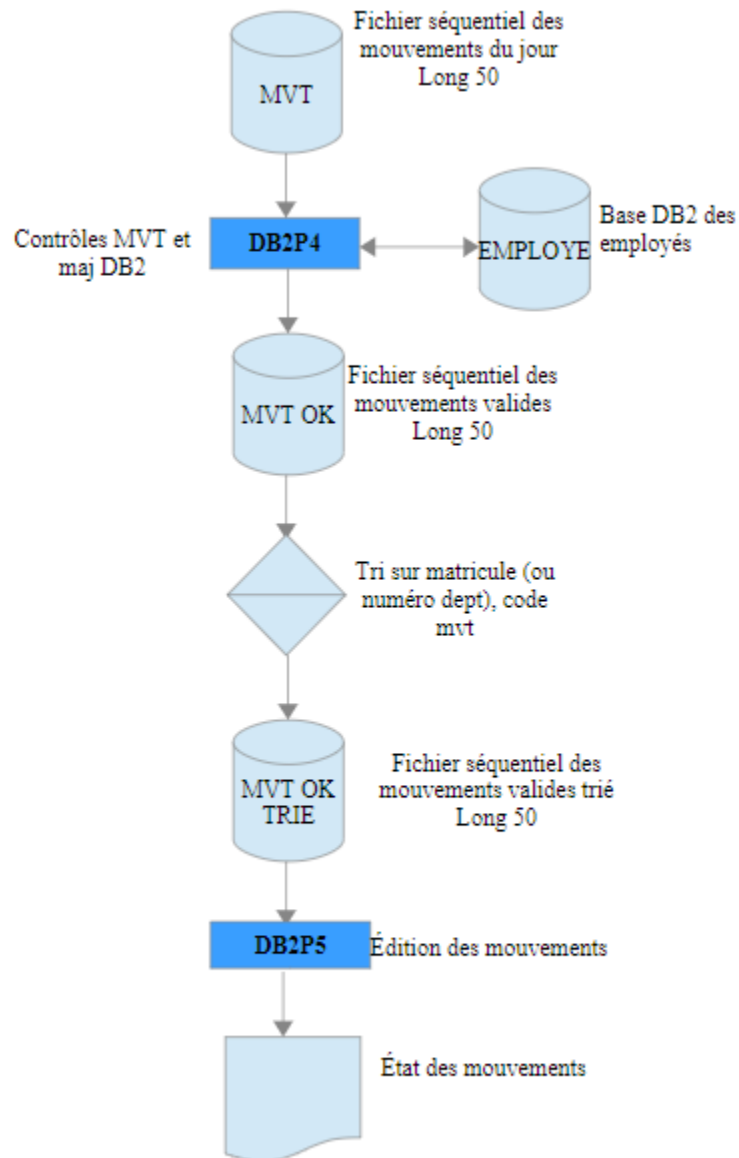


PROJET COBOL/DB2

Énoncé : Mise à jour de la base EMPLOYE à partir d'un fichier séquentiel des mouvements du jour et édition des mouvements valides ayant impactés la base.



Programme DB2P4

Dans la bibliothèque **userid.SOURCE.DB2**, écrire le programme **DB2P4** qui met à jour la base EMPLOYE à partir du fichier séquentiel mouvements **API0.COB.MVT.BASE.EMPLOYE** en fonction d'un code mouvement indiquant l'action à réaliser (création, modification, suppression) et d'un code indiquant la table à mettre à jour.

Règles de gestion :

Contrôle de l'enregistrement :

Contrôler chaque champ de chaque enregistrement (cf règles ci-dessous).

Afficher toutes les erreurs rencontrées (il peut y avoir plusieurs erreurs par enregistrement).

Ne pas faire la mise à jour dans la table adéquate s'il y a au moins une erreur dans l'enregistrement.

Arrêter le programme pour toute erreur technique.

Incrémenter le compteur d'enregistrements en erreur.

Contrôles par champ :

- code mouvement : doit être égal à 'C', 'M' ou 'S' sinon afficher erreur 1
- code table : doit être égal à 'D' ou 'E' sinon afficher erreur 2

Ne pas poursuivre les contrôles si le code mouvement ou le code table est erroné

1) Si code mouvement = 'C' et code table = 'E' (création employé)

- matricule : doit être renseigné sinon erreur 3, ne doit pas exister en table EMPLOYE sinon erreur 4
- nom : doit être renseigné sinon erreur 5
- numéro de département : doit être renseigné sinon erreur 6, doit exister en table DEPARTEMENT sinon erreur 7
- date entrée : doit être numérique et au format AAAAMMJJ, doit être cohérente sinon erreur 8
- salaire : doit être numérique supérieur à zéro sinon erreur 9
- commission : doit être numérique, peut être égal à zéro sinon erreur 10

2) Si code mouvement = 'M' et code table = 'E' (modification employé)

- matricule : doit être renseigné sinon erreur 3, doit exister en table EMPLOYE sinon erreur 11
- nom : doit être renseigné sinon erreur 5
- numéro de département : doit être renseigné sinon erreur 6 , doit exister en table DEPARTEMENT sinon erreur 7
- date entrée : doit être numérique et au format AAAAMMJJ, doit être cohérente sinon erreur 8
- salaire : doit être numérique supérieur à zéro sinon erreur 9

- commission : doit être numérique, peut être égal à zéro sinon erreur 10

3) Si code mouvement = 'S' et code table = 'E' (suppression employé)

- matricule : doit être renseigné sinon erreur 3, doit exister en table EMPLOYE sinon erreur 11, **ne doit pas être référencé en tant que chef dans la table DEPARTEMENT sinon erreur 17**

pas de contrôle sur les autres zones

4) Si code mouvement = 'C' et code table = 'D' (création département)

- numéro : doit être renseigné sinon erreur 6, ne doit pas exister en table DEPARTEMENT sinon erreur 12

- nom : doit être renseigné sinon erreur 13

- bâtiment : doit être renseigné sinon erreur 14

- matricule du chef : doit être renseigné sinon erreur 15, doit exister en table EMPLOYE sinon erreur 16

5) Si code mouvement = 'M' et code table = 'D' (modification département)

- numéro : doit être renseigné sinon erreur 6, doit exister en table DEPARTEMENT sinon erreur 7

- nom : doit être renseigné sinon erreur 13

- bâtiment : doit être renseigné sinon erreur 14

- matricule du chef : doit être renseigné sinon erreur 15, doit exister en table EMPLOYE sinon erreur 16

6) Si code mouvement = 'S' et code table = 'D' (suppression département)

- numéro : doit être renseigné sinon erreur 6, doit exister en table DEPARTEMENT sinon erreur 7, **ne doit pas être référencé dans la table EMPLOYE sinon erreur 18**

pas de contrôle sur les autres zones

Mise à jour :

Pas de mise à jour si l'enregistrement comporte une ou plusieurs erreurs

7) Si code mouvement = 'C' et code table = 'E' (création employé)

créer l'employé dans la table EMPLOYE avec les champs de l'enregistrement

Mettre la commission à NULL dans la table si elle est à zéro dans l'enregistrement

incrémenter le compteur des employés créés

8) Si code mouvement = 'M' et code table = 'E' (modification employé)

modifier l'employé avec les champs de l'enregistrement

Mettre la commission à NULL dans la table si elle est à zéro dans l'enregistrement

incrémenter le compteur des employés modifiés

9) Si code mouvement = 'S' et code table = 'E' (suppression employé)

supprimer l'employé de la table EMPLOYE

incrémenter le compteur des employés supprimés

10) Si code mouvement = 'C' et code table = 'D' (création département)

créer le département dans la table DEPARTEMENT avec les champs de l'enregistrement

incrémenter le compteur des départements créés

11) Si code mouvement = 'M' et code table = 'D' (modification département)

modifier le département avec les champs de l'enregistrement

incrémenter le compteur des départements modifiés

12) Si code mouvement = 'S' et code table = 'D' (suppression département)

supprimer le département de la table DEPARTEMENT

incrémenter le compteur des départements supprimés

**13) Reconduire l'enregistrement traité dans le fichier des mouvements valides
userid.COB.MVT.VALIDE**

incrémenter le compteur des enregistrements valides créés

En fin de programme :

Afficher les compteurs :

- nb enregistrements lus
- **nb enregistrements valides créés**
- nb enregistrements erronés
- nb employés créés
- nb employés modifiés
- nb employés supprimés
- nb départements créés
- nb départements modifiés
- nb départements supprimés

Programme DB2P5
Édition des mouvements valides

Créer l'état des mouvements valides du jour :

MOUVEMENTS DU JOUR

LE 2 mai 2023

TABLE EMPLOYE

ACTION	MAT	NOM EMP	NOD	NOM DEPT	DATE ENT	SALAIRE	COMMISSION
CREATION	80	SARAH	P01	INFO	27/04/2023	2 500,00	1 000,00
MODIFICATION	10	DURAND	E10	PAIE	10/02/2000	12 000,00	5 000,00
...							

TABLE DEPARTEMENT

ACTION	NOD	NOM DEPT	LIEU	MAT CHEF	NOM CHEF
CREATION	M02	MARKET	BAT2	70	NOEL
MODIFICATION	E10	PAIE	BAT1	40	JAVA
...					

ANNEXE

Caractéristiques du fichier mouvements :

- format des enregistrements : fixe bloqué
- longueur d'un enregistrement : 50 octets

Description des enregistrements :

- code mouvement : 1 caractère alphanumérique, valeurs possibles 'C' pour création, 'M' pour modification, 'S' pour suppression
- code table : 1 caractère alphanumérique, valeurs possibles 'D' pour table DEPARTEMENT, 'E' pour table EMPLOYE
- la suite de l'enregistrement dépend de la table à mettre à jour :
 - si code table = 'D' la suite de l'enregistrement est :
 - numéro de département : 3 caractères alphanumériques
 - nom du département : 6 caractères alphanumériques
 - bâtiment : 4 caractères alphanumériques
 - matricule du chef : 3 caractères alphanumériques
 - si code table = 'E' la suite de l'enregistrement est :
 - matricule : 3 caractères alphanumériques
 - nom : 7 caractères alphanumériques
 - numéro de département : 3 caractères alphanumériques
 - date entrée : 8 caractères numériques
 - salaire : 5 entiers, 2 décimales
 - commission : 5 entiers, 2 décimales

Créer la clause COPY de la description l'enregistrement dans la bibliothèque
userid.SOURCE.COPY

TABLE DES ERREURS

Num	Libellé
1	CODE MOUVEMENT ERRONE
2	CODE TABLE ERRONE
3	MATRICULE NON RENSEIGNE
4	MATRICULE DEJA EXISTANT
5	NOM NON RENSEIGNE
6	DEPARTEMENT NON RENSEIGNE
7	DEPARTEMENT INEXISTANT
8	DATE ENTREE ERRONEE
9	SALAIRE NON NUMERIQUE OU NUL
10	COMMISSION NON NUMERIQUE
11	MATRICULE INEXISTANT
12	DEPARTEMENT DEJA EXISTANT
13	NOM DEPARTEMENT NON RENSEIGNE
14	BATIMENT NON RENSEIGNE
15	MATRICULE DU CHEF NON RENSEIGNE
16	MATRICULE DU CHEF INEXISTANT
17	SUPPRESSION EMPLOYE IMPOSSIBLE, CONTRAINTE SUR DEP
18	SUPPRESSION DEP IMPOSSIBLE, CONTRAINTE SUR EMPLOYE

CONTENU DU FICHIER DES MOUVEMENTS

***** Top of Data *****

CD
CE
CDP01
CE10
MDAAA 100
MEAAA
SDAAA
SEAAA
ME100AAAAAAA 0000000000000000000000
CE80 SARAH M012023042702500000100000
CE80 SARAH P012023042702500000100000
ME10 DURAND E102000021012000000500000
ME60 PARIS E1020080422100000000000000
MDE10PAIE BAT140
CE90 JEAN M012023011010780000000000
CDM02MARKETBAT270
CDM03BIDON BAT470
SDM03
SDM01
SDE10
SE30
SE20

***** Bottom of Data *****

DESCRIPTION DES ENRGISTREMENTS

FD FICMVT.
01 ENR-MVT PIC X(50).

