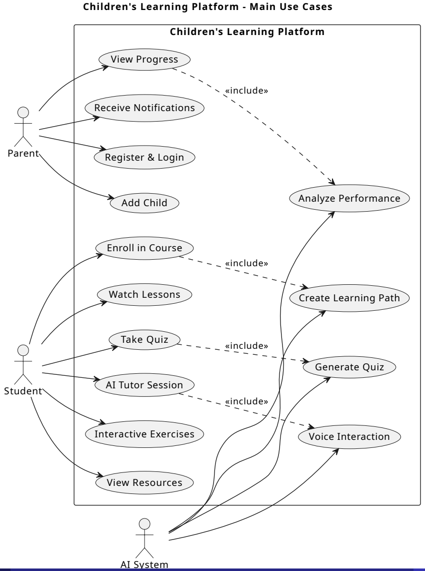
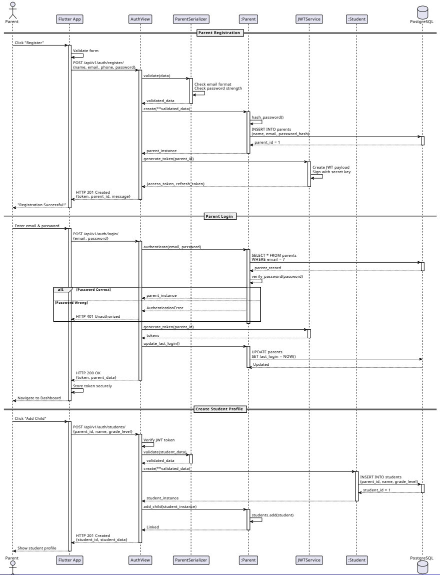
UML Diagrams & Database Schema

February 16, 2026

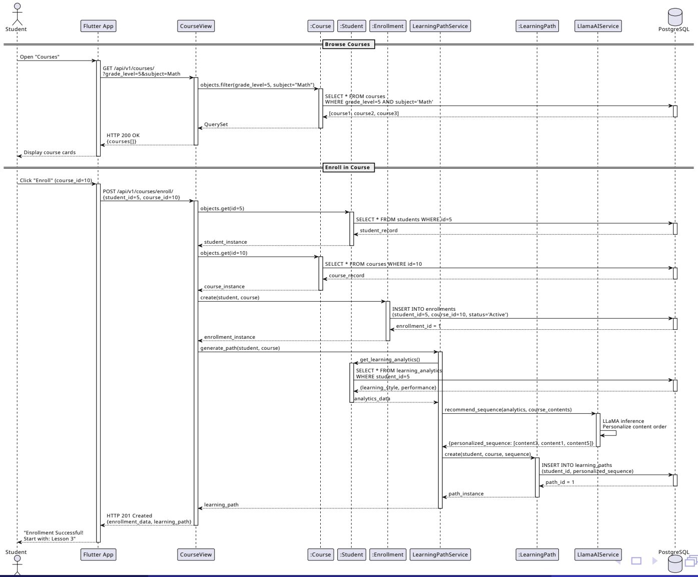
Use Case Diagram



Sequence Diagram - Authentication



Sequence Diagram - Enrollment



Student

Flutter App

CourseView

:Course

:Student

:Enrollment

LearningPathService

:LearningPath

LlamaAIService

PostgreSQL

Browse Courses

Open "Courses"

GET /api/v1/courses/
?grade_level=5&subject=Math

objects.filter(grade_level=5, subject="Math")

SELECT * FROM courses
WHERE grade_level=5 AND subject='Math'

[course1, course2, course3]

QuerySet

HTTP 200 OK
{courses[]}

Display course cards

Enroll in Course

Click "Enroll" (course_id=10)

POST /api/v1/courses/enroll/
{student_id=5, course_id=10}

objects.get(id=5)

SELECT * FROM students WHERE id=5

student_record

student_instance

objects.get(id=10)

SELECT * FROM courses WHERE id=10

course_record

course_instance

create(student, course)

INSERT INTO enrollments
(student_id=5, course_id=10, status='Active')

enrollment_id = 1

enrollment_instance

generate_path(student, course)

get_learning_analytics()

SELECT * FROM learning_analytics
WHERE student_id=5

{learning_style, performance}

analytics_data

recommend_sequence(analytics, course_contents)

LLaMA inference
Personalize content order

{personalized_sequence: [content3, content1, content5]}

create(student, course, sequence)

INSERT INTO learning_paths
(student_id, personalized_sequence)

path_id = 1

path_instance

learning_path

HTTP 201 Created
{enrollment_data, learning_path}

"Enrollment Successful!
Start with: Lesson 3"

