

Database Management System

PROJECT REPORT

Team ID: 5

Personal Knowledge Graph Vault

Name	SRN
Diya D Bhat	PES2UG23CS183
Dhanya Prabhu	PES2UG23CS169

1. Abstract / Introduction

The *Personal Knowledge Graph Vault* is a database-driven web application built using MySQL and Streamlit (Python). It helps users organize interconnected information like concepts, notes, tasks, collaborators, and tags in a structured relational format. The project demonstrates major DBMS concepts such as normalization, triggers, stored procedures, functions, and SQL queries integrated into a real-time interactive interface.

2. Objectives

- To design a normalized relational database for managing interconnected knowledge.
- To implement CRUD operations and automate data handling using triggers.
- To demonstrate stored procedures and functions through GUI interaction.
- To apply aggregate, join, and nested queries for analytical insights.
- To integrate backend MySQL with a user-friendly Streamlit interface.

3. Problem Statement

Managing ideas, notes, and related work across multiple platforms can be unorganized. The Knowledge Graph Vault solves this by storing all information in a centralized, relational structure, linking related data and automating updates. This helps students and researchers maintain, retrieve, and visualize their knowledge efficiently.

4. System Design

The system follows a modular and normalized architecture, separating data into related tables (Users, Concepts, Notes, Tasks, Tags, Collaborators, etc.). Each module performs distinct operations, ensuring data consistency and easy maintenance through SQL relations and GUI-based forms.

