

### List of assumptions:

1. Members will be fined \$2/day if they haven't still returned the book after the return due date (21 days after the borrowed date).
2. There will be no fine calculation, membership termination or clearance of the fine fees for the expired memberships, however there will be a suspension record created for every overdue item they had in the past even if the membership is expired, just for the sake of keeping a record so in case they renew their membership we still have the data recorded. So, to test the procedure and trigger correctly I needed to insert new data (memberships with a later expiry date) and borrowedby records so I can use them for the test.
3. There will be a suspension record created every time for the members who haven't returned the books before the return due date however their membership won't be changed to SUSPENDED unless their fine fees exceed \$30 (15 days after the return due date) so they have 15 days after the return due to bring back the borrowed books before their account gets suspended but they will be still fined for the 15 days before getting suspended.

Schema:

BRANCH ( BranchID [PK], BranchSuburb, BranchState )

MEMBER ( MemberID [PK], MemberStatus, MemberName, MemberAddress, MemberSuburb, MemberState, MemberExpDate, MemberPhone, FineFee )

PUBLISHER ( PublisherID [PK], PublisherName, PublisherAddress )

BOOK ( BookID [PK], BookTitle, PublisherID [FK], PublishedYear, Price )

AUTHOR ( AuthorID [PK], AuthorName, AuthorAddress )

AUTHOREDBY ( BookID [PK, FK], AuthorID [PK, FK] )

HOLDING ( BranchID [PK, FK], BookID [PK, FK], InStock, OnLoan )

BORROWEDBY ( BookIssueID [PK], BranchID [FK], BookID [FK], MemberID [FK], DateBorrowed, DateReturned, ReturnDueDate )

SUSPENSIONRECORD ( SuspensionID [PK], MemberID [PK, FK], FineFee, SuspensionStartDate )

### Task 1:

For this task I needed to add a new column to Member table to allow logging overdue items FineFee so I simply used the ALTER syntax to add a column to the table "ALTER TABLE Member ADD FineFee DECIMAL(10,2) DEFAULT 0;" then I had to calculate the fine fees for the members who have overdue items or had returned their item after the return due date. I used a procedure to do this, in the body of the procedure there are few business rules being followed. First, it will calculate the fine fees based on the BR.7 then it will be implementing the BR.6 by changing the MemberStatus of the membership with an outstanding fee of \$30 or above to SUSPENDED. After calling the procedure I have the Member table updated (fine fees are calculated for the members with a non-expired membership) and members with an outstanding amount of \$30 or above are suspended. This

procedure will also create and insert the suspension records to the newly created table that I want to use later for keeping track of membership suspensions.

Task1 prove: as you can see there are few members I've inserted to the dataset each has different data and the calculated fine and the membership status is changed according to their data for example as you can see, there is no change done to the members with expired membership the fine fees and member status stays intact however, there was a suspension record created for them and added to the SuspensionRecord table. but for other new members that I've inserted there are different scenarios implemented, for example Member ID 7,8,9 have outstanding fine fees of above \$30 so their memberships were changed to SUSPENDED from REGULAR, next scenario is when the Member ID 10 has an outstanding fine fee of less than \$30 so their member status stays intact.

MemberID	MemberStatus	MemberName	MemberAddress	MemberSuburb	MemberState	MemberExpDate	MemberPhone	FineFee
1	REGULAR	Joe	4 Nowhere St	Here	NSW	2021-09-30	0434567811	0.00
2	REGULAR	Pablo	10 Somewhere St	There	ACT	2022-09-30	0412345678	0.00
3	REGULAR	Chen	23/9 Faraway Cl	Far	QLD	2020-11-30	0412346578	0.00
4	REGULAR	Zhang	Dunno St	North	NSW	2020-12-31		0.00
5	REGULAR	Saleem	44 Magnolia St	South	SA	2020-09-30	1234567811	0.00
6	SUSPENDED	Homer	Middle of Nowhere	North Ryde	NSW	2020-09-30	1234555811	0.00
7	SUSPENDED	Amir	23 Stan St	West	NSW	2025-09-30	0447854894	796.00
8	SUSPENDED	Joe	4 Lily St	Left	VIC	2024-09-30	0447854424	66.00
9	SUSPENDED	Leo	17 Rsp St	south	Far	2025-10-30	0447254834	310.00
10	REGULAR	Carl	6 WW St	Macquarie	NSW	2024-10-30	0417894464	26.00
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Task2:

In this task I need to write a trigger that will implement the BR.8 "When a suspended member clears their fine (i.e, paid all the outstanding fees) and has no or has returned all overdue items, reset the member's membership status to REGULAR." To implement this, I must write a trigger to be checking those conditions every time there is a change to the member table however there are some cases where we don't want the change to be even done to the member table, so I handled those situations with error handlers. The exceptions are two, the first one is when there is an attempt to update the expired memberships and the second situation that will raise the error is when there is an attempt to pay more than the total fine fees of a membership (which means the new fine fee should not be negative before the update on Member table). I decided to use BEFORE UPDATE trigger to avoid causing an infinite loop. So, the trigger will be looking at the new fine fee before updating the table and if the new fine fee is equal to 0 then it will be changing the status to REGULAR.

Task3:

In this task I am going to write a procedure to "list the members that currently have an overdue item and their (individual) membership has been suspended twice in the past three years End these members' membership by setting their MemberStatus to TERMINATED." For this I need few things, the first one is to check if there is any member with an overdue item for this I decided to declare a cursor for members with overdue items whose membership hasn't expired which will be going through the joined version of member table and borrowby table since I need access to both member IDs and the members expiry date so if the membership is already expired it wont be terminated, I also declared a not found handler for the cursor so it will set the v\_finished to 1 when there is no more records in the cursor and this will be going to set the conditions for the REPEAT UNTIL loop to stop. After fetching the data from the cursor I wrote an error handler for when there is no member with overdue items found, so in that case the variable v\_MemberID wouldn't be changed at all and will be remained NULL so the error with the custom message will be raised if the v\_MemberID is NULL. The other important thing that is supposed to be handled by this procedure is to list out the members that are going to be terminated so I created a temporary table in the

procedure and stored the members that are going to be terminated in that then I used a simple SELECT statement to print out the values stored the in the table.

#### Task 4:

Task2 test plan:

To correctly test this trigger first I need to come up with a method to update the fine fee values in the table so I created this new procedure called PayFineFees:

```
DELIMITER //
DROP PROCEDURE IF EXISTS PayFineFees //
CREATE PROCEDURE PayFineFees(IN deductAmount DECIMAL(10,2), IN inputMemberID INT)
BEGIN
    DECLARE newFineFee DECIMAL(10,2);
    DECLARE curFineFee DECIMAL(10,2);
    DECLARE msg VARCHAR(255);
    SELECT FineFee - deductAmount INTO newFineFee FROM Member WHERE MemberID = inputMemberID;
    #deduct the amount from FineFee for the respective member in Member table
    UPDATE Member
    SET FineFee = newFineFee
    WHERE MemberID = inputMemberID;
END //
DELIMITER ;
```

Now I will use this procedure to test the functionality of my trigger this is the current stage of the Member table

	MemberID	MemberStatus	MemberName	MemberAddress	MemberSuburb	MemberState	MemberExpDate	MemberPhone	FineFee
▶	1	REGULAR	Joe	4 Nowhere St	Here	NSW	2021-09-30	0434567811	0.00
	2	REGULAR	Pablo	10 Somewhere St	There	ACT	2022-09-30	0412345678	0.00
	3	REGULAR	Chen	23/9 Faraway Cl	Far	QLD	2020-11-30	0412346578	0.00
	4	REGULAR	Zhang	Dunno St	North	NSW	2020-12-31		0.00
	5	REGULAR	Saleem	44 Magnolia St	South	SA	2020-09-30	1234567811	0.00
	6	SUSPENDED	Homer	Middle of Nowhere	North Ryde	NSW	2020-09-30	1234555811	0.00
	7	SUSPENDED	Amir	23 Stan St	West	NSW	2025-09-30	0447854894	796.00
	8	SUSPENDED	Joe	4 Lily St	Left	VIC	2024-09-30	0447854424	66.00
	9	SUSPENDED	Leo	17 Rsp St	south	Far	2025-10-30	0447254834	310.00
	10	REGULAR	Carl	6 WW St	Macquarie	NSW	2024-10-30	0417894464	26.00

