

Website URL: <http://classwork.engr.oregonstate.edu:6851/>

Executive Summary

Project Evolution and Feedback Integration

The Study Application database project underwent significant refinement from initial concept to final implementation. Our group responded to instructor and peer feedback across five project steps, resulting in a robust, normalized database system supporting flashcard-based learning with full CRUD functionality across seven entities.

Initial design (steps 1-2): Our original proposal focused on basic user-flashcard relationships. Feedback prompted us to expand our M:N relationships, particularly clarifying how study sets could be shared between users and how quiz attempts would be tracked. We restructured our schema to include three junction tables (User_StudySets, User_Quizzes, Quiz_Flashcards) rather than embedding these relationships directly, improving data integrity and flexibility.

Schema Refinement: Step 2 feedback identified potential partial dependencies and inconsistent naming conventions. We unified all naming across ERD, outline, and DDL files, and changed numeric fields from FLOAT to DECIMAL for precise grade tracking. We clarified in our normalization analysis that junction table attributes (role, question_order) depend on composite keys, eliminating concerns about 2NF compliance.

Throughout all the development stages of our Study Application, we implemented a few changes targeted at improving our UI/UX based on the feedback we received, and things we thought might be better overall. One of the major improvements we added was repositioning all the data tables to the top of each page. This allows users to immediately view their existing records before performing any CREATE actions, which aligns with any standard database application and their design patterns. We also moved the RESET button into the main navigation bar so that it visually and functionally matched the rest of the site's controls, buttons, and theme overall. In addition to those changes, we refined the RESET workflow by adding a confirmation popup before its execution and a success popup after the action was completed. This was then followed by an automatic page refresh, rather than directing users to a different page. Finally, we improved the date formatting in the Users table by removing the unnecessary time component (00:00:00), which makes the join date much easier to read and more meaningful to the users.

AI tool usage - Strengths and Weaknesses:

Its strengths lie in generating bulky mundane code like a CSS style sheet, effectively suggesting improvements in lieu of peer feedback, and offering explanations of complex concepts. Its weaknesses lie in overcomplicating solutions, the constant oversight that was needed.

The final implementation exceeds minimum requirements with full CRUD operations across all seven entities, two UPDATE M:N operations, three DELETE M:N operations, and a functioning database RESET feature using stored procedures. The application successfully handles dynamic foreign key selection through drop downs, maintains referential integrity through cascading deletions, and provides immediate user feedback through client-side JavaScript.

Key lessons learned include the importance of maintaining consistent naming conventions across all documentation, the value of peer feedback in identifying usability issues, and the necessity of thoroughly understanding database schema before implementing logic.

Study Application

Group 69: Sasan Pourassef, Jeremy Dempsey

Website URL: <http://classwork.engr.oregonstate.edu:6851/>

Overview

Students from various universities and schools can often struggle to organize their study materials through all of their different subjects they have to track. Many students rely on scattered tools, a combination of Google Docs, Quizlet, and even physical flashcards in order to prepare and practice for exams. This results in duplication, lost progress data, and very limited personalization.

Our study app aims to create a database driven study website that allows users to create, review, and track flashcards, quizzes, and topic-filtered vocabulary lists all in one place. Each user will be able to keep multiple study sets for different modules and topics, which allows a streamlined organization and an easy way to get learning materials. This system will support hundreds of users, with each maintaining dozens of study sets across various topics.

The platform will be able to handle up to 10,000 flashcards, 5,000 quiz attempts, and over 1,000 sessions at a time, ensuring scalability for classroom or institutional use. The platform allows users to track their learning progress and performance across quizzes, offering both personalization and analytics based on insights.

In this initial phase, we will focus on implementing core user functionality: flashcard creation, quiz generation, and progress tracking. These features will be built on a robust, scalable relational database structure that can later support advanced features such as analytics, collaborative study groups or adaptive learning recommendations.

