAP on trouve la création de base de données et sa manipulation avec un code dans ABAP EDITOR. Ces étapes seront bien détaillées dans la partie des ateliers : la création, insertion, recherche et la mise à jour d'une table.

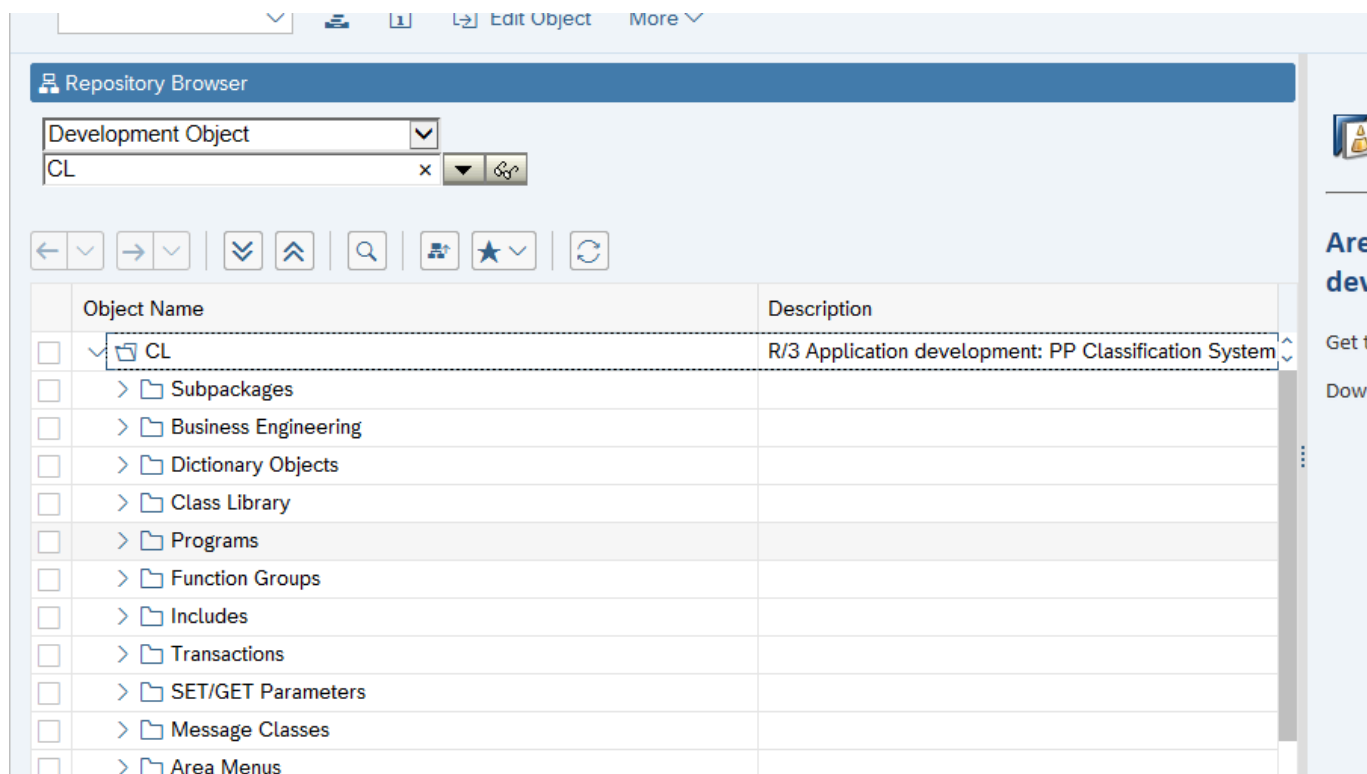
II. Ateliers :

Comme atelier donné en premier de créer une base de données contenant l'id de chaque stagiaire, son nom et prénom, la date du début de stage et la date de fin, le sujet de stage, l'état d'avancement (en formation ou en production) et le statut de ce stagiaire (initiation, application ou fin d'étude).