4.1 ER Diagram

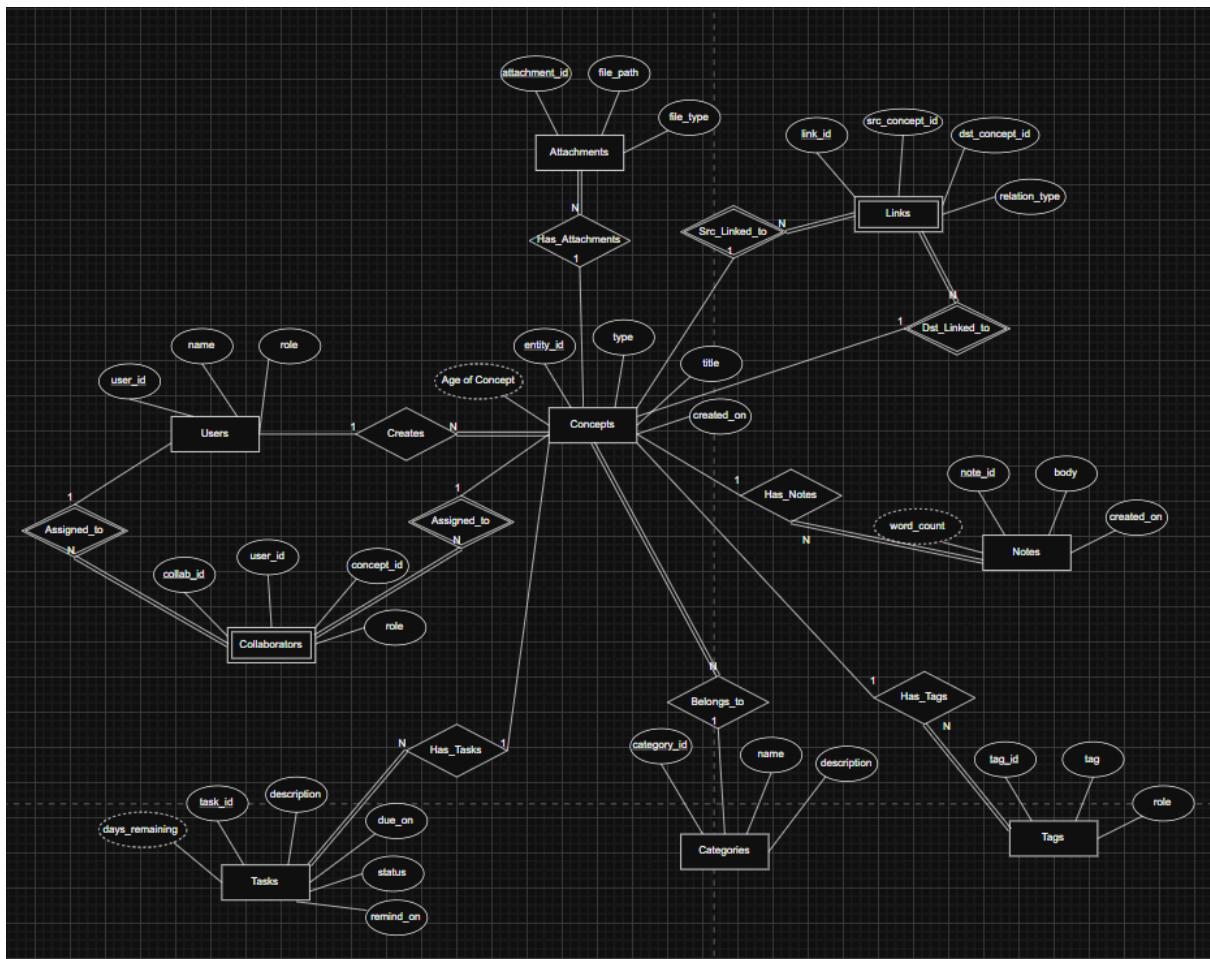


Fig 1: ER Diagram

4.2 Relational Schema

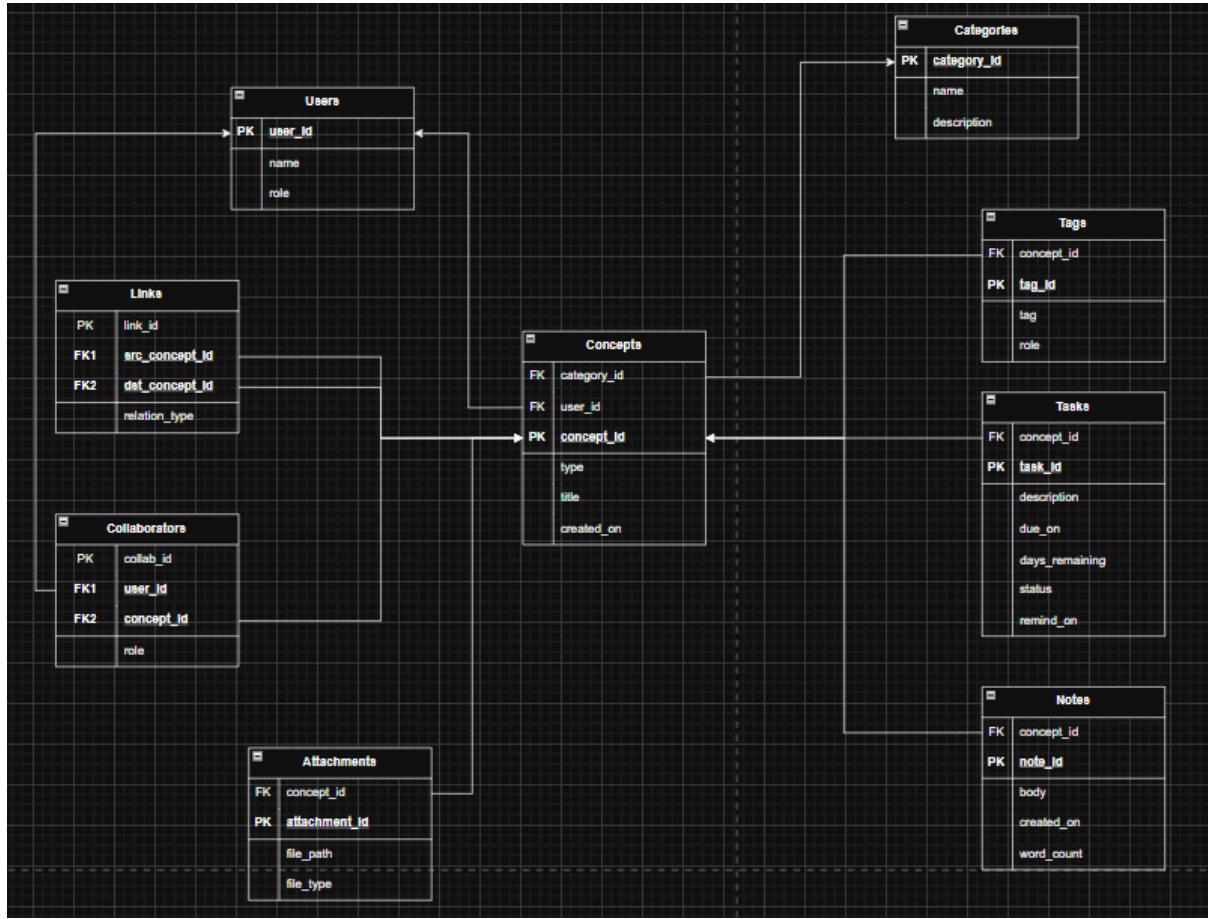


Fig 2: Relational Schema

5. Normalization

All tables in this project follow Third Normal Form (3NF) to avoid redundancy and ensure data consistency.

- 1NF: Each attribute holds atomic values. For example, every Task has a single description such as `due_date`, and `status`.
- 2NF: All non-key attributes depend fully on the primary key. In Tasks, every column depends only on `task_id`, and in Concepts, on `entity_id`.
- 3NF: There are no transitive dependencies. Tables like Collaborators, Concept_Tags, and Links connect data through foreign keys instead of repeating information.

