Zadanie 2.

Stwórz tabele

1. Employee

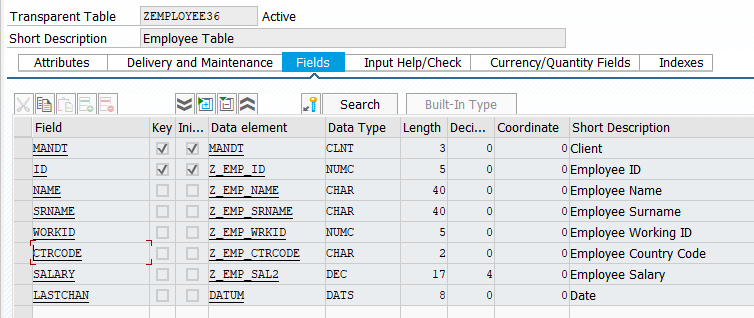
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| USERID | NAME | SURNAME | WORKID | COUNTRYCODE | SALARY | LAST\_CHANGED |

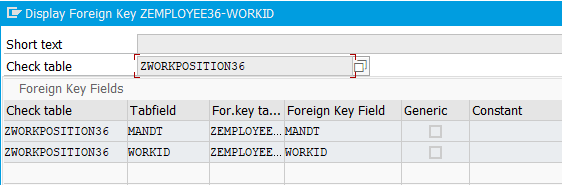
1. WorkPosition

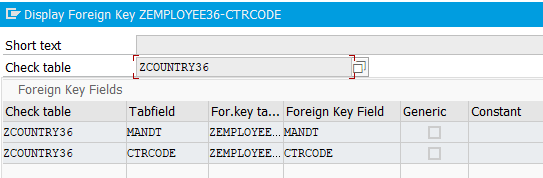
|  |  |  |  |
| --- | --- | --- | --- |
| WORKID | POSIITION | DEFAULT\_SALARY | LAST\_CHANGED |

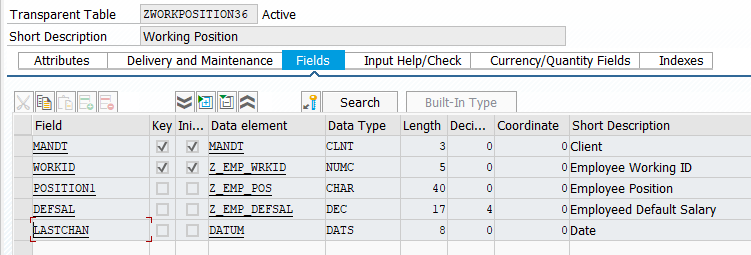
1. Country ZMJ\_COUNTRY\_T

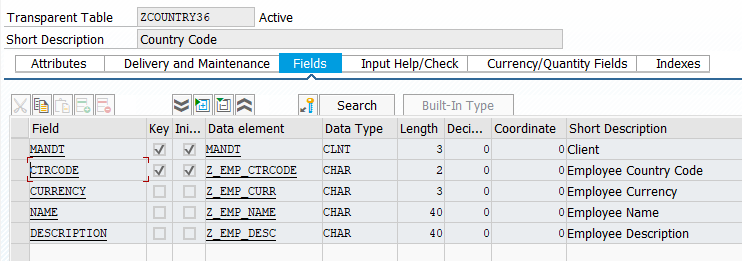
|  |  |  |  |
| --- | --- | --- | --- |
| COUNTRYCODE | CURRENCY | NAME | Description |





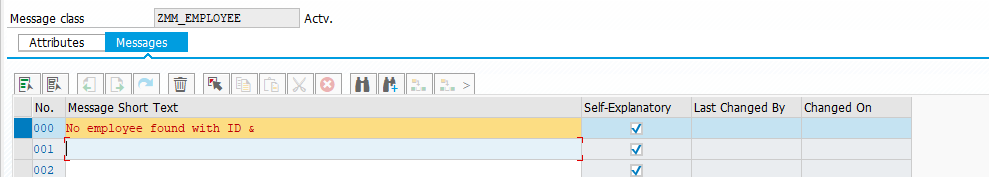






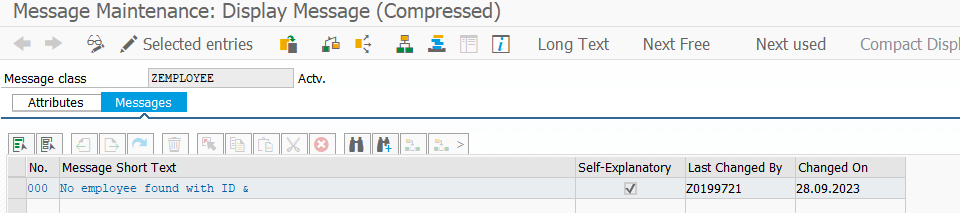
3.Stwórz raport przerobowy ZMJ\_EMPLOYEE\_MANAGEMENT w którym będziesz mógł operować na wartościach z tabeli w oparciu o opcje checkboxowe:

Pamiętaj o error handlingu po selectach po updateach etc. ( komunikatach se91 - )

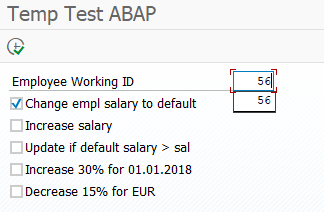


& dla wstawienia wartości dynamicznie w wiadomości.

