

PROJET MAINFRAME

Chaîne complète de traitement et de gestion des produits et ventes sur mainframe

FAUCHER Aymeric
KALILOU SYLLA Mahamadou
VERRIER Alexis

Début : 7 août 2025
Fin : 4 septembre 2025
Total : 13 jours



Sommaire

- Présentation de l'équipe
- Outils
- Contexte
- Cahier des charges
- Etapes du projet
- Présentation des parties
- Démo
- Tests Unitaires
- Conclusion

Présentation de l'équipe



FAUCHER Aymeric



KALILOU SYLLA
Mahamadou



VERRIER Alexis

Outils

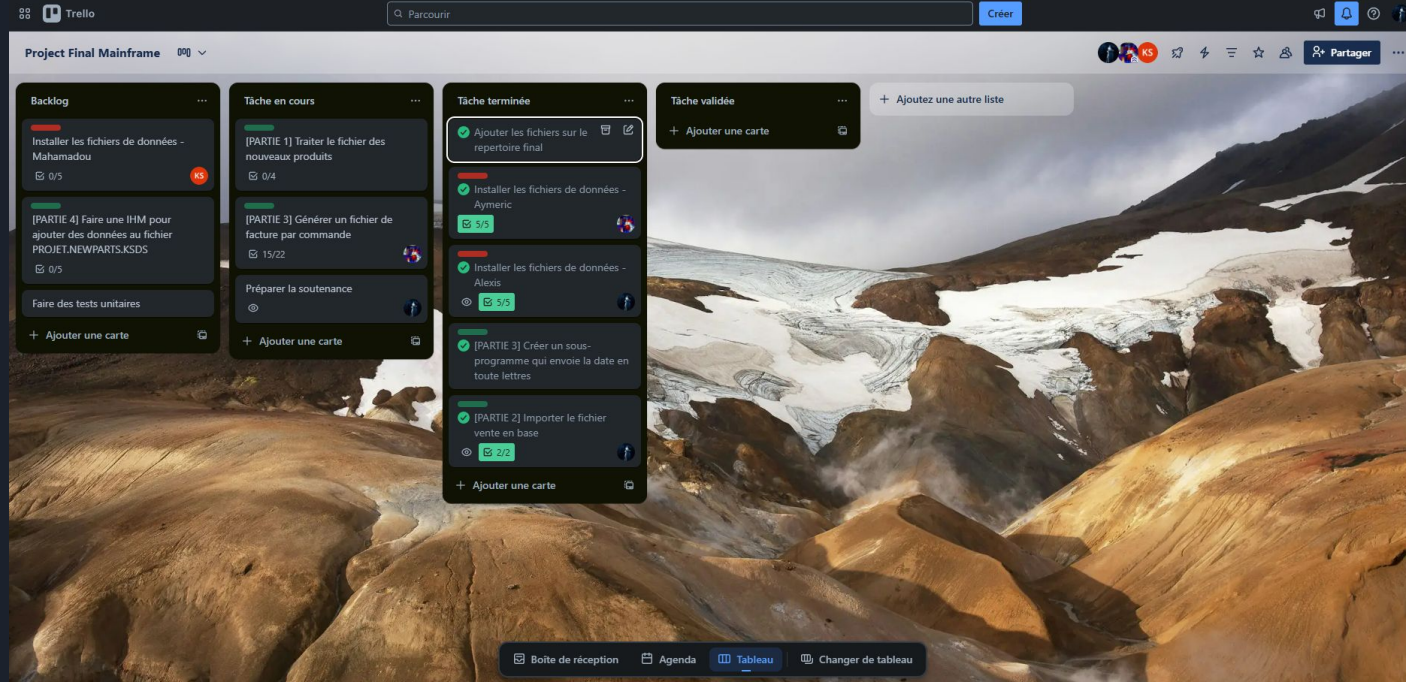


Lien vers le [Github](#)
→ Partage de fichiers



→ Partage de fichiers
→ Communication

Outils



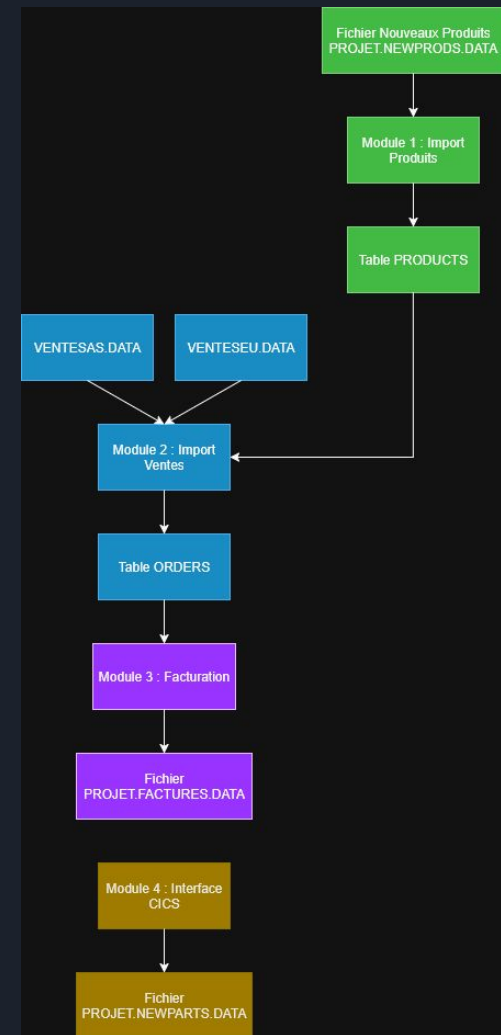
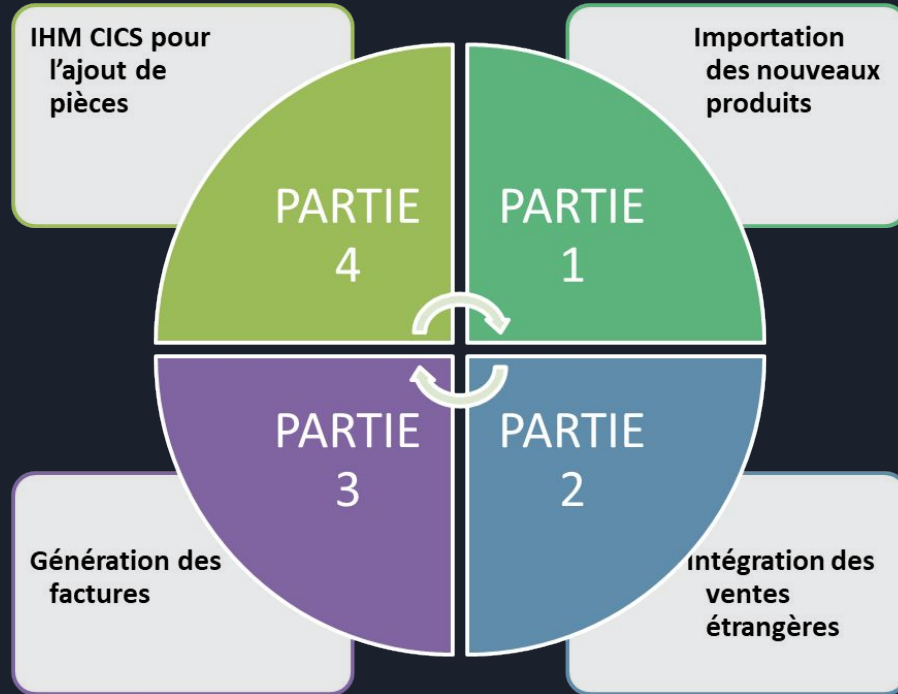
Lien vers le [Trello](#)

→ Visualisation de l'avancement du projet



Contexte

Cahier des charges



Organisation



README

🔧 Objectifs du projet

Le projet se décline en 4 grandes parties :

📁 Partie 1 – Importation des nouveaux produits

- Lecture du fichier `PROJET.NEWPRODS.DATA` (fichier CSV semi-structuré).
- Insertion des nouveaux produits dans la base de données après :
 - Formatage des descriptions (capitalisation).
 - Conversion automatique des prix en dollar via des taux de change.
 - Prise en compte des devises supplémentaires via une structure évolutive.

📁 Partie 2 – Intégration des ventes étrangères

- Lecture et traitement des ventes des fichiers :
 - `PROJET.VENTESEU.DATA` (Europe)
 - `PROJET.VENTESAS.DATA` (Asie)
- Insertion des ventes dans la base.
- Mise à jour automatique du chiffre d'affaires de chaque client.