WORKING-STORAGE SECTION.
01 MVT-ENR.
 05 MVT-CODE-MVT PIC X.
 05 MVT-CODE-TAB PIC X.
 05 MVT-SUITE PIC X(48).
 05 MVT-EMPLOYE REDEFINES MVT-SUITE.
 10 MVT-MAT PIC X(3).
 10 MVT-NOM PIC X(7).
 10 **MVT-NOD1** PIC X(3).
 10 MVT-DAT PIC 9(8).
 10 MVT-SAL PIC 9(5)V99.
 10 MVT-COM PIC 9(5)V99.
 10 PIC X(13).
 05 MVT-DEPARTEMENT REDEFINES MVT-SUITE.
 10 **MVT-NOD2** PIC X(3).
 10 MVT-NDE PIC X(6).
 10 MVT-LIE PIC X(4).
 10 MVT-CHE PIC X(3).
 10 PIC X(32).

PROCEDURE DIVISION.
 READ FICMVT INTO MVT-ENR.

CONTENU DES TABLES AVANT MAJ

```
-----+-----+-----+-----+-----+-----
SELECT * FROM EMPLOYE;
-----+-----+-----+-----+-----+-----
MAT  NOM      NOD  DAT      SAL      COM
-----+-----+-----+-----+-----+-----
10   DURAND    E10  2000-02-10  11000.00  5000.00
20   DUPOND    P01  1998-01-11  15100.50  -----
30   BARI      P02  2001-01-04   6000.00  3000.00
40   JAVA      C04  2005-12-03  21000.50  1000.50
50   CICS      P02  2006-06-20   8500.00  3500.50
60   PARIS     P01  2008-04-22   9000.00  3500.00
70   NOEL      P02  2007-05-18  10500.00   900.50
DSNE610I NUMBER OF ROWS DISPLAYED IS 7
```

```
-----+-----+-----+-----+-----+-----
SELECT * FROM DEPARTEMENT;
-----+-----+-----+-----+-----+-----
NOD  NDE      LIE  CHE
-----+-----+-----+-----+-----+-----
C04  COMPTA   BAT1  40
E10  PAIE     BAT2  10
P01  INFO     BAT3  20
P02  COMMER   BAT2  70
DSNE610I NUMBER OF ROWS DISPLAYED IS 4
```

CONTENU DES TABLES APRES MAJ

```
-----+-----+-----+-----+-----+-----
      SELECT * FROM EMPLOYE;
-----+-----+-----+-----+-----+-----
MAT  NOM          NOD  DAT          SAL          COM
-----+-----+-----+-----+-----+-----
10   DURAND        E10  2000-02-10    12000.00      5000.00
20   DUPOND        P01  1998-01-11    15100.50      -----
30   BARI          P02  2001-01-04    6000.00      3000.00
40   JAVA          C04  2005-12-03    21000.50      1000.50
50   CICS          P02  2006-06-20    8500.00      3500.50
60   PARIS         E10  2008-04-22    10000.00      -----
70   NOEL          P02  2007-05-18    10500.00      900.50
80   SARAH         P01  2023-04-27    2500.00      1000.00
DSNE610I NUMBER OF ROWS DISPLAYED IS 7
```

```
-----+-----+-----+-----+-----+-----
      SELECT * FROM DEPARTEMENT;
-----+-----+-----+-----+-----+-----
NOD  NDE          LIE  CHE
-----+-----+-----+-----+-----+-----
C04  COMPTA        BAT1  40
E10  PAIE          BAT1  40
P01  INFO          BAT3  20
P02  COMMER        BAT2  70
M02  MARKET        BAT2  70
M03  BIDON        BAT4  70
DSNE610I NUMBER OF ROWS DISPLAYED IS 5
```

```
FICHER DES MOUVEMENTS VALIDES ISSU DU TRAITEMENT DE MISE A JOUR
***** Top of Data *
CE80 SARAH  P0120230427025000000100000
ME10 DURAND E1020000210120000000500000
ME60 PARIS  E1020080422100000000000000
MDE10PAIE  BAT140
CDM02MARKETBAT270
CDM03BIDON BAT470
SDM03
SE30
***** Bottom of Data
```

CONTENU DE LA SYSOUT

ENR 01 :

ERREUR 01 : CODE MOUVEMENT ERRONE
ERREUR 02 : CODE TABLE ERRONE

ENR 02 : CD

ERREUR 06 : DEPARTEMENT NON RENSEIGNE
ERREUR 07 : departement non présent dans la table département
ERREUR 13 : NOM DEPARTEMENT NON RENSEIGNE
ERREUR 14 : BATIMENT NON RENSEIGNE
ERREUR 15 : MATRICULE DU CHEF NON RENSEIGNE

ENR 03 : CE

ERREUR 03 : MATRICULE NON RENSEIGNE
ERREUR 05 : NOM NON RENSEIGNE
ERREUR 06 : DEPARTEMENT NON RENSEIGNE
ERREUR 07 : DEPARTEMENT INEXISTANT
ERREUR 08 : DATE ENTREE ERRONEE
ERREUR 09 : SALAIRE NON NUMERIQUE OU NUL
ERREUR 10 : COMMISSION NON NUMERIQUE

ENR 04 : CDP01

ERREUR 12 : DEPARTEMENT DEJA EXISTANT
ERREUR 13 : NOM DEPARTEMENT NON RENSEIGNE
ERREUR 14 : BATIMENT NON RENSEIGNE
ERREUR 15 : MATRICULE DU CHEF NON RENSEIGNE

ENR 05 : CE10

ERREUR 04 : MATRICULE DEJA EXISTANT
ERREUR 05 : NOM NON RENSEIGNE
ERREUR 06 : DEPARTEMENT NON RENSEIGNE
ERREUR 08 : DATE ENTREE ERRONEE
ERREUR 09 : SALAIRE NON NUMERIQUE OU NUL
ERREUR 10 : COMMISSION NON NUMERIQUE

ENR 06 : MDAAA 100

ERREUR 07 : DEPARTEMENT INEXISTANT
ERREUR 13 : NOM DEPARTEMENT NON RENSEIGNE
ERREUR 14 : BATIMENT NON RENSEIGNE
ERREUR 16 : MATRICULE DU CHEF INEXISTANT

ENR 07 : MEAAA

ERREUR 11 : MATRICULE INEXISTANT
ERREUR 05 : NOM NON RENSEIGNE
ERREUR 06 : DEPARTEMENT NON RENSEIGNE
ERREUR 08 : DATE ENTREE ERRONEE
ERREUR 09 : SALAIRE NON NUMERIQUE OU NUL
ERREUR 10 : COMMISSION NON NUMERIQUE

ENR 08 : SDAAA

ERREUR 07 : DEPARTEMENT INEXISTANT

ENR 09 : SEAAA

ERREUR 11 : MATRICULE INEXISTANT

ENR 10 : ME100AAAAAAA 00000000000000000000

ERREUR 11 : MATRICULE INEXISTANT
ERREUR 06 : DEPARTEMENT NON RENSEIGNE
ERREUR 08 : DATE ENTREE ERRONEE
ERREUR 09 : SALAIRE NON NUMERIQUE OU NUL

ENR 11 : CE80 SARAH M012023042702500000100000

ERREUR 07 : DEPARTEMENT INEXISTANT

ENR 12 : CE80 SARAH P012023042702500000100000

```

-----> CREATION EMPLOYE
HVE-EMPLOYE : 80 SARAH P012023-04-2702500000100000

ENR 13 : ME10 DURAND E102000021012000000500000
          2000 BISSEXTILE
-----> MODIFICATION EMPLOYE
HVE-EMPLOYE : 10 DURAND E102000-02-1012000000500000

ENR 14 : ME60 PARIS E102008042210000000000000
-----> MODIFICATION EMPLOYE
HVE-EMPLOYE : 60 PARIS E102008-04-2210000000000000
ENR 15 : MDE10PAIE BAT140
-----> MODIFICATION DEPARTEMENT
HVD-DEPARTEMENT : E10PAIE BAT140

ENR 16 : CE90 JEAN M012023011010780000000000
          ERREUR 07 : DEPARTEMENT INEXISTANT

ENR 17 : CDM02MARKETBAT270
-----> CREATION DEPARTEMENT
HVD-DEPARTEMENT : M02MARKETBAT270

ENR 18 : CDM03BIDON BAT470
-----> CREATION DEPARTEMENT
HVD-DEPARTEMENT : M03BIDON BAT470

ENR 19 : SDM03
-----> SUPPRESSION DEPARTEMENT : M03BIDON BAT470

ENR 20 : SDM01
          ERREUR 07 : DEPARTEMENT INEXISTANT

ENR 21 : SDE10
          ERREUR 18 : SUPPRESSION DEP IMPOSSIBLE, CONTRAINTE SUR EMPLOYE

ENR 22 : SE30
-----> SUPPRESSION EMPLOYE : 30 BARI P02 2001-01-04

ENR 23 : SE20
          ERREUR 17 : SUPPRESSION EMPLOYE IMPOSSIBLE, CONTRAINTE SUR DEP
NB ENR LUS : 23
NB ENR OK : 08
NB ENR ERR : 15
NB EMP CRE : 01
NB EMP MOD : 02
NB EMP SUP : 01
NB DEP CRE : 02
NB DEP MOD : 01
NB DEP SUP : 01

