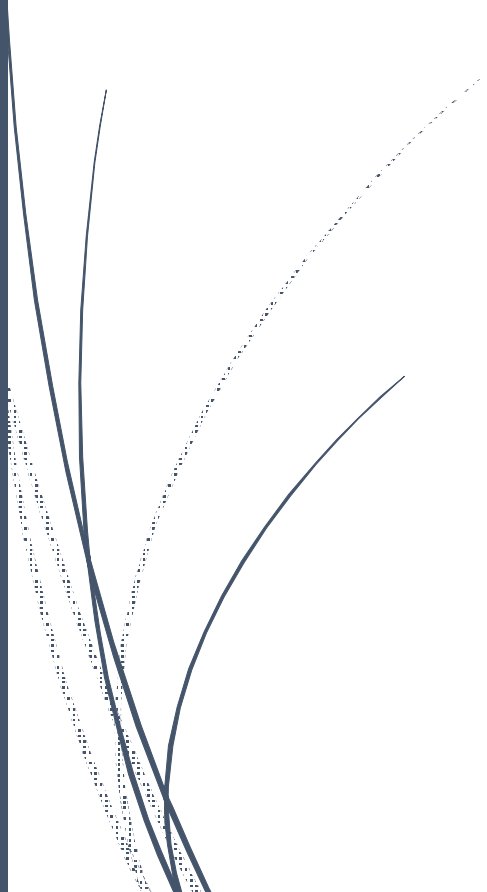
6/6/2014

Assignment # 2

SMART CARD FINANCIAL SETTLEMENT SYSTEM REPORT



Asad ur Rehman
11528015

Contents

Introduction

This programme has been developed to replace the age old manual transaction settlement system. It will automate the transaction settlement process and will produce various reports to keep monitoring staff updated.

Description of task

- This programme is run once every day to settle new transactions
- During each run, programme produces, deskbank file (computer readable file), daily settlement report (human readable file) and fraud report.
- Each produced file is then analyzed by their respective recipients.
- Both deskbank file and daily settlement report contain same meaning; deskbank file is for computer to process settled transactions and daily settlement report is for bank people for monitoring purposes.
- As far as the Fraud report is concerned, it contains all those transactions which were caught as fraudulent transactions during settlement process.
- Each fraudulent transactions is settled and is informed security people about via fraudulent report generated during each settlement run.
- Total settlement amounts that are less than the designated minimum amount are not be settled in the daily transactions until the total settlement amount reaches the minimum settlement amount. At the end of the month,

any transaction that has not been processed during the month is finalized, irrespective of the amount.

Instructions

- It is worth mentioning at this stage that due to some technical difficulties, programme is not able to email daily settlement report. Produced daily settlement report can be found in directory called MY_DIR which points to the location '/oralab/loz' in oracle database.
- Due to the insufficient privileges, files placed in this directory can only be read.
- In order to manipulate files, database administrator can be requested to grant extra privileges.
- DailySettlement procedure which is placed in package Pkg_FSS_Settlement should be executed using statement 'EXEC Pkg_FSS_Settlement.DailySettlement' in any integrated development environment such as oracle SQL developer
- During the execution of this statement, FSS_DAILY_TRANSACTIONS table will be populated with new transactions and Deskbank file, Daily settlement report and Fraud report, using their respective naming conventions suggested, will be created and placed in MY_DIR.

- Daily settlement report for any given date can be generated using DailyBankingSummary(date) placed in package Pkg_FSS_Settlement . It is important that the input date should only be in the format: DD-Mon-YYYY.

- Fraud report can be generated any time using the procedure FraudReport placed in package Pkg_FSS_Settlement.

Technical design of programme

All the functions, procedures and packages and tables that make up this entire programme are given below:

Packages

1. Pkg_FSS_Settlement

Functions

1. F_CENTRE
2. SETTLEMENT_DONE_TODAY
3. SETTLING_TRANSACTIONS

Procedures

PUBLIC	PRIVATE
DAILYBANKINGSUMMARY	ASSIGN_LODGE_REF_NUM
DAILYSETTLEMENT	CHECK_CURRENT_DATE
FRAUDREPORT	CREATE_DESKBANK_FILE
COMMON.LOG	DOWNLOAD_NEW_DATA
-	MARK_FRAUDULENT_TRANSACTIONS

Tables

1. FSS_DAILY_TRANSACTIONS
2. FSS_TRANSACTIONS
3. FSS_DAILY_SETTLEMENT
4. PARAMETER
5. FSS_RUN_TABLE

All the functions, tables, packages and procedures are discussed below:

Package

PKG_FSS_SETTLEMENT

- This package contains all the private and public functions and procedures that are used to accomplish the various objectives.