Pour cela, on accède à SAP logon on entre les données spécifiées à chaque utilisateur puis on crée un package dans lequel on affecte la table créer, pour le créer on tape se80 ici :



Puis on obtient cette page :



On crée le package dans subpackage :

Package: Saved

Properties Use Accesses Package Interfaces Subpackages Package Hierarchy

Basic Data

Short Description:

Application Component: Classes

Person Responsible:

Created by:

Created On:

Last Changed By:

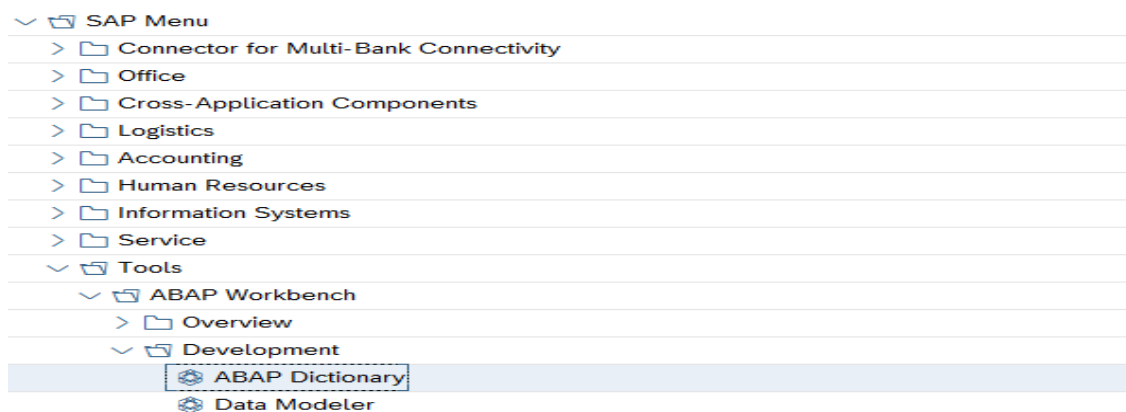
Changed on:

Package Properties

Superpackage:

CL est donné par le système de SAP.

Puis on crée la table de base de données comme suit :



Après une fenêtre de cette forme s'ouvre on fait entrer le nom de la table qui doit commencer avec « z ou y » comme le programme, en l'affectant au package créer on a comme suit :

Transparent Table: Active

Short Description:

Attributes Delivery and Maintenance Fields Input Help/Check Currency/Quantity Fields Indexes

Last Changed:

Package:

Original language:

Dans la partie Fields on crée les données de notre table comme suit :

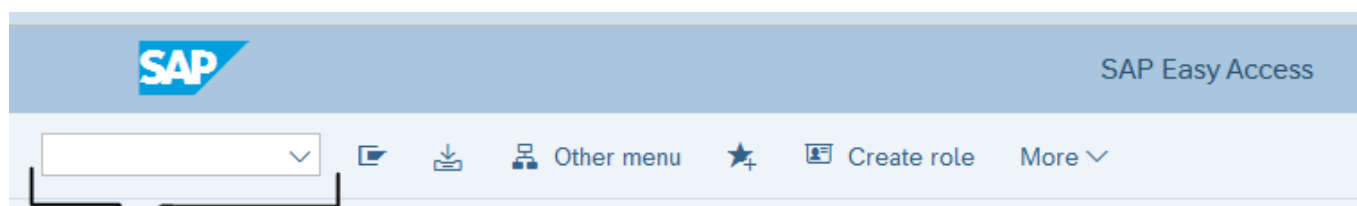
Field	Key	Initia...	Data element	Data Type	Length	Decima...	Coordinate	Short Description
<input type="checkbox"/> ID	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		INT1	3	0	0	
<input type="checkbox"/> NOM	<input type="checkbox"/>	<input type="checkbox"/>		CHAR	256	0	0	
<input type="checkbox"/> PRENOM	<input type="checkbox"/>	<input type="checkbox"/>		CHAR	256	0	0	
<input type="checkbox"/> SUJET	<input type="checkbox"/>	<input type="checkbox"/>		CHAR	256	0	0	
<input type="checkbox"/> DATEDEBUT	<input type="checkbox"/>	<input type="checkbox"/>		DATS	8	0	0	
<input type="checkbox"/> DATEFIN	<input type="checkbox"/>	<input type="checkbox"/>		DATS	8	0	0	
<input type="checkbox"/> ETATAVANCEMENT	<input type="checkbox"/>	<input type="checkbox"/>		CHAR	256	0	0	
<input type="checkbox"/> STATUTSTAGIAIRE	<input type="checkbox"/>	<input type="checkbox"/>		CHAR	256	0	0	