Now I will call the procedure and try different scenarios with it.

call PayFineFees(796,7);

call PayFineFees(40,8);

SELECT \* FROM Member;

	MemberID	MemberStatus	MemberName	MemberAddress	MemberSuburb	MemberState	MemberExpDate	MemberPhone	FineFee
▶	1	REGULAR	Joe	4 Nowhere St	Here	NSW	2021-09-30	0434567811	0.00
	2	REGULAR	Pablo	10 Somewhere St	There	ACT	2022-09-30	0412345678	0.00
	3	REGULAR	Chen	23/9 Faraway Cl	Far	QLD	2020-11-30	0412346578	0.00
	4	REGULAR	Zhang	Dunno St	North	NSW	2020-12-31		0.00
	5	REGULAR	Saleem	44 Magnolia St	South	SA	2020-09-30	1234567811	0.00
	6	SUSPENDED	Homer	Middle of Nowhere	North Ryde	NSW	2020-09-30	1234555811	0.00
	7	REGULAR	Amir	23 Stan St	West	NSW	2025-09-30	0447854894	0.00
	8	SUSPENDED	Joe	4 Lily St	Left	VIC	2024-09-30	0447854424	26.00
	9	SUSPENDED	Leo	17 Rsp St	south	Far	2025-10-30	0447254834	310.00
	10	REGULAR	Carl	6 WW St	Macquarie	NSW	2024-10-30	0417894464	26.00
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

As you can see the trigger changed the status of member ID 7 to regular but the others remained intact. Now I will try to see if the exceptions are being handled properly with the trigger so I try to

pay more than the total amount of the member ID 8.

✖ 4444 04:10:44 call PayFineFees(27,8)

Error Code: 1644. You cannot pay more than your total finefees

It is working as expected and raised the error with the custom message now lets try the other exception. I try to pay for the fine fees of a membership that is expired.

✖ 4445 04:12:30 call PayFineFees(29,1)

Error Code: 1644. Membership has expired.

The trigger behaviour is as expected for this situation as well. Now I try to clear the rest of the fine fee of the member ID 8.

call PayFineFees(26,8);

MemberID	MemberStatus	MemberName	MemberAddress	MemberSuburb	MemberState	MemberExpDate	MemberPhone	FineFee
1	REGULAR	Joe	4 Nowhere St	Here	NSW	2021-09-30	0434567811	0.00
2	REGULAR	Pablo	10 Somewhere St	There	ACT	2022-09-30	0412345678	0.00
3	REGULAR	Chen	23/9 Faraway Cl	Far	QLD	2020-11-30	0412346578	0.00
4	REGULAR	Zhang	Dunno St	North	NSW	2020-12-31		0.00
5	REGULAR	Saleem	44 Magnolia St	South	SA	2020-09-30	1234567811	0.00
6	SUSPENDED	Homer	Middle of Nowhere	North Ryde	NSW	2020-09-30	1234555811	0.00
7	REGULAR	Amir	23 Stan St	West	NSW	2025-09-30	0447854894	0.00
8	REGULAR	Joe	4 Lily St	Left	VIC	2024-09-30	0447854424	0.00
9	SUSPENDED	Leo	17 Rsp St	south	Far	2025-10-30	0447254834	310.00
10	REGULAR	Carl	6 WW St	Macquarie	NSW	2024-10-30	0417894464	26.00
	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

As you can see it worked as expected and changed the member status to REGULAR successfully.