```

CORRECTION

PROGRAMME DB2P4

```
CBL DYNAM
IDENTIFICATION DIVISION.
PROGRAM-ID. DB2P4.
*****
* BUT : MISE A JOUR DE LA BASE EMPLOYE *
* PRINCIPALES FONCTIONS : *
* - CONTROLE DES MOUVEMENTS DU JOUR *
* - CREATION, MODIFICATION, SUPPRESSION DANS LES TABLES *
* EMPLOYE ET DEPARTEMENT *
* - RECONDUCTION DES MOUVEMENTS VALIDES POUR EDITION ULTERIEURE *
*****
AUTHOR. GIUSEPPE.
DATE-WRITTEN. 03/05/23.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.
    DECIMAL-POINT IS COMMA.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT MVT          ASSIGN MVT
        ORGANIZATION IS SEQUENTIAL
        ACCESS MODE   IS SEQUENTIAL
        FILE STATUS   IS FS-MVT.
    SELECT MVTVAL       ASSIGN MVTVAL
        ORGANIZATION IS SEQUENTIAL
        ACCESS MODE   IS SEQUENTIAL
        FILE STATUS   IS FS-MVTVAL.

DATA DIVISION.
FILE SECTION.
FD  MVT
    RECORDING MODE IS F.
01  ENR-MVT          PIC X(50) .
FD  MVTVAL
    RECORDING MODE IS F.
01  ENR-MVTVAL       PIC X(50) .

WORKING-STORAGE SECTION.
01  MVT-ENR.
* CODE MVT : C, M, S
    05 MVT-CODE-MVT    PIC X.
* CODE TAB : D, E
    05 MVT-CODE-TAB    PIC X.
* SUITE
    05 MVT-SUITE       PIC X(48) .
*
    05 MVT-EMPLOYE      REDEFINES MVT-SUITE.
        10 MVT-MAT      PIC X(3) .
        10 MVT-NOM      PIC X(7) .
        10 MVT-NOD1     PIC X(3) .
        10 MVT-DAT      PIC 9(8) .
        10 MVT-SAL      PIC 9(5)V99.
        10 MVT-COM      PIC 9(5)V99.
        10              PIC X(13) .
*
    05 MVT-DEPARTEMENT REDEFINES MVT-SUITE.
        10 MVT-NOD2     PIC X(3) .
        10 MVT-NDE      PIC X(6) .
        10 MVT-LIE      PIC X(4) .
        10 MVT-CHE      PIC X(3) .
        10              PIC X(32) .

* FILE STATUS MVT
```

```

01  FS-MVT                      PIC 99 VALUE ZEROES.

* FILE STATUS MVTVAL
01  FS-MVTVAL                  PIC 99 VALUE ZEROES.

* INDICATEUR FIN DE FICHIER MVT
01                                PIC X  VALUE SPACES.
    88 FIN-MVT                  VALUE '5'.

* DECLARATION DE LA SQLCA
    EXEC SQL INCLUDE SQLCA      END-EXEC.

* DECLARATION DE LA TABLE EMPLOYE
    EXEC SQL INCLUDE EMPLOYE    END-EXEC.

* DECLARATION DE LA TABLE DEPARTEMENT
    EXEC SQL INCLUDE DEPART     END-EXEC.

* DECLARATION DES HOST VARIABLES DE LA TABLE EMPLOYE
01  HVE-EMP.
    05 HVE-MAT                  PIC X(3) .
    05 HVE-NOD                  PIC X(3) .
    05 HVE-NOM                  PIC X(7) .
    05 HVE-SAL                  PIC S9(5)V99 COMP-3.
    05 HVE-COM                  PIC S9(5)V99 COMP-3.
    05 HVE-DAT                  PIC X(10) .

* DECLARATION DES HOST VARIABLES DE LA TABLE DEPARTEMENT
01  HVD-DEP.
    05 HVD-NOD                  PIC X(3) .
    05 HVD-LIE                  PIC X(4) .
    05 HVD-CHE                  PIC X(3) .
    05 HVD-NDE                  PIC X(6) .

* INDICATEURS DE NULLITE
01  IND-COM                     PIC S9(4) COMP VALUE ZEROES.

* COMPTEURS
01  COMPTEURS.
    05 CPT-MVT-LUS              PIC 99  VALUE ZEROES.
    05 CPT-MVT-ERR              PIC 99  VALUE ZEROES.
    05 CPT-MVT-VAL              PIC 99  VALUE ZEROES.
    05 CPT-EMP-CRE              PIC 99  VALUE ZEROES.
    05 CPT-EMP-MAJ              PIC 99  VALUE ZEROES.
    05 CPT-EMP-SUP              PIC 99  VALUE ZEROES.
    05 CPT-DEP-CRE              PIC 99  VALUE ZEROES.
    05 CPT-DEP-MAJ              PIC 99  VALUE ZEROES.
    05 CPT-DEP-SUP              PIC 99  VALUE ZEROES.

* SQLCODE EN PICTURE D'EDITION
01  ED-SQLCODE                  PIC +(10) VALUE ZEROES.

* TABLE DES ERREURS
01  TAB-ERR.
    05                          PIC X(50) VALUE
        'CODE MOUVEMENT ERRONE'.
    05                          PIC X(50) VALUE
        'CODE TABLE ERRONE'.
    05                          PIC X(50) VALUE
        'MATRICULE NON RENSEIGNE'.
    05                          PIC X(50) VALUE
        'MATRICULE DEJA EXISTANT'.
    05                          PIC X(50) VALUE
        'NOM NON RENSEIGNE'.
    05                          PIC X(50) VALUE
        'DEPARTEMENT NON RENSEIGNE'.

```

```

05          PIC X(50) VALUE
'DEPARTEMENT INEXISTANT'.
05          PIC X(50) VALUE
'DATE ENTREE ERRONEE'.
05          PIC X(50) VALUE
'SALAIRE NON NUMERIQUE OU NUL'.
05          PIC X(50) VALUE
'COMMISSION NON NUMERIQUE'.
05          PIC X(50) VALUE
'MATRICULE INEXISTANT'.
05          PIC X(50) VALUE
'DEPARTEMENT DEJA EXISTANT'.
05          PIC X(50) VALUE
'NOM DEPARTEMENT NON RENSEIGNE'.
05          PIC X(50) VALUE
'BATIMENT NON RENSEIGNE'.
05          PIC X(50) VALUE
'MATRICULE DU CHEF NON RENSEIGNE'.
05          PIC X(50) VALUE
'MATRICULE DU CHEF INEXISTANT'.
05          PIC X(50) VALUE
'SUPPRESSION EMPLOYE IMPOSSIBLE, CONTRAINTE SUR DEP'.
05          PIC X(50) VALUE
'SUPPRESSION DEP IMPOSSIBLE, CONTRAINTE SUR EMPLOYE'.
01 REDEFINES TAB-ERR.
05 POSTE-ERR          PIC X(50) OCCURS 18.

* INDICATEUR DE BALAYAGE DE LA TABLE D'ERREUR
01 I          PIC 99      VALUE ZEROES.

* INDICATEUR D'ERREUR
01          PIC XX      VALUE SPACES.
    88 OK          VALUE 'OK'.
    88 ERREUR      VALUE 'KO'.

* INDICATEUR D'EXISTENCE DE L'EMPLOYE EN TABLE EMPLOYE
01          PIC X      VALUE SPACES.
    88 EMPLOYE-EXISTANT      VALUE '0'.
    88 EMPLOYE-INEXISTANT    VALUE '1'.

* DESCRIPTION DE LA ZONE DE COMMUNICATION AVEC LE SOUS-PROGRAMME
* DE CONTROLE DE DATE
01 ZCOM.
05 ZDATE.
    10 ZANNEE          PIC 9(4).
    10 ZMOIS           PIC 9(2).
    10 ZJOUR           PIC 9(2).
05 ZCDRET             PIC 9.
    88 ZCDRET-OK       VALUE 0.
    88 ZCDRET-KO       VALUE 1.
05 ZDATE-EDIT         PIC X(17).
01 SPDAT              PIC X(8) VALUE 'SPDAT'.

PROCEDURE DIVISION.
* OUVERTURE DES FICHIERS, 1ERE LECTURE MVT
PERFORM DEBUT

* BOUCLE DE CONTROLE ET TRAITEMENT D'UN MVT, LECTURE MVT SUIVANT
PERFORM UNTIL FIN-MVT
PERFORM CONTROLES
IF OK
PERFORM MAJ
PERFORM RECONDUCTION
END-IF
PERFORM READ-MVT
END-PERFORM

```

```

* FERMETURE DES FICHIERS, AFFICHAGE DES COMPTEURS
  PERFORM FIN
.

DEBUT.
* OUVERTURE MVT EN LECTURE
  OPEN INPUT MVT
  IF FS-MVT NOT = ZEROES
    DISPLAY 'ERREUR OPEN MVT, FS : ' FS-MVT
    PERFORM FIN-ERREUR-GRAVE
  END-IF

* OUVERTURE MVTVAL EN ECRITURE
  OPEN OUTPUT MVTVAL
  IF FS-MVTVAL NOT = ZEROES
    DISPLAY 'ERREUR OPEN MVTVAL, FS : ' FS-MVTVAL
    PERFORM FIN-ERREUR-GRAVE
  END-IF

* 1ERE LECTURE MVT, TEST FICHER VIDE
  PERFORM READ-MVT
  IF FIN-MVT
    DISPLAY 'ERREUR FICHER MVT VIDE'
    PERFORM FIN-ERREUR-GRAVE
  END-IF
.

* CONTROLE DU MVT
CONTROLES.
  DISPLAY SPACES
  DISPLAY 'ENR ' CPT-MVT-LUS ' : ' MVT-ENR

* INITIALISATION INDICATEUR ERREUR
  SET OK TO TRUE

* CONTROLE CODE MOUVEMENT
  IF NOT (MVT-CODE-MVT = 'C' OR 'M' OR 'S')
*   IF MVT-CODE-MVT NOT = 'C' AND 'M' AND 'S'
* --> CODE MOUVEMENT ERRONE
    DISPLAY POSTE-ERR(1)
    SET ERREUR TO TRUE
  END-IF

* CONTROLE CODE TABLE
  IF NOT (MVT-CODE-TAB = 'D' OR 'E')
*   IF MVT-CODE-TAB NOT = 'D' AND 'E'
* --> CODE TABLE ERRONE
    DISPLAY POSTE-ERR(2)
    SET ERREUR TO TRUE
  END-IF

* SI PAS D'ERREUR ON TESTE LES CHAMPS SUIVANTS EN FONCTION
* DU CODE TABLE (EMPLOYE OU DEPARTEMENT) ET DU CODE MOUVEMENT
* (CREATION, MODIFICATION, SUPPRESSION)
  IF OK
    EVALUATE MVT-CODE-MVT ALSO MVT-CODE-TAB
      WHEN      'C'          ALSO 'E'
        PERFORM CTRL-MAT-INEX
        PERFORM CTRL-NOM
        PERFORM CTRL-NOD1
        PERFORM CTRL-DAT
        PERFORM CTRL-SAL
        PERFORM CTRL-COM
      WHEN      'M'          ALSO 'E'
        PERFORM CTRL-MAT-EXIST

```



```

        PERFORM CTRL-NOM
        PERFORM CTRL-NOD1
        PERFORM CTRL-DAT
        PERFORM CTRL-SAL
        PERFORM CTRL-COM
    WHEN      'S'                ALSO 'E'
        PERFORM CTRL-MAT-EXIST
        PERFORM CTRL-CIF-DEPARTEMENT
    WHEN      'C'                ALSO 'D'
        PERFORM CTRL-NOD2-INEX
        PERFORM CTRL-NDE
        PERFORM CTRL-LIE
        PERFORM CTRL-CHE
    WHEN      'M'                ALSO 'D'
        PERFORM CTRL-NOD2-EXIST
        PERFORM CTRL-NDE
        PERFORM CTRL-LIE
        PERFORM CTRL-CHE
    WHEN      'S'                ALSO 'D'
        PERFORM CTRL-NOD2-EXIST
        PERFORM CTRL-CIF-EMPLOYE
    END-EVALUATE
END-IF

* INCREMENTATION COMPTEURS MVT OK, MVT ERREUR
    IF OK
        ADD 1 TO CPT-MVT-VAL
    ELSE
        ADD 1 TO CPT-MVT-ERR
    END-IF
.

* CONTROLE MATRICULE INEXISTANT
CTRL-MAT-INEX.
    PERFORM SELECT-EMPLOYE
    IF EMPLOYE-EXISTANT
* ---> MATRICULE DEJA EXISTANT
        DISPLAY POSTE-ERR(4)
        SET ERREUR TO TRUE
    END-IF
.

* CONTROLE MATRICULE EXISTANT
CTRL-MAT-EXIST.
    PERFORM SELECT-EMPLOYE
    IF EMPLOYE-INEXISTANT
* ---> MATRICULE INEXISTANT
        DISPLAY POSTE-ERR(11)
        SET ERREUR TO TRUE
    END-IF
.

SELECT-EMPLOYE.
    IF MVT-MAT = SPACES
* ---> MATRICULE NON RENSEIGNE
        DISPLAY POSTE-ERR(3)
        SET ERREUR TO TRUE
    ELSE
        MOVE MVT-MAT TO HVE-MAT
        EXEC SQL
            SELECT MAT
            INTO :HVE-MAT
            FROM EMPLOYE
            WHERE MAT = :HVE-MAT
        END-EXEC
        EVALUATE SQLCODE

