

Library Management System

I. Business rules

1. Books and copies

- Each book can have many authors or many keywords and each author or keyword can relate to many words.
- Each book will have one publisher and one DDC code.
- DDC code are used to categorize book and can be used to locate the book in the library.

2. Magazines

- All magazines are not for rented.
- Each magazine many have many issues, each issue may have many copies.
- Each magazine has one publisher and may have many keywords.

3. Rent

- Each copy can only be borrowed by 1 member at a time.
- Members must not have any unpaid fine at the time they rent a copy.
- Each member can only borrow 2 copies simultaneously and these copies must be available or reserved by the member at that time.
- The copies must be return after 14 days, the due date can be extended twice, which means that the copies can be returned after 45 days maximum.

4. Reserve

- Members can reserve a copy if it is available. The reservation can be cancelled by the member or automatically after 2 days.
- Each member can only reserve 3 copies at a time.

5. Fine

- If the member return copies late, they will be fined. The amount of the fine is equal to the number of days overdue * 1.00 USD.
- Members will also be fined if they damage or lost the copy.
- Any rent that is 45-day overdue will be considered lost.

6. Waitlist

- Members can enter waitlist for a book if none of the copies of that book is available.

- When a book has available copies, the first member on the waitlist for that book will get the reservation for it automatically.

7. Subscribe

- Each member can subscribe to many magazines and will receive emails whenever a new issue of that magazine arrives at the library.

II. Data Glossary

Entity 1	Entity 2	Relationship
Books	Copies	1:N
Books	Author	N:N
Publisher	Books	1:N
Books	Keyword	N:N
DDC	Books	1:N
Member	Rent	1:N
Member	Waitlist	1:N
Member	Fine	1:N
Member	Reserve	1:N
Books	Waitlist	1:N
Copies	Rent	1:N
Copies	Reserve	1:N
Rent	Fine	1:N
Magazine	Keyword	N:N
Magazine	Magazine copies	1:N
Member	Subscribe	1:N
Magazine copies	Subscribe	1:N
Fine invoice	Fine	1:1
Staff	Rent	2:N

3. Book

BOOK_ID	TITLE	DDC_ID	PUBLISHER_ID	PUBLISHED_YEAR	PAGES	SIZE	PRICE
AF5732	Review of early childhood policy and programs in...	362	FH838	1997-01-01	80	27	26.9
AG1102	Garbage in the cities: Refuse, reform, and the en...	363	AI405	2005-01-01	303	23	35.6
AG2005	Concepts of fitness and wellness: A comprehensi...	613	DI458	2000-01-01	482	28	14.5
AG4978	The Lion of Wall Street: The two lives of Jack Dre...	332	GH194	1996-01-01	328	23	16.3
AG9174	The economic life of refugees	330	CG206	2005-01-01	131	23	22.7
BB5794	Oil painting: Pure and Simple	751	FB389	1986-01-01	128	23	4
BB7245	Teaching foreign language skills	407	HA178	1968-01-01	403	22	23.8
BF1978	Wild Decembers: A novel	823	IC519	1999-01-01	257	23	30.9

4. Copies

COPIES_ID	BOOK_ID	RENT_STAT	BOOK_STAT	ADDED_DATE
AL05	CA5426	Available	Fine	2011-07-28 00:00:00.000
AL0515	ED5081	Available	Water damaged	2012-11-06 00:00:00.000
AL0810	CD7165	Available	Torn page(s)	2016-02-08 00:00:00.000
AL1009	DA2309	Available	Fine	2012-11-16 00:00:00.000
AL10773	EC2302	Available	Fine	2010-11-29 00:00:00.000
AL1090	CD7165	Available	Ink stained	2012-02-07 00:00:00.000
AL1111	HE9027	Available	Fine	2016-07-12 00:00:00.000
AL11332	EA1463	Available	Fine	2012-08-04 00:00:00.000

Rent status: include Available, Reserved, Rented, Lost

5. Publisher

PUBLISHER_ID	PUBLISHER
AA291	Kumarian
AC266	W. W. Norton
AE836	Island
AI405	Anthony Mott
BB328	MacMillan Press
BG567	BFI Publishing
BH639	Abrams

6. Keyword

KEYWORD_ID	KEYWORD
A17204	Travelers
A28095	Fiction
A30136	Storms
A40879	Language
A43833	Ocean travel
A49199	Realationship
A49410	Pollution

7. Author

AUTHOR_ID	AUTHOR
A230844	Jim Stafford
A310867	John Stephen Doherty
A700043	Bernard Harrison
A903623	Thomas Carlyle
B125790	Ruth Lindsey
B369237	Peter M. Dunn
B392334	Christopher Hibbert
B501428	John Gardner

8. DDC

DDC_ID	CATEGORY	SECTION
000	Computer science, information & general works	A1
001	Knowledge	A1
002	The book	A2
003	Systems	A2
004	Data processing & computer science	A2
005	Computer programming, programs & data	A3
006	Special computer methods	A4
010	Bibliography	A5

9. Book_Author (Author_inline)

BOOK_ID	AUTHOR_ID
AF5732	F261763
AF5732	H188829
AG1102	G485660
AG2005	B125790
AG2005	C119763
AG2005	E218711
AG4978	D435478
AG9174	G572594

10. Book_KW (Book_inline)

BOOK_ID	KEYWORD_ID
AF5732	B54386
AF5732	C19110
AF5732	C47841
AF5732	G40529
AG1102	E36656
AG2005	C14827
AG2005	D90169
AG2005	F15359