Outline

1. Users

- Purpose: store registered student accounts and their profile details
- Attributes:
 - user_id (INT, PK, auto_increment, not NULL)
 - username (VARCHAR(50), unique, not NULL)
 - email (VARCHAR(100), unique, not NULL)
 - password_hash (VARCHAR(255), not NULL)
 - join_date (DATE, not NULL)
- Relationships:
 - 1:M & M:N with StudySets (a user can create and share many sets, and sets may be shared by multiple users. The user doesn't have to share their study sets)
 - 1:M with quizzes (a user can take many quizzes)

2. StudySets

- Purpose: A named collection of flashcards grouped by topic or course
- Attributes:
 - set_id (INT, PK, auto_increment, not NULL)
 - user_id (INT, FK -> users.user_id, not NULL)
 - title (VARCHAR(100), not NULL)
 - subject (VARCHAR(50), NULL)
 - created_at (DATETIME, not NULL)
- Relationships:
 - 1:M with Flashcards (each set contains many flashcards)
 - M:N with Quizzes (a quiz may include flashcards from multiple sets)
 - M:N with Users (allows study set sharing between multiple users)

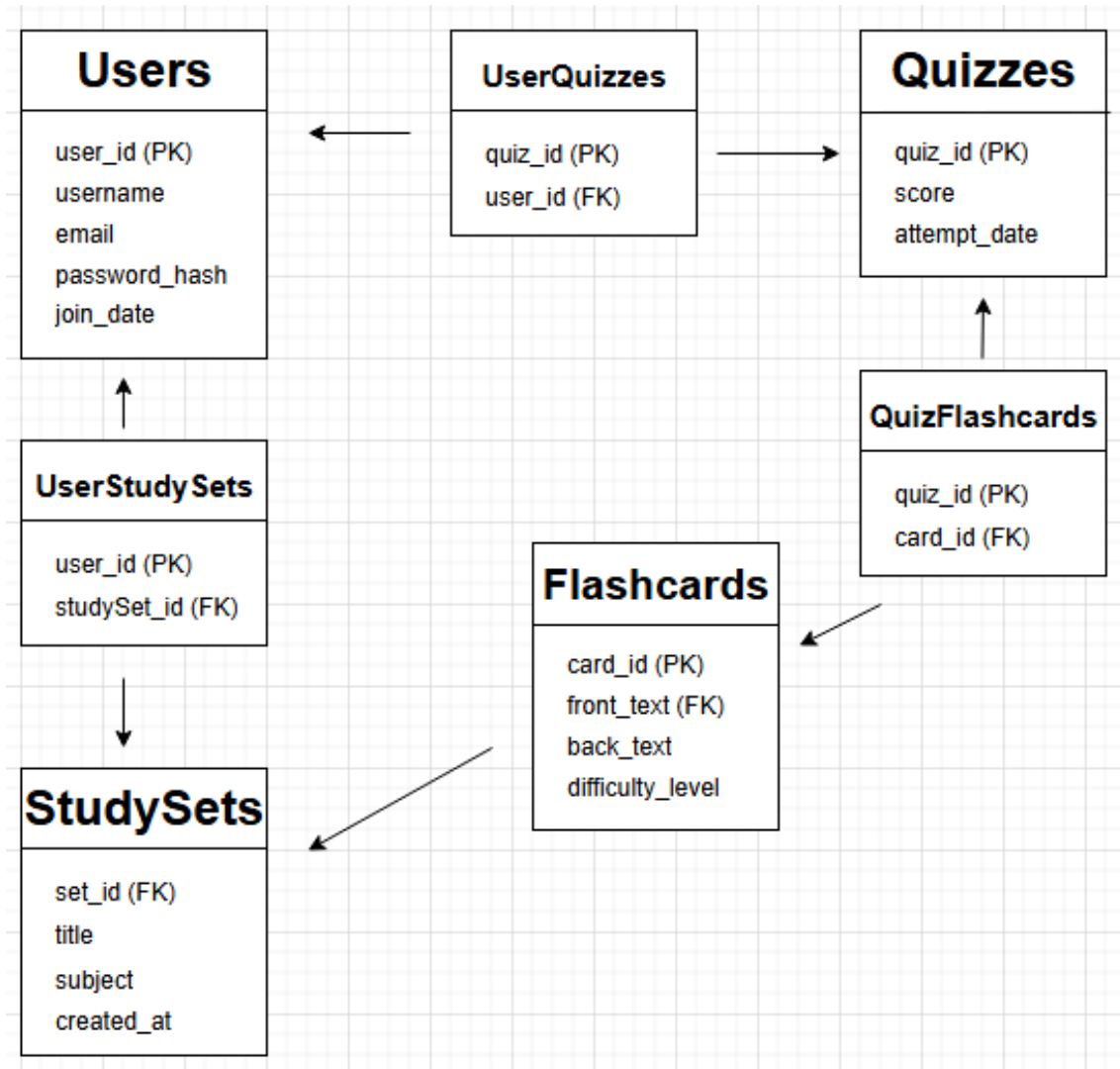
3. Flashcards

- Purpose: Store individual study questions and answers.
- Attributes:
 - card_id (INT, PK, auto_increment, not NULL)
 - set_id (INT, FK -> StudySets.set_id, not NULL)
 - front_text (TEXT, not NULL)
 - back_text (TEXT, not NULL)
 - difficulty_level (ENUM('easy', 'medium', 'hard'))
- Relationships:
 - Belongs to one StudySet
 - M:N with Quizzes (flashcards can appear in multiple quizzes)