📄 Partie 3 – Génération des factures

- Extraction des commandes depuis la base vers un fichier `PROJET.EXTRACT.DATA`.
- Génération des factures au format texte (`PROJET.FACTURES.DATA`) en respectant :
 - Un formatage structuré par commande.
 - La conversion des dates en toutes lettres (via un sous-programme).
 - Ajout de la TVA (taux fourni en `SVSIN`) et de la commission.

💻 Partie 4 – IHM CICS pour l'ajout de pièces

- Interface de saisie sécurisée (authentification via `AJC.EMPLOVE.KSDS`).
- Ajout de pièces dans le fichier `PROJET.NEWPARTS.KSDS` .
- Respect d'une nomenclature stricte pour les ressources CICS (Mapsets, Transactions...).

 PROJ-FINAL

- > Demo/SOURCE
- > Partie1
- > Partie2
- > Partie3
- > Partie4
- > SOURCES
- Projet-MAINFRAME.pdf
- README.md

Lien vers le [Github](#)
→ Partage de fichiers



Etapes du projet

- Création d'un dépôt [Github](#) commun
- Démarche
 - Lecture de l'énoncé
 - Création d'un tableau sur [Trello](#)
 - Répartition des tâches
- Mise en oeuvre
 - Développement
 - Commit Github
 - Test unitaire
- Review globale
- Présentation

Répartition :

- Partie 1 → KALILOU SYLLA Mahamadou
- Partie 2 → VERRIER Alexis
- Partie 3 → FAUCHER Aymeric
- Partie 4 → FAUCHER Aymeric
- Démonos → FAUCHER Aymeric
- Tests unitaires → VERRIER Alexis



PARTIE 1

KALILOU SYLLA Mahamadou




OBJECTIF ET TRAITEMENT

1. Lecture des taux en SYSIN et stockage en tableau
2. Ouverture du fichier CSV
3. Pour chaque ligne :
 - Découpage (UNSTRING)
 - Normalisation de la description
 - Conversion du prix texte → numérique
 - Application du taux de change
 - Insertion dans DB2

```
P10;USB FLASH DRIVE;15;EU
P11;HEADPHONES;30.5;DO
P12;MICRO;25.75;YU
P13;TABLET;125.20;YU
P14;LAPTOP;899.99;DO
P15;MOTHERBOARD;60;EU
P06;DESKTOP COMPUTER;350.55;DO
P07;DOCKING STATION;200;EU
P18;NETWORK SWITCH;150.75;YU
P19;LAPTOP GAMER;900;EU
P20;SSD HARD DISK;152.50;YU
```

EXTRAIT DE CODE (Simplifié)



```
UNSTRING PRODUCT-REC DELIMITED BY ' ; '  
    INTO PRO-P-NO  
        PRO-DESCRIPTION-TEXT  
        WS-PRICE-TXT  
        W-CURRENCY-CODE  
END-UNSTRING
```

```
COMPUTE PRO-PRICE-NUM = FUNCTION NUMVAL-C(W'S-PRICE-TXT)
```

```
MOVE FUNCTION LOWER-CASE(PRO-DESCRIPTION-TEXT)  
    TO PRO-DESCRIPTION-TEXT  
PERFORM VARYING W-IDX FROM 1 BY 1  
    UNTIL W-IDX > LENGTH OF PRO-DESCRIPTION-TEXT  
    IF W-IDX = 1 OR PRO-DESCRIPTION-TEXT(W-IDX - 1:1) = ' '  
        MOVE FUNCTION UPPER-CASE(PRO-DESCRIPTION-TEXT(W-IDX:1))  
            TO PRO-DESCRIPTION-TEXT(W-IDX:1)  
    END-IF
```

SORTIE ET RÉSULTATS

EU : 1,10
DO : 1.00
YU : 0.14

Exemple:
P13 : 125,22 YU * 0.14 (Tx) = 17.52 DO
P14 : 899,99 YU * 1.00 (Tx) = 899,99 DO

NUMERO PRODUIT	DESCRIPTION	PRIX	DEVISE
P10	USB FLASH DRIVE	15	EU
P11	HEADPHONES	30.5	DO
P12	MICRO	25.75	YU
P13	TABLET	125.20	YU
P14	LAPTOP	899.99	DO
P15	MOTHERBOARD	60	EU
P06	DESKTOP COMPUTER	350.55	DO
P07	DOCKING STATION	200	EU
P18	NETWORK SWITCH	150.75	YU
P19	LAPTOP GAMER	900	EU
P20	SSD HARD DISK	152.50	YU

P_NO	DESCRIPTION	PRICE	DEVISE
P10	Usb Flash Drive	16,5	DO
P11	Headphones	30,5	DO
P12	Micro	3,6	DO
P13	Tablet	17,5	DO
P14	Laptop	899,9	DO
P15	Motherboard	66,0	DO
P06	Desktop Computer	350,5	DO
ERREUR INSERT : DOUBLON SUR CODE P06			
P07	Docking Station	220,0	DO
ERREUR INSERT : DOUBLON SUR CODE P07			
P18	Network Switch	21,1	DO
P19	Laptop Gamer	990,0	DO
P20	Ssd Hard Disk	21,3	DO

MISE À JOUR DES PRODUITS

```
SELECT * FROM PRODUCTS WHERE P_NO > 'P05';
```

P_NO	DESCRIPTION	PRICE
P06	Printer XP427	50.55
P07	USB Drive	10.25
P10	Usb Flash Drive	16.50
P11	Headphones	30.50
P12	Micro	3.60
P13	Tablet	17.52
P14	Laptop	899.99
P15	Motherboard	66.00
P18	Network Switch	21.10
P19	Laptop Gamer	990.00
P20	Ssd Hard Disk	21.35

Problèmes rencontrés + solutions

```
//*  
//  
//      SET  SYSUID=API3,  
//      NOMPGM=PRO3  
//*  
//APPROC  EXEC  COMPDB2  
//STEPDB2.SYSLIB      DD  DSN=&SYSUID..COB.CPY,DISP=SHR  
//STEPDB2.SYSIN      DD  DSN=&SYSUID..COB.SRC(&NOMPGM),DISP=SHR  
//STEPDB2.DBRMLIB    DD  DSN=&SYSUID..COB.DBRM(&NOMPGM),DISP=SHR  
//STEPCOB.SYSLIB     DD  DSN=&SYSUID..COB.CPY,DISP=SHR  
//              DD  DSN=API3.COB.CPY,DISP=SHR  
//STEPLNK.SYSLMOD    DD  DSN=&SYSUID..COB.LOAD(&NOMPGM),DISP=SHR  
//*
```

CONCLUSION

- ✓ Automatisation du traitement CSV
- ✓ Conversion multi-devises via SYSIN
- ✓ Normalisation des données produit
- ✓ Intégration directe en DB2

→ Programme modulaire et extensible



PARTIE 2

VERRIER Alexis



- Objectif
- Fichiers traités :
 - PROJET.VENTESEU.DATA
 - PROJET.VENTESAS.DATA
 - Format fixe avec : numéro commande, date, client, produit, quantité, prix
- Traitement
- Résultat

