

# E-GENTING PROGRAMMING COMPETITION, 2004

## WORKSHOP HANDOUT, WEEK 1, VERSION 1

### 1 BACKGROUND

#### 1.1 Formalities

```
exec sql include sqlca;
//...
int main ()
{
    //...
    exec sql connect to 'custDb';
    //...
}
```

#### 1.2 Singleton Select

```
exec sql begin declare section;
    char nameVar[41];    // Customer name variable
    long idVar;         // Customer id variable
exec sql end declare section;
// ...

idVar = 100127;
exec sql  select      custName
           into        :nameVar
           from        custFile
           where       custId = :idVar;
if (SQLCODE != 0)
    { /* process row not found error*/ }
cout << "Customer name is " << nameVar << '\n';
```

## 1.3 Cursor Select

```
exec sql declare custCur cursor for
      select      custId, custName
      from        custFile;
exec sql open custCur;
for (;;) {
    exec sql fetch custCur
          into :idVar, :nameVar;
    if (SQLCODE != 0) break;
    cout << idVar << " " << nameVar << '\n';
}
exec sql close custCur;
```

## 2 EXERCISES

### 2.1 A Trivial Problem

Write an embedded SQL program that lists customer identifier, name and total value of the customer's purchases from a database with the following schema.

```
// Customer file

create table custFile (
    custId    integer not null,    // Customer identifier
    custName  char(40) not null    // Customer name
);

// Purchases file

create table purFile (
    purId     integer not null,    // Customer identifier
    purNo     integer not null,    // Purchase number
    purVal    integer not null     // Purchase value
);
```

The purchase value is stored in cents, but must be displayed in dollars and cents with a decimal point. Each customer may have made zero or more purchases.

The database name is 'custDb'.

### 2.2 Microbank without Time Constraints

If you are able to solve the trivial problem before the end of the workshop, attempt to solve the Microbank problem from last year's competition, disregarding the execution time constraints.

This is the Microbank problem (minus the execution time constraints):

The government of a struggling third-world economy has established a bank, known as the 'Microbank' to provide relatively small loans to would-be entrepreneurs to assist them to set up new businesses. So far, around a million of these micro-loans have been made. Most of the loans are being serviced (i.e. interest, such as it is, is being collected and/or the capital of the loan is being repaid), but a good proportion of the loans are non-performing (i.e. not being serviced).

Before a loan was approved, two authorisations were required. First, an executive of the Microbank had to recommend a loan be made to a customer. Second, another executive had to approve the granting of the loan.

The Microbank now suspects that some of its executives might have been recommending and approving significant numbers of loans to their friends and relatives without taking any reasonable steps to ensure that the borrowers had either the capacity to service the loans or intention of doing so. The Microbank would like to know the number and value of non-performing loans for each executive, irrespective of whether the executive recommended or approved the loan.

Specifically, it requires a report with the following contents:

1. for each executive:
  - a. executive code (6 digits),
  - b. executive name (20 characters),
  - c. number of loans recommended or approved by the executive,
  - d. dollar value of loans recommended or approved by the executive,
  - e. number of non-performing loans recommended or approved by the executive,
  - f. dollar value of non-performing loans recommended or approved by the executive;
2. total number of loans issued,
3. total value of loans issued,
4. total number of non-performing loans,
5. total value of non-performing loans.

The totals cannot be a simple sum of the corresponding values for each executive because each loan will usually be associated with two executives. If the values for each executive were simply totalled up, they would be approximately twice the required total. They will not be exactly twice because some executives are no longer with the bank and the records for those executives have been deleted.

The data for the report is stored on an SQL database. The database is called 'microDb'. The relevant parts of the schema are:

```
create table executives (  
    execCode integer not null,  
    execName char(20) not null,  
);
```

```

create table loans (
    loanId          integer not null,
    loanCust        char(20) not null,
    loanAmount      double precision not null,
    loanStatus      smallint not null
);
create table approvals (
    appLoanId       integer not null,
    appExecCode     integer not null,
    appType         smallint not null
);
create index appLoanIdInd on approvals(appLoanId);

```

execCode

is the six-digit executive code.

execName

is the executive's name.

loanId

is an 8-digit loan identifier. Each loan has a unique loan identifier.

loanCust

is the customer's name.

loanAmount

is the loan amount in cents. I.e. a \$100.00 loan would have  
loanAmount=10000.0.

loanStatus

is 1 if the loan is being serviced, 0 if the loan is not being serviced. Values other  
than 1 and 0 are invalid.

appLoanId

is the loan identifier to which the approval relates. It is in the same domain as the  
loans.loanId column.

appExecCode

is the executive code of the Microbank executive who recommended or approved  
the loan. It is in the same domain as the executives.execCode column.

appType

is 0 for a recommendation or 1 for an approval. Values other than 0 and 1 are  
invalid.

Your task is to write the reporting program so that it satisfies all the above requirements.