On indique le type et la longueur de chaque variable entrée.

Alors après la création de la table on va y faire des modifications avec la programmation ABAP, on va faire l'insertion, la recherche et la modification sur un élément créé.

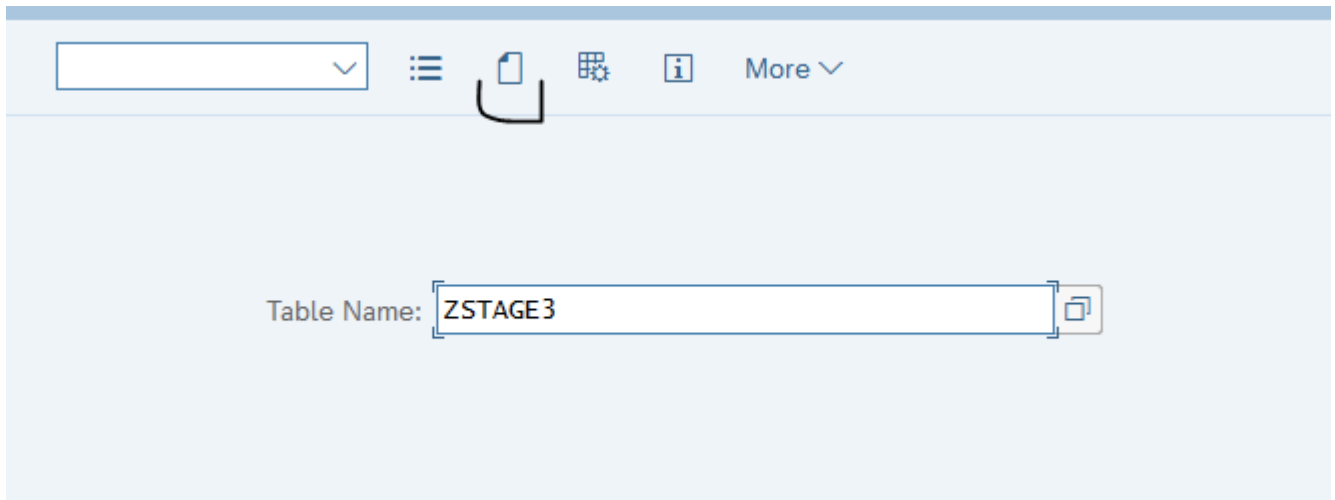
1- Insertion :

Pour l'insertion il y a deux façons d'y faire soit par la programmation en ABAP ou en tapant se16 dans cette partie :



The screenshot shows the SAP Easy Access menu. At the top, there is a search bar and the text 'SAP Easy Access'. Below this, there is a row of icons and text: a magnifying glass icon, a document icon, a download icon, a person icon, the text 'Other menu', a star icon, a person icon, the text 'Create role', and a dropdown arrow icon. Below this row, there is a large empty rectangular box, likely for a command or search input.

On obtient cette page :



Après on tape dans la partie entourée et on remplit les données mais si le id existe déjà les données ne seront pas écraser et va demander de changer le id.