Partie 2

O_NO = Numéro de commande

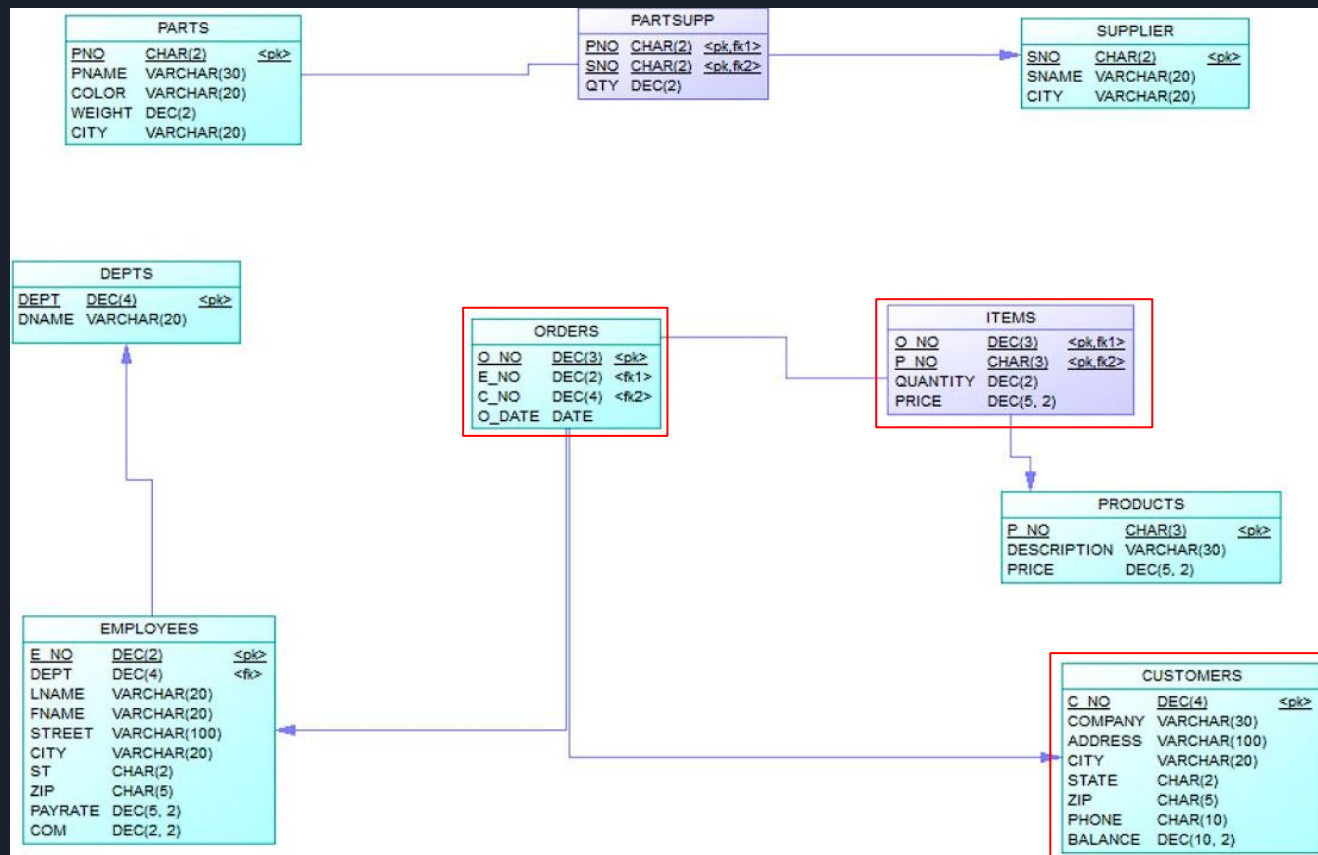
C_NO = Numéro de client

VENTESAS

1	50115/10/2022200003P020154910
2	50115/10/2022200003P030357502
3	50202/11/2022300002P050502507
4	50305/11/2022500001P15 10
5	50517/11/2022400004P10 01
6	50517/11/2022400004P12 04

VENTESEU

1	50010/10/2022100004P010259903
2	50010/10/2022100004P030357502
3	50010/10/2022100004P040100005
4	50202/11/2022300002P020154903
5	50202/11/2022300002P030357505
6	50305/11/2022500001P11 05
7	50407/11/2022400003P14 01
8	50407/11/2022400003P18 04



Update Table ORDERS

```

016500 UPDATE-ORDER.
016600     CALL WS-DATEPROG USING ENR-O-DATE WS-FORMAT
016700
016800     MOVE ENR-O-NO      TO ORD-O-NO
016900     MOVE ENR-E-NO      TO ORD-S-NO
017000     MOVE ENR-C-NO      TO ORD-C-NO
017100     MOVE ENR-O-DATE    TO ORD-O-DATE
017200
017300     EXEC SQL
017400         INSERT INTO
017500             API2.ORDERS(O_NO, S_NO, C_NO, O_DATE)
017600             VALUES (:ORD-O-NO, :ORD-S-NO, :ORD-C-NO, :ORD-O-DATE)
017700     END-EXEC
017800     PERFORM TEST-SQLCODE.

```

Update Table ITEMS

```

018000 UPDATE-ITEMS.
018100     MOVE ENR-PRICE TO WS-PRICE
018200     IF WS-PRICE = 0 THEN
018300         PERFORM FETCH-PRICE
018400         MOVE PRO-PRICE TO ENR-PRICE
018500     END-IF
018600
018700     MOVE ENR-O-NO      TO ITE-O-NO
018800     MOVE ENR-P-NO      TO ITE-P-NO
018900     MOVE ENR-QUANTITY  TO ITE-QUANTITY
019000     MOVE ENR-PRICE     TO ITE-PRICE
019100     EXEC SQL
019200         INSERT INTO
019300             API2.ITEMS(O_NO, P_NO, QUANTITY, PRICE)
019400             VALUES (:ITE-O-NO, :ITE-P-NO, :ITE-QUANTITY, :ITE-PRICE)
019500     END-EXEC
019600     MOVE 'UPDATING ITEMS' TO WS-MSG
019700     PERFORM TEST-SQLCODE
019800     PERFORM CUMUL-CA-CUSTOMER.

```

Cumul du chiffre d'affaires + Update BDD

```

019900 CUMUL-CA-CUSTOMER.
020000     COMPUTE CUST-COMM = ENR-PRICE * ENR-QUANTITY
020100     ADD CUST-COMM TO CUST-COMM-TOTAL.
020200
020300 UPDATE-CA-CUSTOMER.
020400     MOVE CUST-COMM-TOTAL TO CUST-COMM-UP
020500     MOVE ORD-C-NO TO CUS-C-NO
020600     EXEC SQL
020700         UPDATE API2.CUSTOMERS
020800         SET BALANCE = BALANCE + :CUST-COMM-UP
020900         WHERE C_NO = :CUS-C-NO
021000     END-EXEC
021100     PERFORM TEST-SQLCODE.

```

Afficher les codes SQL + Programme d'abandon

```

021300 TEST-SQLCODE.
021400     EVALUATE TRUE
021500         WHEN SQLCODE = ZERO
021600             CONTINUE
021700         WHEN SQLCODE = -803
021800             DISPLAY
021900                 'ERREUR INSERT : DOUBLON SUR CODE '
022000         WHEN SQLCODE > ZERO
022100             IF SQLCODE = +100
022200                 DISPLAY 'CODE XX INTROUVABLE POUR OPERATION '
022300             END-IF
022400             MOVE SQLCODE TO ED-SQLCODE
022500             DISPLAY 'WARNING : ' ED-SQLCODE
022600         WHEN OTHER
022700             DISPLAY 'MSG -> ' WS-MSG
022800             MOVE SQLCODE TO ED-SQLCODE
022900             DISPLAY 'ANOMALIE ' ED-SQLCODE
023000             PERFORM ABEND-PROG
023100     END-EVALUATE.
023200
023300 ABEND-PROG.
023400     EXEC SQL ROLLBACK END-EXEC
023500     DISPLAY 'ROLLING BACK TO PREV TABLE STATE'
023600     COMPUTE WS-ANO = 1 / WS-ANO.

```

Table ORDERS

O_NO	S_NO	C_NO	O_DATE
101.	10.	1.	2022-02-15
102.	20.	2.	2018-03-20
103.	30.	3.	2020-02-25
500.	10.	4.	2022-10-10
501.	20.	3.	2022-10-15
502.	30.	2.	2022-11-02
503.	50.	1.	2022-11-05
504.	40.	3.	2022-11-07
505.	40.	4.	2022-11-17

DSNE610I NUMBER OF ROWS DISPLAYED IS 9

Table ITEMS

O_NO	P_NO	QUANTITY	PRICE
101.	P01	10.	25.99
101.	P03	15.	35.75
102.	P03	10.	35.75
103.	P05	10.	50.25
500.	P01	3.	25.99
500.	P03	2.	35.75
500.	P04	5.	10.00
501.	P02	10.	15.49
501.	P03	2.	35.75
502.	P02	3.	15.49
502.	P03	5.	35.75
502.	P05	7.	50.25
503.	P11	5.	30.50
503.	P15	10.	66.00
504.	P14	1.	899.99
504.	P18	4.	21.10
505.	P10	1.	16.50
505.	P12	4.	3.60

