



MACQUARIE
University
SYDNEY • AUSTRALIA

DEPARTMENT OF COMPUTING
ISYS224/ITEC624 2018 – ASSIGNMENT 2
(weight: 20%)

Due: 11:55pm Friday, 2nd November 2018

The background knowledge for the assignments is given in the textbook(s), lectures, any other components of the unit, and in the readings provided on iLearn. However, some parts of the assignments may not be answered without prior independent research and searching for other sources of information.

Case Background

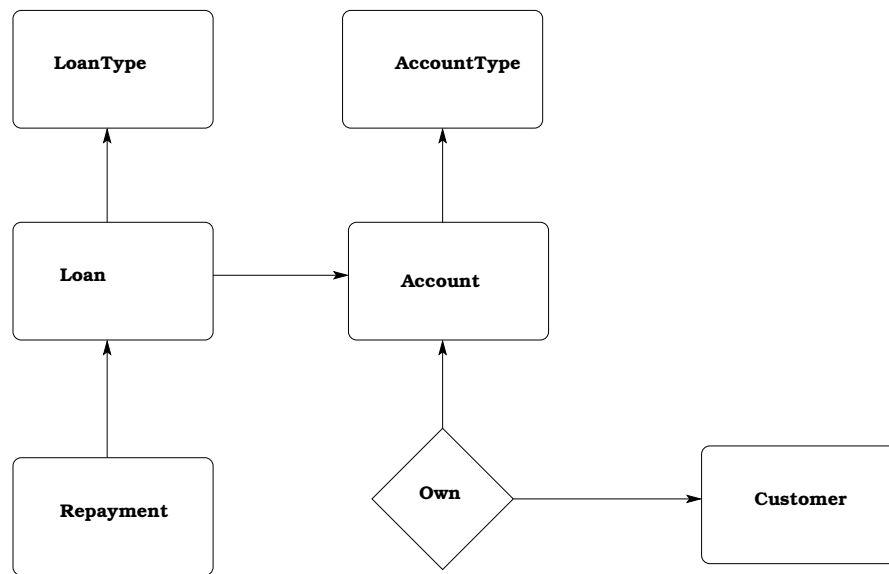
The general background is same as for Assignment 1. However the focus recently has shifted to the loans aspect of the Spock Bank because of the banking royal commission. For your convenience, the following information is provided as part of the domain description:

1. Relevant business rules (*slightly altered*)
2. A *simplified* entity-relationship data model for the problem domain. In this, $X \rightarrow Y$ represents a many-one relationship between object X and object Y.
3. A corresponding relational model/schema is also provided.

