

## 15.11 Lab for W7 FOR GROUP A

Consider the following entities:

1. Citizen < ID, Name, DOB >
2. Salary < CID, Dt, Amount >
3. AirportsLog < Dt, Tp (type) , PName (name of desti) >

Now write a function (or more) to satisfy the following requirements:

Input: Citizen ID.

Output: Status

Status Domain:<CIP, VIP, ORDINARY>

Algorithm for Status evaluation:

It is CIP if: 1) His average salary is above 100000 for the last 5 years. and 2)He has made a total of 10 departures in that period.

It is VIP if: 1) His average salary is between 50000 and 100000 for the last 5 years. and 2)He has made a total of 5 departures in that period.

Otherwise set his status ORDINARY.

*NOTE: Make the solution as modular as possible.*

## 15.12 Lab for W7 FOR GROUP B

Consider the following entities:

1. `Acc_type < ID, Name, IRate, GraceP >`

GraceP: 1= each 3 months, 2= each 6 months

2. `Customers < ID, Name, DOB, Address >`

3. `Accounts < AccNo, CID, Acc_type (FK), OpenDt, Current_Balance, LastDtIgr`

4. `Transactions < Dt, AccNo (FK), Tp (type: deposit or withdraw), Amount >`

**Task A:** Now write a function (or more) to satisfy the following requirements:

Input: Customer ID.

Output: Status

Status Domain: <CIP, VIP, ORDINARY>

Algorithm for Status evaluation:

If his current balance is more than 100000 then his status is VIP.

If his current balance is between 40000 and 100000 then his status is IMPORTANT.

Otherwise ORDINARY.

**Task B:** Write a subprogram to update a customer's current based on:

If today is between 1 and 3 months comparing with LastDtIgr then calculate his interest

based on the corresponding table (i.e. Acc\_type).

If today is between 3 and 6 months comparing with LastDtIgr then calculate his interest based on the corresponding table (i.e. Acc\_type)

## 15.13 Lab for W7 FOR GROUP C

Consider the following entities:

1. `Acc_type < ID, Name, IRate, GraceP >`

GraceP: 1= each 3 months, 2= each 6 months

2. `Customers < ID, Name, DOB, Address >`

3. `Defaulters < CID (FK), Dt (of entry), Tp (Major or minor) >`

3. `Accounts < AccNo, CID, Acc_type (FK), OpenDt, Current_Balance, LastDtIg`

4. `Transactions < Dt, AccNo (FK), Tp (type: deposit or withdraw), Amount >`

**Task A:** Now write a function (or more) to satisfy the following requirements:

Input: Customer ID.

Output: Status

Status Domain: `< CIP, VIP, ORDINARY >`

Algorithm for Status evaluation:

1. **VIP:** If his current balance is more than 200000 and he has no allegation at all (major or minor) then his status is VIP.
2. **IMPORTANT:** If his current balance is between 40000 and 200000 and he has no major allegation (i.e. Defaulters Table) then his status is IMPORTANT.
3. **ORDINARY:** If his current balance is between 10000 and 40000 and he has only one major allegation (i.e. Defaulters Table) then his status is ORDINARY.
4. **CRIMINAL:** If (his current balance is irrelevant) he has 3 or more major allegations and 5 or more minor allegations (i.e. Defaulters Table) then his status is CRIMINAL.

**Task B:** Write a subprogram to update a customer's current balance based on:

If today is between 1 and 3 months comparing with LastDtIg then calculate his interest

based on the corresponding table (i.e. Acc\_type).

If today is between 3 and 6 months comparing with LastDtIg then calculate his interest based on the corresponding table (i.e. Acc\_type)

## 15.17 Lab for W9 FOR GROUP A,B,C

Consider the following scenario:

A bank issues loan to its customers. There are three loan schemes: i) S-A ii) S-B and iii) S-C. They are defined as follows:

Scheme	No. of Installment	Service Charge for remaining loan	Eligibility
S-A	30	5%	Total Transaction in the last 12 months $\geq$ 2000000
S-B	20	10%	Total Transaction in the last 12 months $\geq$ 1000000
S-C	15	15%	Total Transaction in the last 12 months $\geq$ 500000

Your tasks are as follows:

1. Design the table definitions and issue the required DDLs. Additional assumptions are welcome in design phase.
2. Write a function to assign a customer to a specific category of loans as mentioned.
3. Once a customer is assigned to a specific loan scheme, write a procedure to schedule his loan. Assume each loan must be paid after 3 months interval.