Rozwiązanie



Błede ID:





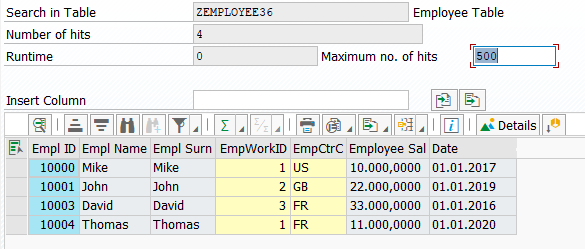
KOD

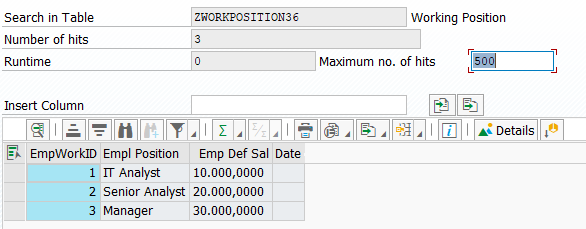
SELECTION-SCREEN BEGIN OF BLOCK b1 WITH FRAME TITLE TEXT-000.  
  PARAMETERS:  
    p\_id     TYPE z\_emp\_id MODIF ID eid,  
    p\_wid    TYPE z\_emp\_wrkid MODIF ID wid,  
    p\_salar  TYPE int4 MODIF ID sa2,  
    p\_newsur TYPE z\_emp\_name MODIF ID nsu,  
    p\_newsal TYPE z\_emp\_sal2 MODIF ID nsa.  
SELECTION-SCREEN END OF BLOCK b1.  
  
SELECTION-SCREEN BEGIN OF BLOCK b2 WITH FRAME TITLE TEXT-001.  
  PARAMETERS:  
    p\_sala1 AS CHECKBOX USER-COMMAND pay DEFAULT 'X' MODIF ID sc1,  
    p\_sala2 AS CHECKBOX USER-COMMAND sur MODIF ID sc2,  
    p\_sala3 AS CHECKBOX USER-COMMAND sta MODIF ID sc3,  
    p\_sala4 AS CHECKBOX USER-COMMAND sxa MODIF ID sc4,  
    p\_sala5 AS CHECKBOX USER-COMMAND sxa MODIF ID sc5.  
SELECTION-SCREEN END OF BLOCK b2.  
  
DATA: zdefsal  TYPE z\_emp\_defsal,  
      zdefsal3 TYPE z\_emp\_defsal,  
      zdefsal4 TYPE z\_emp\_defsal,  
      date1    TYPE d VALUE '20180101',  
      date2    TYPE d,  
      code     TYPE z\_emp\_ctrcode,  
      count    TYPE i,  
      defid    TYPE z\_emp\_id VALUE '10000',  
      curr     TYPE z\_emp\_curr.  
  
AT SELECTION-SCREEN OUTPUT.  
  LOOP AT SCREEN.  
    IF p\_sala3 EQ abap\_true.  
      p\_sala1 = abap\_false.  
      p\_sala2 = abap\_false.  
      IF screen-group1 EQ 'SC1' OR screen-group1 EQ 'SC3' OR screen-group1 = 'WID' OR screen-group1 EQ 'SC2' OR screen-group1 = 'SC4' OR screen-group1 = 'SC5'.  
        screen-active = 1.  
      ELSE.  
        screen-active = 0.  
      ENDIF.  
      MODIFY SCREEN.  
    ENDIF.  
    IF p\_sala1 EQ abap\_true.  
      p\_sala3 = abap\_false.  
      p\_sala2 = abap\_false.  
      IF screen-group1 EQ 'SC1' OR screen-group1 EQ 'SC3' OR screen-group1 = 'WID' OR screen-group1 EQ 'SC2' OR screen-group1 = 'SC4' OR screen-group1 = 'SC5'.  
        screen-active = 1.  
      ELSE.  
        screen-active = 0.  
      ENDIF.  
      MODIFY SCREEN.  
    ENDIF.  
    IF p\_sala2 EQ abap\_true.  
      p\_sala3 = abap\_false.  
      p\_sala1 = abap\_false.  
      IF screen-group1 EQ 'WID' OR screen-group1 EQ 'SC1' OR screen-group1 EQ 'SC3' OR screen-group1 = 'SA2' OR screen-group1 EQ 'SC2' OR screen-group1 = 'SC4' OR screen-group1 = 'SC5'.  
        screen-active = 1.  
      ELSE.  
        screen-active = 0.  
      ENDIF.  
      MODIFY SCREEN.  
    ENDIF.  
    IF p\_sala4 EQ abap\_true.  
      p\_sala1 = abap\_false.  
      p\_sala2 = abap\_false.  
      p\_sala3 = abap\_false.  
      IF screen-group1 EQ 'SC1' OR screen-group1 EQ 'SC3' OR screen-group1 = 'SC4' OR screen-group1 EQ 'SC2' OR screen-group1 = 'SC5'.  
        screen-active = 1.  
      ELSE.  
        screen-active = 0.  
      ENDIF.  
      MODIFY SCREEN.  
    ENDIF.  
    IF p\_sala5 EQ abap\_true.  
      p\_sala1 = abap\_false.  
      p\_sala2 = abap\_false.  
      p\_sala3 = abap\_false.  
      p\_sala4 = abap\_false.  
      IF screen-group1 EQ 'SC1' OR screen-group1 EQ 'SC3' OR screen-group1 = 'SC4' OR screen-group1 EQ 'SC2' OR screen-group1 = 'SC5'.  
        screen-active = 1.  
      ELSE.  
        screen-active = 0.  
      ENDIF.  
      MODIFY SCREEN.  
    ENDIF.  
  ENDLOOP.  
  