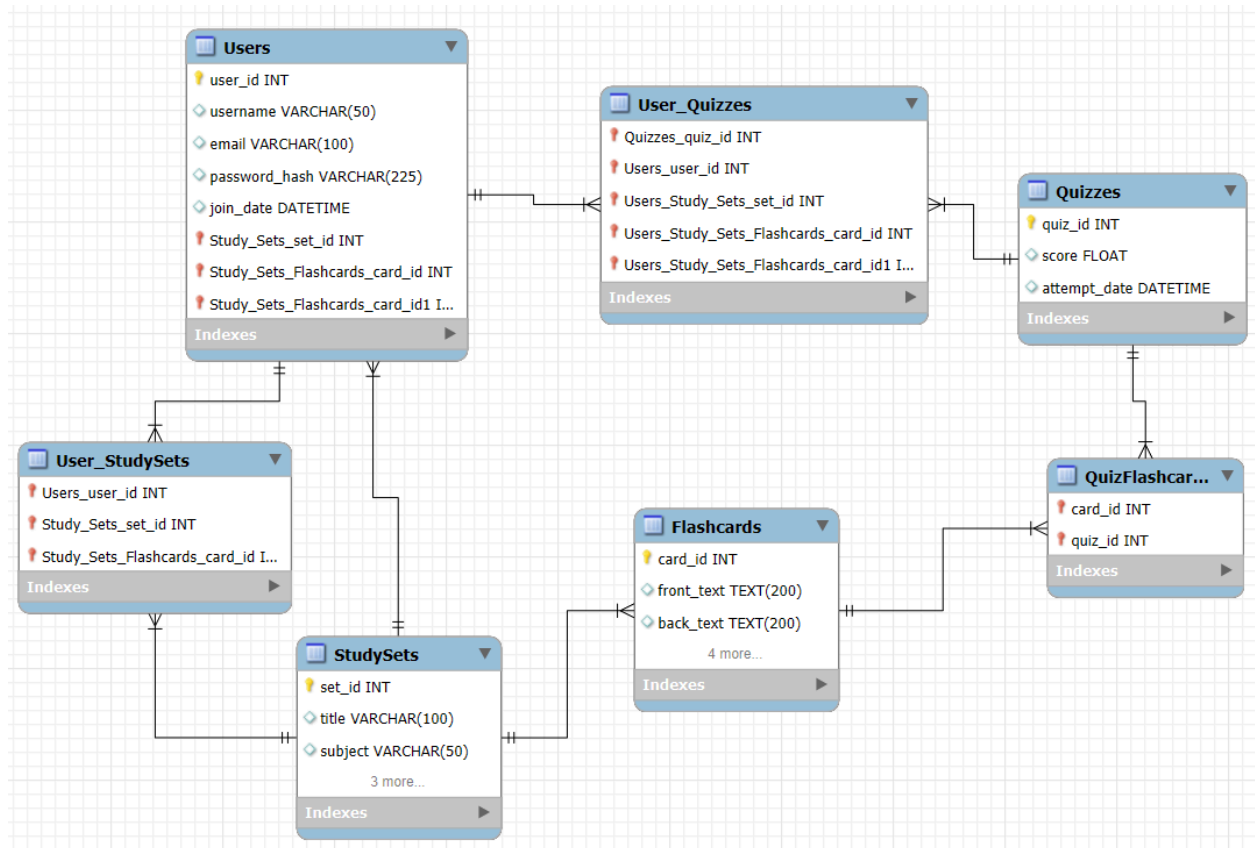
4. Quizzes

- Purpose: Represent generated quizzes, including metadata about when they were taken.
- Attributes:
 - quiz_id (INT, PK, auto_increment, not NULL)
 - user_id (INT, FK -> Users.user_id, not NULL)
 - score (FLOAT(4,2), NULL)
 - attempt_date (DATETIME, not NULL)
- Relationships:
 - M:N with Flashcards (intersection table: QuizFlashcards)
 - M:N with Users (through a linking table: User_Quizzes)