```

```

        WHEN ZEROES
            SET EMPLOYE-EXISTANT TO TRUE
        WHEN +100
            SET EMPLOYE-INEXISTANT TO TRUE
        WHEN OTHER
            MOVE SQLCODE TO ED-SQLCODE
            DISPLAY 'ERREUR SELECT EMPLOYE, SQLCODE : ' ED-SQLCODE
            PERFORM FIN-ERREUR-GRAVE
        END-EVALUATE
    END-IF
.

* CONTROLE D'INTEGRITE FONCTIONNEL
CTRL-CIF-DEPARTEMENT.
*     L'EMPLOYE NE DOIT PAS ETRE CHEF DE DEPARTEMENT
    MOVE MVT-MAT TO HVD-CHE

    EXEC SQL
        SELECT NOD
        INTO :HVD-NOD
        FROM DEPARTEMENT
        WHERE CHE = :HVD-CHE
    END-EXEC
    EVALUATE SQLCODE
        WHEN ZEROES
* SI L'EMPLOYE EST CHEF DE PLUSIEURS DEPARTEMENTS, ALORS DB2
* RETOURNE PLUSIEURS LIGNES, MAIS COMME IL NE PEUT PAS IL MET
* UN SQLCODE A -811
        WHEN -811
* ---> SUPPRESSION EMPLOYE IMPOSSIBLE, CONTRAINTE SUR DEP
            DISPLAY POSTE-ERR(17)
            SET ERREUR TO TRUE
            WHEN +100
                CONTINUE
            WHEN OTHER
                MOVE SQLCODE TO ED-SQLCODE
                DISPLAY 'ERREUR SELECT DEPART, SQLCODE : ' ED-SQLCODE
                PERFORM FIN-ERREUR-GRAVE
            END-EVALUATE
        .

* CONTROLE NOM EMPLOYE
CTRL-NOM.
    IF MVT-NOM = SPACES
* ---> NOM NON RENSEIGNE
        DISPLAY POSTE-ERR(5)
        SET ERREUR TO TRUE
    END-IF
.

* CONTROLE NUMERO DEPARTEMENT
CTRL-NOD1.
    IF MVT-NOD1 = SPACES
* ---> DEPARTEMENT NON RENSEIGNE
        DISPLAY POSTE-ERR(6)
        SET ERREUR TO TRUE
    ELSE
        MOVE MVT-NOD1 TO HVD-NOD
        EXEC SQL
            SELECT NOD
            INTO :HVD-NOD
            FROM DEPARTEMENT
            WHERE NOD = :HVD-NOD
        END-EXEC
        EVALUATE SQLCODE
            WHEN ZEROES

```

```

        CONTINUE
        WHEN +100
* ----> DEPARTEMENT INEXISTANT
        DISPLAY POSTE-ERR(7)
        SET ERREUR TO TRUE
        WHEN OTHER
        MOVE SQLCODE TO ED-SQLCODE
        DISPLAY 'ERREUR SELECT DEPART, SQLCODE : ' ED-SQLCODE
        PERFORM FIN-ERREUR-GRAVE
        END-EVALUATE
    END-IF
.

* CONTROLE DATE ENTREE
CTRL-DAT.
    IF MVT-DAT NOT NUMERIC
    OR MVT-DAT = ZEROES
* ----> DATE ENTREE ERRONEE
        DISPLAY POSTE-ERR(8)
        SET ERREUR TO TRUE
    ELSE
        MOVE MVT-DAT TO ZDATE
        CALL SPDAT USING ZCOM
        IF ZCDRET-KO
* ----> DATE ENTREE ERRONEE
            DISPLAY POSTE-ERR(8)
            SET ERREUR TO TRUE
        END-IF
    END-IF
.

* CONTROLE SALAIRE
CTRL-SAL.
    IF MVT-SAL NOT NUMERIC
    OR MVT-SAL = ZEROES
* ----> SALAIRE NON NUMERIQUE OU NUL
        DISPLAY POSTE-ERR(9)
        SET ERREUR TO TRUE
    END-IF
.

* CONTROLE COMMISSION
CTRL-COM.
    IF MVT-COM NOT NUMERIC
* ----> COMMISSION NON NUMERIQUE
        DISPLAY POSTE-ERR(10)
        SET ERREUR TO TRUE
    END-IF
.

* CONTROLE NUMERO DEPARTEMENT DOIT ETRE INEXISTANT
CTRL-NOD2-INEX.
    IF MVT-NOD2 = SPACES
* ----> NUMERO DEPARTEMENT DOIT ETRE RENSEIGNE
        DISPLAY POSTE-ERR(06)
        SET ERREUR TO TRUE
    ELSE
* ----> NE DOIT PAS EXISTER EN TABLE DEPARTEMENT
        MOVE MVT-NOD2 TO HVD-NOD
        EXEC SQL
            SELECT NOD
            INTO :HVD-NOD
            FROM DEPARTEMENT
            WHERE NOD = :HVD-NOD
        END-EXEC
        EVALUATE SQLCODE

```

```

        WHEN ZEROES
* ----> DEPARTEMENT EXISTANT : ERREUR
        DISPLAY POSTE-ERR(12)
        SET ERREUR TO TRUE
        WHEN +100
* ----> DEPARTEMENT INEXISTANT : OK
        CONTINUE
        WHEN OTHER
        MOVE SQLCODE TO ED-SQLCODE
        DISPLAY 'ERREUR SELECT DEPART, SQLCODE : ' ED-SQLCODE
        PERFORM FIN-ERREUR-GRAVE
        END-EVALUATE
    END-IF
.

* CONTROLE NUMERO DEPARTEMENT DOIT EXISTER
CTRL-NOD2-EXIST.
    IF MVT-NOD2 = SPACES
* ----> NUMERO DEPARTEMENT DOIT ETRE RENSEIGNE
        DISPLAY POSTE-ERR(06)
        SET ERREUR TO TRUE
    ELSE
* ----> DOIT EXISTER EN TABLE DEPARTEMENT
        MOVE MVT-NOD2 TO HVD-NOD
        EXEC SQL
            SELECT NOD
            INTO :HVD-NOD
            FROM DEPARTEMENT
            WHERE NOD = :HVD-NOD
        END-EXEC
        EVALUATE SQLCODE
        WHEN ZEROES
* ----> DEPARTEMENT EXISTANT : OK
        CONTINUE
        WHEN +100
* ----> DEPARTEMENT INEXISTANT : ERREUR
        DISPLAY POSTE-ERR(07)
        SET ERREUR TO TRUE
        WHEN OTHER
        MOVE SQLCODE TO ED-SQLCODE
        DISPLAY 'ERREUR SELECT DEPART, SQLCODE : ' ED-SQLCODE
        PERFORM FIN-ERREUR-GRAVE
        END-EVALUATE
    END-IF
.

* CONTROLE NOM DEPARTEMENT
CTRL-NDE.
    IF MVT-NDE = SPACES
* ----> NOM DEPARTEMENT DOIT ETRE RENSEIGNE
        DISPLAY POSTE-ERR(13)
        SET ERREUR TO TRUE
    END-IF
.

* CONTROLE BATIMENT
CTRL-LIE.
    IF MVT-LIE = SPACES
* ----> BATIMENT DOIT ETRE RENSEIGNE
        DISPLAY POSTE-ERR(14)
        SET ERREUR TO TRUE
    END-IF
.

* CONTROLE MATRICULE DU CHEF
CTRL-CHE.

```

```

        IF MVT-CHE = SPACES
* ---> MATRICULE DU CHEF DOIT ETRE RENSEIGNE
        DISPLAY POSTE-ERR(15)
        SET ERREUR TO TRUE
    ELSE
        MOVE MVT-CHE TO HVE-MAT
        EXEC SQL
            SELECT MAT
            INTO :HVE-MAT
            FROM EMPLOYE
            WHERE MAT = :HVE-MAT
        END-EXEC
        EVALUATE SQLCODE
            WHEN ZEROES
                CONTINUE
            WHEN +100
* ---> MATRICULE DU CHEF INEXISTANT EN TABLE EMPLOYE
                DISPLAY POSTE-ERR(16)
                SET ERREUR TO TRUE
                WHEN OTHER
                    MOVE SQLCODE TO ED-SQLCODE
                    DISPLAY 'ERREUR SELECT EMPLOYE, SQLCODE : ' ED-SQLCODE
                    PERFORM FIN-ERREUR-GRAVE
                END-EVALUATE
        END-IF
    .

* CONTROLE DEPARTEMENT DANS TABLE EMPLOYE
CTRL-CIF-EMPLOYE.
    MOVE MVT-NOD2 TO HVE-NOD

    EXEC SQL
        SELECT NOD INTO :HVE-NOD
        FROM EMPLOYE
        WHERE NOD = :HVE-NOD
    END-EXEC

    EVALUATE SQLCODE
        WHEN ZEROES
            WHEN -811
* ---> SUPPRESSION IMPOSSIBLE: DEPARTEMENT EXISTE EN TABLE EMPLOYE
                DISPLAY POSTE-ERR(18)
                SET ERREUR TO TRUE
                WHEN +100
                    CONTINUE
                WHEN OTHER
                    MOVE SQLCODE TO ED-SQLCODE
                    DISPLAY 'ERREUR SELECT EMPLOYE, SQLCODE : ' ED-SQLCODE
                    PERFORM FIN-ERREUR-GRAVE
                END-EVALUATE
    .

*****
*****
* MISE A JOUR DES TABLES
*****
*****
MAJ.
    DISPLAY 'ICI ON MET A JOUR LA TABLE'
    EVALUATE MVT-CODE-MVT ALSO MVT-CODE-TAB
* CREATION D'UN EMPLOYE
    WHEN      'C'              ALSO 'E'
        PERFORM CREATION-EMPLOYE
* MODIFICATION EMPLOYE
    WHEN      'M'              ALSO 'E'
        PERFORM MODIFICATION-EMPLOYE

```

```

* SUPPRESSION EMPLOYE
    WHEN      'S'              ALSO 'E'
        PERFORM SUPPRESSION-EMPLOYE
* CREATION DEPARTEMENT
    WHEN      'C'              ALSO 'D'
        PERFORM CREATION-DEPARTEMENT
* MODIFICATION DEPARTEMENT
    WHEN      'M'              ALSO 'D'
        PERFORM MODIFICATION-DEPARTEMENT
* SUPPRESSION DEPARTEMENT
    WHEN      'S'              ALSO 'D'
        PERFORM SUPPRESSION-DEPARTEMENT
END-EVALUATE
.

CREATION-EMPLOYE.
* ALIMENTATION DES COLONNES DE LA TABLE PAR LES CHAMPS DE L'ENREG
    PERFORM ALIM-EMPLOYE

* INSERT DE L'EMPLOYE
    EXEC SQL
        INSERT INTO EMPLOYE
        VALUES
            (:HVE-MAT,
            :HVE-NOM,
            :HVE-NOD,
            :HVE-DAT,
            :HVE-SAL,
            :HVE-COM :IND-COM)
    END-EXEC

    IF SQLCODE = ZEROES
        ADD 1 TO CPT-EMP-CRE
        DISPLAY '              CREATION EMPLOYE'
    ELSE
        MOVE SQLCODE TO ED-SQLCODE
        DISPLAY 'ERREUR INSERT EMPLOYE, SQLCODE : ' ED-SQLCODE
        PERFORM FIN-ERREUR-GRAVE
    END-IF
.

MODIFICATION-EMPLOYE.
* ALIMENTATION DES COLONNES DE LA TABLE PAR LES CHAMPS DE L'ENREG
    PERFORM ALIM-EMPLOYE

* INSERT DE L'EMPLOYE
    EXEC SQL
        UPDATE EMPLOYE
        SET
            NOM = :HVE-NOM,
            NOD = :HVE-NOD,
            DAT = :HVE-DAT,
            SAL = :HVE-SAL,
            COM = :HVE-COM :IND-COM
        WHERE
            MAT = :HVE-MAT
    END-EXEC

    IF SQLCODE = ZEROES
        ADD 1 TO CPT-EMP-MAJ
        DISPLAY '              MODIFICATION EMPLOYE'
    ELSE
        MOVE SQLCODE TO ED-SQLCODE
        DISPLAY 'ERREUR UPDATE EMPLOYE, SQLCODE : ' ED-SQLCODE
        PERFORM FIN-ERREUR-GRAVE
    END-IF