START-OF-SELECTION.  
  
  
  IF p\_sala1 IS NOT INITIAL.  
  
    SELECT defsal FROM zworkposition36 WHERE workid = @p\_wid INTO TABLE @DATA(lt\_employee).  
  
    LOOP AT lt\_employee INTO DATA(ls\_employee).  
      zdefsal = ls\_employee-defsal.  
    ENDLOOP.  
  
    SELECT SINGLE \* FROM zemployee36 WHERE workid = @p\_wid INTO @DATA(ls\_emp2).  
    ls\_emp2-salary = zdefsal.  
    UPDATE zemployee36 FROM ls\_emp2.  
  
    IF sy-subrc <> 0.  
      MESSAGE e000(zemployee) WITH p\_wid.  
    ENDIF.  
  
  ENDIF.  
  
  IF p\_sala2 IS NOT INITIAL.  
  
    SELECT SINGLE \* FROM zemployee36 WHERE workid = @p\_wid INTO @DATA(ls\_employee2).  
    ls\_employee2-salary = ls\_employee2-salary + p\_salar.  
    UPDATE zemployee36 FROM ls\_employee2.  
  
    IF sy-subrc <> 0.  
      MESSAGE e000(zemployee) WITH p\_wid.  
    ENDIF.  
  
  ENDIF.  
  
  
  IF p\_sala3 IS NOT INITIAL.  
  
    SELECT defsal FROM zworkposition36 WHERE workid = @p\_wid INTO TABLE @DATA(lt\_employee3).  
  
    LOOP AT lt\_employee3 INTO DATA(ls\_employee3).  
      zdefsal3 = ls\_employee3-defsal.  
    ENDLOOP.  
  
    SELECT SINGLE \* FROM zemployee36 WHERE workid = @p\_wid INTO @DATA(ls\_emp3).  
    zdefsal4 = ls\_emp3-salary.  
  
    IF zdefsal3 > zdefsal4.  
      ls\_emp3-salary = zdefsal3.  
      UPDATE zemployee36 FROM ls\_emp3.  
      WRITE: 'Salary updated.'.  
    ENDIF.  
  
    IF sy-subrc <> 0.  
      MESSAGE e000(zemployee) WITH p\_wid.  
    ENDIF.  
  
  ENDIF.  
  
  
  IF p\_sala4 IS NOT INITIAL.  
  
    SELECT COUNT( id ) FROM zemployee36 INTO @count.  
  
    DO count TIMES.  
      SELECT lastchan FROM zemployee36 WHERE id = @defid INTO TABLE @DATA(lt\_employee4).  
  
      LOOP AT lt\_employee4 INTO DATA(ls\_employee4).  
        date2 = ls\_employee4-lastchan.  
      ENDLOOP.  
  
      SELECT SINGLE \* FROM zemployee36 WHERE id = @defid INTO @DATA(ls\_emp4).  
  
      IF date2 < date1.  
        ls\_emp4-salary = ls\_emp4-salary \* '1.3'.  
        UPDATE zemployee36 FROM ls\_emp4.  
      ENDIF.  
  
      defid = defid + 1.  
    ENDDO.  
  
    WRITE: 'Salary updated.'.  
  
  ENDIF.  
  
  
  IF p\_sala5 IS NOT INITIAL.  
  
    defid = 10000.  
  
    SELECT COUNT( id ) FROM zemployee36 INTO @count.  
  
    DO count TIMES.  
  
      SELECT ctrcode FROM zemployee36 WHERE id = @defid INTO TABLE @DATA(lt\_employee5).  
  