DSNE610I NUMBER OF ROWS DISPLAYED IS 18

Problèmes rencontrés + solutions

Display Filter View Print Options Help										Display Filter View Print Options Help									
SDSF OUTPUT DISPLAY API2COMP JOB07046 DSID 102 LINE 313 COLUMNS 17- 96										SDSF OUTPUT DISPLAY API2COMP JOB07046 DSID 103 LINE 0 COLUMNS 17- 96									
COMMAND INPUT ==> █ SCROLL ==> CSR										COMMAND INPUT ==> █ SCROLL ==> CSR									
DSNHLEX LINE 50 COL 38 MEMBER "ORD" " COULD NOT BE INCLUDED										***** TOP OF DATA *****									
DSNHLEX LINE 51 COL 38 MEMBER "ITE" " COULD NOT BE INCLUDED										SNHLEX LINE 50 COL 38 MEMBER "ORD" " COULD NOT BE INCLUDED									
DSNHLEX LINE 52 COL 38 MEMBER "PRO" " COULD NOT BE INCLUDED										SNHLEX LINE 51 COL 38 MEMBER "ITE" " COULD NOT BE INCLUDED									
DSNHSMUD LINE 173 COL 21 UNDEFINED OR UNUSABLE HOST VARIABLE "PRO-PRICE"										SNHLEX LINE 52 COL 38 MEMBER "PRO" " COULD NOT BE INCLUDED									
DSNHSMUD LINE 175 COL 29 UNDEFINED OR UNUSABLE HOST VARIABLE "PRO-P-NO"										SNHSMUD LINE 173 COL 21 UNDEFINED OR UNUSABLE HOST VARIABLE "PRO-PRICE"									
DSNHSMUD LINE 225 COL 23 UNDEFINED OR UNUSABLE HOST VARIABLE "ORD-O-NO"										SNHSMUD LINE 175 COL 29 UNDEFINED OR UNUSABLE HOST VARIABLE "PRO-P-NO"									
DSNHSMUD LINE 225 COL 34 UNDEFINED OR UNUSABLE HOST VARIABLE "ORD-S-NO"										0:PRO-PRICE FROM API2.PRODUCTS WHERE P_NO=:PRO-P-NO									
DSNHSMUD LINE 225 COL 45 UNDEFINED OR UNUSABLE HOST VARIABLE "ORD-C-NO"										SNHSMUD LINE 225 COL 23 UNDEFINED OR UNUSABLE HOST VARIABLE "ORD-O-NO"									
DSNHSMUD LINE 225 COL 56 UNDEFINED OR UNUSABLE HOST VARIABLE "ORD-O-DATE"										SNHSMUD LINE 225 COL 34 UNDEFINED OR UNUSABLE HOST VARIABLE "ORD-S-NO"									
DSNHSMUD LINE 243 COL 26 UNDEFINED OR UNUSABLE HOST VARIABLE "ITE-O-NO"										SNHSMUD LINE 225 COL 45 UNDEFINED OR UNUSABLE HOST VARIABLE "ORD-C-NO"									
DSNHSMUD LINE 243 COL 37 UNDEFINED OR UNUSABLE HOST VARIABLE "ITE-P-NO"										SNHSMUD LINE 225 COL 56 UNDEFINED OR UNUSABLE HOST VARIABLE "ORD-O-DATE"									
DSNHSMUD LINE 243 COL 48 UNDEFINED OR UNUSABLE HOST VARIABLE "ITE-QUANTITY"										F1=HELP F2=SPLIT F3=END F4=RETURN F5=IFIND F6=BOOK F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE									
DSNHSMUD LINE 243 COL 63 UNDEFINED OR UNUSABLE HOST VARIABLE "ITE-PRICE"										F1=HELP F2=SPLIT F3=END F4=RETURN F5=IFIND F6=BOOK F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE									
ILER STATISTICS										F1=HELP F2=SPLIT F3=END F4=RETURN F5=IFIND F6=BOOK F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE									
CS										F1=HELP F2=SPLIT F3=END F4=RETURN F5=IFIND F6=BOOK F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE									
READ: 285										F1=HELP F2=SPLIT F3=END F4=RETURN F5=IFIND F6=BOOK F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE									
BOLS: 62										F1=HELP F2=SPLIT F3=END F4=RETURN F5=IFIND F6=BOOK F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE									
BYTES EXCLUDING ATTRIBUTES: 4352										F1=HELP F2=SPLIT F3=END F4=RETURN F5=IFIND F6=BOOK F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE									
F1=HELP F2=SPLIT F3=END F4=RETURN F5=IFIND F6=BOOK F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE										F1=HELP F2=SPLIT F3=END F4=RETURN F5=IFIND F6=BOOK F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE									
4 X 1 Sess-1 54.38.29.76 §14/21										4 X 1 Sess-1 54.38.29.76 §14/21									

Problèmes rencontrés + solutions

====>

DCLGEN

SSID: DSN1

Enter table name for which declarations are required:

1 SOURCE TABLE NAME ==> **Orders** (Unqualified)

2 TABLE OWNER ==> **API2** (Optional)

3 AT LOCATION ==> (Optional)

Enter destination data set: (Can be sequential or partitioned)

4 DATA SET NAME ... ==> **'API2.SOURCE.DCLGEN(ORD)'**

5 DATA SET PASSWORD ==> (If password protected)

Enter options as desired:

6 ACTION ==> **REPLACE** (ADD new or REPLACE old declaration)

7 COLUMN LABEL ==> **YES** (Enter YES for column label)

8 STRUCTURE NAME .. ==> **st-ord** (Optional)

9 FIELD NAME PREFIX ==> **ord-** (Optional)

10 DELIMIT DBCS ==> **YES** (Enter YES to delimit DBCS identifiers)

11 COLUMN SUFFIX ... ==> **YES** (Enter YES to append column name)

12 INDICATOR VARS .. ==> **NO** (Enter YES for indicator variables)

Menu Functions Confirm Utilities Help

EDIT API2.COB.CPY Row 00001 of 00013

Command ==> █ Scroll ==> 0006

Name	Prompt	Size	Created	Changed	ID
CHGBOOK		18	2024/11/19	2024/11/26 13:43:23	API2
CURRENCY		3	2024/11/12	2024/11/13 10:46:05	API2
CUS		3	2025/08/28	2025/08/28 15:10:22	API2
DAYMONTH		32	2024/11/15	2024/11/15 11:50:17	API2
ITE		5	2025/08/28	2025/08/28 15:09:20	API2
OCHGBOOK		10	2024/11/26	2024/11/26 12:47:40	API2
ORD		5	2025/08/28	2025/08/28 15:08:05	API2
PRO		3	2025/08/28	2025/08/28 15:09:54	API2
STXML		27	2024/11/19	2024/11/22 11:53:45	API2
TESTCONT		4	2024/10/25	2024/11/20 11:14:24	API2
VTAS		11	2024/11/16	2024/11/18 12:10:12	API2
VTEU		11	2024/11/16	2024/11/16 09:59:50	API2
VTSTRUCT		9	2024/11/17	2024/11/17 20:31:08	API2

End

F1=HELP F7=UP

F2=SPLIT F8=DOWN

F3=END F9=DEL

F4=RETURN F10=LEFT

F5=RFIND F11=RIGHT

F6=RCHANGE F12=RETRIEVE

F1=Help F10=Left

F2=Split F11=Right

F3=Exit F12=Cancel

F5=Rfind

F7=Up

F8=Down

F9=Swap

4 X

1 Sess-1

54.38.29.76

15/29

4 X

1 Sess-1

54.38.29.76

14/15

Problèmes rencontrés + solutions

Problème API2 -> passage API1

Format FR/US sur API1



PARTIE 3

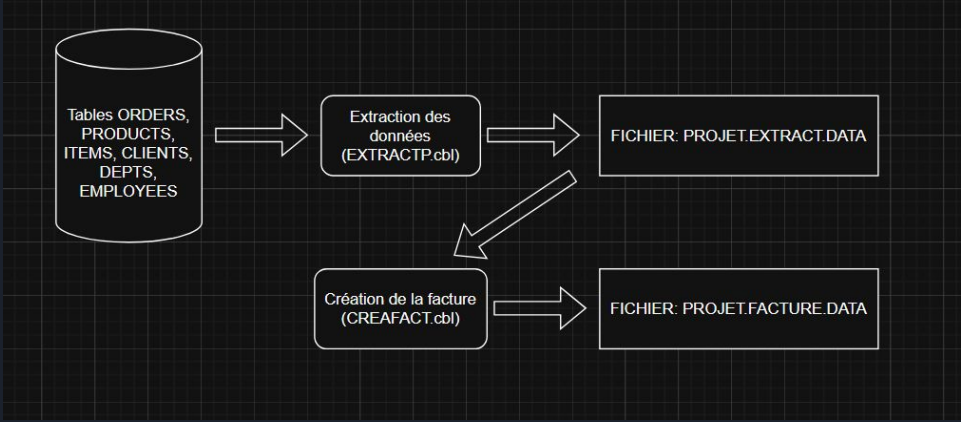
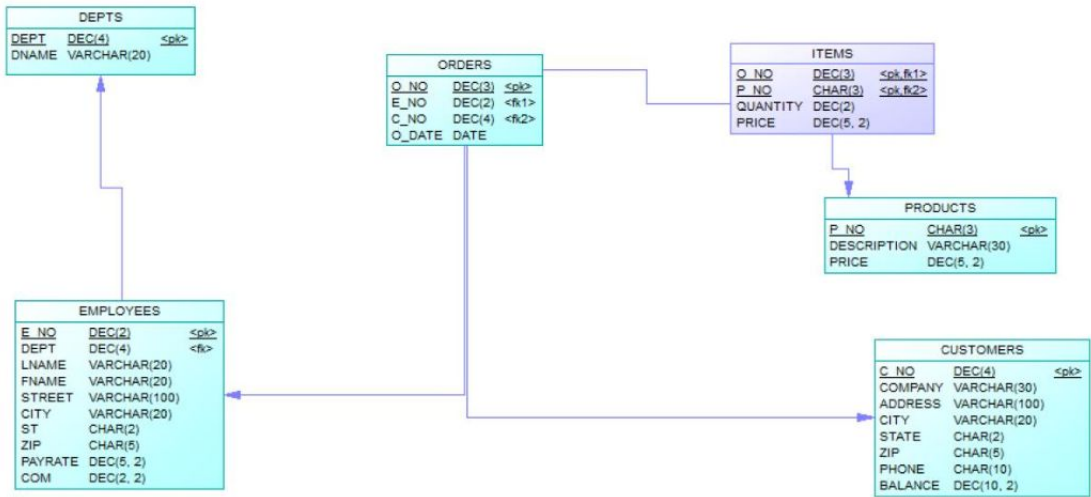
FAUCHER Aymeric