- **Insertion par code :**

```

1  *-----*
2  * Report YSTAGE
3  *-----*
4  *
5  *-----*
6  REPORT YSTAGE.
7  TABLES zstage3 .
8
9  DATA tableau TYPE zstage3.
10
11 PARAMETERS: id_cand TYPE zstage3-id,
12              nom_st TYPE zstage3-nom,
13              prenom_s TYPE zstage3-prenom,
14              D_deb TYPE zstage3-datedebut,
15              D_f TYPE zstage3-datefin,
16              top TYPE zstage3-sujet,
17              stat TYPE zstage3-statutstagiaire,
18              etat TYPE zstage3-etatavancement.
19
20 tableau-id = id_cand.
21 */au-nom= nom_st .
22
23 tableau-nom = nom_st .
24 tableau-prenom = prenom_s.
25 tableau-datedebut = d_deb .
26 tableau-datefin = d_f.

```



```

27  tableau-sujet = top.
28  tableau-etatavancement = etat.
29  tableau-statutstagiaire = stat.
30
31  insert INTO zstage3 VALUES tableau.
32
33  IF sy-subrc = 0.
34  write 'bonne insertion'.
35  else.
36  write 'mauvaise insertion'.
37
38
39  ENDIF.

```

On enregistre, on teste s'il y a erreur par ctrl+f2, on l'active par ctrl+f3 et on exécute, dans l'exécution si le id existe déjà ça donne mauvaise insertion sinon l'insertion est réussie par exemple :

ID_CAND:

NOM_ST:

PRENOM_S:

D_DEB:

D_F:

TOP:

STAT:

ETAT:

Cette personne est insérée si on exécute une deuxième fois on obtient ce message :

Insertion

mauvaise insertion

2- La recherche :

Pour la recherche on utilise le code suivant :

```

1  *-----*
2  *& Report ZSTAGEINITIATION
3  *-----*
4  *&
5  *-----*
6  REPORT ZSTAGEINITIATION.
7
8  PARAMETERS id TYPE zstage3-id.
9  PARAMETERS Nom TYPE zstage3-nom.
10 PARAMETERS Prenom TYPE zstage3-prenom .
11 PARAMETERS D_Deb TYPE zstage3-datedebut.
12 PARAMETERS D_Fin TYPE zstage3-datefin.
13
14 DATA gs_nom TYPE zstage3.
15
16 select  nom prenom  datedebut datefin id sujet etatavancement statutstagiaire
17 FROM zstage3
18 INTO CORRESPONDING FIELDS OF gs_nom
19 WHERE id = Id OR prenom = Prenom OR nom = Nom OR datedebut = d_deb or datefin = d_fin.
20 ENDSELECT.

```

```

if sy-subrc = 0 .
write: 'les donnees de stagiaires sont:'.
NEW-LINE.
WRITE: 'ID=' , gs_nom-id.
new-LINE.
WRITE: 'Nom =' , gs_nom-nom.
NEW-LINE.
WRITE: 'prenom=' , gs_nom-prenom.
NEW-LINE.
WRITE: 'date debut : ' ,
gs_nom-datedebut.
NEW-LINE.
WRITE: 'date fin de stage : ' ,
gs_nom-datefin ,
'etat avancement de stagiaire',
gs_nom-etatavancement ,
'sujet de stage:' ,
gs_nom-sujet ,
'statut de stagiaire',
gs_nom-statutstagiaire.
ELSE .
WRITE 'nom pas trouvé' .
ENDIF.

```

Dans ce code on utilise la commande select dont on sélectionne tous les données du tableau et on les compare avec ce qui est entrée par l'utilisateur puis si le candidat existe le programme affiche ses données sinon il affiche qu'il ne se trouve pas.