11. Rent

RENT_ID	MEMBER_ID	COPIES_ID	RENT_DATE	DUE_DATE	RETURN_DATE	EXTEND	RENT_STAFF	RETURN_STAFF
1	GH667043	AV236	2021-01-01 00:00:00.000	2021-01-16 00:00:00.000	2021-01-16 00:00:00.000	0	1543C	4199A
2	GI405583	AV138	2021-01-03 00:00:00.000	2021-01-18 00:00:00.000	2021-01-17 00:00:00.000	0	5928D	2447F
3	FB787898	AI11878	2021-01-03 00:00:00.000	2021-02-17 00:00:00.000	2021-02-28 00:00:00.000	2	2447F	5889G
4	AE456176	AV121	2021-01-03 00:00:00.000	2021-01-18 00:00:00.000	2021-01-16 00:00:00.000	0	4199A	2447F
5	HB855934	AV587	2021-01-04 00:00:00.000	2021-02-03 00:00:00.000	2021-02-03 00:00:00.000	1	5928D	1543C
6	CH741977	AL141	2021-01-05 00:00:00.000	2021-02-04 00:00:00.000	2021-02-03 00:00:00.000	1	5452E	4199A
7	AA477915	AL7040	2021-01-08 00:00:00.000	2021-02-22 00:00:00.000	2021-02-04 00:00:00.000	2	1543C	4806A
8	HB361155	AV413	2021-01-10 00:00:00.000	2021-02-24 00:00:00.000	2021-02-20 00:00:00.000	2	2447F	5452E

12. Reserve

RESERVE_ID	COPIES_ID	MEMBER_ID	RESERVE_DATE	DUE_DATE	RESERVE_STATUS
1	AV138	GI405583	2021-01-03 00:00:00.000	2021-01-05 00:00:00.000	Finished
2	AL8300	GF448228	2021-01-25 00:00:00.000	2021-01-27 00:00:00.000	Cancelled
3	AL11879	CA163551	2021-01-28 00:00:00.000	2021-01-30 00:00:00.000	Finished
4	AL1152	DG216156	2021-02-02 00:00:00.000	2021-02-04 00:00:00.000	Finished
5	AL14723	EA362778	2021-02-05 00:00:00.000	2021-02-07 00:00:00.000	Cancelled
6	AV1346	AC666869	2021-02-13 00:00:00.000	2021-02-15 00:00:00.000	Finished
7	AV135	GD917626	2021-02-14 00:00:00.000	2021-02-16 00:00:00.000	Cancelled
8	AL11943	CA240857	2021-02-14 00:00:00.000	2021-02-16 00:00:00.000	Cancelled

RESERVE_STATUS includes Finished (member rented the reserved copy), Reserved (the copy is being reserved), Cancelled (member did not rent the copy before due date or member cancelled the reservation)

13. Waitlist

WAITLIST_ID	MEMBER_ID	BOOK_ID	WAIT_DATE	WAIT_STATUS
1	GF448228	GA2256	2021-01-20 00:00:00.000	Finished
2	AE456176	IH8572	2021-02-05 00:00:00.000	Cancelled
3	HB965115	HB9973	2021-02-07 00:00:00.000	Cancelled
4	HF488022	GE9413	2021-02-24 00:00:00.000	Cancelled
5	HG282354	GE9413	2021-02-24 00:00:00.000	Finished
6	IC699356	IH7722	2021-03-03 00:00:00.000	Finished
7	BH540477	AG1102	2021-03-04 00:00:00.000	Finished
8	DB228004	EI3500	2021-03-13 00:00:00.000	Cancelled

WAIT_STATUS includes Finished (member's turn had finished), Waiting (member is waiting for the book), Cancelled (member cancelled the waitlist)

14. Fine

FINE_ID	RENT_ID	FINE_AMOUNT	MEMBER_ID	FINE_DATE	REASON	AFTER_STATUS	FINE_STATUS
1	11	4	FC163629	2021-02-02 00:00:00.000	Overdue	NULL	Paid
2	10	5	HG282354	2021-02-03 00:00:00.000	Overdue	NULL	Paid
3	18	4	ID935100	2021-02-18 00:00:00.000	Overdue	NULL	Paid
4	24	2	DG216156	2021-02-20 00:00:00.000	Overdue	NULL	Paid
5	3	11	FB787898	2021-02-28 00:00:00.000	Overdue	NULL	Paid
6	19	6	GI396298	2021-03-08 00:00:00.000	Overdue	NULL	Paid
7	26	3	HE722514	2021-03-18 00:00:00.000	Overdue	NULL	Paid
8	23	3	DB228004	2021-03-20 00:00:00.000	Overdue	NULL	Paid

REASON includes Overdue, Damaged copy, Lost copy.

AFTER_STATUS refer to the condition of the returned book, it is only required when the reason for the penalty is 'Damaged copy'.

FINE_STATUS includes Paid, Unpaid.