Task3 test plan:

To test this procedure firstly I needed to add a new member with a different number of overdue items to the data, so I can test the procedure for different scenarios now I have a member with one suspension record another member with two suspension record and another with 3 suspension records

MemberID	MemberStatus	MemberName	MemberAddress	MemberSuburb	MemberState	MemberExpDate	MemberPhone	FineFee
1	REGULAR	Joe	4 Nowhere St	Here	NSW	2021-09-30	0434567811	0.00
2	REGULAR	Pablo	10 Somewhere St	There	ACT	2022-09-30	0412345678	0.00
3	REGULAR	Chen	23/9 Faraway Cl	Far	QLD	2020-11-30	0412346578	0.00
4	REGULAR	Zhang	Dunno St	North	NSW	2020-12-31		0.00
5	REGULAR	Saleem	44 Magnolia St	South	SA	2020-09-30	1234567811	0.00
6	SUSPENDED	Homer	Middle of Nowhere	North Ryde	NSW	2020-09-30	1234555811	0.00
7	REGULAR	Amir	23 Stan St	West	NSW	2025-09-30	0447854894	0.00
8	REGULAR	Joe	4 Lily St	Left	VIC	2024-09-30	0447854424	0.00
9	SUSPENDED	Leo	17 Rsp St	south	Far	2025-10-30	0447254834	310.00
10	REGULAR	Carl	6 WW St	Macquarie	NSW	2024-10-30	0417894464	26.00
11	SUSPENDED	Sam	6 Pendi St	Quakers	TEH	2026-10-30	0417594564	890.00

Now I can call the procedure and see if its going to be modifying the correct memberships

CALL TerminateRepeatOffenders();

	MemberID	MemberStatus	MemberName	MemberAddress	MemberSuburb	MemberState	MemberExpDate	MemberPhone	FineFee
▶	1	REGULAR	Joe	4 Nowhere St	Here	NSW	2021-09-30	0434567811	0.00
	2	REGULAR	Pablo	10 Somewhere St	There	ACT	2022-09-30	0412345678	0.00
	3	REGULAR	Chen	23/9 Faraway Cl	Far	QLD	2020-11-30	0412346578	0.00
	4	REGULAR	Zhang	Dunno St	North	NSW	2020-12-31		0.00
	5	REGULAR	Saleem	44 Magnolia St	South	SA	2020-09-30	1234567811	0.00
	6	SUSPENDED	Homer	Middle of Nowhere	North Ryde	NSW	2020-09-30	1234555811	0.00
	7	REGULAR	Amir	23 Stan St	West	NSW	2025-09-30	0447854894	0.00
	8	REGULAR	Joe	4 Lily St	Left	VIC	2024-09-30	0447854424	0.00
	9	TERMINATE	Leo	17 Rsp St	south	Far	2025-10-30	0447254834	310.00
	10	REGULAR	Carl	6 WW St	Macquarie	NSW	2024-10-30	0417894464	26.00
	11	TERMINATE	Sam	6 Pendi St	Quakers	TEH	2026-10-30	0417594564	890.00
•	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL

The procedure is indeed working as expected and modified only the memberships with an overdue item who have two or more suspension records. Below is the list printed out after calling the procedure which is showing only the two terminated memberships so this is working as expected as well.

	MemberID	Name	MemberStatus
▶	9	Leo	TERMINATE
	11	Sam	TERMINATE

Now I want to try the scenario when there is no member with an overdue item to terminate in that case the procedure should be raising the error with custom message, for this I simply removed all the inserted data that was leading to have overdue item for any of the members with a non-expired membership and then I called the procedure in this scenario again and the result was as expected. The error with the custom message no member currently have overdue items was raised. The table below indicates that there is no non-expired membership with any overdue items.