      LOOP AT lt\_employee5 INTO DATA(ls\_employee5).  
        code = ls\_employee5-ctrcode.  
      ENDLOOP.  
  
      SELECT currency FROM zcountry36 WHERE ctrcode = @code INTO TABLE @DATA(lt\_employee6).  
  
      LOOP AT lt\_employee6 INTO DATA(ls\_employee6).  
        curr = ls\_employee6-currency.  
      ENDLOOP.  
  
      IF curr = 'EUR'.  
        SELECT SINGLE \* FROM zemployee36 WHERE id = @defid INTO @DATA(ls\_emp5).  
        ls\_emp5-salary = ls\_emp5-salary \* '0.85'.  
        UPDATE zemployee36 FROM ls\_emp5.  
      ENDIF.  
  
      defid = defid + 1.  
  
    ENDDO.  
  
    WRITE: 'Salary updated.'.  
  
  ENDIF.

Przykładowe akcje.

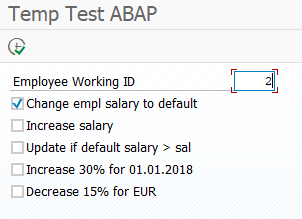
1.Wstaw do tabeli employee pole salary wartości defualt\_salary z tabeli workposition. – checkbox

Przed uruchomieniem:

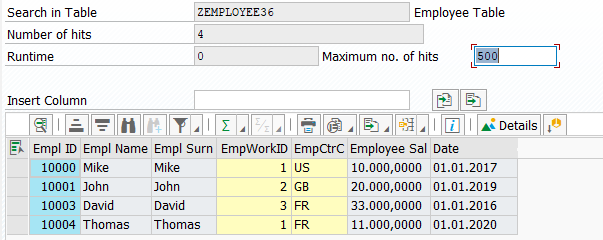




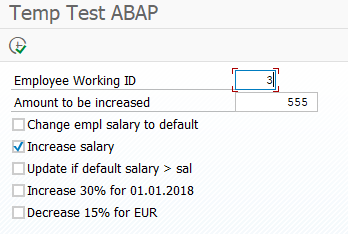
Uruchomienie:



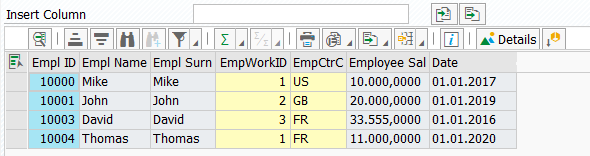
Po uruchomieniu (wartość w 2 linii sie zmieniła)



2.Zwiększ default salary dla zadanej pozycji o wartość podaną na selection screenie – checkbox

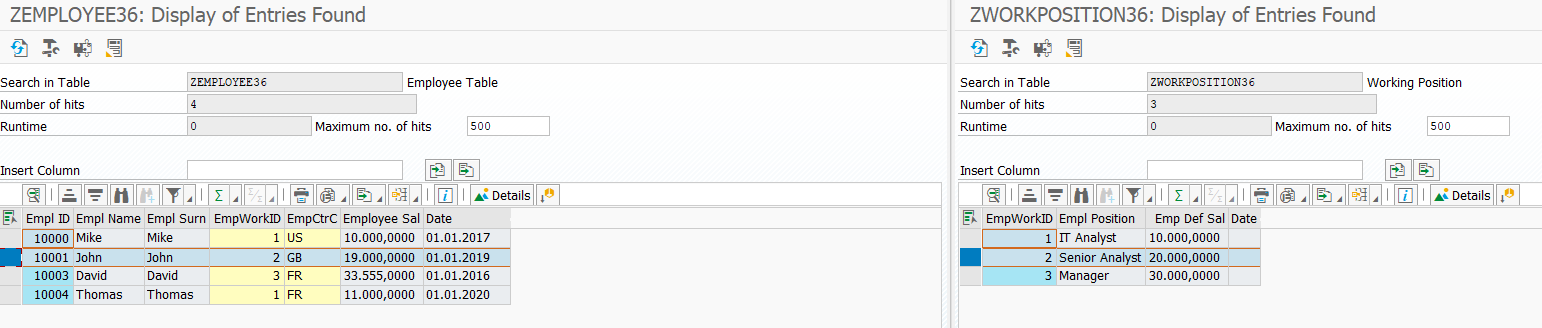
Uruchomienie:  


Po uruchomieniu wartość Salary dla working ID 3 wzrosla o 555

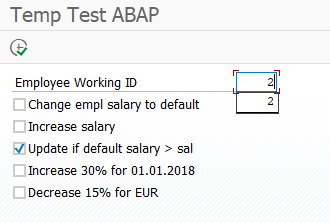


3.Checkbox update salary który nadpisze istniejace salary w tabeli employee jeśli płaca w tabeli work\_position jest większa niż salary w employee dla danej pozycji – checkbox