Objectif : Éditer automatiquement une facture par commande passée, au format lisible et structuré.

Etape 1: Extraction des données de la facture dans un fichier (PROJET.EXTRACT.DATA)

Etape 2: Création d'un fichier facture (PROJET.FACTURE.DATA) en utilisant les données du fichier d'extrait.



Deux requêtes

```

*** DECLARATION DU CURSEUR ORDER ***
EXEC SQL
  DECLARE CORDER CURSOR
  FOR
  SELECT O.O_NO, O.O_DATE, E.E_NO, E.DEPT, E.LNAME,
         E.FNAME, E.COM, C.C_NO, C.COMPANY, C.ADDRESS,
         C.CITY, C.ZIP, C.STATE, D.DNAME
  FROM ORDERS O
  INNER JOIN EMPLOYEES E
  ON E.E_NO = O.S_NO
  INNER JOIN CUSTOMERS C
  ON C.C_NO = O.C_NO
  INNER JOIN DEPTS D
  ON D.DEPT = E.DEPT
  ORDER BY O.O_NO
END-EXEC

*** DECLARATION DU CURSEUR PRODUCTS ***
EXEC SQL
  DECLARE CPROD CURSOR
  FOR
  SELECT O_NO, P.P_NO, QUANTITY, DESCRIPTION, I.PRICE,
         P.PRICE
  FROM ITEMS I
  INNER JOIN PRODUCTS P
  ON I.P_NO = P.P_NO
  WHERE O_NO = :ORD-O-NO
  ORDER BY O_NO
END-EXEC

```

FETCH-ORDER.

```

MOVE SPACE TO DCLEMPLOYEES
MOVE SPACE TO DCLORDERS
MOVE SPACE TO DCLCUSTOMERS
MOVE SPACE TO DCLDEPTS
EXEC SQL
  FETCH CORDER
  INTO :ORD-O-NO, :ORD-O-DATE, :EMP-E-NO, :DEP-DEPT
      , :EMP-LNAME, :EMP-FNAME, :EMP-COM
      , :CUS-C-NO, :CUS-COMPANY
      , :CUS-ADDRESS, :CUS-CITY, :CUS-ZIP, :CUS-STATE
      , :DEP-DNAME
END-EXEC
PERFORM TEST-SQLCODE.

```

FETCH-PRODUCT.

```

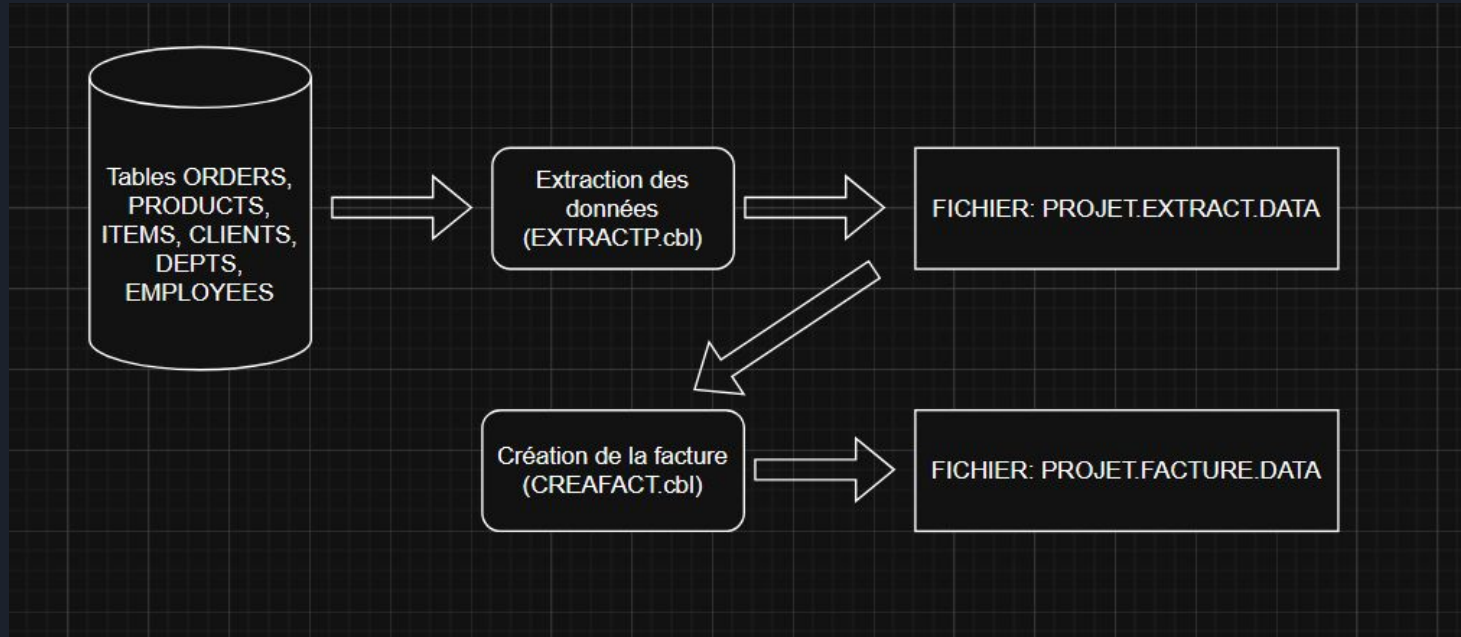
MOVE SPACE TO DCLITEMS
MOVE SPACE TO DCLPRODUCTS
EXEC SQL
  FETCH CPROD
  INTO :ITE-O-NO, :PRO-P-NO, :ITE-QUANTITY
      , :PRO-DESCRIPTION, :ITE-PRICE, :PRO-PRICE
END-EXEC
PERFORM TEST-SQLCODE.