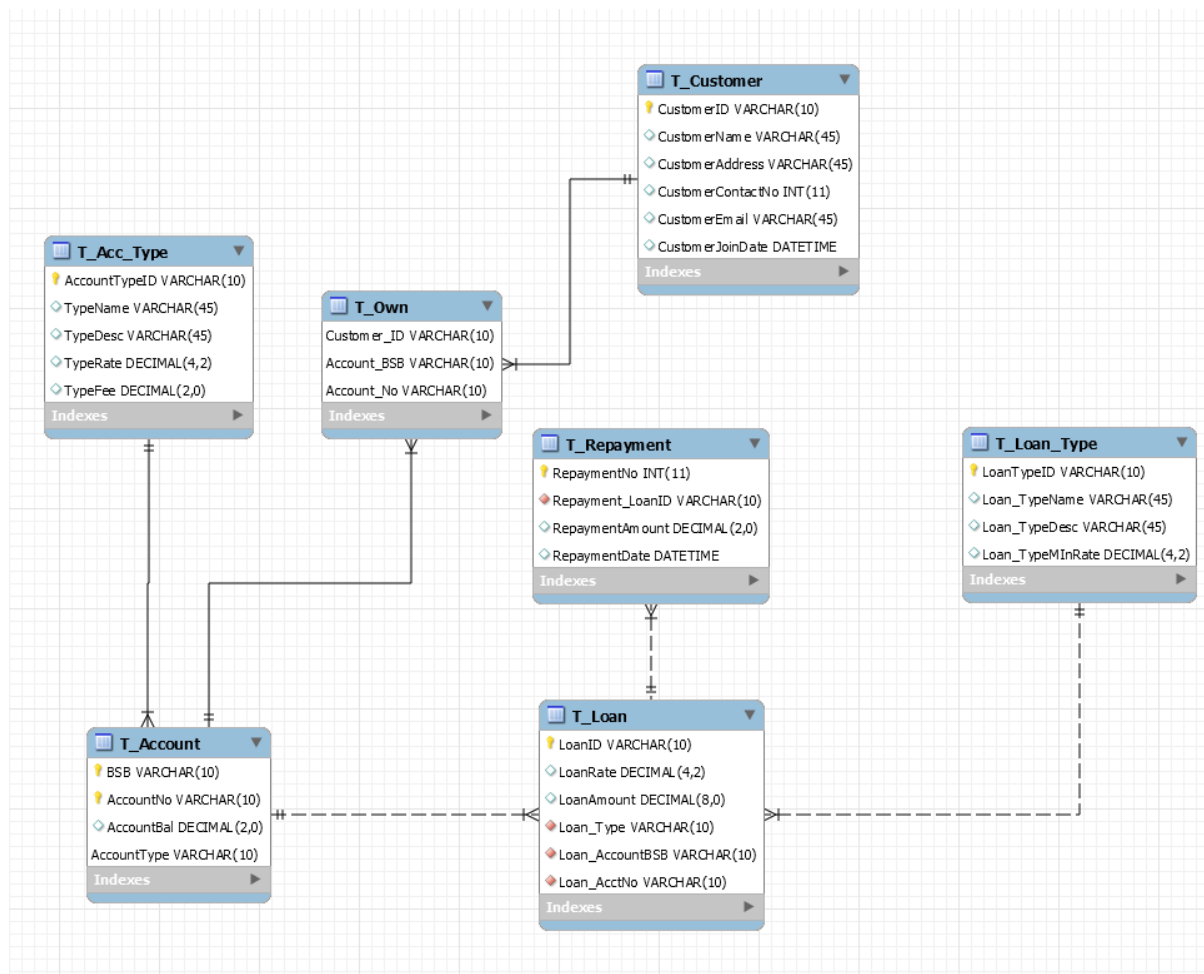
Business rules applicable to loans at Spock Bank:

- Each customer is provided with a unique identifier. Details of a customer include: name, address, contact number, email address, and join date.
- An account can be one of three types: SAV (savings), CHK (Checking) or LON (Loan). Each type of account has a unique identifier. Each account type needs to include: name (e.g., SAV/ CHK/ LON), a description, interest rate that customers receive on deposits, and a yearly service fee. No interest is paid for any money left in Loan accounts.
- An account is identified by a combination of BSB and an account number. Details of account such as type of the account and current balance are stored.
- A loan can be one of three types: HL (Home Loan), IL (Investment Loan) and PL (Personal Loan). Each type of loan has a unique identifier. Each loan type needs to include: name (e.g., HL/ IL/ PL) and description
- A loan is identified by a unique identifier. Details of loan such as interest rate customers pay and total (original, not outstanding) loan amount are stored.
- Every loan repayment is identified by a repayment identifier. Other details of the repayment would include: the loan that the repayment is for, repayment amount, and date and time of repayment.
- An account type may have multiple accounts. An account will be of only one particular type.
- A customer may have multiple accounts. An account could be shared by multiple customers (joint account).
- A loan type may have multiple loans. A loan will be of only one particular type.
- Every loan is associated with an account. An account may not be linked with a loan.
- A repayment is done for a particular loan. A loan may have multiple repayments.

Relevant Entity-Relationship (E-R) Diagram



Relational Schema (in a diagrammatic form)



Tasks

Task 1: Table Generation and Population

(1 mark)

Generate seven tables and populate them using the script provided in the assignment folder. You can optionally create and populate them by yourself, but make sure there is strong alignment between the tables created and rows populated by the script and those by you. In particular, look carefully at how the primary key of the table for repayment (T_Repayment) is automatically generated, and how rows are inserted to it.

Task 2: Procedure for Loan Repayment

(5 marks)

At Spock Bank, loan repayments are made purely online by transferring fund from another account – typically a savings account -- to the loan account. The customers normally use the bank's app or webpage and specify from which account they would want to make the repayment, and towards which loan this payment should count. This action of the customer will effectively call a stored procedure as:

```
Repay_loan(from_BSB, from_accountNo, to_loan, amount)
```

with relevant parameters, and make appropriate changes to the relevant tables.