This design ensures efficient data updates, prevents duplication, and maintains relational integrity across concepts, tasks, and users.

6. Users Creation & Privileges (With GUI)

Users can be added or removed through the “Manage Users” interface. Each user is assigned a role (Viewer, Contributor, or Editor). When a user is deleted, all their linked collaborations are automatically removed via triggers — ensuring data integrity and consistency across the database.

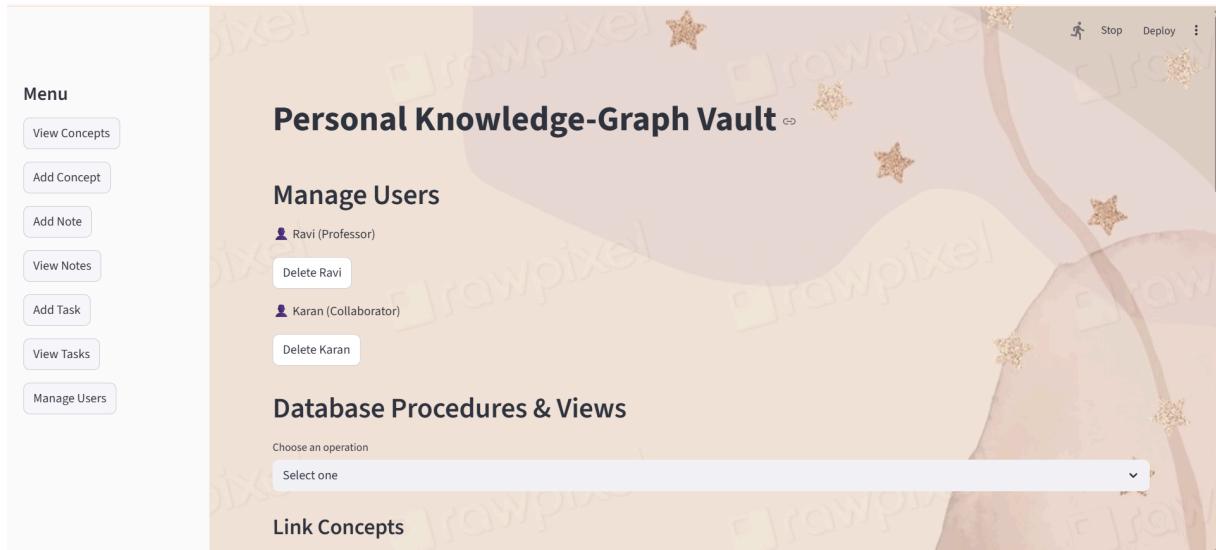


Fig 3: Managing Users

7. Triggers (With GUI Proof)

3 triggers were implemented to automate dependent actions:

- **User Deletion Trigger:** Removes all linked collaborations when a user is deleted.
- **Concept Deletion Trigger:** When a concept is deleted, it deletes related notes, tasks, and attachments automatically.
- **Task Completion Trigger:** Logs into notes automatically when a task’s status changes to “Completed.”

These ensure referential integrity and automatic cascading actions without manual effort, demonstrating real-world automation in database management.

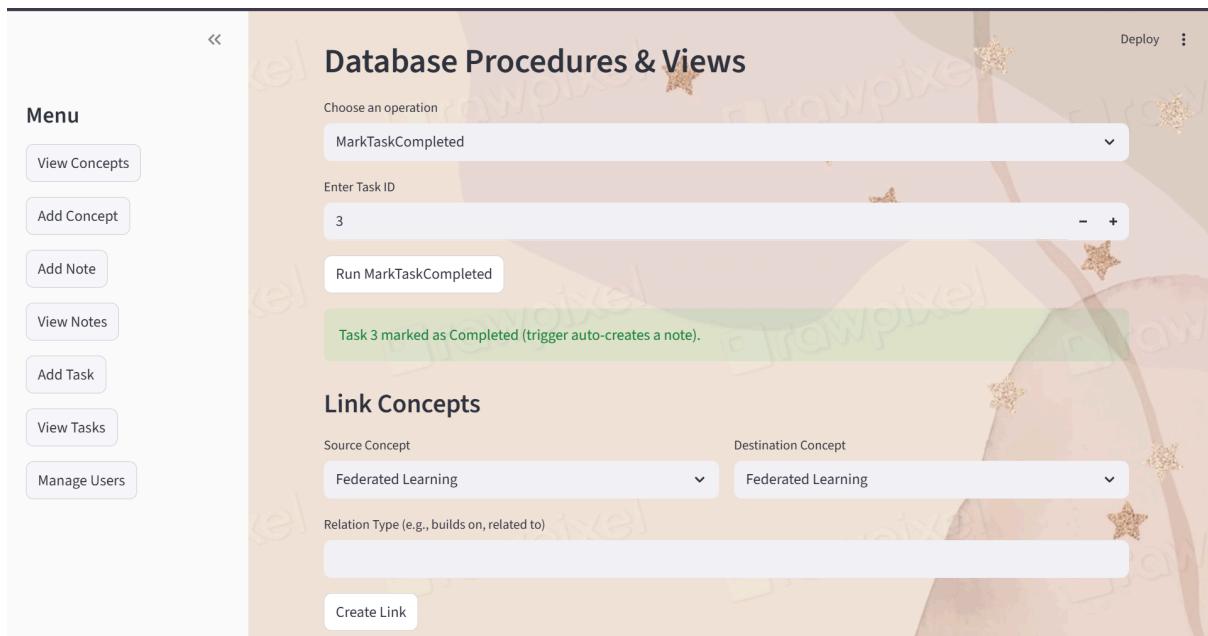


Fig 4: Trigger

8. Stored Procedures & Functions (With GUI)

Stored procedures modularize common database operations:

- **GetConceptDetails(in_entity_id)** – Fetches all notes, tasks, and tags related to a specific concept.
- **GetLinkedConcepts(in_entity_id)** – Retrieves all linked concepts and their relationship types
- **MarkTaskCompleted(in_task_id)** – Updates a task's status to *Completed* and triggers automatic note creation or logging.
- **DaysRemaining(in_task_id)** (*Function*) – Calculates the number of days left until a task's due date.
- **View_Concept_Summary (View)** – Provides a summarized overview of each concept with associated entities for quick access.