```

```

.

SUPPRESSION-EMPLOYE.
    MOVE MVT-MAT TO HVE-MAT
* DELETE DE L'EMPLOYE
    EXEC SQL
        DELETE FROM EMPLOYE
        WHERE
            MAT = :HVE-MAT
    END-EXEC

    IF SQLCODE = ZEROES
        ADD 1 TO CPT-EMP-SUP
        DISPLAY '                SUPPRESSION EMPLOYE '
    ELSE
        MOVE SQLCODE TO ED-SQLCODE
        DISPLAY 'ERREUR DELETE EMPLOYE, SQLCODE : ' ED-SQLCODE
        PERFORM FIN-ERREUR-GRAVE
    END-IF
.

ALIM-EMPLOYE.
* ALIMENTATION DES COLONNES DE LA TABLE PAR LES CHAMPS DE L'ENREG
* CE PARAGRAPHE EST APPELE A PLUSIEURS ENDROITS
    MOVE MVT-MAT      TO HVE-MAT
    MOVE MVT-NOM      TO HVE-NOM
    MOVE MVT-NOD1     TO HVE-NOD
* ALIMENTATION DE LA DATE AU FORMAT AAAA-MM-JJ
* SINON SQLCODE -180 OU -181
    STRING MVT-DAT(1 : 4) DELIMITED BY SIZE
        '-' DELIMITED BY SIZE
        MVT-DAT(5 : 2) DELIMITED BY SIZE
        '-' DELIMITED BY SIZE
        MVT-DAT(7 : 2) DELIMITED BY SIZE
        INTO HVE-DAT
    END-STRING
    MOVE MVT-SAL      TO HVE-SAL
    MOVE MVT-COM      TO HVE-COM

* POSITIONNEMENT DE L'INDICATEUR DE NULLITE
* --> -1 INDIQUE A DB2 DE METTRE LA COLONNE A NULL
* --> 0 INDIQUE A DB2 DE COPIER LA HOST-VARIABLE DANS LA COLONNE
    IF HVE-COM = ZEROES
        MOVE -1      TO IND-COM
    ELSE
        MOVE ZEROES TO IND-COM
    END-IF
.

CREATION-DEPARTEMENT.
* ALIMENTATION DES COLONNES DE LA TABLE PAR LES CHAMPS DE L'ENREG
    PERFORM ALIM-DEPARTEMENT

* INSERT DU DEPARTEMENT
    EXEC SQL
        INSERT INTO DEPARTEMENT
        VALUES
            (:HVD-NOD,
            :HVD-NDE,
            :HVD-LIE,
            :HVD-CHE)
    END-EXEC

    IF SQLCODE = ZEROES
        ADD 1 TO CPT-DEP-CRE
        DISPLAY '                CREATION DEPARTEMENT '

```

```

ELSE
    MOVE SQLCODE TO ED-SQLCODE
    DISPLAY 'ERREUR INSERT DEPARTE, SQLCODE : ' ED-SQLCODE
    PERFORM FIN-ERREUR-GRAVE
END-IF
.

MODIFICATION-DEPARTEMENT.
* ALIMENTATION DES COLONNES DE LA TABLE PAR LES CHAMPS DE L'ENREG
PERFORM ALIM-DEPARTEMENT

* UPDATE DU DEPARTEMENT
EXEC SQL
    UPDATE DEPARTEMENT
    SET
        NDE = :HVD-NDE,
        LIE = :HVD-LIE,
        CHE = :HVD-CHE
    WHERE
        NOD = :HVD-NOD
END-EXEC

IF SQLCODE = ZEROES
    ADD 1 TO CPT-DEP-MAJ
    DISPLAY '                MODIFICATION DEPARTEMENT'
ELSE
    MOVE SQLCODE TO ED-SQLCODE
    DISPLAY 'ERREUR INSERT DEPARTE, SQLCODE : ' ED-SQLCODE
    PERFORM FIN-ERREUR-GRAVE
END-IF
.

SUPPRESSION-DEPARTEMENT.
MOVE MVT-NOD2 TO HVD-NOD

* DELETE DU DEPARTEMENT
EXEC SQL
    DELETE FROM DEPARTEMENT
    WHERE
        NOD = :HVD-NOD
END-EXEC

IF SQLCODE = ZEROES
    ADD 1 TO CPT-DEP-SUP
    DISPLAY '                SUPPRESSION DEPARTEMENT'
ELSE
    MOVE SQLCODE TO ED-SQLCODE
    DISPLAY 'ERREUR INSERT DEPARTE, SQLCODE : ' ED-SQLCODE
    PERFORM FIN-ERREUR-GRAVE
END-IF
.

ALIM-DEPARTEMENT.
* ALIMENTATION DES COLONNES DE LA TABLE PAR LES CHAMPS DE L'ENREG
* CE PARAGRAPHE EST APPELE A PLUSIEURS ENDROITS
MOVE MVT-NOD2    TO HVD-NOD
MOVE MVT-NDE     TO HVD-NDE
MOVE MVT-LIE     TO HVD-LIE
MOVE MVT-CHE     TO HVD-CHE
.

* RECONDUCTION DU MVT VALIDE DANS MVTVAL
RECONDUCTION.
DISPLAY 'ICI ON RECONDUIT LE MVT DANS LE FICHIER DES MVTS OK'
WRITE ENR-MVTVAL FROM ENR-MVT
IF FS-MVTVAL NOT = ZEROES

```



```

        DISPLAY 'ERREUR WRITE MVTVAL, FS : ' FS-MVTVAL
        PERFORM FIN-ERREUR-GRAVE
    END-IF
.

READ-MVT.
* LECTURE MVT
    READ MVT INTO MVT-ENR
    AT END          SET FIN-MVT TO TRUE
    NOT AT END      ADD 1 TO CPT-MVT-LUS
    END-READ
    IF FS-MVT NOT = ZEROES AND 10
*   IF NOT (FS-MVT = ZEROES OR 10)
        DISPLAY 'ERREUR READ MVT, FS : ' FS-MVT
        PERFORM FIN-ERREUR-GRAVE
    END-IF
.

FIN-ERREUR-GRAVE.
* PLUSIEURS POSSIBILITES POUR ARRETER UN PROGRAMME (DEPEND
* DES NORMES DE L'ENTREPRISE)
* --> POSITIONNER UNE VALEUR DANS LE RETURN CODE (MAXCC)
* --> PLANTER LE PROGRAMME EN APPELANT UN SOUS-PROGRAMME
* --> INEXISTANT (ABEND S806)
* --> APPELER UN SOUS-PROGRAMME QUI RECUPERE LES INFORMATIONS ET
* --> FAIT PLANTER LE STEP
    MOVE 15 TO RETURN-CODE
    PERFORM FIN
.

FIN.
    CLOSE MVT MVTVAL

* AFFICHAGE DES COMPTEURS
    DISPLAY 'NB MVT LUS : ' CPT-MVT-LUS
    DISPLAY 'NB MVT ERR : ' CPT-MVT-ERR
    DISPLAY 'NB MVT VAL : ' CPT-MVT-VAL
    DISPLAY 'NB EMP CRE : ' CPT-EMP-CRE
    DISPLAY 'NB EMP MAJ : ' CPT-EMP-MAJ
    DISPLAY 'NB EMP SUP : ' CPT-EMP-SUP
    DISPLAY 'NB DEP CRE : ' CPT-DEP-CRE
    DISPLAY 'NB DEP MAJ : ' CPT-DEP-MAJ
    DISPLAY 'NB DEP SUP : ' CPT-DEP-SUP

    STOP RUN
.

```

JCL DB2P4JCL

```
//API8DB JOB NOTIFY=&SYSUID,CLASS=A,MSGCLASS=H,TIME=(,5)
//*
//PROCLIB JCLLIB ORDER=SDJ.FORM.PROCLIB
//*
//          SET SYSUID=API8,
//          NOMPGM=DB2P4
//*
//APPROC   EXEC COMPDB2
//STEPDB2.SYSLIB DD DSN=API8.SOURCE.DCLGEN,DISP=SHR
//STEPDB2.SYSIN DD DSN=API8.SOURCE.DB2(DB2P4),DISP=SHR
//STEPDB2.DBRMLIB DD DSN=API8.SOURCE.DBRMLIB(DB2P4),DISP=SHR
//STEPCOB.SYSLIB DD DSN=API8.SOURCE.COPY,DISP=SHR
//STEPLNK.SYSLMOD DD DSN=API8.SOURCE.PGMLIB(DB2P4),DISP=SHR
//*
/*----- ETAPE DE BIND -----
/*
//BIND      EXEC PGM=IKJEFT01,COND=(4,LT)
//DBRMLIB DD DSN=API8.SOURCE.DBRMLIB,DISP=SHR
//SYSTSPRT DD SYSOUT=*,OUTLIM=25000
//SYSTSIN DD *
    DSN SYSTEM (DSN1)
    BIND PLAN (DB2P4) -
    QUALIFIER (API8) -
    ACTION (REPLACE) -
    MEMBER (DB2P4) -
    VALIDATE (BIND) -
    ISOLATION (CS) -
    ACQUIRE (USE) -
    RELEASE (COMMIT) -
    EXPLAIN (NO)
/*
/* SUPPRESSION FICHIER DES MVTs VALIDES
//STEP1     EXEC PGM=IDCAMS,COND=(4,LT)
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
    DELETE API8.COB.MVTVAL.BASE.EMPLOYE
    SET MAXCC = 0
/*
/* EXECUTION DU PROGRAMME
//STEPRUN   EXEC PGM=IKJEFT01,COND=(4,LT)
//STEPLIB DD DSN=API8.SOURCE.PGMLIB,DISP=SHR
//          DD DSN=API0.COBOL.LOAD,DISP=SHR
//MVT DD DSN=API0.COB.MVT.BASE.EMPLOYE,DISP=SHR
//MVTVAL DD DSN=API8.COB.MVTVAL.BASE.EMPLOYE,DISP=(,CATLG,DELETE),
//          LIKE=API0.COB.MVT.BASE.EMPLOYE
//SYSOUT DD SYSOUT=*,OUTLIM=1000
//SYSTSPRT DD SYSOUT=*,OUTLIM=2500
//SYSTSIN DD *
    DSN SYSTEM (DSN1)
    RUN PROGRAM(DB2P4) PLAN (DB2P4) PARMS('11001')
//
```