15. Fine_Invoice

INV_ID	FINE_ID	INV_AMOUNT	INV_DATE	INV_STATUS
1	1	4	2021-02-04 00:00:00.000	Finished
2	2	5	2021-02-04 00:00:00.000	Finished
3	3	4	2021-02-21 00:00:00.000	Finished
4	4	2	2021-02-24 00:00:00.000	Finished
5	5	11	2021-02-28 00:00:00.000	Finished
6	6	6	2021-03-09 00:00:00.000	Finished
7	7	3	2021-03-18 00:00:00.000	Finished
8	8	3	2021-03-27 00:00:00.000	Finished

16. Magazine

MAG_ID	TITLE	PUBLISHER_ID	DDC_ID	PERIOD
MAG01	Journal of Natural Products	DA291	052	monthly
MAG02	The Catholic Telegraph	EH960	052	weekly
MAG03	Vogue	GE152	052	monthly
MAG04	Billboard	DK236	052	weekly
MAG05	The Nation	GG801	052	weekly
MAG06	Ecology	FJ994	052	bimonthly
MAG07	Wired	GE152	052	monthly

17. Mag_Copies

MAG_COPIES_ID	MAG_ID	PUBLISHED_DATE	NUMBER	QUANTITY
MC001	MAG06	2020-01-09 00:00:00.000	35	2
MC002	MAG06	2020-03-09 00:00:00.000	36	1
MC003	MAG07	2020-04-10 00:00:00.000	72	1
MC004	MAG03	2020-04-19 00:00:00.000	10	2
MC005	MAG01	2020-05-06 00:00:00.000	10	1
MC006	MAG06	2020-05-09 00:00:00.000	37	3
MC007	MAG07	2020-05-10 00:00:00.000	73	1

18. Mag_KW

MAG_ID	KEYWORD_ID
MAG01	K10657
MAG02	F24075
MAG03	C75351
MAG04	C36197
MAG05	E34299
MAG05	J55516
MAG06	B34739

19. Subscribe

SUB_ID	MEMBER_ID	MAG_ID	SUB_STATUS
1	IC699356	MAG02	Active
2	DE883346	MAG07	Cancelled
3	CC219637	MAG08	Active
4	GI405583	MAG06	Active
5	AC666869	MAG03	Active
6	HE722514	MAG04	Active
7	HF488022	MAG01	Active

SUB_STATUS includes Active and Cancelled.

III. Views

1. WAITLIST_TOP contains information of the earliest member waiting in line for each book, which help promptly find the member whenever a copy of a book becomes available.

BOOK_ID	MEMBER_ID	WAIT_DATE	WAITLIST_ID	WAIT_STATUS
FA1643	DE883346	2021-05-10 00:00:00.000	15	Waiting
GH5181	HG282354	2021-05-08 00:00:00.000	13	Waiting
IE1724	FF107722	2021-05-05 00:00:00.000	12	Waiting

2. AVAILABLE_COPIES contains information of available copies in the library, which assists the librarians in inventory management and renting procedure.

COPIES_ID	BOOK_ID	RENT_STAT	BOOK_STAT	ADDED_DATE
AL05	CA5426	Available	Fine	2011-07-28 00:00:00.000
AL0515	ED5081	Available	Water damaged	2012-11-06 00:00:00.000
AL0810	CD7165	Available	Torn page(s)	2016-02-08 00:00:00.000
AL1009	DA2309	Available	Fine	2012-11-16 00:00:00.000
AL10773	EC2302	Available	Fine	2010-11-29 00:00:00.000
AL1090	CD7165	Available	Ink stained	2012-02-07 00:00:00.000
AL1111	HE9027	Available	Fine	2016-07-12 00:00:00.000

3. OVERDUE_RESERVE contains information of overdue reservations for automatic cancelling at the end of the day.

RESERVE_ID	COPIES_ID	MEMBER_ID	RESERVE_DATE	DUE_DATE	RESERVE_STATUS
20	AV209	BH607561	2021-05-19 00:00:00.000	2021-05-21 00:00:00.000	Reserved

4. POT_LOST_COPIES contains information of potential lost copies (copies that has been overdue for more than 45 days) which will be considered lost if it is not returned at the end of the day.
5. LOST_COPIES contains information of lost copies which will be used as references for book replacement.

COPIES_ID	BOOK_ID	TITLE	ADDED_DATE	FINE_DATE
AV280	GD9487	Catalina: A romance	2017-08-01 00:00:00.000	2021-05-20 00:00:00.000

IV. Queries

1. Information of the top 5 most borrowed sorted by category provides insights into the most popular categories.

CATEGORY	CAT_COUNT	CAT_PERCENT
Social sciences	16	0.24
Literature & rhetoric	10	0.15
History & geography	9	0.13
Language	9	0.13
Natural sciences & mathematics	8	0.12

2. Information about top 5 most borrowed sorted by keyword provides insights into fields of most interest to readers.

KEYWORD	RENT_COUNT	RENT_PERCENT
Fiction	4	0.06
Ecology	4	0.06
Savanna	4	0.06
English language	4	0.06
Forest	4	0.06

- Information about top 5 authors with most borrowed books provides insights into the most favorite authors.

AUTHOR	RENT_COUNT	RENT_PERCENT
Christopher Hibbert	4	0.06
Jack Hellings	4	0.06
Brian Hopkins	4	0.06
B.J Thomas	4	0.06
Jean Ure	4	0.06

- Information of the number of staff for each position suggests if any position has surplus employees.

TITLE	NUMBER	PERCENTS
Cleaning Staff	7	0.23
Guard	6	0.20
IT Staff	5	0.17
Librarian	5	0.17
Staff	4	0.13
Vice President	2	0.07
President	1	0.03

- Information of the number of rents each day of the week shows the days with high volume of visits, which offers guidance on staff management.

DAY_OF_WEEK	DAY_COUNT	DAY_PERCENT
Thursday	13	0.19
Monday	12	0.18
Wednesday	11	0.16
Tuesday	10	0.15
Sunday	10	0.15
Friday	6	0.09
Saturday	6	0.09

6. Information of the members' ages, divided into 4 age groups: under 30 years old (born after 1991), 30 - 45 years old (born between 1976-1990), 45 - 60 years old (born between 1961-1975), above 60 years old (born before 1960).

