

Data Dictionary

USERS

Description: Central authentication and identity table. Stores credentials, basic profile, and role.

Columns

Column	Type	Constraint	Purpose
id	INTEGER	PK, AUTOINCREMENT	Surrogate primary key for user record
name	TEXT	NOT NULL	Full display name
email	TEXT	UNIQUE, NOT NULL	Login email and unique identifier for contact
password_hash	TEXT	NOT NULL	Hashed password
role	TEXT	NOT NULL, CHECK IN (instructor, student, admin)	Primary role used for authorization
is_active	INTEGER	DEFAULT 1, CHECK IN (0,1)	Account active status flag (boolean)
created_at	TEXT	DEFAULT CURRENT_TIMESTAMP	Account creation timestamp
updated_at	TEXT		Last profile update timestamp

Notes: Enforce email uniqueness. The is_active field supports account deactivation without deletion (0=inactive, 1=active).

INSTRUCTORS

Description: Instructor profile (1:1 with USERS where role=instructor).

Columns

Column	Type	Constraint	Purpose
id	INTEGER	PK, AUTOINCREMENT, FK Users(id)	Surrogate primary key, links to users record
instructor_code	TEXT	UNIQUE	Human/legacy code for instructor
bio	TEXT		Instructor biography
office_hours	TEXT		Office hours availability schedule

Notes: id serves as both primary key and foreign key to USERS, ensuring 1:1 relationship.

STUDENTS

Description: Student profile (1:1 with USERS where role=student).

Columns

Column	Type	Constraint	Purpose
id	INTEGER	PK, AUTOINCREMENT, FK Users(id)	Surrogate primary key, links to users record
student_number	TEXT	UNIQUE, NOT NULL	Institutional student ID
program	TEXT		Academic program/major
year_level	INTEGER		Current year level in program

Notes: id serves as both primary key and foreign key to USERS. Use student_number to reference external academic systems.

ADMINS

Description: Admin profile (1:1 with USERS where role=admin).

Columns

Column	Type	Constraint	Purpose
id	INTEGER	PK, FK Users(id)	Surrogate primary key, links to users record
privileges	TEXT	JSON stored as TEXT	Array/list of admin privilege strings

Notes: id serves as both primary key and foreign key to USERS. privileges field stores granular permissions as JSON for role-based access control.

NOTIFICATIONS

Description: In-app or system notifications for users.

Columns

Column	Type	Constraint	Purpose
id	INTEGER	PK, AUTOINCREMENT	Notification ID
user_id	INTEGER	FK Users(id), NOT NULL	Recipient user
message	TEXT	NOT NULL	Notification body
type	TEXT	CHECK IN (info, warning, success, error)	Category
is_read	INTEGER	DEFAULT 0, CHECK IN (0,1)	Read flag (boolean)
link	TEXT		Optional URL for notification action
created_at	TEXT	DEFAULT CURRENT_TIMESTAMP	When created
read_at	TEXT		When user read it

Notes: Index user_id for quick retrieval. link field enables deep linking to relevant pages. is_read: 0=unread, 1=read.

COURSES

Description: Course metadata owned by an instructor.

Columns

Column	Type	Constraint	Purpose
id	INTEGER	PK, AUTOINCREMENT	Course surrogate ID
instructor_id	INTEGER	FK Instructors(id), NOT NULL	Owner/teacher
code	TEXT	UNIQUE, NOT NULL	Human-readable course code
title	TEXT	NOT NULL	Course title
description	TEXT		Long description
semester	TEXT		Semester/term identifier
year	INTEGER		Academic year
status	TEXT	CHECK IN (draft, active, archived)	Course lifecycle state
max_students	INTEGER		Maximum enrollment capacity
created_at	TEXT	DEFAULT CURRENT_TIMESTAMP	When created
updated_at	TEXT		Last update time

Notes: Consider composite unique constraint on (code, semester, year) for uniqueness across terms if needed. max_students supports enrollment caps.

ENROLLMENTS

Description: Junction table for many-to-many student-course enrollments.

Columns

Column	Type	Constraint	Purpose
id	INTEGER	PK, AUTOINCREMENT	Enrollment surrogate ID
student_id	INTEGER	FK Students(id), NOT NULL	Student ID
course_id	INTEGER	FK Courses(id), NOT NULL	Course ID
status	TEXT	CHECK IN (active, dropped, completed)	Enrollment state
final_grade	REAL		Final course grade
enrolled_at	TEXT	DEFAULT CURRENT_TIMESTAMP	Enrollment time
dropped_at	TEXT		Drop time if any

Notes: Consider UNIQUE constraint on (student_id, course_id) to prevent duplicate enrollments. final_grade stores computed course grade.

ASSIGNMENTS

Description: Assignment tasks within a course.