Functions

F_CENTRE

- The return type of this function is VARCHAR2.
- Function takes a parameter of type VARCHAR2.
- Functions uses LPAD function to place the input string in the centre of the page.
- Total width of the page is considered to be 80 units.
- Finally, function returns the input string which will be displaced in the centre of the page.

SETTLEMENT_DONE_TODAY

- The return type of this function is Boolean.
- Function checks whether the settlement has been done on current system date or not.
- A RUNSTART column of FSS_RUN_TABLE is searched for the current system date. For the search, time component of date is not considered.
- If no_data_found exception is raised, v_date variable is set to null
- Finally, function returns true if v_date is not NULL else returns false

SETTLING_TRANSACTIONS

- The return type of this function is Boolean.
- Functions checks if the programme is still settling transactions.
- A RUNEND column of FSS_RUN_TABLE is searched for the record where RUNSTART column is not null and RUNEND column is null. For the search, time component of date is not considered.
- If NO_DATA_FOUND error pops up, it is dealt with in exception part
- In exception part, v_date variable is set to NULL
- Finally, function returns true if v_date is not NULL else returns false.

Procedures

Public:

DAILYBANKINGSUMMARY

- Procedure accepts date as a parameter.
- Input date is required to follow DD-Mon-YYYY format.
- Cursor is used to fetch the records whose date on which those records were settled is equal to the date passed as a parameter. For the search, time component of date is not considered.
- SQL query used for cursor joins tables FSS_DAILY_SETTLEMENT and FSS_MERCHANT to obtain the required set of columns
- Procedure then, set a file pointer to a file whose name follows the format DailyBankingSummary(date) where date is the date for which banking report is requested. This file is placed in MY_DIR directory.
- After writing all the required information from records fetched by cursor, information about organization who bank account will be used to pay merchants is fetched in the variable with type %ROWTYPE.
- All the required information is then extracted from the record stored in variable and is written into a file.

DAILYSETTLEMENT

- Function SETTLEMENT_DONE_TODAY to check whether settlement has been done on the current system date or not.
- If settlement is already done, programme terminates and reason is logged using the COMMON.LOG procedure.
- If settlement has not been done on the current system date, programme checks whether programme is still settling transactions or not.
- If settling transactions, programme exits reason is logged using the COMMON.LOG procedure.
- Next, procedure DOWNLOAD_NEW_DATA is used to insert the new transactions from FSS_TRANSACTIONS table into FSS_DAILY_TRANSACTIONS table.
- RUNSTART column of FSS_RUN_TABLE is updated with the current system date. Time component of the date is also included but not showed in the table.
- Then, procedure MARK_FRAUDULENT_TRANSACTIONS is used to find all the fraudulent transactions.
- Procedure CHECK_CURRENT_DATE is used to determine whether current system date is the last date of the current month or not.

- Then, tables FSS_DAILY_TRANSACTIONS, FSS_TERMINAL, FSS_MERCHANT are joined and the records where TRANSACTIONSTATUS is null (transactions not settled) are selected and grouped based on the MERCHANTID.
- Resulted set of records are inserted into FSS_DAILY_SETTLEMENT.
- Procedure ASSIGN_LODGE_REF_NUM is executed to assign the unique lodgment reference number is assigned to each record in FSS_DAILY_SETTLEMENT
- RUNEND column of the FSS_RUN_TABLE is updated with the current system date. Time component of the date is also included but not showed in the table.
- Daily banking summary and desk bank file are generated and placed in MY_DIR using procedures DAILYBANKINGSUMMARY and CREATE_DESKBANK_FILE.

FRAUDREPORT

- Cursor is used the fetch the records from FSS_DAILY_TRANSACTIONS table where ERRORCODE column reads FRAUDULENT.

- Procedure then set a file pointer to a file whose name follows the format Fraud_Report(date).AUR where date is the date on which fraud report is produced. This file is placed in MY_DIR directory.
- All the relevant information extracted from the records fetched by the cursor is written in file.

COMMON.LOG

- Procedure accepts string as a parameter
- Procedure writes the input string in the log table

Private:

ASSIGN_LODGE_REF_NUM

- Cursor is used to fetch records from FSS_DAILY_SETTLEMENT who have not been assigned.
- Each lodgment reference number generated is the concatenation of current system date (time component not included) and the number generated by the LODGE_SEQ sequence.