All procedures can be executed through Streamlit buttons with displayed results.

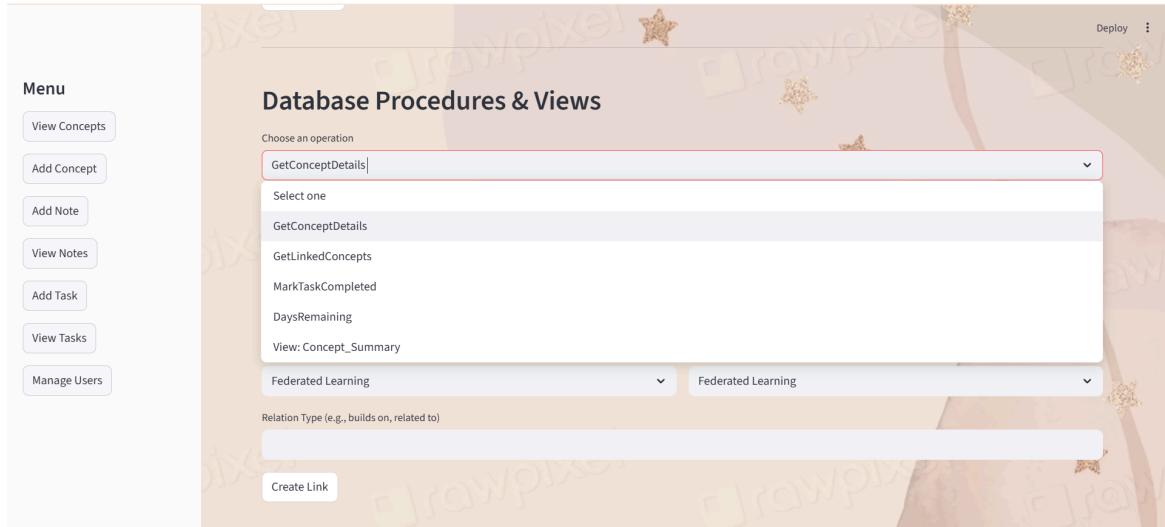


Fig 5: List of all Stored Procedures and Functions

This screenshot shows the same interface as Fig 5, but with a different operation selected. The dropdown now shows "GetConceptDetails". An "Enter Concept ID" input field contains the value "4". Below it, a "Run GetConceptDetails" button is visible. Two tables are displayed: one for notes and one for tasks. The notes table has one row with note_id 9, body "Task \"Integrating Homomorphic Encryption\" completed on 2025-11-12", and created_on "2025-11-12". The tasks table has one row with task_id 13, description "Integrating Homomorphic Encryption", due_on "2025-11-15", status "Completed", and remind_on "2025-11-12". At the bottom, a "Link Concepts" section is shown with "Source Concept" and "Destination Concept" both set to "Federated Learning".

Fig 6: Gets Details of current Concepts

The screenshot shows a web application interface titled "Database Procedures & Views". On the left, a sidebar menu lists various options: View Concepts, Add Concept, Add Note, View Notes, Add Task, View Tasks, and Manage Users. The main content area has a heading "Choose an operation" with a dropdown menu set to "GetLinkedConcepts". Below this is a text input field labeled "Enter Concept ID" containing the value "1". A button labeled "Run GetLinkedConcepts" is present. To the right, a table displays the results of the query:

link_id	related_entity_id	related_title	relation_type
2		Anonymization of Medical Data	Machine Learning
3	2	Membership inference attacks	relation

Below the table, there is a section titled "Link Concepts" with two dropdown menus: "Source Concept" (set to "Federated Learning") and "Destination Concept" (set to "Federated Learning"). A text input field for "Relation Type" is also shown.