PROGRAMME DB2P5

CBL DYNAM

```
IDENTIFICATION DIVISION.
PROGRAM-ID. DB2P5.
*****
* BUT : EDITION DES MOUVEMENTS VALIDES *
*****
AUTHOR. GIUSEPPE.
DATE-WRITTEN. 12/05/23.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.
    DECIMAL-POINT IS COMMA.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT MVTVAL      ASSIGN MVTVAL
        ORGANIZATION IS SEQUENTIAL
        ACCESS MODE   IS SEQUENTIAL
        FILE STATUS   IS FS-MVTVAL.
    SELECT EDIT        ASSIGN EDIT
        ORGANIZATION IS SEQUENTIAL
        ACCESS MODE   IS SEQUENTIAL
        FILE STATUS   IS FS-EDIT.
DATA DIVISION.
FILE SECTION.
* FICHER DES MVTs VALIDES
FD MVTVAL
    RECORDING MODE IS F.
01 ENR-MVTVAL          PIC X(50) .

* FICHER D'EDITION
FD EDIT
    RECORDING MODE IS F.
01 ENR-EDIT            PIC X(132) .

WORKING-STORAGE SECTION.
* FILE STATUS
01 FS-MVTVAL            PIC 99 VALUE ZEROES.
01 FS-EDIT              PIC 99 VALUE ZEROES.

* TABLES DB2
* SQLCA
    EXEC SQL INCLUDE SQLCA      END-EXEC.
* TABLE EMPLOYE
    EXEC SQL INCLUDE EMPLOYE    END-EXEC.
* TABLE DEPARTEMENT
    EXEC SQL INCLUDE DEPART     END-EXEC.

* DECLARATION DES HOST VARIABLES DE LA TABLE EMPLOYE
01 HVE-EMP.
    05 HVE-MAT            PIC X(3) .
    05 HVE-NOD            PIC X(3) .
    05 HVE-NOM            PIC X(7) .
    05 HVE-SAL            PIC S9(5)V99 COMP-3.
    05 HVE-COM            PIC S9(5)V99 COMP-3.
    05 HVE-DAT            PIC X(10) .

* DECLARATION DES HOST VARIABLES DE LA TABLE DEPARTEMENT
01 HVD-DEP.
    05 HVD-NOD            PIC X(3) .
    05 HVD-LIE            PIC X(4) .
    05 HVD-CHE            PIC X(3) .
    05 HVD-NDE            PIC X(6) .

* SQLCODE EN TENUE D'APPARAT
```

```

01  SQLCODE-ED          PIC +(10) VALUE ZEROES.

* DESCRIPTION ENREGISTREMENT DES MVTs VALIDES
01  MVT-ENR.
* CODE MVT : C, M, S
    05 MVT-CODE-MVT      PIC X.
* CODE TAB : D, E
    05 MVT-CODE-TAB      PIC X.
* SUITE
    05 MVT-SUITE          PIC X(48) .
*
    05 MVT-EMPLOYE        REDEFINES MVT-SUITE.
        10 MVT-MAT        PIC X(3) .
        10 MVT-NOM        PIC X(7) .
        10 MVT-NOD1       PIC X(3) .
        10 MVT-DAT        PIC 9(8) .
        10 MVT-SAL        PIC 9(5)V99.
        10 MVT-COM        PIC 9(5)V99.
        10                PIC X(13) .
*
    05 MVT-DEPARTEMENT    REDEFINES MVT-SUITE.
        10 MVT-NOD2       PIC X(3) .
        10 MVT-NDE        PIC X(6) .
        10 MVT-LIE        PIC X(4) .
        10 MVT-CHE        PIC X(3) .
        10                PIC X(32) .

* COMPTEURS
01  CPT-MVT-LUS          PIC 99 VALUE ZEROES.

* LIGNES D'EDITION
01  LIG1.
    05                  PIC X(20) VALUE SPACES.
    05                  PIC X(18) VALUE
        'MOUVEMENTS DU JOUR'.
    05                  PIC X(10) VALUE SPACES.
    05                  PIC X(05) VALUE 'LE : '.
    05 DATE-JOUR-ED      PIC X(17) VALUE SPACES.

01  LIG2.
    05                  PIC X(20) VALUE SPACES.
    05                  PIC X(18) VALUE ALL '-'.

* LIGNES D'EDITION POUR LA TABLE EMPLOYE
01  LIG3E.
    05                  PIC X(13) VALUE 'TABLE EMPLOYE'.

01  LIG4E.
    05                  PIC X(13) VALUE ALL '-'.

01  LIG5E.
    05                  PIC X(85) VALUE ALL '-'.

01  LIG6E.
    05                  PIC X(14) VALUE '| ACTION |'.
    05                  PIC X(06) VALUE ' MAT |'.
    05                  PIC X(10) VALUE ' NOM EMP |'.
    05                  PIC X(06) VALUE ' NOD |'.
    05                  PIC X(11) VALUE ' NOM DEPT |'.
    05                  PIC X(13) VALUE ' DATE ENT |'.
    05                  PIC X(12) VALUE ' SALAIRE |'.
    05                  PIC X(13) VALUE ' COMMISSION |'.

01  LIG7E.
    05                  PIC X(01) VALUE '|'.
    05 ACTIONE-ED        PIC X(12) VALUE SPACES.

```

```

05          PIC X(02) VALUE ' | '.
05 MATE-ED   PIC X(03) VALUE SPACES.
05          PIC X(03) VALUE ' | '.
05 NOM-EMPE-ED PIC X(07) VALUE SPACES.
05          PIC X(03) VALUE ' | '.
05 NODE-ED   PIC X(03) VALUE SPACES.
05          PIC X(05) VALUE ' | '.
05 NDEE-ED   PIC X(07) VALUE SPACES.
05          PIC X(02) VALUE ' | '.
05 DATE-ED   PIC X(10) VALUE SPACES.
05          PIC X(01) VALUE SPACES.
05          PIC X(02) VALUE ' | '.
05 SALE-ED   PIC ZZBZZ9,99 VALUE ZEROES.
05          PIC X(01) VALUE SPACES.
05          PIC X(02) VALUE ' | '.
05 COME-ED   PIC ZZBZZ9,99 VALUE ZEROES.
05          PIC X(03) VALUE ' | '.

* LIGNES D'EDITION POUR LA TABLE DEPARTEMENT
01 LIG3D.
05          PIC X(17) VALUE 'TABLE DEPARTEMENT'.

01 LIG4D.
05          PIC X(17) VALUE ALL '-'.

01 LIG5D.
05          PIC X(60) VALUE ALL '-'.

01 LIG6D.
05          PIC X(14) VALUE ' | ACTION | '.
05          PIC X(06) VALUE ' NOD | '.
05          PIC X(10) VALUE ' NOM DEPT | '.
05          PIC X(06) VALUE ' LIEU | '.
05          PIC X(11) VALUE ' MAT CHEF | '.
05          PIC X(13) VALUE ' NOM CHEF | '.

01 LIG7D.
05          PIC X(01) VALUE ' | '.
05 ACTIOND-ED PIC X(12) VALUE SPACES.
05          PIC X(02) VALUE ' | '.
05 NODD-ED    PIC X(03) VALUE SPACES.
05          PIC X(04) VALUE ' | '.
05 NOM-DEPD-ED PIC X(07) VALUE SPACES.
05          PIC X(02) VALUE ' | '.
05 LIED-ED    PIC X(04) VALUE SPACES.
05          PIC X(05) VALUE ' | '.
05 CHED-ED    PIC X(03) VALUE SPACES.
05          PIC X(03) VALUE SPACES.
05          PIC X(05) VALUE ' | '.
05 NOM-CHEFD-ED PIC X(07) VALUE SPACES.
05          PIC X(02) VALUE ' | '.

* DESCRIPTION DE LA ZONE DE COMMUNICATION AVEC LE SOUS-PROGRAMME
* DE CONTROLE DE DATE
01 ZCOM.
05 ZDATE.
05 10 ZANNEE PIC 9(4).
05 10 ZMOIS  PIC 9(2).
05 10 ZJOUR  PIC 9(2).
05 ZCDRET PIC 9.
05 88 ZCDRET-OK VALUE 0.
05 88 ZCDRET-KO VALUE 1.
05 ZDATE-EDIT PIC X(17).
01 SPDAT PIC X(8) VALUE 'SPDAT'.

* INDICATEUR DE FIN DE FICHIER

```

```

01          PIC X      VALUE SPACES.
      88 FIN-MVTVAL      VALUE 'C'.

* ZONE RESERVE POUR DETECTER LA RUPTURE SUR LE CODE TABLE
01 RES-CODE-TAB      PIC X      VALUE SPACES.

PROCEDURE DIVISION.
  PERFORM DEBUT
  PERFORM UNTIL FIN-MVTVAL
    PERFORM TRT-ENTETE-TABLE
    PERFORM UNTIL MVT-CODE-TAB NOT = RES-CODE-TAB
      OR FIN-MVTVAL
      PERFORM TRAITEMENT
      PERFORM READ-MVTVAL
    END-PERFORM
  END-PERFORM
  PERFORM FIN
.

DEBUT.
* OUVERTURE DES FICHIERS POUR PAS CHANGER
  OPEN INPUT  MVTVAL
  IF FS-MVTVAL NOT = ZEROES
    DISPLAY 'PB OPEN MVTVAL, FS : ' FS-MVTVAL
    PERFORM FIN-ERREUR-GRAVE
  END-IF

  OPEN OUTPUT EDIT
  IF FS-EDIT  NOT = ZEROES
    DISPLAY 'PB OPEN EDIT, FS : ' FS-EDIT
    PERFORM FIN-ERREUR-GRAVE
  END-IF

* 1ERE LECTURE MVTVAL
  PERFORM READ-MVTVAL
  IF FIN-MVTVAL
    DISPLAY 'FICHIER MVTVAL VIDE'
    PERFORM FIN-ERREUR-GRAVE
  END-IF

* EDITION DE L'ENTETE
  MOVE FUNCTION CURRENT-DATE TO ZDATE
  CALL SPDAT USING ZCOM
  IF ZCDRET-OK
    MOVE ZDATE-EDIT TO DATE-JOUR-ED
  END-IF

* EDITION DU TITRE
  MOVE LIG1          TO ENR-EDIT
  WRITE ENR-EDIT AFTER ADVANCING PAGE
  MOVE LIG2          TO ENR-EDIT
  WRITE ENR-EDIT
.

TRT-ENTETE-TABLE.
  EVALUATE MVT-CODE-TAB
    WHEN 'E'

* EDITION DE L'ENTETE POUR LA TABLE EMPLOYE
  MOVE LIG3E          TO ENR-EDIT
  WRITE ENR-EDIT AFTER ADVANCING 1 LINE
  MOVE LIG4E          TO ENR-EDIT
  WRITE ENR-EDIT
  MOVE LIG5E          TO ENR-EDIT
  WRITE ENR-EDIT
  MOVE LIG6E          TO ENR-EDIT
  WRITE ENR-EDIT

```

```

        MOVE LIG5E          TO ENR-EDIT
        WRITE ENR-EDIT
    WHEN 'D'
* EDITION DE L'ENTETE POUR LA TABLE DEPARTEMENT
        MOVE LIG3D          TO ENR-EDIT
        WRITE ENR-EDIT AFTER ADVANCING 1 LINE
        MOVE LIG4D          TO ENR-EDIT
        WRITE ENR-EDIT
        MOVE LIG5D          TO ENR-EDIT
        WRITE ENR-EDIT
        MOVE LIG6D          TO ENR-EDIT
        WRITE ENR-EDIT
        MOVE LIG5D          TO ENR-EDIT
        WRITE ENR-EDIT
    END-EVALUATE

* ALIMENTER LA ZONE RESERVE DE RUPTURE
    MOVE MVT-CODE-TAB TO RES-CODE-TAB
.

TRAITEMENT.
* EDITION DE LA LIGNE DETAIL
    EVALUATE MVT-CODE-TAB
        WHEN 'E' PERFORM TRT-EMPLOYE
        WHEN 'D' PERFORM TRT-DEPARTEMENT
    END-EVALUATE
.

TRT-EMPLOYE.
* LIGNE DETAIL DE L'EMPLOYE
    EVALUATE MVT-CODE-MVT
        WHEN 'C' MOVE 'CREATION'          TO ACTIONE-ED
        WHEN 'M' MOVE 'MODIFICATION' TO ACTIONE-ED
        WHEN 'S' MOVE 'SUPPRESSION' TO ACTIONE-ED
    END-EVALUATE

* INITIALISATION DES VARIABLES AVANT AFFICHAGE
    MOVE SPACES          TO MATE-ED
                                NOM-EMPE-ED
                                NODE-ED
                                NDEE-ED
                                DATE-ED
    MOVE ZEROES          TO SALE-ED
                                COME-ED

* RECHERCHE DU NOM DU DEPARTEMENT
    MOVE MVT-NOD1      TO HVD-NOD

    EXEC SQL
        SELECT NDE INTO :HVD-NDE
        FROM DEPARTEMENT
        WHERE NOD = :HVD-NOD
    END-EXEC
    EVALUATE SQLCODE
        WHEN ZEROES
            CONTINUE
        WHEN +100
            MOVE ALL '.' TO HVD-NOD
        WHEN OTHER
            MOVE SQLCODE TO SQLCODE-ED
            DISPLAY 'ERREUR SELECT DEPARTEMENT, SQLCODE : ' SQLCODE-ED
            PERFORM FIN-ERREUR-GRAVE
    END-EVALUATE

* ALIMENTATION DES ZONES D'EDITION
    MOVE MVT-MAT      TO MATE-ED

```

```

IF MVT-CODE-MVT NOT = 'S'
  MOVE MVT-NOM      TO NOM-EMPE-ED
  MOVE MVT-NOD1     TO NODE-ED
  MOVE HVD-NDE      TO NDEE-ED
  STRING MVT-DAT(7 : 2) DELIMITED BY SIZE
    '/' DELIMITED BY SIZE
  MVT-DAT(5 : 2) DELIMITED BY SIZE
    '/' DELIMITED BY SIZE
  MVT-DAT(1 : 4) DELIMITED BY SIZE
    INTO DATE-ED
  END-STRING
  MOVE MVT-SAL      TO SALE-ED
  MOVE MVT-COM      TO COME-ED
END-IF

```

```

MOVE LIG7E      TO ENR-EDIT
WRITE ENR-EDIT
MOVE LIG5E      TO ENR-EDIT
WRITE ENR-EDIT
.