Entity-Relationship Diagram



Schema Diagram



Normalization Analysis

Our database schema has been analyzed using the normalization process:

- 1NF (first normal form)
 - All tables have no repeating groups
 - Each table has a primary key
 - No duplicate rows exist
- 2NF (second normal form)
 - Schema meets 1NF requirements
 - All non-key attributes are functionally dependent on the primary key
 - No partial dependencies exist
 - All tables with composite keys have attributes dependent on the entire key
 - In Quiz_Flashcards, question_order depends on (quiz_id, card_id) composite key
 - In User_StudySets, role depends on (user_id, set_id) composite key
- 3NF (third normal form)
 - Schema meets 2NF requirements

- No transitive dependencies exist
- All non-key attributes depend only on the primary key
- Potential transitive dependencies examined:
 - StudySets.subject could theoretically depend on title, but we allow multiple sets with same title under different users, so no transitive dependency exists
 - Quizzes.score depends only on quiz_id, not on any intermediate attributes
 - All foreign keys represent relationships, not transitive dependencies

Conclusion: Our schema is in 3NF. No denormalization was required as all relationships are properly modeled and data integrity is maintained without introducing redundancy or anomalies. There is no partial dependency in Users, StudySets, or Flashcards as each table non-key attributes depend on its single-column PK; and junctions have composite PKs and their non-key attributes, such as role or question_order, where they depend on the full composite.

Example data (from DDL.sql):

Users Table:

user_id	username	email	password_hash	join_date
1	sasan	sasan@example.com	hash_sasan	2025-09-01
2	jeremy	jeremy@example.com	hash_jeremy	2025-09-03
3	john	john@example.com	hash_john	2025-09-10

StudySets Table:

set_id	user_id	title	subject	created_at
1	1	CS340 - SQL Basics	Databases	2025-09-15 09:30:00
2	1	Biology - Cell Terms	Biology	2025-09-15 14:00:00
3	2	Algorithms - Sorting	CS	2025-09-25 16:45:00

Flashcards Table:

card_id	set_id	front_text	back_text	difficulty_level
1	1	What is a primary key?	A column or set of columns that uniquely identify a row.	easy
2	1	Define foreign key.	A field referencing a PK in another table; enforces referential integrity	easy
3	1	What does ON DELETE CASCADE do?	Deletes child rows when the parent is deleted.	medium

4	2	Mitochondria function?	Powerhouse of the cell; ATP production.	easy
5	3	Average time complexity of merge sort?	$O(n \log n)$	medium

Quizzes table:

quiz_id	user_id	score	attempt_date
1	1	95.00	2025-10-01 10:00:00
2	2	88.50	2025-10-02 13:20:00

User_StudySets table (demonstrates M:N relationship - sharing study sets):

user_id	set_id	role
2	1	viewer
3	1	editor

User_Quizzes table (demonstrates M:N relationship - users and quizzes):

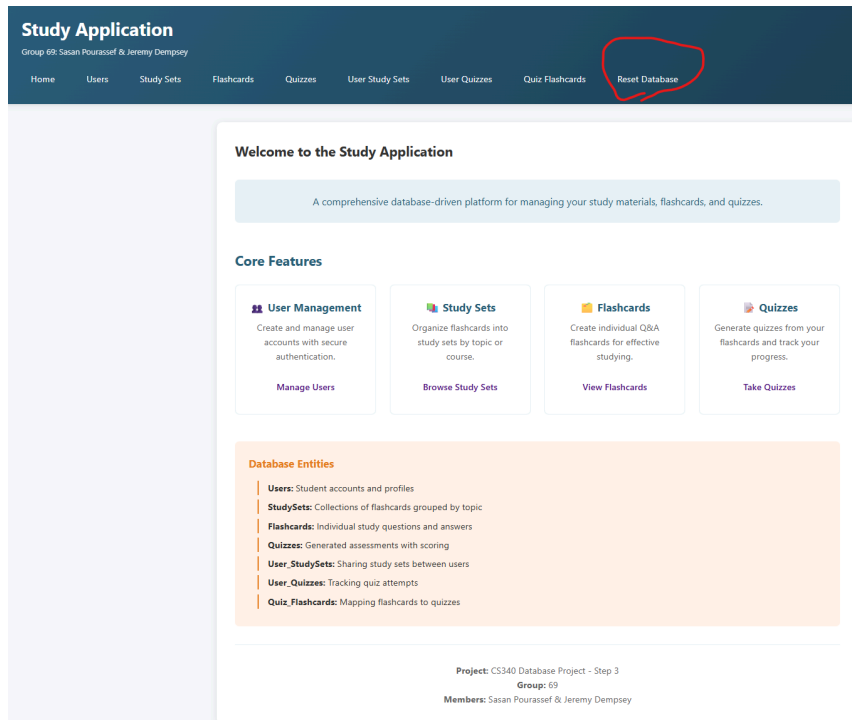
user_id	quiz_id
1	1
2	1
2	2

Quiz_Flashcards table(demonstrates M:N relationship - flashcards in quizzes):

