

Halaby_pkg:

1- Function increase_emp_sal_check (Emp_ID , incr_flag , incr_value)

-> check if the increase amount|Perc is ok or not

Conds :

a- sal after increase should be between min_salary and Max_salary of job

b- Sal should not be greater than his direct manager Salary

return Y|N if return = 'Y' | if 'N' then this is not eligible increase

2- procedure increase_emp_sal (Emp_ID , incr_flag , incr_value)

if Function-1 return 'Y' then call the procedure

-> increase salary + record a log table

[Emp_id , tr_date , tr_user , salary_bfr , salary_aftr]

3- Function : is_number (p_number)

-> this Function to check if the input i a valid numbers

-> Return Value Y or N

EX : 12345 -> Y

12345x -> N

4- Procedure : dist_emp_sal_elm_pro(emp_id)

A- N.B. : you have to create the below setup table at first :

Table : emp_sal_elms_setup

[Sal_cat , elm_id , elm_desc , elm_sign , elm_prc_val]

B- insert the below values per Sal Cat:

('A' , 1 , income_tax , '-' , .20)

('A' , 2 , soc_ins , '-' , .09)

('A' , 3 , Med_ins , '-' , .05)

('A' , 4 , other_ded , '-' , .05)

('A' , 5 , mob_allw , '+' , .03)

('A' , 6 , trns_allw , '+' , .10)

('B' , 1 , income_tax , '-' , .22)

('B' , 2 , soc_ins , '-' , .10)

('B' , 3 , Med_ins , '-' , .08)

('B' , 4 , other_ded , '-' , .07)

('B' , 5 , mob_allw , '+' , .05)

('B' , 6 , trns_allw , '+' , .12)

('C' , 1 , income_tax , '-' , .25)

('C' , 2 , soc_ins , '-' , .10)

('C' , 3 , Med_ins , '-' , .07)

--('C' , 4 , other_ded , '-' , .05)

--('C' , 5 , mob_allw , '+' , .03)

('C' , 6 , trns_allw , '+' , .15)

('D' , 1 , income_tax , '-' , .27)

('D' , 2 , soc_ins , '-' , .14)

('D' , 3 , Med_ins , '-' , .07)

--('D' , 4 , other_ded , '-' , .05)

--('D' , 5 , mob_allw , '+' , .03)

--('D' , 6 , trns_allw , '+' , .10)

('D', 4, instnv_allw, '+', .25)

C- Transaction Table : dist_emp_sal_elms_trans

[Emp_id , Sal_cat , elm_id , elm_amt_val , trans_month {Month_last_day}]

Execution EX : dist_emp_sal_elm_pro(100)

(100 , 'D' , 1 , ??? {elm value from the current Salary} , EOM)

(100 , 'D' , 2 , ??? {elm value from the current Salary} , EOM)

(100 , 'D' , 3 , ??? {elm value from the current Salary} , EOM)

(100 , 'D' , 4 , ??? {elm value from the current Salary} , EOM)

D- Create a view to get :

1- emp_id | Emp_full_name | Sal_cat | Current_sal | Actual_sal_after_elm

5- Function : get_full_address (p_id , p_flag 'E|D')

-> if p_flag = 'E' -> get the Full_address of the employee

-> if p_flag = 'D' -> get the Full_address of the Department

6- Function : get_emp_count_by (p_value , p_ref)

-> get_emp_count_by (10|20|30 , 'D') -> emp count by department

-> get_emp_count_by ('IT_PROG'| 'SA_REP' , 'J') -> emp count by job

-> get_emp_count_by (100 | 120 | 122 , 'M') -> emp count by Manager

-> get_emp_count_by ('A'| 'B' | 'C' | 'D' , 'S') -> emp count by Sal_cat

-> get_emp_count_by ('US' | 'FR' , 'C') -> emp count by Country

7- Function : Get_sal_diff (Emp_id , 1|2|3)

-> to get the salary Diff between the employee and his managers LVL

EX: Get_sal_diff (199 , 1) his LVL 1 manager -> first Level

Get_sal_diff (199 , 2) his LVL 2 manager -> Second Level

Get_sal_diff (199 , 3) his LVL 3 manager -> third Level

N.B. : if there is no mentioned level then Raise a message

8-Function : is_magaer (emp_id , 'E|D')

-> to Check if the emp_id is a Manager or not

IF 'E' -> if he is a manager for any employee

IF 'D' -> if he is a manager for the department

9- Function : get_sal_weight(p_emp_id , 'D|J|M|C')