```

Structure du fichier
EXTRACT.DATA

DE	A	LG	TYPE	NOM	OBSERVATIONS
	1	3	3 A	Type Enregistrement	La valeur est toujours 'ORD'
4	5	2 N	N° Commande	O-NO	
6	14	9 A	Date Commande	O-DATE	
15	17	3 N	N° Employe	E-NO	
18	37	20 A	Nom Employe	LNAME	
38	40	3 N	Nom Employe - Longueur	LNAME-LEN	
41	60	20 A	Prénom Employe	FNAME	
61	63	3 N	Prénom Employe - Longueur	FNAME-LEN	
64	65	2 N	Comission	COM	
66	69	4 A	N° Client	C-NO	
70	99	30 A	Nom entreprise	COMPANY	
100	102	3 N	Nom entreprise - Longueur	COMPANY-LEN	
103	202	100 A	Adresse	ADDRESS	
203	205	3 N	Adresse - Longueur	ADDRESS-LEN	
206	225	20 A	Ville	CITY	
226	228	3 N	Ville - Longueur	CITY - LEN	
229	233	5 A	Code Postal	ZIP	
234	235	2 A	State	STATE	
236	238	3 N	N° Departement Employe	DEPT	
239	258	20 A	Nom departement	DNAME	
259	261	3 N	Nom Departement - Longueur	DNAME-LEN	
262	270	9 FILLER			
DE	A	LG	TYPE	NOM	OBSERVATIONS
	1	3	3 A	Type Produit	La valeur est toujours 'PRO'
4	6	3 A	N° Produit	P-NO	
7	9	3 N	Prix produit	PRICE	
10	39	30 A	Description produit	DESCRIPTION	
40	42	3 N	Description produit - Longueur	DESCRIPTION-LEN	
43	44	2 N	Quantité	QUANTITY	
45	47	3 N	Prix commande	PRICE	
48	270	223 FILLER			

```
01 ENR.  
05 ENR-GENERIC.  
10 TYPE-ENR PIC X(3).  
10 FILLER PIC X(267).  
  
05 ENR-ORDER REDEFINES ENR-GENERIC.  
10 TYPE-ORD PIC X(3).  
10 EORD-O-NO PIC S9(3)V USAGE COMP-3.  
10 EORD-O-DATE PIC X(10).  
10 EEMP-E-NO PIC S9(2)V USAGE COMP-3.  
10 EEMP-LNAME PIC X(20).  
10 EEMP-LNAME-LEN PIC S9(4) USAGE COMP-3.  
10 EEMP-FNAME PIC X(20).  
10 EEMP-FNAME-LEN PIC S9(4) USAGE COMP-3.  
10 EEMP-COM PIC S9(2) USAGE COMP-3.  
10 ECUS-C-NO PIC X(4).  
10 ECUS-COMPANY PIC X(30).  
10 ECUS-COMPANY-LEN PIC S9(4) USAGE COMP-3.  
10 ECUS-ADDRESS PIC X(100).  
10 ECUS-ADDRESS-LEN PIC S9(4) USAGE COMP-3.  
10 ECUS-CITY PIC X(20).  
10 ECUS-CITY-LEN PIC S9(4) USAGE COMP-3.  
10 ECUS-ZIP PIC X(5).  
10 ECUS-STATE PIC X(2).  
10 EDEP-DEPT PIC S9(4)V USAGE COMP-3.  
10 EDEP-DNAME PIC X(20).  
10 EDEP-DNAME-LEN PIC S9(4) USAGE COMP-3.  
10 FILLER PIC X(9).  
  
05 ENR-PRODUCT REDEFINES ENR-GENERIC.  
10 TYPE-PRO PIC X(3).  
10 EPRO-P-NO PIC X(3).  
10 EPRO-PRICE PIC S9(3)V9(2) USAGE COMP-3.  
10 EPRO-DESCRIPTION PIC X(30).  
10 EPRO-DESCRIPTION-LEN PIC S9(4) USAGE COMP-3.  
10 EITE-QUANTITY PIC S9(2)V USAGE COMP-3.  
10 EITE-PRICE PIC S9(3)V9(2) USAGE COMP-3.  
10 FILLER PIC X(223).
```



Le fichier de facture

New York, Tuesday, September 02, 2025				
Order N°101				
Date : 2022-02-15				
Your contact within the department Sales : Doe, John				
NO	DESCRIPTION	QUANTITY	PRICE	LINE TOTAL
P01	Mouse Bluetooth	10	25,99\$	259,90\$
P03	Router	15	35,75\$	536,25\$
SUB TOTAL:				796,15\$
SALES TAX (11,5%) :				91,55\$
COMMISSION (5,0%) :				39,80\$
TOTAL :				887,70\$

```
WRITE-PAGE-FACTURE.  
*SI LA LIGNE ACTUELLE DECRIT UNE COMMANDE ON ECRIT LE HEADER  
  IF TYPE-ENR = 'ORD' THEN  
    MOVE ZERO TO WS-SUB-TOTAL  
*SI ON NE FAIT PAS LA SAUVEGARDE ON LE PERDRA QUAND ON PASSERA  
*AU PRODUIT  
  MOVE EEMP-COM TO WS-EMP-COM  
  PERFORM WRITE-PAGE-HEADER  
  PERFORM WRITE-PAGE-ORDER  
  PERFORM READ-FILE-EXT  
ELSE  
*SI LA LIGNE ACTUELLE DECRIT UN PRODUIT ON ECRIT LES INFOS  
*DU PRODUIT  
  PERFORM WRITE-PAGE-PRODUCT-HEADER  
  PERFORM UNTIL FF-EXT = 1 OR TYPE-ENR = 'ORD'  
*CALCUL DU PRIX TOTAL DU PRODUIT EN FONCTION DE LA QUANTITE  
    COMPUTE WS-LINE-PRICE = EPRO-PRICE  
      * EITE-QUANTITY  
*ET ON L'AJOUTE AU TOTAL  
    ADD WS-LINE-PRICE TO WS-SUB-TOTAL  
    PERFORM WRITE-PAGE-PRODUCT-CONTENT  
    PERFORM READ-FILE-EXT  
  END-PERFORM  
  PERFORM WRITE-PAGE-PRODUCT-FOOTER  
*CALCUL DE LA VALEUR FINALE DE LA COMMANDE  
  COMPUTE WS-ST-VALUE = WS-SUB-TOTAL * WS-ST-RATE  
  COMPUTE WS-COM-VALUE = WS-SUB-TOTAL * WS-EMP-COM  
  DISPLAY WS-COM-VALUE  
  COMPUTE WS-TOTAL = WS-SUB-TOTAL + WS-ST-VALUE  
  DISPLAY WS-TOTAL  
  PERFORM WRITE-PAGE-TOTAL  
  IF FF-EXT NOT EQUAL 1 THEN  
    PERFORM WRITE-JUMP-PAGE-FACT  
  END-IF  
END-IF.
```


ABC Company
123 Main Street
New York, 10001
NY

New York, Tuesday, September 02, 2025

Order N°101
Date : 2022-02-15

Your contact within the department Sales : Doe, John

NO	DESCRIPTION	QUANTITY	PRICE	LINE TOTAL
P01	Mouse Bluetooth	10	25.99\$	259.90\$
P03	Router	15	35.75\$	536.25\$

SUB TOTAL:

796,15\$

SALES TAX (11,5%) :

91,55\$

COMMISSION (5,0%) :

39,80\$

TOTAL :

887,70\$

```
WRITE-PAGE-FACTURE.  
*SI LA LIGNE ACTUELLE DECRIT UNE COMMANDE ON ECRIT LE HEADER  
  IF TYPE-ENR = 'ORD' THEN  
    MOVE ZERO TO WS-SUB-TOTAL  
*SI ON NE FAIT PAS LA SAUVEGARDE ON LE PERDRA QUAND ON PASSERA  
*AU PRODUIT  
  MOVE EEMP-COM TO WS-EMP-COM  
  PERFORM WRITE-PAGE-HEADER  
  PERFORM WRITE-PAGE-ORDER  
  PERFORM READ-FILE-EXT  
ELSE  
*SI LA LIGNE ACTUELLE DECRIT UN PRODUIT ON ECRIT LES INFOS  
*DU PRODUIT  
  PERFORM WRITE-PAGE-PRODUCT-HEADER  
  PERFORM UNTIL FF-EXT = 1 OR TYPE-ENR = 'ORD'  
*CALCUL DU PRIX TOTAL DU PRODUIT EN FONCTION DE LA QUANTITE  
    COMPUTE WS-LINE-PRICE = EPRO-PRICE  
    * EITE-QUANTITY  
*ET ON L'AJOUTE AU TOTAL  
    ADD WS-LINE-PRICE TO WS-SUB-TOTAL  
    PERFORM WRITE-PAGE-PRODUCT-CONTENT  
    PERFORM READ-FILE-EXT  
  END-PERFORM  
  PERFORM WRITE-PAGE-PRODUCT-FOOTER  
*CALCUL DE LA VALEUR FINALE DE LA COMMANDE  
  COMPUTE WS-ST-VALUE = WS-SUB-TOTAL * WS-ST-RATE  
  COMPUTE WS-COM-VALUE = WS-SUB-TOTAL * WS-EMP-COM  
  DISPLAY WS-COM-VALUE  
  COMPUTE WS-TOTAL = WS-SUB-TOTAL + WS-ST-VALUE  
  DISPLAY WS-TOTAL  
  PERFORM WRITE-PAGE-TOTAL  
  IF FF-EXT NOT EQUAL 1 THEN  
    PERFORM WRITE-JUMP-PAGE-FACT  
  END-IF  
END-IF.
```