AGE	AGE_COUNT	AGE_PERCENT
45-60	19	0.38
30-45	16	0.32
Above 60	9	0.18
Under 30	6	0.12

7. Information about frequent members of the library (members who come to borrow books at least 3 times).

MEMBER_ID	FIRST_NAME	LAST_NAME	RENT_COUNT
HB965115	Margie	Graves	5
GH667043	Rosa	Varela	4
HG721320	Claude	Jefferies	4
HE722514	Larry	Larry	3
FB787898	Lawrence	Lewis	3
FF107722	Charles	Amy	3
GA151753	Paul	Parrish	3

8. The average time period that members borrow a book indicates whether the rent period should increase so that the members do not have to extend many times.

RENT_AVG
25.53

9. Information of the common fines acts as baseline for tightening the rules if book losses or damages happen frequently.

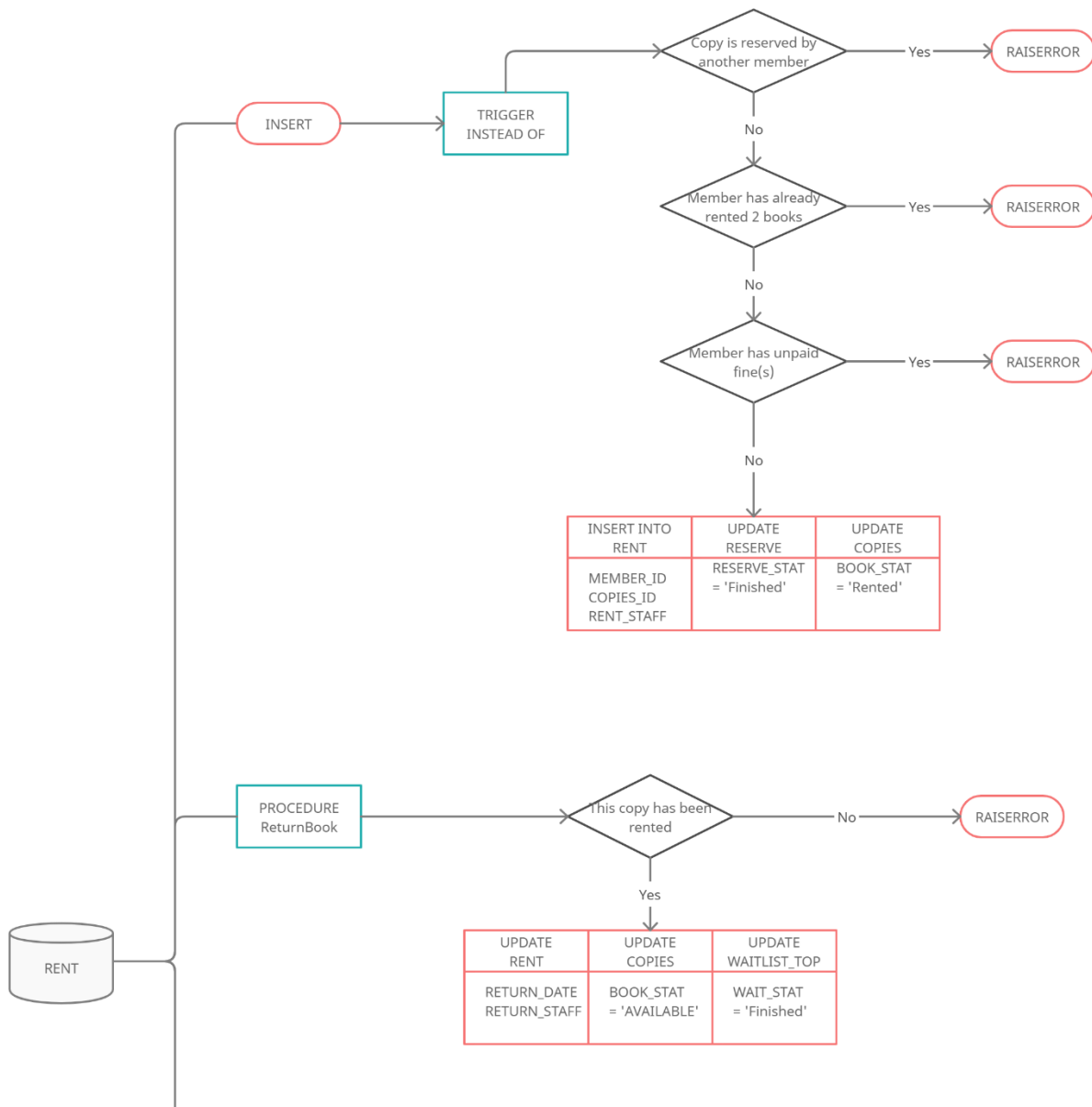
REASON	FINE_COUNT	FINE_PERCENT
Overdue	14	0.21