Exemple on va tester sur le id 4 qu'on vient de créer :

ID:

NOM:

PRENOM:

D_DEB:

D_FIN:



TPStage

les donnees de stagiaires sont:

ID= 4

Nom =

RHARNATI

prenom=

MERYEM

date debut : 07.08.2020

date fin de stage : 04.09.2020 etat avancement de stagiaire

EN FORMATION

sujet de stage:

SAP FIORI

statut de stagiaire

PFE

Si on entre un id qui n'existe pas on a comme suit :

ID:

NOM:



TPStage

nom pas trouvé

3- La mise à jour (update) :

Pour la mise à jour on utilise le code suivant dont on sélectionne les données et on met à jours soit la date de fin ou l'état d'avancement puis on affiche cette mise à jour pour vérifier le code :

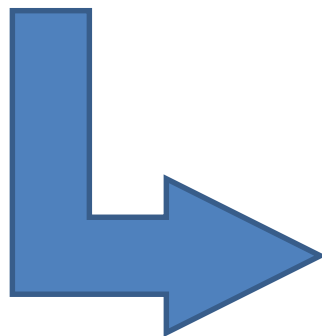
```

1  *-----*
2  * Report YSTAGEUPDATING
3  *-----*
4  *
5  *-----*
6  REPORT YSTAGEUPDATING.
7
8  PARAMETERS id TYPE zstage3-id.
9  PARAMETERS name TYPE zstage3-nom.
10 PARAMETERS:fName TYPE zstage3-prenom,
11          FD TYPE zstage3-datedebut,
12          Ld TYPE zstage3-datefin,
13          subj TYPE zstage3-sujet,
14  etat TYPE zstage3-etatavancement,
15          status TYPE zstage3-statutstagiaire.
16
17
18 DATA gs_nom TYPE zstage3.
19
20 select nom prenom datedebut datefin id sujet etatavancement statutstagiaire
21 FROM zstage3
22 INTO CORRESPONDING FIELDS OF gs_nom
23 WHERE id = Id OR prenom = fname or nom = name or datedebut = fd or datefin = ld .
24 ENDSELECT.
25
26 IF sy-subrc = 0.
27
28   WRITE 'candidat existe'.
29
30   UPDATE zstage3
31   SET: datefin = ld,
32       etatavancement = etat.
33
34   WRITE 'la mise a jour bien faite '.
35
36   write 'le candidat est:'.
37   new-LINE.
38   write gs_nom-id.
39   new-LINE.
40   WRITE gs_nom-nom.
41   new-LINE.
42   WRITE gs_nom-prenom.
43   new-LINE.
44   WRITE gs_nom-datedebut.
45   new-LINE.
46   WRITE gs_nom-datefin.
47   new-LINE.
48   WRITE gs_nom-etatavancement.
49   NEW-LINE.
50   WRITE gs_nom-statutstagiaire.
51
52 NEW-LINE.
53 WRITE gs_nom-sujet.
54
55
56 else.
57   WRITE 'la mise a jour mal faite'.
58
59 ENDIF.

```

On prend le même exemple : on avait date de fin de stage 04/09/2020 on va la rendre 11/09/2020 et on va rendre l'état d'avancement en production au lieu de en formation :

ID:	4
NAME:	
FNAME:	
FD:	
LD:	11.09.2020
SUBJ:	
ETAT:	EN PRODUCTION
STATUS:	



update

candidat existe la mise a jour bien faite le candidat est:

4

RHARNATI

MERYEM

07.08.2020

11.09.2020

EN PRODUCTION

PFE

SAP FIORI

- [Bibliographie :](https://learninghub.sap.com/)

<https://learninghub.sap.com/>

• Conclusion :

En guise de conclusion, j'ai pu bien savoir les bases du langage ABAP comme la création des bases de données dont il y a la création de la table, l'insertion, modification, la recherche et la suppression, aussi déclaration des variables, les conditions... De plus la manipulation et l'application de toutes ces notions dans l'environnement de SAP.

Et ce nouvel langage de programmation m'a donné un esprit de critique et de recherche plus profond dans le domaine d'informatique.