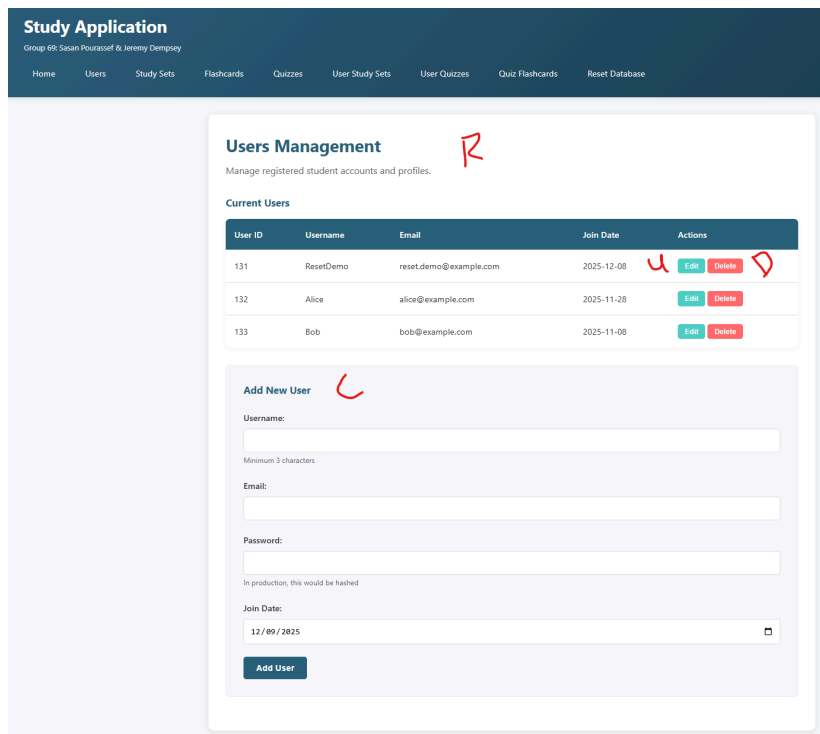
quiz_id	card_id	question_order
1	1	1
1	2	2
1	3	3
2	5	1

UI Screenshots:

Index page: Read (RESET button available on every page at the top of the page)



Users page: Create, Read, Update, Delete



Study Sets page: CRUD

Study Application

Group 69: Sasan Pourassef & Jeremy Dempsey

[Home](#)[Users](#)[Study Sets](#)[Flashcards](#)[Quizzes](#)[User Study Sets](#)[User Quizzes](#)[Quiz Flashcards](#)[Reset Database](#)

Study Sets Management

Manage collections of flashcards grouped by topic or course.

Existing Study Sets

Set ID	Title	Subject	Owner	Created Date	Actions
91	SQL Basics	Databases	ResetDemo	Mon Dec 08 2025 16:33:05 GMT-0800 (Pacific Standard Time)	Edit Delete
92	Biology 101	Biology	Alice	Mon Dec 08 2025 16:33:05 GMT-0800 (Pacific Standard Time)	Edit Delete

Create New Study Set

Owner (User):

-- Select Owner --

Who owns this study set?

Title:

Name of the study set (e.g., "CS340 - SQL Basics")

Subject:

Optional subject or course name

Created Date:

12/08/2025 06:00 PM

Create Study Set

Flashcards page: CRUD

Study Application

Group 69: Sasan Pourassef & Jeremy Dempsey

[Home](#)[Users](#)[Study Sets](#)[Flashcards](#)[Quizzes](#)[User Study Sets](#)[User Quizzes](#)[Quiz Flashcards](#)[Reset Database](#)

Flashcards Management

Create and manage individual study questions and answers.