Fig 7: Get Details of all the concepts linked to one

The screenshot shows a web application interface titled "Database Procedures & Views". The sidebar menu is identical to Fig 7. The main content area has a heading "Choose an operation" with a dropdown menu set to "DaysRemaining". Below this is a text input field labeled "Enter Task ID" containing the value "3". A button labeled "Run DaysRemaining Function" is present. A message box displays "Days Remaining for Task 3: 14". To the right, there is a section titled "Link Concepts" with two dropdown menus: "Source Concept" (set to "Federated Learning") and "Destination Concept" (set to "Federated Learning"). A text input field for "Relation Type" is also shown. A button labeled "Create Link" is at the bottom.

Fig 8: Shows the days remaining for submitting a task

The screenshot shows a user interface titled "Database Procedures & Views". On the left, there is a vertical menu with buttons for "View Concepts", "Add Concept", "Add Note", "View Notes", "Add Task", "View Tasks", and "Manage Users". The main area is titled "Choose an operation" with a dropdown menu set to "View: Concept_Summary". Below this is a button labeled "Show Concept Summary". A table titled "Concept_Summary" displays three rows of data:

entity_id	title	type	created_on	category	owner	notes_count	tasks_count
1	Federated Learning	Idea	2025-09-10	Research Topics	Alice	3	2
2	Membership inference attacks	Paper	2025-09-12	Research Topics	None	1	1
3	Anonymization of Medical Data	Project	2025-09-15	Projects	Ananya	0	0

Below the table is a section titled "Link Concepts" with fields for "Source Concept" (set to "Federated Learning") and "Destination Concept" (set to "Federated Learning"). There is also a dropdown for "Relation Type (e.g., builds on, related to)". At the bottom of this section is a "Create Link" button.

Fig 9: Concept Summary

9. CRUD Operations (Create, Read, Update, Delete)

All CRUD operations are integrated within the GUI:

- **Create:** Add concepts, notes, tasks, users, and attachments.
- **Read:** View and filter stored information via tables and dashboards.
- **Update:** Change task status or edit related records.
- **Delete:** Remove users or concepts, triggering cascade deletions automatically.

10. Queries Based on Application Functionality

Three key queries demonstrate SQL functionality:

- **Aggregate Query:** Counts notes per concept using COUNT() and GROUP BY.
- **Nested Query:** Finds concepts with more than one note using subqueries.
- **Join Query:** Combines Tasks, Concepts, and Users to show task ownership. These highlight relational power and data insights derived from structured queries.

The screenshot shows the 'Queries Showcase' section of the application. On the left, there is a vertical menu with buttons for 'View Concepts', 'Add Concept', 'Add Note', 'View Notes', 'Add Task', 'View Tasks', and 'Manage Users'. The main area displays three types of queries:

- Aggregate Query: Average Tasks per Concept**: A table showing average tasks per concept. The data is as follows:

title	avg_tasks
Federated Learning	1
Adding HE to Federated Learning	0

- Nested Query: Concepts with More Than 1 Note**: A table showing concepts with more than one note. The data is as follows:

title
Adding HE to Federated Learning

- Join Query: Tasks with Concept and User Info**: A table showing tasks with their corresponding concept and owner. The data is as follows:

description	status	concept	owner
Integrating Homomorphic Encryption	Completed	Federated Learning	Ravi

Fig 10: Queries

11. Screenshots of Frontend

The screenshot shows the 'Personal Knowledge-Graph Vault' interface. On the left, there is a vertical menu with buttons for 'View Concepts', 'Add Concept', 'Add Note', 'View Notes', 'Add Task', 'View Tasks', and 'Manage Users'. The main area displays two concepts:

- Federated Learning (Idea)**: Created on: 2025-09-10. Includes a 'Delete Concept 4' button.
- Adding HE to Federated Learning (Idea)**: Created on: 2025-11-04. Includes a 'Delete Concept 5' button.

