

TITLE: PROJECT STAGE 3

NAME: KAREN ADJOA SELASI KOOMSON

DESIGN OF EXTENDED FEATURES: REASONABLE SYSTEM DESIGNS

This stage focuses on extending the current Public Library DBMS to include automation features that make daily library operations more efficient. Both features aim to improve how overdue items and member accounts are handled while maintaining data integrity and reducing manual work for staff.

Feature 1 – Overdue Alert

Purpose:

The goal of this feature is to automatically alert staff each day about materials that are overdue. This helps ensure timely follow-ups with members who have not returned their borrowed items.

Idea:

Every morning the system should list all loans that are overdue (not returned and due date is before today) so staff can act.

How overdue items are identified:

A material is overdue when the **return_date** is NULL and the **due_date** has passed the current date (**due_date < CURRENT_DATE**). This condition can be checked daily using a query or automated process.

When the alert runs:

The alert should be generated **once every day**, preferably in the morning at 9am. This can be done using a scheduler such as pg_cron, pgAgent, or an external system task that calls a simple SQL script.

Database updates or additions:

- A new **view** called **v_overdue_open** will display all overdue materials in real time.
- Optionally, an **overdue_alert_log** table can record daily snapshots of overdue items for reporting or auditing.

```

r T S M T Q D P projects2Queries.sql* 
s X SQL X Statistics X Dependencies X Dependents X Processes X Library_Database/postgres@PostgreSQL 18 
Query History Scratch Pad 
125 WHERE b.return_date IS NULL 
126 AND b.due_date < CURRENT_DATE; 
127 
128 CREATE OR REPLACE VIEW v_overdue_open AS 
129 SELECT b.borrow_id, m.name AS member_name, mat.title AS material_title, 
130 b.borrow_date, b.due_date, CURRENT_DATE - b.due_date AS days_overdue 
131 FROM borrow b 
132 JOIN member m ON b.member_id = m.member_id 
133 JOIN material mat ON b.material_id = mat.material_id 
134 WHERE b.return_date IS NULL 
135 AND b.due_date < CURRENT_DATE; 
136 
Data Output Messages Notifications 
CREATE VIEW 
Query returned successfully in 94 msec.

```

The screenshot shows the pgAdmin 4 interface with the SQL tab active. The left sidebar displays the database schema with tables like author, borrow, catalog, genre, material, member, and staff. The main pane shows a query to create a view named v_overdue_open. The query selects borrow_id, member_name, material_title, borrow_date, due_date, and days_overdue from the borrow, member, and material tables where return_date is null and due_date is less than current_date. The query executes successfully.

This view gives staff an up-to-date list of overdue books they can check anytime.

```

Query History Scratch Pad 
1 SELECT * FROM public.v_overdue_open 
2 
Data Output Messages Notifications 
Showing rows: 1 to 22 Page No: 1 of 1 
borrow_id member_name material_title borrow_date due_date days_overdue 
1 31 Isla Thomas Ulysses 2022-12-28 2023-01-18 
2 32 Alice Johnson The Great Gatsby 2023-01-23 2023-02-13 
3 34 Kate Anderson Lolita 2023-03-01 2023-03-22 
4 35 Luke Jackson Brave New World 2023-03-10 2023-03-31 
5 36 Mia White The Sound and the Fury 2023-03-15 2023-04-05 
6 37 Quinn Hall Catch-22 2023-03-25 2023-04-15 
Total rows: 22 Query complete 00:00:00.225 CRLF Ln 1, Col 1 

```

The screenshot shows the pgAdmin 4 interface with the Data Output tab active. It displays the results of the query SELECT * FROM public.v_overdue_open. The results are shown in a table with columns: borrow_id, member_name, material_title, borrow_date, due_date, and days_overdue. The table contains 22 rows of data, each representing a book that is overdue.

Feature 2 – Membership Deactivation and Reactivation

Purpose:

This feature automatically manages member status based on their overdue history. Members who repeatedly fail to return books are deactivated, and their accounts are reactivated once the overdue issues are resolved.

Defining an overdue occurrence:

One borrowing record that becomes overdue (due_date < today and return_date IS NULL). Each borrow counts as one occurrence, regardless of how long it remains overdue.

How the system reacts:

- A member's account becomes **deactivated** when they reach **three or more** overdue occurrences.
- Once they **return** the materials or **pay** the overdue fees, their account is automatically set back to **active**.

Database changes:

1. Add a **status** column to the **Member** table:

```
ALTER TABLE member
ADD COLUMN status TEXT NOT NULL DEFAULT 'active'
CHECK (status IN ('active','deactivated'));
```

2. We can also add an **overdue_occurrence** table to record overdue events and a table to track fee payments.

Automation - Using Triggers & Procedures:

- A daily procedure counts how many active overdue records each member has and automatically sets **status = 'deactivated'** if the count ≥ 3 .
- A trigger on the **payment** table updates member status back to 'active' when the overdue fees are cleared.

Summary

Both features make the library database smarter and more efficient. The Overdue Alert ensures staff always know which materials are late, while the Membership Deactivation/ Reactivation feature enforces fair borrowing rules automatically. Together, they improve reliability, save staff time, and strengthen data consistency across the system.

References

- [1] R. Elmasri and S. B. Navathe, *Fundamentals of Database Systems*, 7th ed., Pearson, 2016. [2] C. Coronel and S. Morris, *Database Systems: Design, Implementation and Management*, 13th ed., Cengage Learning, 2019.