Existing Flashcards

Card ID	Study Set	Front Text	Back Text	Difficulty	Actions
133	SQL Basics	What does SELECT do?	Reads rows from a table	EASY	Edit Delete
134	SQL Basics	What is a FK?	Ref to a PK in another table	MEDIUM	Edit Delete
135	Biology 101	Cell organelle?	Mitochondria	EASY	Edit Delete

Create New Flashcard

Study Set:

-- Select Study Set --

Which study set does this flashcard belong to?

Front (Question):

The question or prompt

Back (Answer):

The answer or explanation

Difficulty Level:

Easy

Create Flashcard

Quiz page: CRUD

Study Application

Group 69: Sasan Pourassef & Jeremy Dempsey

Home Users Study Sets Flashcards Quizzes User Study Sets User Quizzes Quiz Flashcards Reset Database

Quizzes Management

Manage quiz attempts and track student performance.

Quiz History

Quiz ID	User	Score	Attempt Date	Actions
90	ResetDemo	95%	Mon Dec 08 2025 16:33:05 GMT-0800 (Pacific Standard Time)	Edit Delete
91	Alice	88.5%	Sun Dec 07 2025 16:33:05 GMT-0800 (Pacific Standard Time)	Edit Delete

Create New Quiz

Quiz Taker (User):

-- Select User --

Who is taking this quiz?

Score:

Leave blank if quiz is not yet graded (0-100)

Attempt Date:

12 / 08 / 2025 06 : 15 PM

Create Quiz

User-Studysets page: CRUD, M:N UPDATE/DELETE

Study Application

Group 69: Sasan Pourassef & Jeremy Dempsey

Home Users Study Sets Flashcards Quizzes User Study Sets User Quizzes Quiz Flashcards Reset Database

User Study Sets - Shared Access

Manage which users have access to which study sets. Share study sets with other users to collaborate.

Current Shared Study Sets

ID	User	Study Set	Role	Actions
132-92	Alice	Biology 101	editor	Remove Access
133-91	Bob	SQL Basics	viewer	Remove Access

Share Study Set with User

Select User: [Choose a user...](#) Select Study Set: [Choose a study set...](#) Role: [Viewer](#) [Share Study Set](#)

Note

To modify access, remove the existing relationship and create a new one if needed.

Update User Role (UPDATE M:N) ★

Select User-Set Relationship: [Choose a relationship...](#) New Role: [Viewer](#) [Update Role](#)

User-Quizzes page: CRD

Study Application

Group 69: Sasan Pourassef & Jeremy Dempsey

[Home](#)[Users](#)[Study Sets](#)[Flashcards](#)[Quizzes](#)[User Study Sets](#)[User Quizzes](#)[Quiz Flashcards](#)[Reset Database](#)

User Quiz Attempts

Track quiz attempts by users, including scores and completion dates. View quiz history and performance.

Quiz Attempt History

Attempt ID	User	Quiz	Score	Date Taken	Actions
133-91	Bob	91	88.5%	Sun Dec 07 2025 16:33:05 GMT-0800 (Pacific Standard Time)	Delete Attempt

Record Quiz Attempt

Select User: Select Quiz: [Record Quiz Attempt](#)

CS340 Database Project - Oregon State University

Quiz-Flashcards page: CRUD, M:N UPDATE/DELETE

Study Application

Group 69: Sasan Pourasaf & Jeremy Dempsey

[Home](#)
[Users](#)
[Study Sets](#)
[Flashcards](#)
[Quizzes](#)
[User Study Sets](#)
[User Quizzes](#)
[Quiz Flashcards](#)
[Reset Database](#)

Quiz Flashcard Assignments

Manage which flashcards are included in each quiz. Build quizzes by assigning flashcards from study sets.

Current Quiz Flashcard Assignments

Assignment ID	Quiz	Flashcard Question	Study Set	Actions
90-133	90	What does SELECT do?	SQL Basics	Remove from Quiz
90-134	90	What is a FK?	SQL Basics	Remove from Quiz
91-135	91	Cell organelle?	Biology 101	Remove from Quiz

Add Flashcard to Quiz

Select Quiz: Select Flashcard: [Add Flashcard to Quiz](#)

View by Quiz

Filter by Quiz:

Tip: Flashcards must belong to a study set before they can be added to a quiz. Create flashcards in the Flashcards page first.

Update Question Order (UPDATE M:N)

Select Quiz-Flashcard: New Question Order: [Update Order](#)

Citations

All core work on the project is original and created by Sasan Pourassef and Jeremy Dempsey.
AI was used to generate example data and to generate the base style.css.