```

TRT-DEPARTEMENT.

```

* LIGNE DETAIL DU DEPARTEMENT
  EVALUATE MVT-CODE-MVT
    WHEN 'C' MOVE 'CREATION'      TO ACTIOND-ED
    WHEN 'M' MOVE 'MODIFICATION' TO ACTIOND-ED
    WHEN 'S' MOVE 'SUPPRESSION' TO ACTIOND-ED
  END-EVALUATE

```

* INITIALISATION DES VARIABLES AVANT AFFICHAGE

```

  MOVE SPACES      TO NODD-ED
                      NOM-DEPD-ED
                      LIED-ED
                      CHED-ED
                      NOM-CHEFD-ED

```

* RECUPERATION DU NOM DU CHEF DANS LA TABLE EMPLOYE

```

  MOVE MVT-CHE      TO HVE-MAT

```

```

EXEC SQL
  SELECT NOM INTO :HVE-NOM
  FROM EMPLOYE
  WHERE MAT = :HVE-MAT
END-EXEC

```

```

EVALUATE SQLCODE
  WHEN ZEROES
    CONTINUE
  WHEN +100
    MOVE ALL '.' TO HVE-MAT
  WHEN OTHER
    MOVE SQLCODE TO SQLCODE-ED
    DISPLAY 'ERREUR SELECT EMPLOYE, SQLCODE : ' SQLCODE-ED
    PERFORM FIN-ERREUR-GRAVE
  END-EVALUATE

```

* ALIMENTATION DES ZONES D'EDITION

```

  MOVE MVT-NOD2     TO NODD-ED
  IF MVT-CODE-MVT NOT = 'S'
    MOVE MVT-NDE     TO NOM-DEPD-ED
    MOVE MVT-LIE     TO LIED-ED
    MOVE MVT-CHE     TO CHED-ED
    MOVE HVE-NOM     TO NOM-CHEFD-ED
  END-IF

```

```

MOVE LIG7D      TO ENR-EDIT

```



```

WRITE ENR-EDIT
MOVE LIG5D      TO ENR-EDIT
WRITE ENR-EDIT
.

READ-MVTVAL.
  READ MVTVAL INTO MVT-ENR
  AT END      SET FIN-MVTVAL TO TRUE
  NOT AT END  ADD 1 TO CPT-MVT-LUS
END-READ
IF NOT (FS-MVTVAL = ZEROES OR 10)
  DISPLAY 'PB READ MVTVAL, FS : ' FS-MVTVAL
  PERFORM FIN-ERREUR-GRAVE
END-IF
.

FIN-ERREUR-GRAVE.
MOVE 20 TO RETURN-CODE
PERFORM FIN.

FIN.
CLOSE MVTVAL EDIT
DISPLAY 'NB MVTS LUS : ' CPT-MVT-LUS
STOP RUN.

```

```

JCL DB2P5
//API8DB JOB NOTIFY=&SYSUID,CLASS=A,MSGCLASS=H,TIME=(,5),RESTART=*
//*
//PROCLIB JCLLIB ORDER=SDJ.FORM.PROCLIB
//*
//          SET SYSUID=API8,
//          NOMPGM=DB2P5
//*
//APPROC   EXEC COMPDB2
//STEPDB2.SYSLIB DD DSN=API8.SOURCE.DCLGEN,DISP=SHR
//STEPDB2.SYSIN DD DSN=API8.SOURCE.DB2(DB2P5),DISP=SHR
//STEPDB2.DBRMLIB DD DSN=API8.SOURCE.DBRMLIB(DB2P5),DISP=SHR
//STEPJOB.SYSLIB DD DSN=API8.SOURCE.COPY,DISP=SHR
//STEPLNK.SYSLMOD DD DSN=API8.SOURCE.PGMLIB(DB2P5),DISP=SHR
//*
//*--- ETAPE DE BIND -----
//*
//BIND      EXEC PGM=IKJEFT01,COND=(4,LT)
//DBRMLIB DD DSN=API8.SOURCE.DBRMLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*,OUTLIM=25000
//SYSTSIN DD *
    DSN SYSTEM (DSN1)
    BIND PLAN      (DB2P5)      -
        QUALIFIER (API8)      -
        ACTION     (REPLACE)   -
        MEMBER      (DB2P5)    -
        VALIDATE    (BIND)     -
        ISOLATION   (CS)       -
        ACQUIRE     (USE)      -
        RELEASE      (COMMIT)   -
        EXPLAIN      (NO)
//
//* SUPPRESSION DU FICHIER TRIE
//STEP1      EXEC PGM=IDCAMS,COND=(4,LT)
//SYSPRINT DD SYSOUT=*
//SYSIN      DD *
    DELETE API8.COB.MVTVAL.BASE.EMPLOYE.TRIE
    SET MAXCC = 0
    DELETE API8.COB.MVTVAL.BASE.EMPLOYE.EDIT
    SET MAXCC = 0
//
//* TRI DU FICHIER DES MVTS VALIDES
//* SUR CODE TABLE ORDRE DECROISSANT, MATRICULE ET CODE MOUVEMENT
//* ORDRE CROISSANT
//STEP2      EXEC PGM=SORT,COND=(4,LT)
//SYSOUT DD SYSOUT=*
//SORTIN DD DSN=API8.COB.MVTVAL.BASE.EMPLOYE,DISP=SHR
//SORTOUT DD DSN=API8.COB.MVTVAL.BASE.EMPLOYE.TRIE,
//          DISP=(,CATLG,DELETE),
//          LIKE=API0.COB.MVT.BASE.EMPLOYE
//SYSIN      DD *
    SORT FIELDS=(2,1,CH,D,3,3,CH,A,1,1,CH,A)
//
//* EXECUTION DU PROGRAMME
//STEPRUN EXEC PGM=IKJEFT01,COND=(4,LT)
//STEPLIB DD DSN=API8.SOURCE.PGMLIB,DISP=SHR
//          DD DSN=API8.COBOL.LOAD,DISP=SHR
//          DD DSN=API0.COBOL.LOAD,DISP=SHR
//MVTVAL DD DSN=API8.COB.MVTVAL.BASE.EMPLOYE.TRIE,DISP=SHR
//EDIT DD DSN=API8.COB.MVTVAL.BASE.EMPLOYE.EDIT,
//          DISP=(,CATLG,DELETE),
//          DCB=(DSORG=PS,RECFM=FB,LRECL=133),SPACE=(TRK,(1))
//SYSOUT DD SYSOUT=*,OUTLIM=1000
//SYSPRINT DD SYSOUT=*,OUTLIM=2500
//SYSTSIN DD *
    DSN SYSTEM (DSN1)

```

```
RUN PROGRAM(DB2P5) PLAN (DB2P5) PARM('11001')  
//
```

```

PROGRAMME ALIMDB2 (création des tables)
  IDENTIFICATION DIVISION.
  PROGRAM-ID. ALIMDB2.
  AUTHOR. OTEUR.
  DATE-WRITTEN. 29/03/2023.
  *****
  *   FONCTION DU PROGRAMME: EDITION DES DEPARTEMENTS   *
  *   250429 : CREATION DU PROGRAMME                     *
  *****
  ENVIRONMENT DIVISION.
  CONFIGURATION SECTION.
  SPECIAL-NAMES.
      DECIMAL-POINT IS COMMA.
  DATA DIVISION.
  WORKING-STORAGE SECTION.
  * DECLARATION DE LA TABLE DEPARTEMENT
      EXEC SQL INCLUDE DEPT      END-EXEC.

  * DECLARATION DE LA TABLE EMPLOYE
      EXEC SQL INCLUDE EMPLOYE END-EXEC.

  * DECLARATION DE LA SQLCA
      EXEC SQL INCLUDE SQLCA    END-EXEC.

  PROCEDURE DIVISION.
  *-- SUPPRESSION DE LA TABLE EMPLOYE
      EXEC SQL
          DROP TABLE EMPLOYE
        END-EXEC
      EXEC SQL COMMIT END-EXEC

  *-- CREATION DE LA TABLE EMPLOYE
      EXEC SQL
          CREATE TABLE EMPLOYE
            (MAT      CHAR(3)      NOT NULL,
             NOM      CHAR(7)      NOT NULL,
             NOD      CHAR(3)      NOT NULL,
             DAT      DATE         NOT NULL,
             SAL      DECIMAL(7,2) NOT NULL,
             COM      DECIMAL(7,2) ,
             PRIMARY KEY(MAT) ,
             FOREIGN KEY(NOD) REFERENCES DEPARTEMENT ON DELETE RESTRICT)
        END-EXEC
      EXEC SQL COMMIT END-EXEC

  *-- CREATION DE L'INDEX DE LA CLE PRIMAIRE
      EXEC SQL
          CREATE UNIQUE INDEX IXEMPLOYE ON EMPLOYE (MAT)
        END-EXEC
      EXEC SQL COMMIT END-EXEC

  *-- ALIMENTATION DE LA TABLE EMPLOYE
      EXEC SQL
          INSERT INTO EMPLOYE VALUES
            ('10','DURAND','E10','2000-02-10',11000,5000)
        END-EXEC
      EXEC SQL
          INSERT INTO EMPLOYE VALUES
            ('20','DUPOND','P01','1998-01-11',15100.50,NULL)
        END-EXEC
      EXEC SQL
          INSERT INTO EMPLOYE VALUES
            ('30','BARI','P02','2001-01-04',6000.00,3000)
        END-EXEC
      EXEC SQL
          INSERT INTO EMPLOYE VALUES