Mise en page de la facture

Utilisation de VARCHAR

```
MOVE L-DATE-TEXT TO P-O-DATE
MOVE SPACE      TO P-CONTACT
STRING
    'Your contact within the department ' DELIMITED BY
        SIZE
EDEP-DNAME (1:EDEP-DNAME-LEN) DELIMITED BY SIZE
    ' : '
        DELIMITED BY SIZE
EEMP-LNAME (1:EEMP-LNAME-LEN) DELIMITED BY SIZE
    ' , '
        DELIMITED BY SIZE
EEMP-FNAME (1:EEMP-FNAME-LEN) DELIMITED BY SIZE
INTO P-CONTACT
PERFORM VARYING WS-IDX FROM 1 BY 1 UNTIL WS-IDX > 9
MOVE P-O-LINES(WS-IDX) TO PAGE-Footer
```

```
01 PAGE-ORDER-ARRAY REDEFINES PAGE-ORDER.
05 PO-LINES OCCURS 9 TIMES PIC X(78).
01 PAGE-PRODUCT.
05 PP-LINE-HEADER.
    10 FILLER PIC X(3) VALUE '| +'.
    10 FILLER PIC X(72) VALUE ALL '-'.
    10 FILLER PIC X(3) VALUE '+ |'.
05 PP-LINE-LIB.
    10 FILLER PIC X(3) VALUE '| |'.
    10 FILLER PIC X(2) VALUE 'NO'.
    10 FILLER PIC X(4) VALUE ALL SPACES.
    10 FILLER PIC X(11) VALUE 'DESCRIPTION'.
    10 FILLER PIC X(25) VALUE ALL SPACES.
    10 FILLER PIC X(8) VALUE 'QUANTITY'.
    10 FILLER PIC X(4) VALUE ALL SPACES.
    10 FILLER PIC X(5) VALUE 'PRICE'.
    10 FILLER PIC X(2) VALUE ALL SPACES.
    10 FILLER PIC X(10) VALUE 'LINE TOTAL'.
    10 FILLER PIC X(4) VALUE '| |'.
05 PP-PRODUCT.
    10 FILLER PIC X(3) VALUE '| |'.
    10 P-P-NO PIC X(4).
    10 FILLER PIC X(2) VALUE ALL SPACES.
    10 P-DESCRIPTION PIC X(35).
    10 FILLER PIC X(7) VALUE ALL SPACES.
    10 P-QUANTITY PIC ZZ.
    10 FILLER PIC X(2) VALUE ALL SPACES.
    10 P-PRICE PIC ZZZ,99.
    10 FILLER PIC X VALUE '$'.
    10 FILLER PIC X(2) VALUE ALL SPACES.
    10 P-LINE-TOTAL PIC ZZZZZ,99.
    10 FILLER PIC X(2) VALUE '$ '.
    10 FILLER PIC X(3) VALUE '| |'.
05 PP-LINE-FOOTER.
    10 FILLER PIC X(3) VALUE '| +'.
    10 FILLER PIC X(72) VALUE ALL '-'.
    10 FILLER PIC X(3) VALUE '+ |'.
01 PAGE-TOTAL.
```



DEMO

Partie 1 + 2 + 3



PARTIE 4

FAUCHER Aymeric

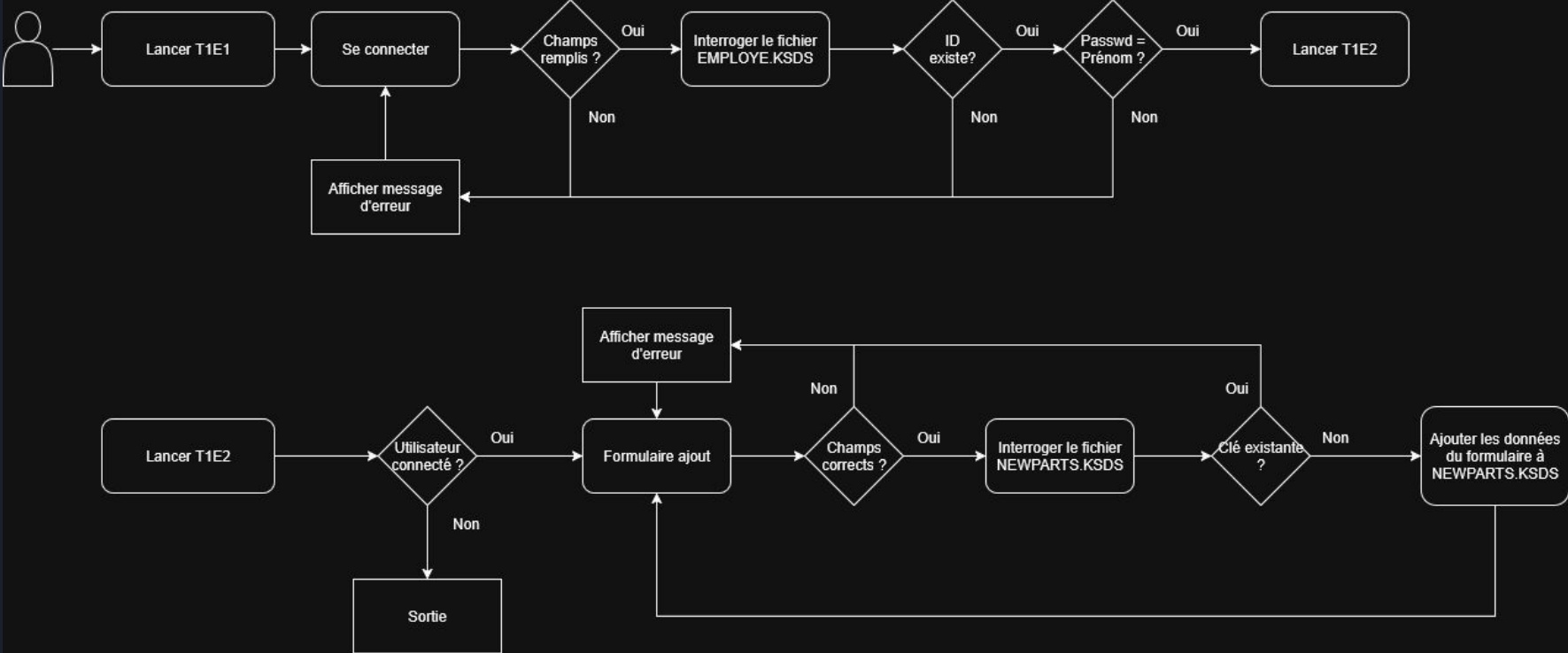
Résumé

- Objectif : Permettre l'ajout de pièces à la base via une interface sécurisée, directement dans le mainframe.
- Authentification :
 - Login = code employé
 - Mot de passe = prénom (via fichier AJC.EMPLOYEE.KSDS)
- Interface CICS :
 - Ajout des nouvelles pièces dans PROJET.NEWPARTS.KSDS
 - Données ensuite transférées en base (hebdomadaire)
 - Interface simple et sécurisée
- Conventions imposées :
 - Noms des fichiers : PARTSX, USERSX
 - Mapset : MSX, Map : MAPX
 - Transaction : TX...

Structure du fichier "PROJET.NEWPARTS.KSDS"

DE	A	LG	TYPE	NOM	OBSERVATIONS	NULLABLE
1	2	2	A	N° Parts	PNO-NPRT	NON
3	32	30	A	Nom Parts	PNAME-NPRT	NON
33	52	20	A	Couleur	COLOR-NPRT	OUI
53	54	2	A	Poids	WEIGHT-NPRT	OUI
55	74	20	N	Ville	CITY-NPRT	OUI
75	80	6	FILLER			

Diagramme IHM



Ecrans

Ecran authentication (T1E1, PGM1LOG)

```
*****
*   DATE : 02/09/25   AJOUT PARTS   HEURE : 22:17:15   *
*****
IDENTIFIANT* : _____
MOT DE PASSE* : _____

*****
* ESC : FIN   ENTER : AJOUTER   * CHAMP OBLIGATOIRE *
* MESSAGE :                                     *
*                                     *
*****
```

Ecran ajout (T1E2, PGM1NPT)

```
*****
*   DATE : 02/09/25   AJOUT PARTS   HEURE : 22:18:14   >
*****
PARTS ID*      : _____

PARTS NAME*    : _____
PARTS COLOR    : _____
PARTS WEIGHT   : _____
PARTS CITY     : _____

*****
* ESC : FIN   ENTER : AJOUTER   * CHAMP OBLIGATOIRE * >
* MESSAGE :                                     >
*                                     >
*****
```