CHECK_CURRENT_DATE

- Procedure checks whether the current system date is the last date of the current month or not.

- If it is the last date, column TRANSACTIONSTATUS of table FSS_DAILY_SETTLEMENT is set to null where TRANSACTIONSTATUS equals P (pending).
- If it is not the last date, column TRANSACTIONSTATUS of table FSS_DAILY_SETTLEMENT is set to P (pending) where TRANSACTIONNR is determined by the sub-query.
- Query presented below has three sub-queries. Innermost query (marked as 1), gives out the minimum allowed amount for transaction to be settled, query marked as 2, groups the transactions based on MERCHANTID attribute and then gives out only those MERCHANTID whose total credit amount is less than produced by the query marked as 1.
- Query marked as 3, gives out TRANSACTIONNR which correspond to the MERCHANTID produced by query marked as 2.

```
update fss_daily_transactions
set TRANSACTIONSTATUS = 'P'
where TRANSACTIONNR in (select tr.transactionnr
                        from fss_transactions tr join fss_terminal te on TR.TERMINALID = TE.TERMINALID
                        join fss_merchant m on M.MERCHANTID = TE.MERCHANTID
                        where m.merchantid in (select m.merchantid
                                               from fss_transactions tr join fss_terminal te on TR.TERMINALID = TE.TERMINALID
                                               join fss_merchant m on M.MERCHANTID = TE.MERCHANTID
                                               group by M.MERCHANTID
                                               having sum(tr.transactionamount)/100 < (select referencevalue
                                                                                       from fss_reference
                                                                                       where referencename = 'Daily Minimum Settlement')));
```

3

2

1

CREATE_DESKBANK_FILE

- Cursor is used to fetch the records where TRANSACTIOSTATUS is null.
- SQL query used by cursor joins the tables FSS_DAILY_SETTLEMENT and FSS_MERCHANT to extract the required information.
- Cursor for loop is used to write every transaction into a file.
- After each transaction is written into a file, current transaction's TRANSACTIONSTATUS column is set to S (settled).
- After writing all the required information from records fetched by cursor, information about organization who bank account will be used to pay merchants is fetched in the variable with type %ROWTYPE.
- All the required information is then extracted from the record stored in variable and is written into a file.

DOWNLOAD_NEW_DATA

- Procedure loads the set of transactions in the FSS_DAILY-SETTLEMENT table from FSS_TRANSACTIONS
- SQL MINUS function has been used to find only those transactions that are not present in the FSS_DAILY_TRANSACTIONS but in FSS_TRANSACTIONS.

MARK_FRAUDULENT_TRANSACTIONS

- Cursor is used to fetch all the transactions from the FSS_DAILY_TRANSACTIONS.
- All the records are ordered by CARDID and TRANSACTIONNR.
- Purpose of ordering this way allows the programme to check all the transactions for a particular CADRDID before start checking for the other CARDID.
- Cursor for loop is used to process all the transactions.
- Since all the transactions are ordered by the TRANSACTIONNR as well, difference in the CARDNEWVALUE of previous transaction and the CARDOLDVALUE of the current transaction will mean that that some fraud has been committed.
- Transactions that matches this criteria, will be marked as Fraudulent.
- ERRORCODE column of the fraudulent transaction is updated with the string Fraudulent.

Tables

FSS_DAILY_TRANSACTIONS

- This table stores the transactions that need to be settled
- The columns in this table are as follows:

  - TRANSACTIONNR
  - DOWNLOADDATE
  - TERMINALID
  - CARDID
  - TRANSACTIONDATE
  - CARDOLDVALUE
  - TRANSACTIONAMOUNT
  - CARDNEWVALUE
  - TRANSACTIONSTATUS
  - ERRORCODE

PARAMETER

- This table along with other information stores the email addresses of the recipients of daily banking summary.

FSS_TRANSACTION

- This table has the same columns as those in FSS_DAILY_TRANSACTIONS

RUN_TABLE

- This table used to ensure that the settlement process is not run more than once a day.

- Columns in this table are as follows:

  - RUNSTART
  - RUNEND
  - RUNOUTCOME
  - REMARKS

- RUNSTART contains the date on which the settlement process started.

- RUNEND contains the date on which the settlement process ended.
- RUNOUTCOME contains the outcome of the settlement process: was settlement process successfully completed?

ERD DIAGRAM REPRESENTING THE DATABASE STRUCTURE