```

```

        ('40','JAVA','C04','2005-12-03',21000.50,1000.50)
END-EXEC
EXEC SQL
    INSERT INTO EMPLOYE VALUES
        ('50','CICS','P02','2006-06-20',8500,3500.50)
END-EXEC
EXEC SQL
    INSERT INTO EMPLOYE VALUES
        ('60','PARIS','P01','2008-04-22',9000,3500)
END-EXEC
EXEC SQL
    INSERT INTO EMPLOYE VALUES
        ('70','NOEL','P02','2007-05-18',10500,900.50)
END-EXEC
EXEC SQL COMMIT END-EXEC

*-- SUPPRESSION DE LA TABLE DEPARTEMENT
EXEC SQL
    DROP TABLE DEPARTEMENT
END-EXEC
EXEC SQL COMMIT END-EXEC

*-- CREATION DE LA TABLE DEPARTEMENT
EXEC SQL
    CREATE TABLE DEPARTEMENT
        (NOD          CHAR(3)          NOT NULL,
         NDE          CHAR(6)          NOT NULL,
         LIE          CHAR(4)          NOT NULL,
         CHE          CHAR(3)          NOT NULL,
         PRIMARY KEY (NOD),
         FOREIGN KEY (CHE) REFERENCES EMPLOYE ON DELETE RESTRICT)
END-EXEC

*-- CREATION DE L'INDEX DE LA CLE PRIMAIRE
EXEC SQL
    CREATE UNIQUE INDEX IXDEPARTEMENT ON DEPARTEMENT (NOD)
END-EXEC
EXEC SQL COMMIT END-EXEC

*-- ALIMENTATION DE LA TABLE DEPARTEMENT
EXEC SQL
    INSERT INTO DEPARTEMENT VALUES
        ('C04','COMPTA','BAT1','40')
END-EXEC
EXEC SQL
    INSERT INTO DEPARTEMENT VALUES
        ('E10','PAIE','BAT2','10')
END-EXEC
EXEC SQL
    INSERT INTO DEPARTEMENT VALUES
        ('P01','INFO','BAT3','20')
END-EXEC
EXEC SQL
    INSERT INTO DEPARTEMENT VALUES
        ('P02','COMMER','BAT2','70')
END-EXEC
EXEC SQL COMMIT END-EXEC

* ARRET DU PROGRAMME
STOP RUN
.
```

```

JCL ALIMDB2J
//API8DB JOB NOTIFY=&SYSUID,CLASS=A,MSGCLASS=H
//*
//PROCLIB JCLLIB ORDER=SDJ.FORM.PROCLIB
//*
//          SET SYSUID=API8,
//          NOMPGM=ALIMDB2
//*
//APPROC   EXEC COMPDB2
//STEPDB2.SYSLIB DD DSN=API8.SOURCE.DCLGEN,DISP=SHR
//          DD DSN=API8.SOURCE.COPY,DISP=SHR
//STEPDB2.SYSIN DD DSN=API8.SOURCE.DB2 (ALIMDB2) ,DISP=SHR
//STEPDB2.DBRMLIB DD DSN=API8.SOURCE.DBRMLIB (ALIMDB2) ,DISP=SHR
//STEPLNK.SYSLMOD DD DSN=API8.SOURCE.PGMLIB (ALIMDB2) ,DISP=SHR
//
//*--- ETAPE DE BIND -----
//
//BIND      EXEC PGM=IKJEFT01,COND=(4,LT)
//DBRMLIB DD DSN=API8.SOURCE.DBRMLIB,DISP=SHR
//SYSTSPRT DD SYSOUT=*,OUTLIM=25000
//SYSTSIN DD *
DSN SYSTEM (DSN1)
BIND PLAN      (ALIMDB2) -
  QUALIFIER (API8) -
  ACTION     (REPLACE) -
  MEMBER      (ALIMDB2) -
  VALIDATE    (BIND) -
  ISOLATION   (CS) -
  ACQUIRE    (USE) -
  RELEASE     (COMMIT) -
  EXPLAIN     (NO)
//
//* EXECUTION DU PROGRAMME
//STEPRUN EXEC PGM=IKJEFT01,COND=(4,LT)
//STEPLIB DD DSN=API8.SOURCE.PGMLIB,DISP=SHR
//SYSOUT DD SYSOUT=*,OUTLIM=1000
//SYSTSPRT DD SYSOUT=*,OUTLIM=2500
//SYSTSIN DD *
DSN SYSTEM (DSN1)
RUN PROGRAM(ALIMDB2) PLAN (ALIMDB2) PARMS('11001')
//

```

```

JCL D'EXECUTION DE LA CHAINE
//API8PRJ1 JOB NOTIFY=&SYSUID,CLASS=A,MSGCLASS=H,TIME=(,10)
//*
//* SUPPRESSION FICHER DES MVTS VALIDES
//STEP1 EXEC PGM=IDCAMS,COND=(4,LT)
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE API8.COB.MVTVAL.BASE.EMPLOYE
SET MAXCC = 0
//*
//* EXECUTION DU PROGRAMME DE SUPPRESSION ET CREATION DES TABLES
//STEPRUN EXEC PGM=IKJEFT01,COND=(4,LT)
//STEPLIB DD DSN=API8.SOURCE.PGMLIB,DISP=SHR
//SYSOUT DD SYSOUT=*,OUTLIM=1000
//SYSTSPRT DD SYSOUT=*,OUTLIM=2500
//SYSTSIN DD *
DSN SYSTEM (DSN1)
RUN PROGRAM(ALIMDB2) PLAN (ALIMDB2) PARMS('11001')
//*
//* EXECUTION DU PROGRAMME DE MISE A JOUR DES TABLES
//STEPRUN EXEC PGM=IKJEFT01,COND=(4,LT)
//STEPLIB DD DSN=API8.SOURCE.PGMLIB,DISP=SHR
// DD DSN=API0.COBO.LOAD,DISP=SHR
//MVT DD DSN=API0.COBO.MVT.BASE.EMPLOYE,DISP=SHR
//MVTVAL DD DSN=API8.COB.MVTVAL.BASE.EMPLOYE,DISP=(,CATLG,DELETE),
// LIKE=API0.COBO.MVT.BASE.EMPLOYE
//SYSOUT DD SYSOUT=*,OUTLIM=1000
//SYSTSPRT DD SYSOUT=*,OUTLIM=2500
//SYSTSIN DD *
DSN SYSTEM (DSN1)
RUN PROGRAM(DB2P4) PLAN (DB2P4) PARMS('11001')
//*
//* SUPPRESSION DU FICHER TRIE
//STEP1 EXEC PGM=IDCAMS,COND=(4,LT)
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE API8.COB.MVTVAL.BASE.EMPLOYE.TRIE
SET MAXCC = 0
DELETE API8.COB.MVTVAL.BASE.EMPLOYE.EDIT
SET MAXCC = 0
//*
//* TRI DU FICHER DES MVTS VALIDES
//* SUR CODE TABLE ORDRE DECROISSANT, MATRICULE ET CODE MOUVEMENT
//* ORDRE CROISSANT
//STEP2 EXEC PGM=ISORT,COND=(4,LT)
//SYSOUT DD SYSOUT=*
//SORTIN DD DSN=API8.COB.MVTVAL.BASE.EMPLOYE,DISP=SHR
//SORTOUT DD DSN=API8.COB.MVTVAL.BASE.EMPLOYE.TRIE,
// DISP=(,CATLG,DELETE),
// LIKE=API0.COBO.MVT.BASE.EMPLOYE
//SYSIN DD *
SORT FIELDS=(2,1,CH,D,3,3,CH,A,1,1,CH,A)
//*
//* EXECUTION DU PROGRAMME D'EDITION DES MVTS DU JOUR
//STEPRUN EXEC PGM=IKJEFT01,COND=(4,LT)
//STEPLIB DD DSN=API8.SOURCE.PGMLIB,DISP=SHR
// DD DSN=API8.COBO.LOAD,DISP=SHR
// DD DSN=API0.COBO.LOAD,DISP=SHR
//MVTVAL DD DSN=API8.COB.MVTVAL.BASE.EMPLOYE.TRIE,DISP=SHR
//EDIT DD DSN=API8.COB.MVTVAL.BASE.EMPLOYE.EDIT,
// DISP=(,CATLG,DELETE),
// DCB=(DSORG=PS,RECFM=FB,LRECL=133),SPACE=(TRK,(1))
//SYSOUT DD SYSOUT=*,OUTLIM=1000
//SYSTSPRT DD SYSOUT=*,OUTLIM=2500
//SYSTSIN DD *
DSN SYSTEM (DSN1)

```

```
RUN PROGRAM(DB2P5) PLAN (DB2P5) PARM('11001')  
//
```


EXEMPLES D'ORDRES SQL DE MISE A JOUR

pour les colonnes qui sont déclarées NULL dans le CREATE TABLE, les mettre à NULL quand la host-variable n'est pas renseignée

exemple . Colonne COM

déclarer en working-storage un indicateur par colonne au format demi-mot binaire
01 IND-COM PIC S9(4) COMP VALUE ZEROES.

ORDRE INSERT

renseigner toutes les host-variables

```
MOVE ... TO HVE-MAT
MOVE ... TO HVE-NOM
MOVE ... TO HVE-NOD
MOVE ... TO HVE-DAT doit être au format AAAA-MM-JJ
MOVE ... TO HVE-SAL doit être au format S9(5)V99 COMP-3
MOVE ... TO HVE-COM doit être au format S9(5)V99 COMP-3
```

Renseigner l'indicateur de nullité pour la colonne COM

```
IF HVE-COM = ZEROES
    MOVE -1 TO IND-COM
ELSE
    MOVE ZEROES TO IND-COM
END-IF
```

1ère façon : en spécifiant les colonnes, l'ordre n'est pas important

```
EXEC SQL
    INSERT INTO EMPLOYE
        (MAT, NOD, NOM, SAL, COM, DAT)
    VALUES
        (:HVE-MAT, :HVE-NOD, :HVE-NOM, :HVE-SAL, :HVE-COM, :HVE-DAT)
END-EXEC
```

2ème façon : sans spécifier les colonnes, l'ordre des host-variables doit correspondre à celui du CREATE TABLE

```
EXEC SQL
    INSERT INTO EMPLOYE
    VALUES
        (:HVE-MAT, :HVE-NOM, :HVE-NOD, :HVE-DAT, :HVE-SAL, :HVE-COM)
END-EXEC
```

```
IF SQLCODE = ZEROES
    ADD 1 TO CPT-EMP-CRE
ELSE
    MOVE SQLCODE TO ED-SQLCODE
    DISPLAY 'ERREUR INSERT EMPLOYE ; SQLCODE : ' ED-SQLCODE
    PERFORM ERREUR-GRAVE
END-IF
```

ORDRE UPDATE

renseigner toutes les host-variables

```
MOVE ... TO HVE-MAT
MOVE ... TO HVE-NOM
MOVE ... TO HVE-NOD
MOVE ... TO HVE-DAT doit être au format AAAA-MM-JJ
MOVE ... TO HVE-SAL doit être au format S9(5)V99 COMP-3
MOVE ... TO HVE-COM doit être au format S9(5)V99 COMP-3
```

Renseigner l'indicateur de nullité pour la colonne COM

```
IF HVE-COM = ZEROES
    MOVE -1 TO IND-COM
ELSE
    MOVE ZEROES TO IND-COM
END-IF
```

```
EXEC SQL
    UPDATE EMPLOYE
    SET     NOM =:HVE-NOM,
           COM =:HVE-COM:IND-COM,
           SAL =:HVE-SAL,
           DAT =:HVE-DAT,
           NOD =:HVE-NOD
    WHERE MAT =:HVE-MAT
END-EXEC
```

ORDRE DELETE

Renseigner les host-variables de la clé primaire
MOVE ... TO HVE-MAT

```
EXEC SQL
    DELETE FROM EMPLOYE
    WHERE MAT =:HVE-MAT
END-EXEC
```

SUPPLÉMENT

But : augmenter les salaires et commissions des employés selon le département, l'année d'entrée dans l'entreprise et le statut de l'employé. Les informations sont stockées dans un fichier KSDS. Appliquer la règle de calcul correspondante.

Programme : DB2P6

Fichier : API0.COB.AUGMENT.KSDS

Organisation : KSDS

Longueur enregistrement : 20 octets

Clé :

- numéro département 3 caractères alphanumériques
- année d'entrée 4 caractères numériques

Champs : chaque règle de calcul est stockée dans 1 caractère alphanumérique

- règle de calcul de l'augmentation du salaire des employés (hors chef de département)
- règle de calcul de l'augmentation de la commission des employés (hors chef de département)
- règle de calcul de l'augmentation du salaire du chef de département
- règle de calcul de l'augmentation de la commission du chef de département

Règles de calcul :

- règle A : augmenter d'un forfait de 200
- règle B : selon le montant :
 - inférieur à 5000, augmenter de 2%
 - de 5000 à 10000, augmenter de 1,5%
 - de 10000 à 15000, augmenter de 1%
 - supérieur à 15000, augmenter de 0,5%
- règle C : inférieur à 5000, augmenter de 300
 - de 5000 à 10000, augmenter de 200
 - de 10000 à 15000, augmenter de 100
 - supérieur à 15000, pas d'augmentation
- règle D : augmenter de 5%

Contenu du fichier API0.COB.AUGMENT.KSDS

C041998ABCD
C042000ABCD
C042001BCBA
C042005AABB
C042006CBDA
C042007CCAB
C042008BDDD
E101998ABCD
E102000ABCD
E102001BCBA
E102005AABB
E102006CBDA
E102007CCAB
E102008BDDD
P011998ABCD
P012000CDDA
P012001BCBA
P012005AABB
P012006CBDA
P012007CCAB
P012008BDDD
P021998ABCD
P022000CDDA
P022001BCBA
P022005AABB
P022006CBDA
P022007CCAB
P022008BDDD