Columns

Column	Type	Constraint	Purpose
id	INTEGER	PK, AUTOINCREMENT	Assignment ID
course_id	INTEGER	FK Courses(id), NOT NULL	Owning course
title	TEXT	NOT NULL	Assignment title
description	TEXT		Full assignment text
release_date	TEXT		When made available
due_date	TEXT	NOT NULL	Deadline
max_points	INTEGER	DEFAULT 100	Maximum points
late_submission_penalty	REAL	DEFAULT 0.0	Penalty percentage for late submissions
allow_late_submissions	INTEGER	DEFAULT 0, CHECK IN (0,1)	Whether late submissions accepted (boolean)
is_published	INTEGER	DEFAULT 0, CHECK IN (0,1)	Visibility flag (boolean)
created_at	TEXT	DEFAULT CURRENT_TIMESTAMP	Created time
updated_at	TEXT		Last update time

Notes: late_submission_penalty can be percentage (0.0-1.0) or points deducted. Booleans: 0=false, 1=true.

TEST_CASES

Description: Per-assignment test cases for autograding.

Columns

Column	Type	Constraint	Purpose
id	INTEGER	PK, AUTOINCREMENT	Test case ID
assignment_id	INTEGER	FK Assignments(id), NOT NULL	Which assignment this belongs to
name	TEXT	NOT NULL	Short test case name
description	TEXT		Detailed test case description
stdin	TEXT		Standard input for the test
expected_output	TEXT	NOT NULL	Expected output to compare
timeout_ms	INTEGER	DEFAULT 5000	Max runtime in milliseconds

Column	Type	Constraint	Purpose
memory_limit_mb	INTEGER	DEFAULT 256	Maximum memory limit in megabytes
points	INTEGER	DEFAULT 0	Points if test passed
is_visible	INTEGER	DEFAULT 1, CHECK IN (0,1)	Student-visible flag (boolean)
sort_order	INTEGER	DEFAULT 0	Ordering for display or scoring
created_at	TEXT	DEFAULT CURRENT_TIMESTAMP	Created time

Notes: expected_output often needs normalization (trimmed comparison rules). is_visible: 0=hidden, 1=visible to students.

SUBMISSIONS

Description: Student code submissions for assignments.

Columns

Column	Type	Constraint	Purpose
id	INTEGER	PK, AUTOINCREMENT	Submission ID
assignment_id	INTEGER	FK Assignments(id), NOT NULL	Which assignment this belongs to
student_id	INTEGER	FK Students(id), NOT NULL	Submitting student
version	INTEGER	DEFAULT 1	Incrementing version number
language	TEXT	CHECK IN (python, java, cpp, javascript, c)	Submission language
status	TEXT	CHECK IN (pending, queued, running, graded, failed, error)	Submission status
score	REAL	DEFAULT 0.0	Final score
is_late	INTEGER	DEFAULT 0, CHECK IN (0,1)	Late flag (boolean)
created_at	TEXT	DEFAULT CURRENT_TIMESTAMP	Created time
updated_at	TEXT		Last modification time
graded_at	TEXT		When grading completed

Notes: Index (assignment_id, student_id) for queries like "latest submission per student". updated_at tracks re-grading. is_late: 0=on time, 1=late.

RESULTS

Description: Result rows from running a submission against test cases (one row per submission/test-case execution).

Columns

Column	Type	Constraint	Purpose
id	INTEGER	PK, AUTOINCREMENT	Result ID
submission_id	INTEGER	FK Submissions(id), NOT NULL	Related submission
test_case_id	INTEGER	FK Test_cases(id), NOT NULL	Test case executed
passed	INTEGER	NOT NULL, CHECK IN (0,1)	Whether the test passed (boolean)
stdout	TEXT		Captured standard output
stderr	TEXT		Captured standard error
runtime_ms	INTEGER		Execution time in ms
memory_kb	INTEGER		Memory used in kilobytes
exit_code	INTEGER		OS/process exit code
error_message	TEXT		Human-readable error description
created_at	TEXT	DEFAULT CURRENT_TIMESTAMP	Execution time

Notes: Use FK cascade rules to keep results in sync with submissions. passed: 0=failed, 1=passed.

HINTS

Description: AI-generated hints associated with submissions.

Columns

Column	Type	Constraint	Purpose
id	INTEGER	PK, AUTOINCREMENT	Hint ID
submission_id	INTEGER	FK Submissions(id), NOT NULL	Which submission produced/received the hint
model_used	TEXT		Model identifier/version used to generate hint
confidence	REAL	CHECK (confidence >= 0.0 AND confidence <= 1.0)	Model confidence score
hint_text	TEXT	NOT NULL	The actual hint content
is_helpful	INTEGER	CHECK IN (0,1)	Feedback whether hint helped (boolean)
feedback	TEXT		Student feedback on hint quality
created_at	TEXT	DEFAULT CURRENT_TIMESTAMP	Generation time

Notes: feedback field allows students to provide qualitative assessment. is_helpful: 0=not helpful, 1=helpful, NULL=not rated.

EMBEDDINGS

Description: Vector references for submissions used for similarity search. Modeled 1:1 with submission.