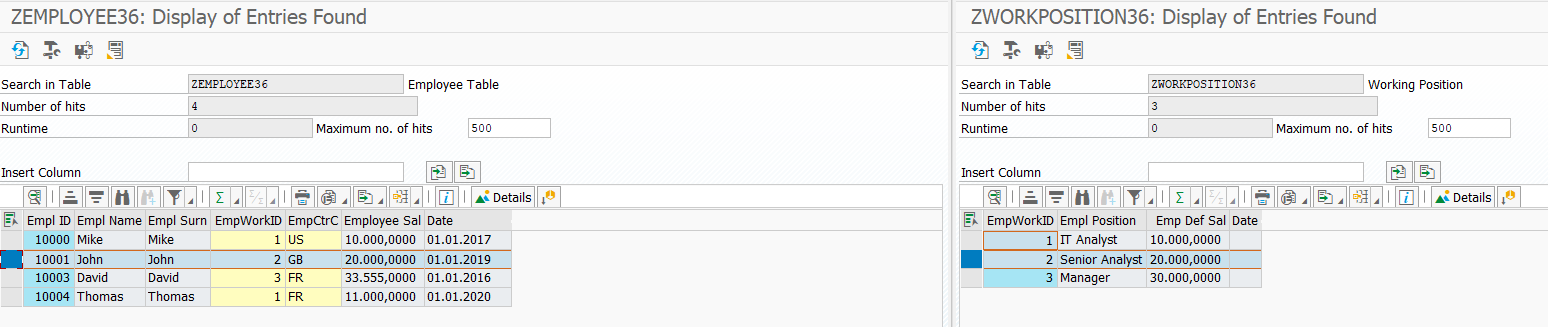
Przed uruchomieniem – wartość salary w 2 jest mniejsza niż w default



Uruchomienie:

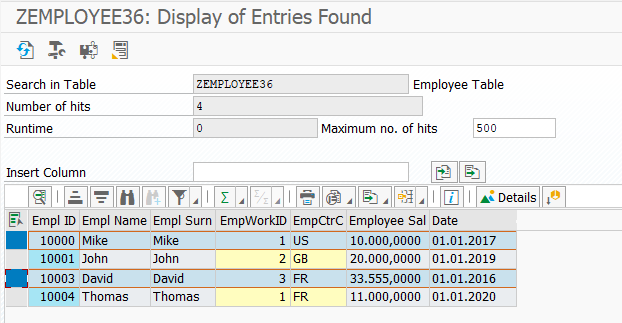


Po uruchomieniu:

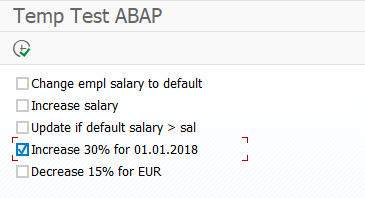


1. Zwiększenie płacy dla pracowników zatrudnionych od 01.01.2018(może być dowolna data jakie masz zdefiniowane w tabeli pracowników ) o 30%

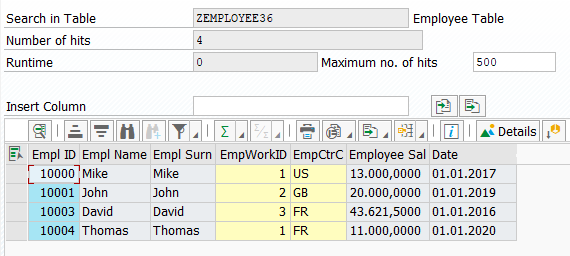
Przed uruchomieniem – dwóch pracowników prze 01 01 2018 wiec im sie powinno zmienic:

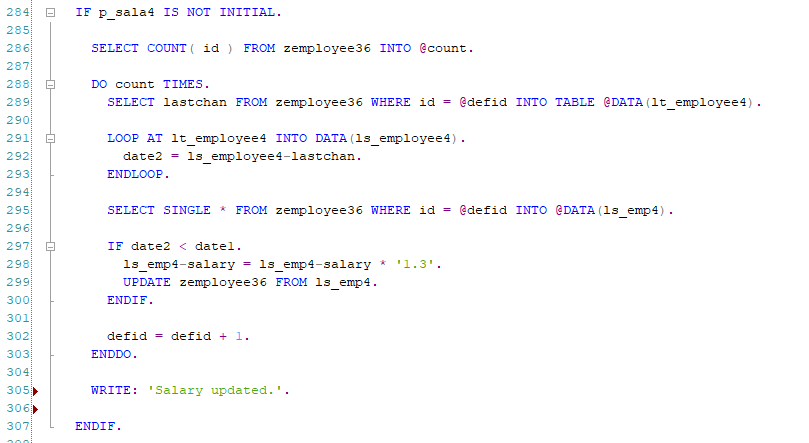


Uruchomienie:



Rezultat:

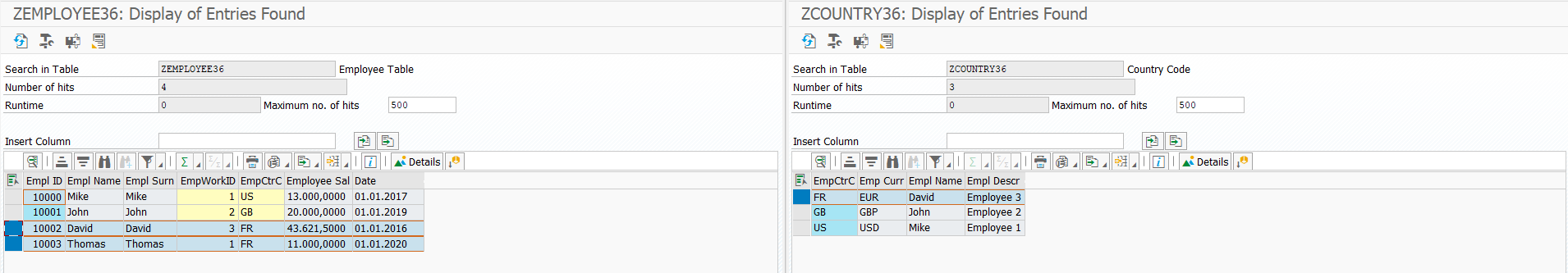




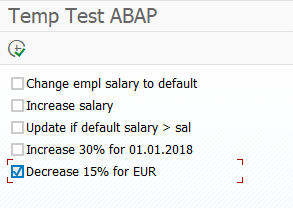
5. Dla ludzi którzy dostają wypłate w EURO zmniejsz ich płacę o 15% , waluta niech będzie zdefiniowana w tabelii ZMJ\_COUNTRY\_T pole CURRENCY przykładowy wpis w tabeli:

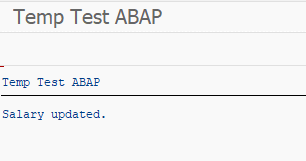
|  |  |  |  |
| --- | --- | --- | --- |
| DE | EUR | Germany | Europe |

Przed uruchomieniem – w EUR jest dwóch pracowników

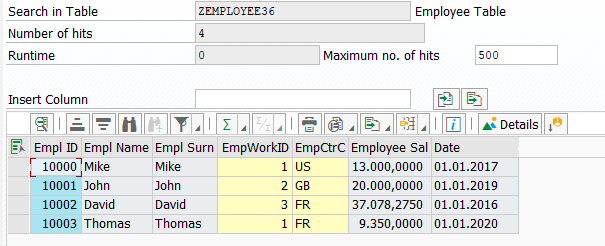


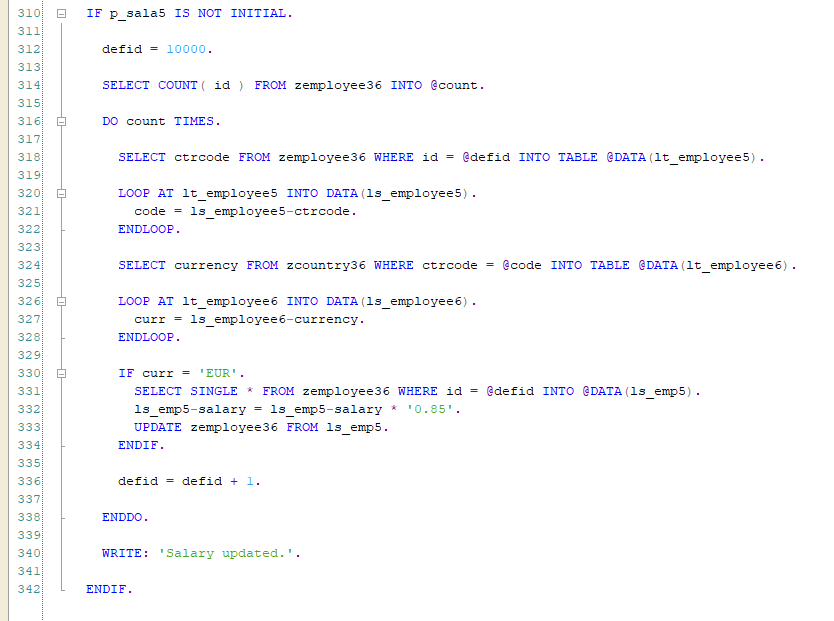
Uruchomienie:





Resultat





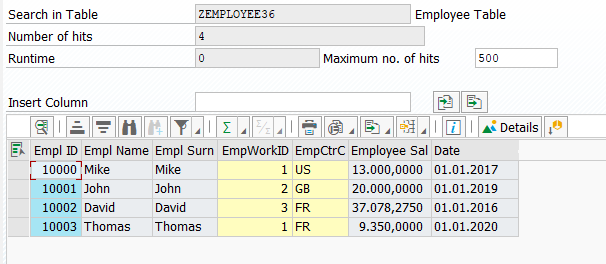
6

Stwórz raport „overview” w którym użytkownik będzie miał opcje wyświetlenia:

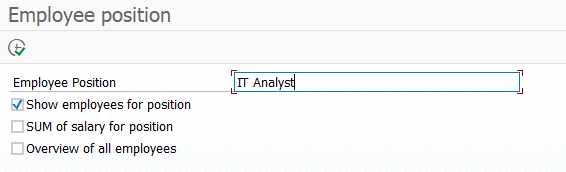
KOD

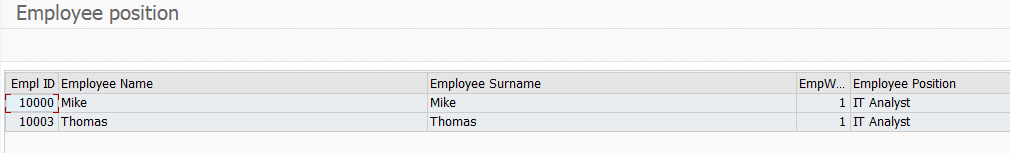
SELECTION-SCREEN BEGIN OF BLOCK b1 WITH FRAME TITLE TEXT-000.  
  PARAMETERS:  
    p\_wid   TYPE z\_emp\_wrkid MODIF ID wid,  
    p\_posit TYPE z\_emp\_pos MODIF ID pos.  
SELECTION-SCREEN END OF BLOCK b1.  
  
SELECTION-SCREEN BEGIN OF BLOCK b2 WITH FRAME TITLE TEXT-001.  
  PARAMETERS:  
    p\_chx1 AS CHECKBOX USER-COMMAND one DEFAULT 'X' MODIF ID sc1,  
    p\_chx2 AS CHECKBOX USER-COMMAND two MODIF ID sc2,  
    p\_chx3 AS CHECKBOX USER-COMMAND thr MODIF ID sc3.  
SELECTION-SCREEN END OF BLOCK b2.  
  
DATA: woid       TYPE z\_emp\_wrkid,  
      go\_alv     TYPE REF TO cl\_salv\_table,  
      itab       TYPE TABLE OF i,  
      sum\_result TYPE i.  
  
AT SELECTION-SCREEN OUTPUT.  
  LOOP AT SCREEN.  
    IF p\_chx1 EQ abap\_true.  
      p\_chx2 = abap\_false.  
      p\_chx3 = abap\_false.  
      IF screen-group1 EQ 'POS' OR screen-group1 EQ 'SC1' OR screen-group1 EQ 'SC2' OR screen-group1 EQ 'SC3'.  
        screen-active = 1.  
      ELSE.  
        screen-active = 0.  
      ENDIF.  
      MODIFY SCREEN.  
    ENDIF.  
    IF p\_chx2 EQ abap\_true.  
      p\_chx1 = abap\_false.  
      p\_chx3 = abap\_false.  
      IF screen-group1 EQ 'POS' OR screen-group1 EQ 'SC1' OR screen-group1 EQ 'SC2' OR screen-group1 EQ 'SC3'.  
        screen-active = 1.  
      ELSE.  
        screen-active = 0.  
      ENDIF.  
      MODIFY SCREEN.  
    ENDIF.  
    IF p\_chx3 EQ abap\_true.  
      p\_chx1 = abap\_false.  
      p\_chx2 = abap\_false.  
      IF screen-group1 EQ 'SC1' OR screen-group1 EQ 'SC2' OR screen-group1 EQ 'SC3'.  
        screen-active = 1.  
      ELSE.  
        screen-active = 0.  
      ENDIF.  
      MODIFY SCREEN.  
    ENDIF.  
  ENDLOOP.  
  
START-OF-SELECTION.  
  
  
  IF p\_chx1 IS NOT INITIAL.  
  
    SELECT workid FROM zworkposition36 WHERE position1 = @p\_posit INTO TABLE @DATA(lt\_employee).  
  
    LOOP AT lt\_employee INTO DATA(ls\_employee).  
      woid = ls\_employee-workid.  
    ENDLOOP.  
  
    SELECT a~id, a~name,  
      a~srname,  
      a~workid,  
      b~position1  
      FROM zemployee36 AS a  
      INNER JOIN zworkposition36 AS b  
      ON a~workid = b~workid  
      WHERE a~workid = @woid  
      INTO TABLE @DATA(lt\_employee2).  
  
    TRY.  
        cl\_salv\_table=>factory( *"*  
          IMPORTING  
            r\_salv\_table   = go\_alv  
          CHANGING  
            t\_table        = lt\_employee2  
        ).  
      CATCH cx\_salv\_msg.  
    ENDTRY.  
  
    go\_alv->display( ).  
  
  ENDIF.  
  
  IF p\_chx2 IS NOT INITIAL.  
  