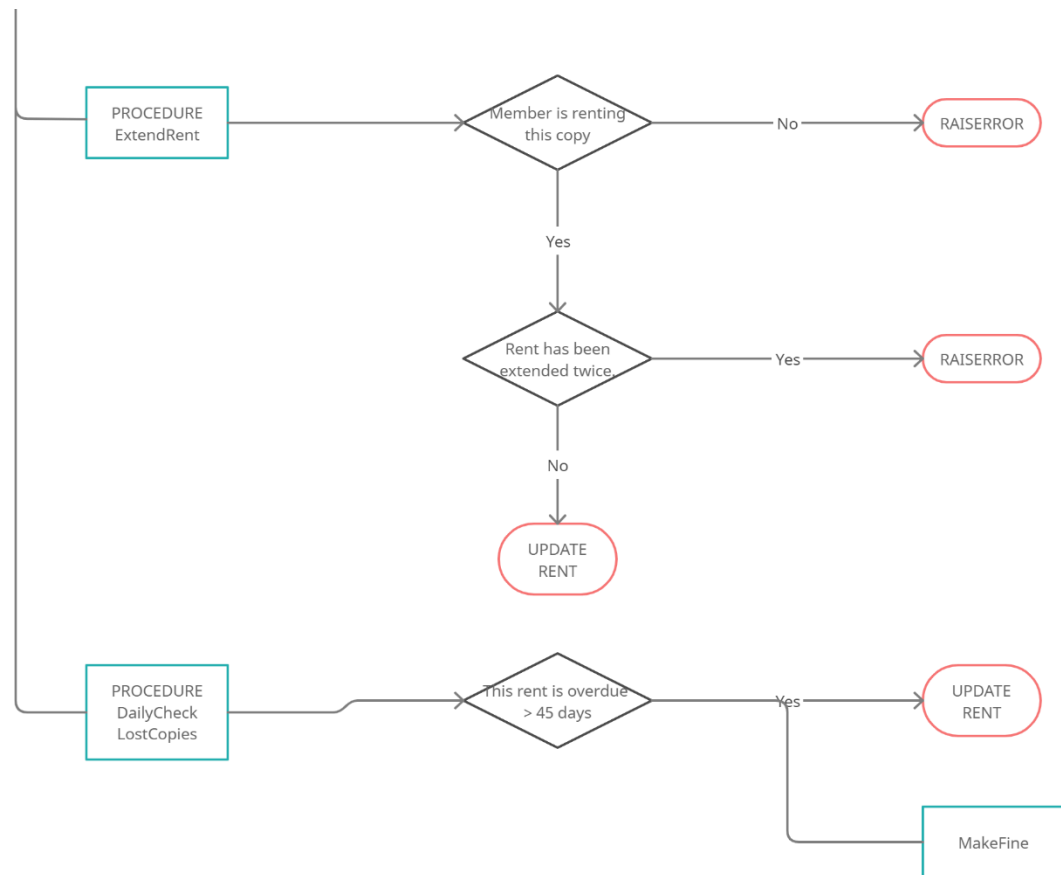
10. Information of the quality of the books helps the librarians make sure that the majority of the books in the library are in good condition.

BOOK_STAT	STAT_COUNT	STAT_PERCENT
Fine	138	0.76
Torn page(s)	26	0.14
Ink stained	11	0.06
Water damaged	7	0.04

V. Triggers and Procedures

1. Rent





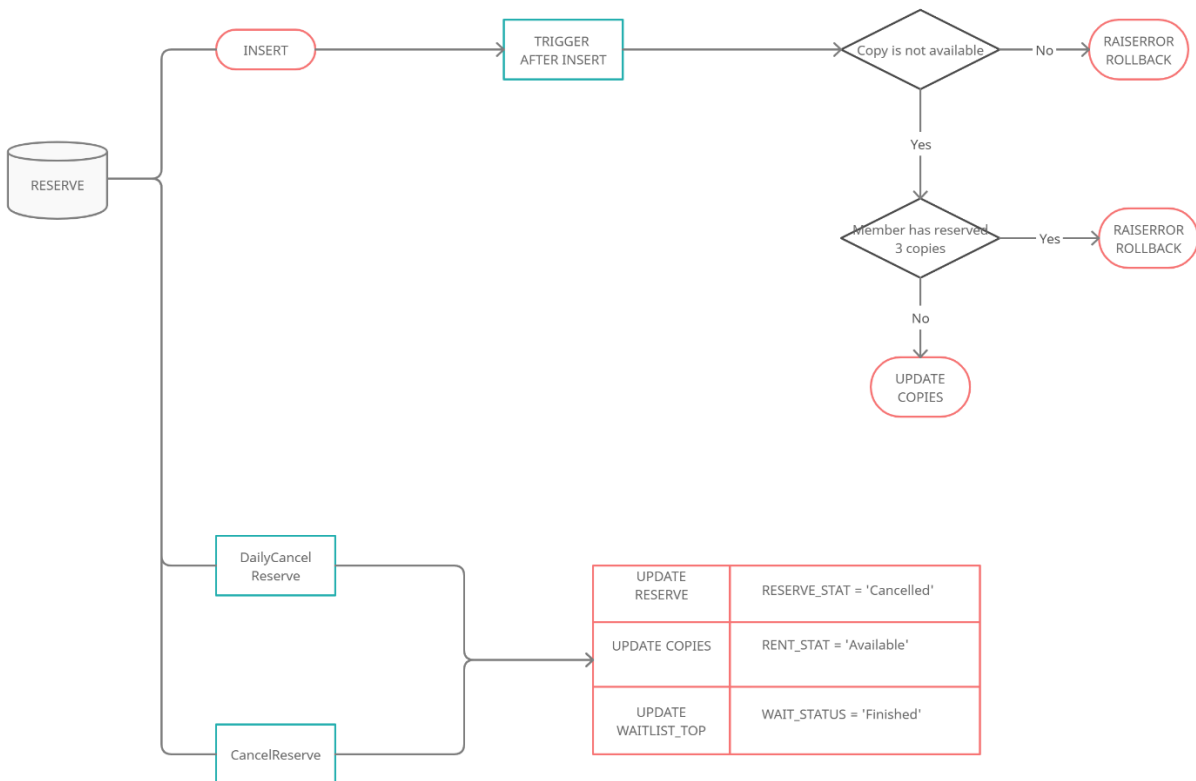
The trigger `reserve_check_and_update_status` runs prior to any insert into table `RENT`. It verifies whether the rent satisfies the three conditions as shown above; if any of these conditions are not met, the `RAISERROR` function is called. Otherwise, the insertion into table `RENT` is proceeded along with an update to table `COPIES` (changes `BOOK_STAT` to `Rented`) and another to table `RESERVE` (changes `RESERVE_STAT` of the rented copies by that member to `Finished`).