-> to get the salary weight for the employee vs D-> His Department

J-> His job

M->His Manager

C-> the Company

10- Function : get_dept_name(p_id , 'E|D')

-> to get department_name for E-> Employee | D-> Department

11- Function : Get_Manager_name(p_id , 'E|D')

-> to get Manager full name for E-> Employee | D-> Department

12- Function : My_to_char (p_Date , p_format_id)

A -> at first you have to create table with below structure

[Format_id , Date_format]

EX:

(1, 'dd-mm-yyyy')

(2 , 'mm-dd-yyyy')

(3, 'Mon, dd / yyyy')

.... etc -> please add more records

B-> create the function to give it the date and format to return the right date format as oer the setup

Execution EX : My_to_char (sysdate , 3)

13- function to return multi value based on emp_ID: get_top_sal (p_emp_id , PD , PJ , PL ,PS)

A -> to get the top salary for :

'PD' -> the top salary in the Emp Department

'PJ' -> the top salary in the Emp job

'PL' -> the top salary in the Emp Location

'PS' -> the top salary in the Emp Salary Cat

and return the top salary in the whole Company

B- Record the Above outputs in a table with below Structure :

Table Name : Emps_counts

[tr_number , tr_date , emp_id , comp_top_sal , emp_dept_top_Sal , emp_job_top_Sal ,
emp_loc_top_Sal , emp_cat_top_Sal]

**14- **Procedure : calc_premiums(p_itm , p_price , p_months ,p_dep, p_disc ,
p_mop[M|Q|S|Y])**

N.B. : MOP = Mode of Payment M-> Monthly | Q-> Quarterly | S-> Semi Annual | Y-> Annual

N.B. : p_months : 12 or 24 or 36 etc

N.B. : Create a tables to record the trans

N.B. : p_dep = deposit , if p_dep = 0 then there is no deposit Amount

N.B. : p_disc = Discount , if p_disc = 0 then there is no Discount Amount

before starting the Code you have to create the below 2 tables :

Table 1 : Prem_H

[itm_id , itm_price , prms_no[As per MOP] , dep_amount , disc_amount , mop , inv_no]

N.B. : inv_no = Invoice_number -> Auto Generated by Seq starting from 10000

Table 2: Prem_D

(Item_id , prm_no , prm_amt , prm_date , prm_flag[P|D|C])

N.B. : P-> Premium

D-> Deposit -> if 0 then no records in Prem_D

C-> Discount -> if 0 then no records in Prem_D

Execution EX: calc_premiums('iphone-16' , 70000 , 10 ,1000,2000, 'Q')

Prem_H :

('iphone-16' , 70000 , 10 ,1000,2000, 'M', 10000)

Prem_D :

('iphone-16' , 0 , 1000 , sysdate , 'D') -> this is the Deposit record

('iphone-16' , 0 , 2000 , sysdate , 'C') -> this is the Discount record

N.B. : After minus 1000 deposit and 2000 Discount the remaining amount is $70000 - 3000 = 67000$ so we will dist 67000 EGP

->>> how to Calculate the Premium ??

-> $67000 / 36 = 1,861.111$, the Payment should be in Quarter bases so the $1,861.111 * 3 = 5,583.333$

-> $67000 / 5,583.333 = 12$ Premium

('iphone-16' , 1 , 5,583.33 , next EOM , 'P') -> this is the First Premium

('iphone-16' , 2 , 5,583.33 , next EOM , 'P') -> this is the Month Premium

('iphone-16' , 3 , 5,583.33 , next EOM , 'P') -> this is the Month Premium

('iphone-16' , 4 , 5,583.33 , next EOM , 'P') -> this is the Month Premium

('iphone-16' , 5 , 5,583.33 , next EOM , 'P') -> this is the Month Premium

('iphone-16' , 6 , 5,583.33 , next EOM , 'P') -> this is the Month Premium

('iphone-16' , 7 , 5,583.33 , next EOM , 'P') -> this is the Month Premium

('iphone-16' , 8 , 5,583.33 , next EOM , 'P') -> this is the Month Premium

('iphone-16' , 9 , 5,583.33 , next EOM , 'P') -> this is the Month Premium

('iphone-16' , 10 , 5,583.33 , next EOM , 'P') ->this is the Month Premium

('iphone-16' , 11 , 5,583.33 , next EOM , 'P') ->this is the Month Premium

('iphone-16' , 12 , 5,583.33 , next EOM , 'P') ->this is the Last Premium

->>> the Calculation Logic should be aplicable for all Cases

15- ** Search Task :

i need a Procedure to Export any Table or view Data in a csv File

Procedure : Export_data (p_table_view_name)