

7. DATABASE

7.1 Database Overview

The Study in Woods platform uses PostgreSQL 15.x as its primary relational database management system, storing all persistent application data across 30+ tables. The database schema follows normalization principles (3NF) to minimize data redundancy while maintaining referential integrity through foreign key constraints. GORM (Go Object Relational Mapping) library manages database operations, automatic migrations, and relationship handling.

7.2 Database Schema Summary

Category	Tables	Purpose
User Management	users, jwt_token_blacklist	Authentication, authorization, session management
Academic Hierarchy	universities, courses, semesters, subjects	Educational structure and curriculum organization
Document Management	documents, syllabuses, syllabus_units, syllabus_topics	File storage, syllabus extraction, content indexing
Chat System	chat_sessions, chat_messages, chat_memories, chat_compacted_contexts	AI conversations, context management, memory optimization
PYQ System	pyq_papers, pyq_questions, pyq_question_choices, pyq_crawler_sources, pyq_crawled_papers	Previous year questions management and crawling
System & Audit	api_keys, api_key_usage_logs, user_activities, admin_audit_logs, app_settings	API management, activity tracking, configuration
Background Jobs	indexing_jobs, indexing_job_items, cron_job_logs	Asynchronous processing, job scheduling
User Engagement	user_notifications, user_courses	Notifications, enrollment tracking

7.3 Core Table Definitions

7.3.1 *users*

Primary table for user authentication and profile management.

Column	Type	Key/Constraint
id	SERIAL	PRIMARY KEY
email	VARCHAR(255)	UNIQUE, NOT NULL
password_hash, password_salt	VARCHAR(255), BYTEA	NOT NULL (bcrypt)
name	VARCHAR(255)	NOT NULL
role	VARCHAR(20)	DEFAULT 'student' (student/admin)
semester, token_version	INTEGER	DEFAULT 1, 0
created_at, updated_at, deleted_at	TIMESTAMP	Soft delete support

Relations: Has many ChatSessions, ChatMessages, UserCourses, AdminAuditLogs

7.3.2 Academic Hierarchy (*universities -> courses -> semesters -> subjects*)

Table	Key Columns	Foreign Key
universities	id, name, code (UNIQUE), location, is_active	-
courses	id, name, code (UNIQUE), duration, description	university_id -> universities(id)
semesters	id, number, name	course_id -> courses(id)
subjects	id, name, code, credits, knowledge_base_uuid, agent_uuid	semester_id -> semesters(id)

All tables include created_at, updated_at timestamps. CASCADE delete propagates through hierarchy.

7.3.3 documents

Stores uploaded PDFs and tracks indexing status with DigitalOcean Knowledge Base.

Column	Type	Purpose
id	SERIAL	PRIMARY KEY
subject_id	INTEGER	FK -> subjects(id)
type	VARCHAR(20)	'syllabus', 'pyq', 'book', 'reference', 'notes'
filename, file_size, page_count	VARCHAR, BIGINT, INT	File metadata
spaces_url, spaces_key	TEXT, VARCHAR	DigitalOcean Spaces storage
data_source_id, indexing_job_id	VARCHAR(100)	Knowledge Base integration
indexing_status	VARCHAR(20)	'pending', 'in_progress', 'completed', 'failed'

7.3.4 Syllabus Structure (*syllabuses* -> *syllabus_units* -> *syllabus_topics*)

Table	Key Columns	Foreign Key
syllabuses	id, subject_name, subject_code, total_credits, extraction_status, raw_extraction	subject_id, document_id
syllabus_units	id, unit_number, title, description, hours	syllabus_id -> syllabuses(id)
syllabus_topics	id, topic_number, title, description, keywords	unit_id -> syllabus_units(id)

7.3.5 Chat System

chat_sessions

Column	Type	Key/Constraint
id	SERIAL	PRIMARY KEY
user_id	INTEGER	FK -> users(id)
subject_id	INTEGER	FK -> subjects(id)
title	VARCHAR(255)	Auto-generated or user-set

chat_messages

Column	Type	Purpose
id, session_id, subject_id, user_id	INTEGER	PK and Foreign Keys
role	VARCHAR(20)	'user', 'assistant', 'system'
content	TEXT	Message content
citations	JSONB	Knowledge Base citations array
tokens_used, model_used, response_time	INT, VARCHAR, INT	Usage analytics
is_streamed	BOOLEAN	SSE streaming flag

Additional Chat Tables: chat_memories (conversation context), chat_memory_batches (batch processing), chat_compacted_contexts (compressed long-term memory)

7.3.6 System & Audit Tables

Table	Key Columns	Purpose
api_keys	id, user_id, key_hash, name, is_active	Encrypted API key storage
api_key_usage_logs	id, user_id, service, endpoint, status_code	API consumption tracking
user_activities	id, user_id, action, resource_type, resource_id, ip_address	User action tracking
admin_audit_logs	id, admin_id, action, target_type, target_id, changes (JSONB)	Admin action audit trail
app_settings	id, key (UNIQUE), value, description	Application configuration
jwt_token_blacklist	id, user_id, token_hash (UNIQUE), expires_at	Invalidated token tracking

7.4 Entity Relationships

The database follows a hierarchical structure with cascading relationships:

Primary Relationships:

- universities (1) -> (M) courses -> (M) semesters -> (M) subjects
- subjects (1) -> (M) documents, syllabuses, chat_sessions
- users (1) -> (M) chat_sessions -> (M) chat_messages
- syllabuses (1) -> (M) syllabus_units -> (M) syllabus_topics
- users (1) -> (M) api_keys, user_activities, admin_audit_logs

ER Diagram Reference: The complete Entity-Relationship diagram is available in the Design section (Chapter 6), showing all 30+ tables with their relationships, cardinality, and key constraints.

7.5 Indexes and Performance

Strategic indexing optimizes query performance:

- **B-tree indexes:** All foreign keys (user_id, subject_id, session_id, course_id, semester_id)
- **Unique indexes:** Email addresses, codes, token hashes
- **Composite indexes:** (user_id, created_at) for user activity queries
- **Partial indexes:** indexing_status='pending' for background job processing
- **GIN indexes:** JSONB columns (citations, metadata) for @> containment queries

Connection Management: pgx driver maintains 25-100 concurrent connections with 1-hour max lifetime. Query optimization includes prepared statement caching, GORM preloading to prevent N+1 queries, and Redis caching with 5-minute TTL.