Fig 11: View all Concepts/ideas

Link Concepts

Source Concept: Federated Learning

Destination Concept: Adding HE to Federated Learning

Relation Type (e.g., builds on, related to): Machine Learning

Create Link

Linked 'Federated Learning' → 'Adding HE to Federated Learning' as 'Machine Learning'

Existing Links

link_id	source	destination	relation_type
4	Federated Learning	Adding HE to Federated Learning	Machine Learning

Manage Collaborators

Role: Contributor

Add Collaborator

Fig 12: Link Concepts to make a graph

Manage Collaborators

Select User: Ravi

Assign to Concept: Federated Learning

Role: Contributor

Add Collaborator

Current Collaborations

user	concept	role
Ravi	Federated Learning	Contributor

Add Tags to Concepts

Select Tag: AI

Select Concept: Federated Learning

Assign Tag

Fig 13: Manage Collaborators

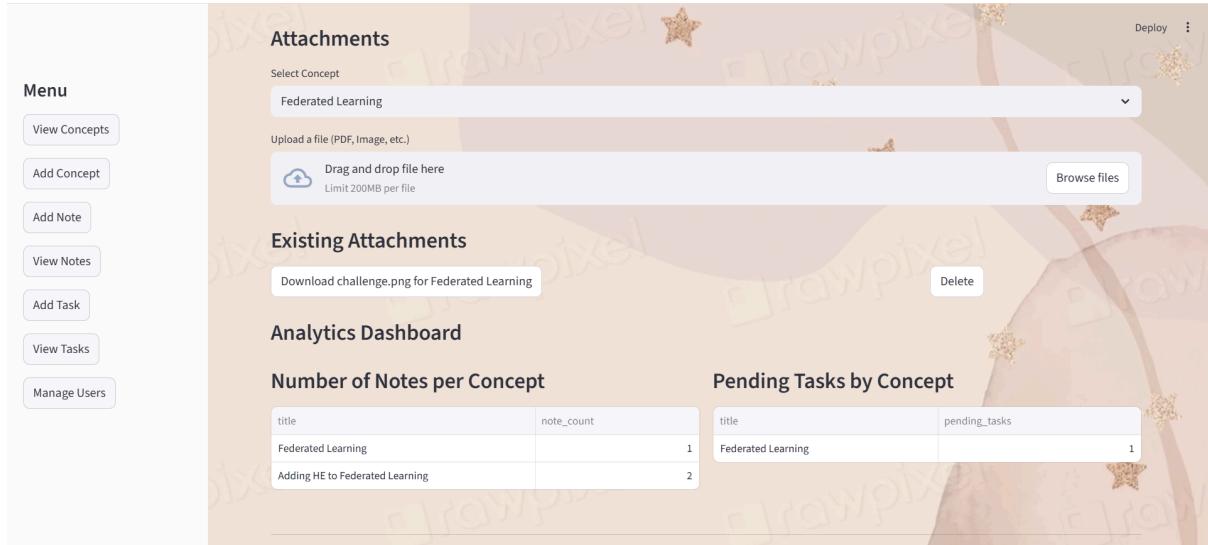


Fig 14: Attachments and Analytics Dashboard

12. Results & Discussion

The Knowledge Graph Vault successfully integrates core DBMS concepts into a practical knowledge management system. It maintains data integrity through triggers, supports modular operations via stored procedures, and offers a smooth, interactive GUI using Streamlit. Users can efficiently manage concepts, notes, tasks, and collaborations — demonstrating how relational design and normalization enable structured, consistent, and meaningful data handling.

13. Conclusion

The project showcases how database-driven systems can organize and interlink information intelligently. By combining SQL features like triggers, functions, and procedures with a user-friendly front end, the system ensures both usability and robustness. Overall, the Knowledge Graph Vault proves effective in demonstrating real-world DBMS implementation through a cohesive, well-normalized, and interactive knowledge management platform.