## 15.18 Lab for W10 for Group A

Based on Explicit cursor.

Consider the following entities for automation of a banking system:

1. AccountProperties < ID (pk), Name, iRATE, GP (grace period) >

GP=1 for daily

GP=2 for monthly

GP=3 for quarterly

GP=4 for biyearly

GP=5 for yearly

2. Accounts < ID (pk), Name, AccCode (fk ref to AccountsProperties), Balance, LastDateInterest (date on which last interest was given) >

Insert some values in AccountProperties and Accounts (but maintain the reference).

Your tasks are :

1. Write a function to calculate the profit based on profitRate, amount and duration.
2. Write a procedure to update all accounts current balance (i.e. interest will be calculated and added if it satisfies conditions). Use the *traditional cursor loop* for this problem.

## 15.19 Lab for W10 for Group B

Based on Explicit cursor

Consider the following entities for automation of a banking system:

1. AccountProperties < ID (pk), Name, iRATE, GP (grace period) >

GP=1 for daily

GP=2 for monthly

GP=3 for quarterly

GP=4 for biyearly

GP=5 for yearly

2. Accounts < ID (pk), Name, AccCode (fk ref to AccountsProperties), Balance, LastDateInterest (date on which last interest was given) >

Insert some values in AccountProperties and Accounts (but maintain the reference).  
Your tasks are :

1. Write a function to calculate the profit based on irate, amount and duration. Remember the duration could be in number of *Days* or *Months* or *Year*. Your program should handle it smoothly.
2. Write a procedure to update all accounts current balance (i.e. profit will be calculated and added if it satisfies conditions). Use the *cursor for loop* for this problem.

## 15.20 Lab for W10 for Group C

Based on Explicit cursor

Consider the following entities for automation of a banking system:

1. AccountProperties < ID (pk), Name, profitRate, GP (grace period) >

GP=1 for daily

GP=2 for monthly

GP=3 for quarterly

GP=4 for biyearly

GP=5 for yearly

2. Accounts < ID (pk), Name, AccCode (fk ref to AccountsProperties),  
LastDateInterest (date on which last interest was given) >

3. Transactions < TID (pk), accNo (fk ref to Accounts), Amount, Transctio  
dt (date of trans.) >

4. Amounts < AccNo, Pricipalamount, ProfitAmount, dt>

Insert some values in AccountProperties and Accounts (but maintain the reference).  
Your tasks are :

1. Write a function to calculate the profit based on profitRate, amount and duration.
2. You have to write another function to calculate current balance from the transacitons.
3. Write a procedure to calculate all accounts' profit (i.e. profit will be calculated if it satisfies conditions). Use the *cursor for loop* for this problem. The procedure will insert appropriate record in its Ammounts table.

## 15.21 Lab for W11 FOR GROUP A

Consider the main problem for your group as mentioned in 15.3. In this lab you are asked to generate the ID of each citizen. ID generation Algorithm is given below:

**ID Formate:** *DVYYYYDDMM.XXXXXXX*

**Algorithm:** DV is the division code of the citizen was born. YYYYDDMM is the numeric representation of his Date of Birth. And XXXXXXXX is the serial no (in increasing order).

Your task is to write a function to generate the ID as described. It takes two input parameters: DOB and DV code and returns the ID.

## 15.22 Lab for W11 FOR GROUP B

Consider the main problem for your group as mentioned in 15.4. In this lab you are asked to generate the Account No in a specified format given below:

**ACC No Format:** *TTBBBYYYYMMDD.XXXXXX*

where T=account type

B=Branch Code

X=Serial No

YMD=Year Month Day of DOB of the citizen.