The procedure `ReturnBook` takes 2 arguments, `copies_id` and `return_staff` as input. If there is a record of this copy being rented in the table, updates are performed on three tables, `RENT`, `COPIES` and `WAITLIST_TOP`. The first update adds the return date and return staff's ID, the second changes `BOOK_STAT` to 'Available', and the last one changes to `WAIT_STATUS` of the corresponding book to `Finished`.

The procedure `ExtendRent` takes `rent_id` and checks if the rent meets two requirements (as shown above) in order to proceed the update of 'Extend' on table `RENT`.

The procedure `DailyCheckLostCopies` runs daily at the end of the day to terminate the any rent that is 45-day overdue and penalize the corresponding member.

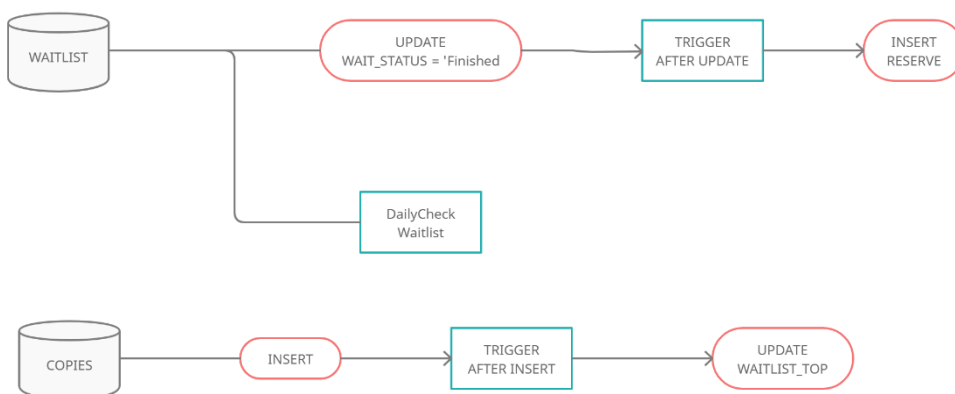
2. Reserve



After an insert into table RESERVE, the trigger change_status_reserves is pulled. A rollback and error raise operates if any of the two conditions are not met; else an update to the RENT_STAT on table COPIES is executed.

The procedure DailyCancelReserve and CancelReserve essentially performs the same tasks, apart from a minor difference which is that the former runs every day to cancel overdue reservations while the latter is used at members' disposal. Boths procedures performed updates on RESERVE, COPIES, and WAITLIST_TOP.

3. Waitlist

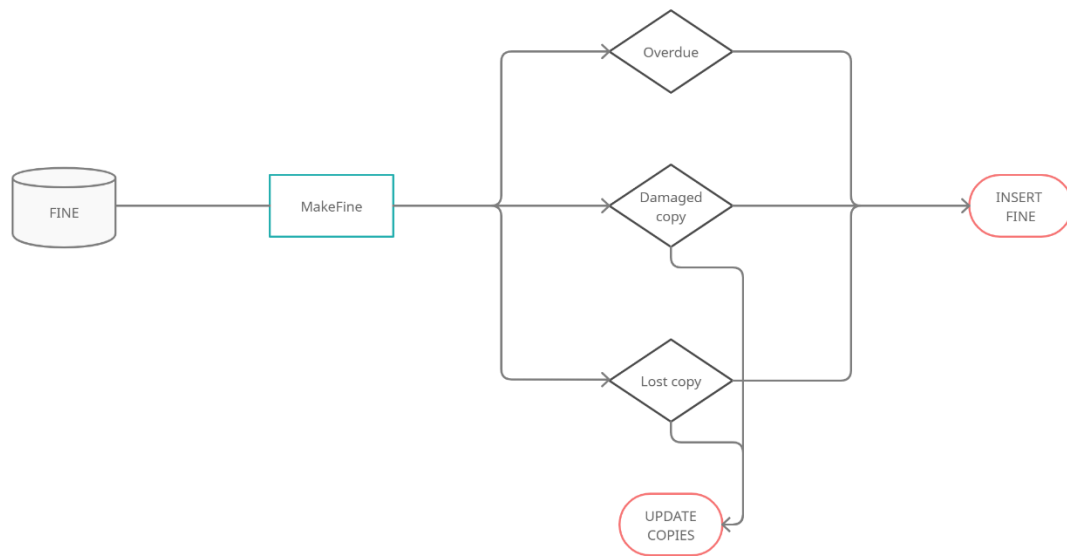


Trigger `auto_reserve` is called whenever an update to `WAIT_STATUS` on table `WAITLIST` from 'Waiting' to 'Finished'. A reservation is created automatically for the member.