This procedure is intended to transfer the amount of money in question from the account identified by (from_BSB, from_accountNo) to the loan account associated with the to_loan. The procedure must comply with the following business rules:

1. The repayment towards a loan must be made from an account such that at least one of the borrower individually (or jointly with some other entities) holds the account from which the money is being drawn.
2. There must be adequate amount of money in the relevant account for a repayment to be made – if there is not enough money, the repayment transaction cannot be completed.
3. If the repayment goes through,
 - a) the balance is accordingly reduced in the account identified by (from_BSB, from_accountNo), and
 - b) an appropriate new record is inserted into the repayment account.
4. If for some reason the repayment transaction could not be completed, the failure is gracefully handled, and an appropriate message is displayed.

For this task, you need to do the following:

- a) **Write the procedure** Repay_loan
- b) **Test the procedure** Repay_loan. Your test should include:
 - i. one case when the repayment is successful,
 - ii. one case where it fails because someone is trying to pay off someone else's loan, and
 - iii. one case where there is not enough money to make the repayment in the relevant account.

Task 3: Database Trigger

(5 marks)

Triggers are used to implement complex business rules. Spoc Bank has some rules that have not been outlined earlier. One of them concerns the number of loans customers can have. The rules are:

- a) A customer cannot individually have more than five loans.
- b) The maximum number of loans that a customer can have, including both individual and joint loans, is eight.
- c) A customer cannot have more than one personal loan.
- d) Total amount of all the loans that an individual is involved in cannot exceed 10 million dollars. This calculation takes into account the original loan amounts, not the remaining loan balances.
- e) An individual cannot be involved in more than three home loans.

You have decided to implement these rules by writing a trigger.

- (i) **Explain** what sort of trigger you would use for this purpose, and why.
- (ii) **Discuss** how you will test this trigger.
- (iii) **write and adequately test this trigger.**

Task 4: Procedure for Interest Calculation

(4 marks)

At Spock Bank, on the 25th of every month, interest is calculated on every loan and is added to the corresponding loan account. Interest calculation is done on a “daily basis” in the sense that it takes into account how many days are there in that month, and interest is calculated on a pro rata basis. Furthermore, the remaining loan amount left to be paid, as reflected in the corresponding loan account, could vary day to day depending on the repayments made. Also note that, often a loan has an associated special savings account called an *offset* account. The balance in the savings account offsets the loan in the sense that in calculating the interest on the loan, interest is calculated only on the difference between the remaining loan amount left to be paid and the amount in the offset account. For instance, if the remaining loan amount is \$5,000 on a given day, and there is \$1,000 in the offset account on that day, interest is charged only on \$4,000 (i.e. \$5,000 - \$1,000) for that day.

- (1) Can the given relational schema (without any modification) support the implementation this procedure? If not, why not? If yes, how? **Discuss.**
- (2) If this schema, as it stands, will not support a procedure for interest calculation as specified above, what modifications in it will you introduce to this schema in order that the above interest calculation procedure can be supported? How will that modification help? **Discuss.**
- (3) Write the code for modifying the tables and inserting rows into them if required.
- (4) Write the procedure for interest calculation.
- (5) Test the procedure you wrote for March 25, May 25 and August 25 of 2018.

Task 5: Theory

(5 marks)

In about one page (single space), with help of examples in the context of the the Spoc Bank domain:

- a) Briefly explain what a transaction is;
- b) Explain what a schedule is, and how it is different from a transaction;
- c) Define and explain the notion of a conflict serializable schedule. Does a conflict serializable schedule guarantee database consistency? Explain why or why not.
- d) Do different conflict-serializable schedules made up of the same set of transactions leave a database in the same state (with same value of the corresponding data items) if they started with the same state of the database? Explain your answer.
- e) Explain what view serializability is, and why its being a weaker notion than conflict serializability is also its strength.

Report Preparation and Assignment Submission

You will submit **two files**:

- a) A report in pdf format, named Report_224A2_<YourLastname>_<Your Firstname>.pdf. For instance, the report by Jane Smith will be named Report_224A2_Smith_Jane.pdf
- b) An sql file named Codes_224A2_<YourLastname>_<Your Firstname>.sql

The two files should be submitted on iLearn submission site for Assignment 2. They should be submitted as two separate files, not as a zipped folder.

In order to prepare the report, you will need to do, **for each task**,

- a) Copy and paste the code (for procedure, trigger, etc.) that you wrote (if applicable)
- b) Copy and paste the query you wrote to test the code (if applicable) and evidence that the code works, and
- c) Provide answer to any question asked in that task

Most likely you will do these using a word file, and then convert it to pdf before submitting.