	MemberID	MemberStatus	MemberName	MemberAddress	MemberSuburb	MemberState	MemberExpDate	MemberPhone	FineFee
▶	1	REGULAR	Joe	4 Nowhere St	Here	NSW	2021-09-30	0434567811	0.00
	2	REGULAR	Pablo	10 Somewhere St	There	ACT	2022-09-30	0412345678	0.00
	3	REGULAR	Chen	23/9 Faraway Cl	Far	QLD	2020-11-30	0412346578	0.00
	4	REGULAR	Zhang	Dunno St	North	NSW	2020-12-31		0.00
	5	REGULAR	Saleem	44 Magnolia St	South	SA	2020-09-30	1234567811	0.00
•	6	SUSPENDED	Homer	Middle of Nowhere	North Ryde	NSW	2020-09-30	1234555811	0.00
•	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL

Member 112	Member 113	×	Apply
Output			
Action Output			
#	Time	Action	Message
5368	05:29:03	SELECT * FROM Member LIMIT 0, 1000	6 row(s) returned
5369	05:29:04	call TerminateRepeatOffenders()	Error Code: 1644. No members currently have overdue items.

## Inserted Data:

INSERT INTO Member

(MemberID,MemberStatus,MemberName,MemberAddress,MemberSuburb,MemberState,MemberExpDate,MemberPhone)

VALUES ('7','REGULAR','Amir','23 Stan St','West','NSW','2025-09-30','0447854894');

INSERT INTO Member

(MemberID,MemberStatus,MemberName,MemberAddress,MemberSuburb,MemberState,MemberExpDate,MemberPhone)

VALUES ('8','REGULAR','Joe','4 Lily St','Left','VIC','2024-09-30','0447854424');

```
INSERT INTO Member
(MemberID,MemberStatus,MemberName,MemberAddress,MemberSuburb,MemberState,MemberE
xpDate,MemberPhone)
```

```
VALUES ('9','REGULAR','Leo','17 Rsp St','south','Far','2025-10-30','0447254834');
```

```
INSERT INTO Member
(MemberID,MemberStatus,MemberName,MemberAddress,MemberSuburb,MemberState,MemberE
xpDate,MemberPhone)
```

```
VALUES ('10','REGULAR','Carl','6 WW St','Macquarie','NSW','2024-10-30','0417894464');
```

```
INSERT INTO Borrowedby
(BranchID,BookID,MemberID,DateBorrowed,DateReturned,ReturnDueDate)
```

```
VALUES ('1','1','7','2023-09-22',NULL,'2023-10-13');
```

```
INSERT INTO Borrowedby
(BranchID,BookID,MemberID,DateBorrowed,DateReturned,ReturnDueDate)
```

```
VALUES ('1','2','7','2022-09-19',NULL,'2022-10-10');
```

```
INSERT INTO Borrowedby
(BranchID,BookID,MemberID,DateBorrowed,DateReturned,ReturnDueDate)
```

```
VALUES ('1','1','8','2023-09-22',NULL,'2023-10-13');
```

```
INSERT INTO Borrowedby
(BranchID,BookID,MemberID,DateBorrowed,DateReturned,ReturnDueDate)
```

```
VALUES ('2','1','8','2023-09-22',NULL,'2023-10-10');
```

```
INSERT INTO Borrowedby
(BranchID,BookID,MemberID,DateBorrowed,DateReturned,ReturnDueDate)
```

```
VALUES ('2','4','9','2023-09-22',NULL,'2023-8-13');
```

```
INSERT INTO Borrowedby
(BranchID,BookID,MemberID,DateBorrowed,DateReturned,ReturnDueDate)
```

```
VALUES ('3','5','9','2023-09-22',NULL,'2023-8-10');
```

```
INSERT INTO Borrowedby
(BranchID,BookID,MemberID,DateBorrowed,DateReturned,ReturnDueDate)
```

```
VALUES ('2','4','10','2023-09-22','2023-8-26','2023-8-13');
```

```
INSERT INTO Borrowedby
(BranchID,BookID,MemberID,DateBorrowed,DateReturned,ReturnDueDate)
```

```
VALUES ('2','4','11','2023-03-22',NULL,'2023-4-13');
```

```
INSERT INTO Borrowedby
(BranchID,BookID,MemberID,DateBorrowed,DateReturned,ReturnDueDate)
```

```
VALUES ('3','5','11','2023-02-22',NULL,'2023-3-10');
```

```
INSERT INTO Borrowedby VALUES ('2','4','11','2023-09-22','2023-8-28','2023-8-13');
```