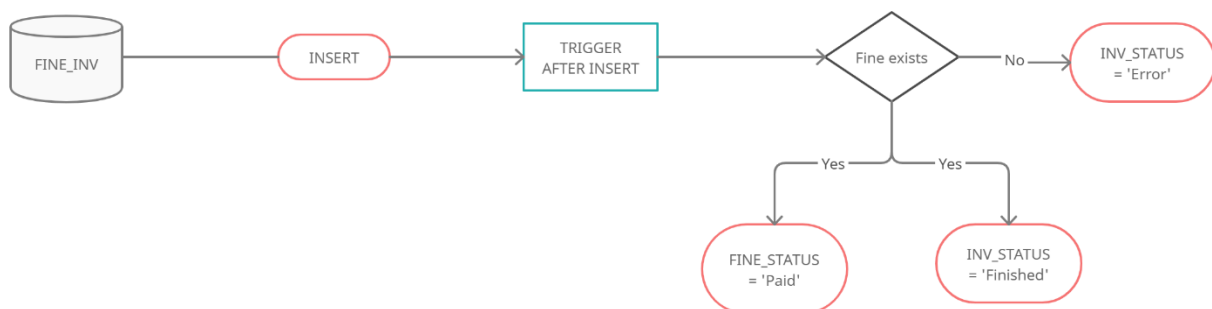
The trigger `CheckWaitlist`, which runs after every insertion into table `COPIES`, make changes table `WAITLIST_TOP` by shifting `WAIT_STATUS` of the corresponding book to Finished.

The procedure `DailyCheckWaitlist` operates on a daily basis to make sure that members in `WAITLIST_TOP` are granted access to the available books that they are in line for.

4. Fine

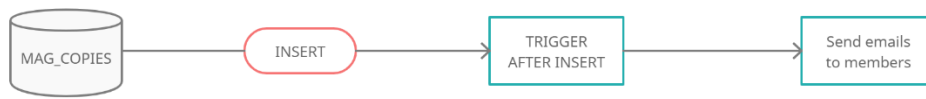


The procedure `MakeFine` takes in three arguments, `rent_id`, `reason`, and `after_stat` (only required if the reason for the fine is Damaged copy). Only when the reason falls in one of the three reasons (Overdue, Damaged copy, Lost copy) is the insertion into table `FINE` executed. If the reason is Damaged copy or Lost copy, the insertion is followed up by an update to `BOOK_STAT` and `RENT_STAT` on table `COPIES`.



The trigger `check_invoice` provides verifications for any insertion into table `FINE_INV`. If the insertion has the right `fine_id` and `fine_amount`, it updates table `FINE_STATUS` and `INV_STATUS`; else, the `INV_STATUS` is changed to 'Error'.

5. Subscribe



The trigger `check_subscribe` send emails to subscribers of a particular magazine whenever a record of a new copy of that magazine is inserted into `MAG_COPIES`.

VI. Logging

1. RENT_AUDIT

❖ Table CHANGERENT

```
CREATE TABLE CHANGERENT
(
    RENT_ID VARCHAR(50),
    FIELD NVARCHAR(30),
    OLD_DATA VARCHAR(50),
    NEW_DATA VARCHAR(50),
    ACTION_TIME smalldatetime);
```

FORMAT:

- `RENT_ID`: “RENT” primary key
- `FIELD`: Columns have changed information
- `OLD_DATA`: Old data before editing or deleting. If adding new data, `OLD_DATA = NULL`
- `NEW_DATA`: New data after editing or deleting. If deleting data, `NEW_DATA = NULL`
- `ACTION_TIME`: return a value of date time.

PURPOSE: Store all the changes about the data

❖ Trigger RENT_AUDIT

```
create trigger rent_audit
on rent
after update, insert
```

- `DECLARE`: a function to declare variable
- `SELECT`: a function to assign value to a variable
- `@var_name`: the name of the variable, always starting with ‘@’

```
declare @new_rentid as INT;

select @new_rentid = i.RENT_ID from inserted as i;
```

- After editing, the changed information is saved in “CHANGERENT” table.
- `CAST`: function converts a value into a specified datatype

```
begin
if @new_rentid <> @old_rentid
or (@old_rentid is NULL and @new_rentid is not NULL)
or (@old_rentid is not NULL and @new_rentid is NULL)
insert into CHANGERENT
values (CAST(@new_rentid AS VARCHAR(50)), 'RENT_ID', CAST(@old_rentid AS VARCHAR(50)), CAST(@new_rentid AS VARCHAR(50)), SYSDATETIME())
```

❖ EXAMPLE: MEMBER_ID GH667043 wants to extend the book once

```
UPDATE RENT
SET EXTEND = 1
WHERE MEMBER_ID = 'GH667043';
```

Then, table “RENT” looks like:

	RENT_ID	MEMBER_ID	COPIES_ID	RENT_DATE	DUE_DATE	RETURN_DATE	EXTEND	RENT_STAFF	RETURN_STAFF
1	1	GH667043	AV236	2021-01-01 00:00:00.000	2021-01-16 00:00:00.000	2021-01-16 00:00:00.000	1	1543C	4199A
2	2	GI405583	AV138	2021-01-03 00:00:00.000	2021-01-18 00:00:00.000	2021-01-17 00:00:00.000	1	5928D	2447F
3	3	FB787898	AI11878	2021-01-03 00:00:00.000	2021-02-17 00:00:00.000	2021-02-28 00:00:00.000	2	2447F	5889G

And the data in table “CHANGERENT”:

	RENT_ID	FIELD	OLD_DATA	NEW_DATA	ACTION_TIME
1	1	EXTEND	2	1	2021-06-04 22:57:00

2. RESERVE_AUDIT

❖ CHANGERESERVE

```
CREATE TABLE CHANGERESERVE
(
    RESERVE_ID VARCHAR(50),
    FIELD NVARCHAR(30),
    OLD_DATA VARCHAR(50),
    NEW_DATA VARCHAR(50),
    ACTION_TIME smalldatetime);
```

❖ RESERVE_AUDIT

```
create trigger reserve_audit
on reserve
after update, insert
```

❖ EXAMPLE: Reserve status of RESERVE_ID = 1 has changed to ‘Cancelled’

```
UPDATE RESERVE
SET RESERVE_STATUS = 'Cancelled'
WHERE RESERVE_ID = 1
```

Table RESERVE after editing:

	RESERVE_ID	COPIES_ID	MEMBER_ID	RESERVE_DATE	DUE_DATE	RESERVE_STATUS
1	1	AV138	GI405583	2021-01-03 00:00:00.000	2021-01-05 00:00:00.000	Cancelled
2	2	AL8300	GF448228	2021-01-25 00:00:00.000	2021-01-27 00:00:00.000	Cancelled
3	3	AL11879	CA163551	2021-01-28 00:00:00.000	2021-01-30 00:00:00.000	Finished

Then, table CHANGERESERVE looks like:

	RESERVE_ID	FIELD	OLD_DATA	NEW_DATA	ACTION_TIME
1	1	RESERVE_STATUS	Finished	Cancelled	2021-05-31 18:29:00

3. WAITLIST_AUDIT

❖ CHANGEWAITLIST

```
CREATE TABLE CHANGEWAITLIST
(
    WAITLIST_ID VARCHAR(50),
    FIELD NVARCHAR(30),
    OLD_DATA VARCHAR(50),
    NEW_DATA VARCHAR(50),
    ACTION_TIME smalldatetime);
```

❖ WAITLIST_AUDIT

```
create trigger waitlist_audit
on waitlist
after update, insert
```

❖ EXAMPLE:


```

UPDATE WAITLIST
SET WAIT_STATUS = 'Cancelled'
WHERE MEMBER_ID = 'GF448228'

```

Then, table WAITLIST looks like:

	WAITLIST_ID	MEMBER_ID	BOOK_ID	WAIT_DATE	WAIT_STATUS
1	1	GF448228	GA2256	2021-01-20 00:00:00.000	Cancelled
2	2	AE456176	IH8572	2021-02-05 00:00:00.000	Cancelled
3	3	HB965115	HB9973	2021-02-07 00:00:00.000	Cancelled

And the edited data is stored in table CHANGEWAITLIST:

	WAITLIST_ID	FIELD	OLD_DATA	NEW_DATA	ACTION_TIME
1	1	WAIT_STATUS	Finished	Cancelled	2021-05-31 18:28:00

4. FINE_AUDIT

❖ CHANGEFINE

```

CREATE TABLE CHANGEFINE
(
  FINE_ID VARCHAR(50),
  FIELD NVARCHAR(30),
  OLD_DATA VARCHAR(50),
  NEW_DATA VARCHAR(50),
  ACTION_TIME smalldatetime);

```

❖ FINE_AUDIT

```

create trigger fine_audit
on fine
after update, insert

```

- ❖ EXAMPLE: The reason for the fine of MEMBER_ID = FC163629 has RENT_ID = 11 changed to 'Lost copy'

```

UPDATE FINE
SET REASON = 'Lost copy'
WHERE RENT_ID = 11;

```

Table FINE after updating:

	FINE_ID	RENT_ID	FINE_AMOUNT	MEMBER_ID	FINE_DATE	REASON	AFTER_STATUS	FINE_STATUS
1	1	11	4	FC163629	2021-02-02 00:00:00.000	Overdue	NULL	Paid
2	2	10	5	HG282354	2021-02-03 00:00:00.000	Overdue	NULL	Paid
3	3	18	4	ID935100	2021-02-18 00:00:00.000	Overdue	NULL	Paid

And the table CHANGEFINE:

	FINE_ID	FIELD	OLD_DATA	NEW_DATA	ACTION_TIME
1	1	REASON	Overdue	Lost copy	2021-06-04 23:26:00

5. COPIES_AUDIT

❖ CHANGECOPIES

```

CREATE TABLE CHANGECOPIES
(
  COPIES_ID VARCHAR(50),
  FIELD NVARCHAR(30),
  OLD_DATA VARCHAR(50),
  NEW_DATA VARCHAR(50),
  ACTION_TIME smalldatetime);

```

❖ COPIES_AUDIT

```
create trigger copies_audit
on copies
after update, insert
```

- ❖ EXAMPLE: The status of BOOK_ID = ED5081 has COPIES_ID = AL0515 changed to 'Ink stained'

```
UPDATE COPIES
SET BOOK_STAT = 'Ink stained'
WHERE COPIES_ID = 'AL0515';
```

After updating, the table COPIES looks like:

	COPIES_ID	BOOK_ID	RENT_STAT	BOOK_STAT	ADDED_DATE
1	AL05	CA5426	Available	Fine	2011-07-28 00:00:00.000
2	AL0515	ED5081	Available	Ink stained	2012-11-06 00:00:00.000
3	AL0810	CD7165	Available	Torn page(s)	2016-02-08 00:00:00.000

And table CHANGECOPIES:

	COPIES_ID	FIELD	OLD_DATA	NEW_DATA	ACTION_TIME
1	AL0515	BOOK_STAT	Fine	Ink stained	2021-06-04 23:34:00