Code vérification formulaires

```
VERIF-SAISIE-LOG.
```

```
  INSPECT LOGINLGI  REPLACING ALL '_' BY SPACES
```

```
  SET ALL-ZONES-OK TO TRUE
```

```
  IF LOGINLGI = SPACES
```

```
    SET MANQUE-ZONE TO TRUE
```

```
    MOVE DFHRED TO LOGINLGC
```

```
    MOVE UNDERSCORE TO LOGINLGI
```

```
    MOVE 'VEUILLEZ ENTRER UN IDENTIFIANT' TO MESS1LGI
```

```
  END-IF
```

```
  IF PASSLGI = SPACES
```

```
    SET MANQUE-ZONE TO TRUE
```

```
    MOVE DFHRED TO PASSLGC
```

```
    MOVE 'VEUILLEZ ENTRER UN MOT DE PASSE' TO MESS1LGI
```

```
  END-IF
```

```
  .
```

```
VERIF-SAISIE.
```

```
  INSPECT PNAMENPI  REPLACING ALL '_' BY SPACES
```

```
  INSPECT COLORNPI  REPLACING ALL '_' BY SPACES
```

```
  INSPECT CITYNPI   REPLACING ALL '_' BY SPACES
```

```
  INSPECT WEIGHTNPI REPLACING ALL '_' BY SPACES
```

```
  INSPECT PNONPI    REPLACING ALL '_' BY SPACES
```

```
  SET ALL-ZONES-OK TO TRUE
```

```
  IF WEIGHTNPI IS NOT NUMERIC OR WEIGHTNPI = SPACES
```

```
    SET MANQUE-ZONE TO TRUE
```

```
    MOVE DFHRED TO WEIGHTNPC
```

```
    MOVE 'WEIGHT DEVRAIT ETRE NUMERIQUE !' TO MESS1NPI
```

```
    MOVE ZERO TO WEIGHTNPI
```

```
  END-IF
```

```
  IF PNAMENPI = SPACES
```

```
    SET MANQUE-ZONE TO TRUE
```

```
    MOVE DFHRED TO PNAMENPC
```

```
    MOVE UNDERSCORE TO PNAMENPI
```

```
    MOVE 'LE CHAMP NAME EST OBLIGATOIRE !' TO MESS1NPI
```

```
  END-IF
```

```
  IF PNONPI = SPACES
```

```
    SET MANQUE-ZONE TO TRUE
```

```
    MOVE DFHRED TO PNONPC
```

```
    MOVE 'LE CHAMP ID EST OBLIGATOIRE !' TO MESS1NPI
```

```
  END-IF
```

```
  .
```

LECTURE-LOG.

```
EXEC CICS
  READ DATASET ('USERS1')
    INTO (ENR-EMP)
    RIDFLD (LOGINLGI)
    RESP (WS-CD-ERR)
END-EXEC
EVALUATE WS-CD-ERR
  WHEN ZERO
    CONTINUE
  WHEN 13
    MOVE 'IDENTIFIANT / PASSWORD INCORRECT(S)'
      TO MESS1LGI
    MOVE DFHRED TO LOGINLGC
    MOVE SPACES TO PASSLGI
    PERFORM ENVOI-ECRAN
  WHEN OTHER
    MOVE 'ERREUR READ USERS1' TO WS-ERR-MESS
    PERFORM FIN-TOTALE
END-EVALUATE.
IF PASSLGO = PREN-EMP
  SET LOGGED TO TRUE
  MOVE 'IDENTIFICATION SUCCES' TO MESS1LGI
  MOVE SPACES TO PASSLGI
  PERFORM ENVOI-ECRAN
ELSE
  MOVE 'IDENTIFIANT / PASSWORD INCORRECT(S)'
    TO MESS1LGI
  MOVE DFHRED TO LOGINLGC
  MOVE SPACES TO PASSLGI
  PERFORM ENVOI-ECRAN
END-IF.
```

AJOUT-NPT.

```
IF ALL-ZONES-OK
  MOVE PNONPI TO PNO-NPT
  MOVE PNAMENPI TO PNAME-NPT
  MOVE COLORNPI TO COLOR-NPT
  MOVE WEIGHTNPI TO WEIGHT-NPT
  MOVE CITYNPI TO CITY-NPT

EXEC CICS
  WRITE DATASET ('PARTS1')
    FROM (ENR-NPT)
    RIDFLD(PNO-NPT)
    RESP (WS-CD-ERR)
    RESP2(WS-CD-ERR2)
END-EXEC
EVALUATE WS-CD-ERR
  WHEN ZERO
    MOVE 'AJOUT EFFECTUE ' TO MESS1NPI
    MOVE DFHBLUE TO PNONPC
    PERFORM ENVOI-ECRAN
  WHEN 14
    MOVE 'DOUBLON SUR ' TO MESS1NPI
    MOVE PNONPI TO MESS2NPI
    PERFORM ENVOI-ECRAN
  WHEN OTHER
    STRING
      'ERREUR WRITE PARTS1 '
      WS-CD-ERR DELIMITED BY SIZE
      SPACE DELIMITED BY SIZE
      WS-CD-ERR2 DELIMITED BY SIZE
    INTO WS-ERR-MESS
    PERFORM FIN-TOTALE
  END-EVALUATE
END-IF.
```

Transition entre T1E1 et T1E2

```

IF LOGGED THEN
    MOVE 'PGM1LOG' TO PREV-PGM
    EXEC CICS
        RETURN TRANSID('T1E2')
        COMMAREA(ZONE)
        LENGTH(LENGTH OF ZONE)
    END-EXEC
ELSE
    * REAFFICHAGE DE LA TRANSACTION T1E1
    EXEC CICS
        RETURN TRANSID ('T1E1')
        COMMAREA (ZONE)
        LENGTH (LENGTH OF ZONE)
    END-EXEC
END-IF.

```

On vérifie si T1E2 a été lancée correctement

```

EVALUATE EIBTRNID
    WHEN 'T1E2'
        IF EIBCALEN = ZERO
            MOVE 'NON AUTHENTIFIE!'
            TO WS-ERR-MESS
            PERFORM FIN-TOTALE
        ELSE
            MOVE DFHCOMMAREA TO ZONE
            IF LOGGED AND PREV-PGM = 'PGM1LOG'
                MOVE LOW-VALUE TO MAP1NPTO
                PERFORM ENVOI-ECRAN
            END-IF
            PERFORM GESTION-TOUCHES
        END-IF
    WHEN OTHER
        MOVE LOW-VALUE TO MAP1NPTO
END-EVALUATE
PERFORM ENVOI-ECRAN.

```



DEMO

Partie 4

Tests unitaires

```
000100 IDENTIFICATION DIVISION.
000200 PROGRAM-ID. TESTUNCD.
000210
000300 DATA DIVISION.
000400 WORKING-STORAGE SECTION.
000500 01 WS-DATE          PIC X(10).
000600 01 WS-FORMAT        PIC X(02).
000730 01 EXPECTED-DATE   PIC X(10).
000740
000750 01 LIB              PIC X(30).
000751 01 L-SEP         PIC X(30) VALUE ALL "*".
000760
000770 LINKAGE SECTION.
000780 COPY TESTCONT.
000791
000800 PROCEDURE DIVISION USING TEST-CONTEXT.
000900
001000     PERFORM TEST-FR-DATE
001100     DISPLAY 'RUN ', TESTS-RUN, ' OK ', PASSES, ' KO ', FAILURES
001200     DISPLAY L-SEP
001300
001620     PERFORM TEST-US-DATE
001630     DISPLAY 'RUN ', TESTS-RUN, ' OK ', PASSES, ' KO ', FAILURES
001640     DISPLAY L-SEP
001700
001800     GOBACK.
```

```
002000 APPELS.
002100     CALL 'CONVDATE' USING WS-DATE, WS-FORMAT
002200     DISPLAY '***** ' LIB ' *****'
002300     CALL 'ASSEQX' USING BY REFERENCE TEST-CONTEXT
002400                               BY CONTENT LIB
002500                               BY CONTENT EXPECTED-DATE
002600                               BY CONTENT WS-DATE.
002700
002800 TEST-FR-DATE.
002900     MOVE '10/10/2022' TO WS-DATE
003000     MOVE 'FR'        TO WS-FORMAT
003100     MOVE '10-10-2022' TO EXPECTED-DATE
003200     MOVE 'TEST-FR-DATE' TO LIB
003300     PERFORM APPELS.
003400
003500 TEST-US-DATE.
003600     MOVE '26/04/2023' TO WS-DATE
003700     MOVE 'US'         TO WS-FORMAT
003800     MOVE '2023-04-26' TO EXPECTED-DATE
003900     MOVE 'TEST-US-DATE' TO LIB
004000     PERFORM APPELS.
```



Tests unitaires

```
***** TOP OF DATA *****
***** TEST-FR-DATE *****
RUN 01 OK 01 KO
*****
***** TEST-US-DATE *****
RUN 02 OK 02 KO
*****
***** BOTTOM OF DATA *****
```



Conclusion

Ce traitement assure une gestion complète et cohérente du cycle de vie des produits et des ventes.

L'ajout d'une interface CICS sécurisée renforce l'efficacité et la fiabilité du système.

Travail en équipe et organisation de groupe.