Columns

Column	Type	Constraint	Purpose
id	INTEGER	PK, FK Submissions(id)	Embedding ID, links to submission (1:1)
vector_ref	TEXT	NOT NULL	Reference pointer to vector store or serialized vector
model_version	TEXT		Embedding model version
dimensions	INTEGER		Vector dimensionality
created_at	TEXT	DEFAULT CURRENT_TIMESTAMP	Generation time

Notes: id serves as both primary key and foreign key to SUBMISSIONS, ensuring 1:1 relationship. Store large vectors outside DB (object store / vector DB) and keep vector_ref as pointer.

SIMILARITY_FLAGS

Description: Plagiarism/similarity detection results per submission.

Columns

Column	Type	Constraint	Purpose
id	INTEGER	PK, AUTOINCREMENT	Flag ID
submission_id	INTEGER	FK Submissions(id), NOT NULL	Submission flagged
similarity_score	REAL	NOT NULL, CHECK (similarity_score >= 0.0 AND similarity_score <= 1.0)	Highest similarity score
highlighted_spans	TEXT	JSON stored as TEXT	JSON describing matched spans for UI
is_reviewed	INTEGER	DEFAULT 0, CHECK IN (0,1)	Has human reviewed it (boolean)
reviewed_by	INTEGER	FK Users(id)	Reviewer user ID
review_notes	TEXT		Human reviewer notes
created_at	TEXT	DEFAULT CURRENT_TIMESTAMP	Detection time
reviewed_at	TEXT		When review completed

Notes: highlighted_spans stores structured JSON data for highlighting matched code sections. is_reviewed: 0=pending, 1=reviewed.

SIMILARITY_COMPARISONS

Description: Junction table that records which other submissions were compared against for a given similarity flag.

Columns

Column	Type	Constraint	Purpose
similarity_id	INTEGER	FK Similarity_flags(id), NOT NULL	The parent similarity flag
compared_submission_id	INTEGER	FK Submissions(id), NOT NULL	Other submission being compared
match_score	REAL	CHECK (match_score >= 0.0 AND match_score <= 1.0)	Score for this pairwise match
matched_segments	TEXT	JSON stored as TEXT	JSON describing matched code segments
note	TEXT		Optional note about this comparison

Primary key: Composite (similarity_id, compared_submission_id)

Notes: matched_segments provides detailed segment-level matching information as JSON. note field allows annotations on specific comparisons.

PEER_REVIEWS

Description: Peer review records where a student reviews a submission.

Columns

Column	Type	Constraint	Purpose
id	INTEGER	PK, AUTOINCREMENT	Peer review ID
submission_id	INTEGER	FK Submissions(id), NOT NULL	Submission being reviewed
reviewer_student_id	INTEGER	FK Students(id), NOT NULL	Reviewing student
rubric_scores	TEXT	JSON stored as TEXT, NOT NULL	Structured rubric scores
comments	TEXT		Free text comments
is_submitted	INTEGER	DEFAULT 0, CHECK IN (0,1)	Reviewer saved vs submitted state (boolean)
created_at	TEXT	DEFAULT CURRENT_TIMESTAMP	Creation time
submitted_at	TEXT		When reviewer submitted final review

Notes: Enforce uniqueness with UNIQUE constraint on (submission_id, reviewer_student_id) if reviewer can review a submission only once. is_submitted: 0=draft, 1=submitted.

FILES

Description: Files associated with submissions or uploaded by users (code files, attachments).

Columns

Column	Type	Constraint	Purpose
id	INTEGER	PK, AUTOINCREMENT	File ID
submission_id	INTEGER	FK Submissions(id)	Which submission the file belongs to
uploader_id	INTEGER	FK Users(id)	Who uploaded the file
path	TEXT	NOT NULL	Storage path or object key
filename	TEXT	NOT NULL	Original file name

Column	Type	Constraint	Purpose
content_type	TEXT		MIME type
size_bytes	INTEGER	NOT NULL	Size in bytes
checksum	TEXT	NOT NULL	Hash for integrity (e.g., SHA-256)
storage_url	TEXT		Full URL or signed URL to file location
created_at	TEXT	DEFAULT CURRENT_TIMESTAMP	Upload time

Notes: Keep binary data in object store; DB stores pointer and metadata. storage_url enables direct access to cloud storage objects.

AUDIT_LOGS

Description: Immutable audit log of user actions and system events.

Columns

Column	Type	Constraint	Purpose
id	INTEGER	PK, AUTOINCREMENT	Audit entry ID
actor_user_id	INTEGER	FK Users(id)	Acting user
action	TEXT	NOT NULL	Action name
entity_type	TEXT		Type of entity affected
entity_id	INTEGER		ID of affected entity
details	TEXT	JSON stored as TEXT	Additional structured details
ip_address	TEXT		Source IP
user_agent	TEXT		Browser/client user agent string
created_at	TEXT	DEFAULT CURRENT_TIMESTAMP	Event time

Notes: user_agent field supports device/browser tracking for security auditing. Consider partition strategy or archiving for large audit tables.