Your tasks are:

- Create DDLs as needed to fulfill the requirements.
- Write a function that takes 3 IN parameters such as Branch Code, Acc Type, Customer ID to fetch its DOB. And it returns a number as the new Account No.



## 15.23 Lab for W11 FOR GROUP C

Consider the main problem for your group as mentioned in 15.5. In this lab you are asked to generate the Student ID in a specified format given below:

**Student ID format: YYDPXX**

where, Y=Year of enrollment.

D=Dept Code

P=Program Code

X=Serial No.

Your tasks are:

- Create DDLs as needed to fulfill the requirements.
- Write a function that takes 3 IN parameters such as Date of enrollment, Department Code and Program Code. And it returns a number as the new Student ID.

## 15.24 Lab for W12 FOR GROUP A

Refer to problem in 15.3 for Country Database Management. Now we concentrate on Government Employees. Salary of an employee is defined as :

*Total Monthly Salary=Basic + Housing. 40% + Transport. 10%*

There are 3 different scales defined by Government:

*S1, S2 and S3* each has only one value: Basic amount.

Also add designation to emp.

**Tasks A:** Design a monthly salary transaction (format is : *TPYYYYDDMMHHMISS*). Note partial salary (all basic, housing and transport) must be calculated if one joins in the middle of the month.

**Part B:**

Now you need to design the Tax Module in connection with bank account. Tax offices maintain the following rules for tax collections:

1. If Total Salary Paid (TSP) at the end of a year is  $\leq 100000$  then no tax is applied. (It should consult your salary monthly transaction for it.)
2. If TSP is between 100000 and 400000 then 5% tax is applied.
3. If TSP is between 400000 and 1000000 then 10% tax is applied.
4. Else 20% tax is applied.

## 15.25 Lab for W12 FOR GROUP B

Refer to problem in 15.4. Here you need to work on 3 types of accounts:

1. Single Account: One person one account.
2. Joint Account: More than one persons having one account.
3. Business Account: (opt) Account will be opened for a specific organization but will be operated by one or more persons of that organization.

### Part A:

Your tasks are as follows:

- You need to design ER for the daily transactions with the following two restrictions:
  1. Max amount per transaction  $\leq 100000$ .
  2. Max amount per month against an account number  $\leq 500000$ .
- The format of Daily Transaction is *TPYYYYDDMMHHMISS*, date implies date of opening account. TP is the type of account.

### Part B:

Now you need to design the Tax Module in connection with bank account. Tax offices maintain the following rules for tax collections:

1. If Total Balance (TB) at the end of a year is  $\leq 100000$  then no tax is applied. (It should consult your daily bank transaction considering both credit and debit.)
2. If TB is between 100000 and 400000 then 5% tax is applied.
3. If TB is between 400000 and 1000000 then 10% tax is applied.
4. Else 20% tax is applied.

## 15.26 Lab for W12 FOR GROUP C

Refer to problem in 15.5. Account section will be added here for the salary management of its employees. University maintains 3 different scales as follows:

*S1, S2 and S3*

1. S1: In S1 total salary is calculated as:

*Total Monthly Salary=Basic + Housing. 10% of Basic + Transport. 15% of Basic*

2. S2: In S2 total salary is calculated as:

*Total Monthly Salary=Basic + Housing. 20% of Basic + Transport. 10% of Basic*

3. S3: In S3 total salary is calculated as:

*Total Monthly Salary=Basic + Housing. 30% of Basic + Transport. 5% of Basic*

### Task A:

- Create a function to calculate the salary of any employee. *Note* partial salary (all basic, housing and transport) must be calculated if one joins in the middle of the month.
- Design a monthly salary transaction (format is : YYYYDDMMHHMISS.NNN) with-  
ing the row-level trigger.

### Part B:

Now you need to design the Tax Module in connection with bank account. Tax offices maintain the following rules for tax collections:

1. If Total Salary Paid (TSP) at the end of a year is  $\leq 100000$  then no tax is applied.
2. If TSP is between 100000 and 400000 then 5% tax is applied.
3. If TSP is between 400000 and 1000000 then 10% tax is applied.
4. Else 20% tax is applied.

*You are required to create necessary functions to calculate the total tax payable according to the above rules.*