    SELECT workid FROM zworkposition36 WHERE position1 = @p\_posit INTO TABLE @DATA(lt\_employee3).  
  
    LOOP AT lt\_employee3 INTO DATA(ls\_employee3).  
      woid = ls\_employee3-workid.  
    ENDLOOP.  
  
    SELECT SUM( salary ) FROM zemployee36 WHERE workid = @woid INTO @sum\_result.  
  
    WRITE: 'SUM of salary for this position is: ', sum\_result.  
  
  ENDIF.  
  
  
  IF p\_chx3 IS NOT INITIAL.  
  
    SELECT a~id, a~name,  
      a~srname,  
      a~workid,  
      a~ctrcode,  
      a~salary,  
      a~lastchan,  
      b~position1,  
      b~defsal  
      FROM zemployee36 AS a  
      INNER JOIN zworkposition36 AS b  
      ON a~workid = b~workid  
      INTO TABLE @DATA(lt\_employee5).  
  
    TRY.  
        cl\_salv\_table=>factory( *"*  
          IMPORTING  
            r\_salv\_table   = go\_alv  
          CHANGING  
            t\_table        = lt\_employee5  
        ).  
      CATCH cx\_salv\_msg.  
    ENDTRY.  
  
    go\_alv->display( ).  
  
  ENDIF.

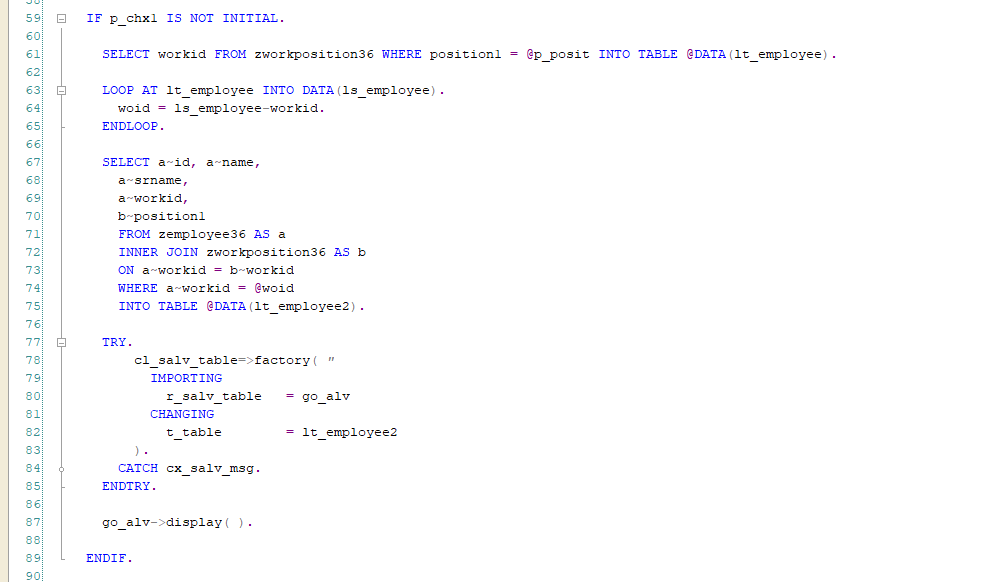
Tabela



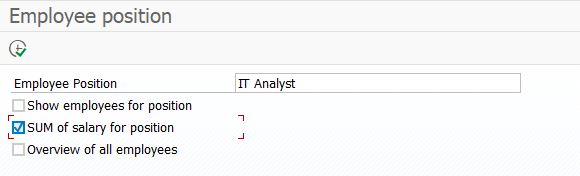
Wszystkich pracowników z zadanej pozycji – checkbox

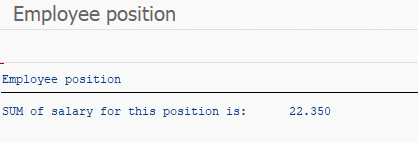


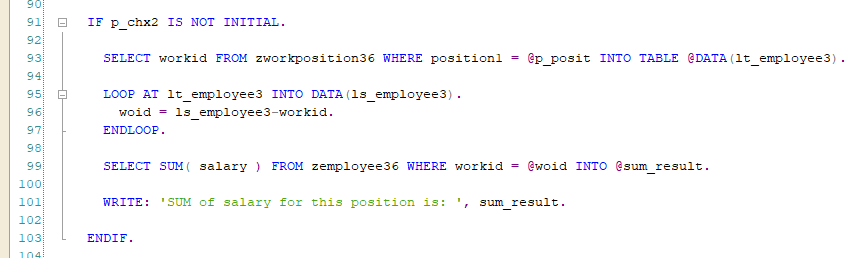




Ile sumarycznie jest płacone osobą na danym stanowisku – użyj keyworda SUM( ) w razie problemów dawaj znać







Overview wszystkich pracowników razem z ich pozycjami + płacami ( wszystkie pola z employee i workposition tabeli )