Student

Flutter App

CourseView

:Course

:Student

:Enrollment

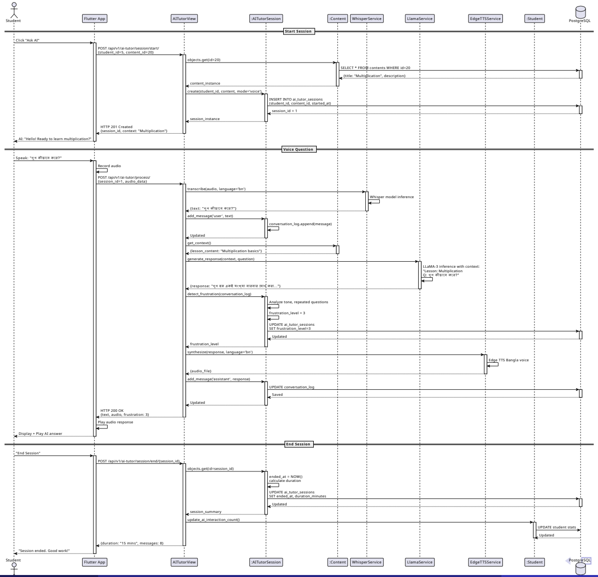
LearningPathService

:LearningPath

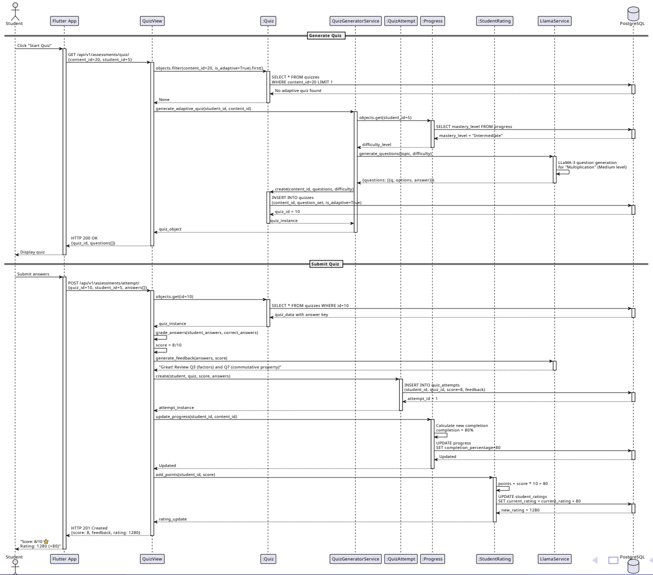
LlamaAIService

PostgreSQL

Sequence Diagram - AI Tutor


Student
Flutter App
AITutorView
:AITutorSession
:Content
WhisperService
LlamaService
EdgeTTSService
:Student
PostgreSQL
Start Session
Click "Ask AI"
POST /api/v1/ai-tutor/session/start/
{student_id=5, content_id=20}
objects.get(id=20)
SELECT * FROM contents WHERE id=20
{title: "Multiplication", description}
content_instance
create(student_id, content, mode='voice')
INSERT INTO ai_tutor_sessions
(student_id, content_id, started_at)
session_id = 1
session_instance
HTTP 201 Created
{session_id, context: "Multiplication"}
AI: "Hello! Ready to learn multiplication?"
Voice Question
Speak: "গুণ কীভাবে করে?"
Record audio
POST /api/v1/ai-tutor/process/
{session_id=1, audio_data}
transcribe(audio, language='bn')
Whisper model inference
{text: "গুণ কীভাবে করে?"}
add_message('user', text)
conversation_log.append(message)
Updated
get_context()
{lesson_content: "Multiplication basics"}
generate_response(context, question)
LLaMA-3 inference with context:
"Lesson: Multiplication
Q: গুণ কীভাবে করে?"
{response: "গুণ হল একই সংখ্যা বারবার যোগ করা..."}
detect_frustration(conversation_log)
Analyze tone, repeated questions
frustration_level = 3
UPDATE ai_tutor_sessions
SET frustration_level=3
Updated
frustration_level
synthesize(response, language='bn')
Edge TTS Bangla voice
{audio_file}
add_message('assistant', response)
UPDATE conversation_log
Saved
Updated
HTTP 200 OK
{text, audio, frustration: 3}
Play audio response
Display + Play AI answer
End Session
"End Session"
POST /api/v1/ai-tutor/session/end/{session_id}
objects.get(id=session_id)
ended_at = NOW()
calculate duration
UPDATE ai_tutor_sessions
SET ended_at, duration_minutes
Updated
session_summary
update_ai_interaction_count()
UPDATE student stats
Updated
{duration: "15 mins", messages: 8}
"Session ended. Good work!"
Student
Flutter App
AITutorView
:AITutorSession
:Content
WhisperService
LlamaService
EdgeTTSService
:Student
PostgreSQL

Sequence Diagram - Quiz



Student
Flutter App
QuizView
:Quiz
QuizGeneratorService
:QuizAttempt
:Progress
:StudentRating
LlamaService
PostgreSQL

Generate Quiz

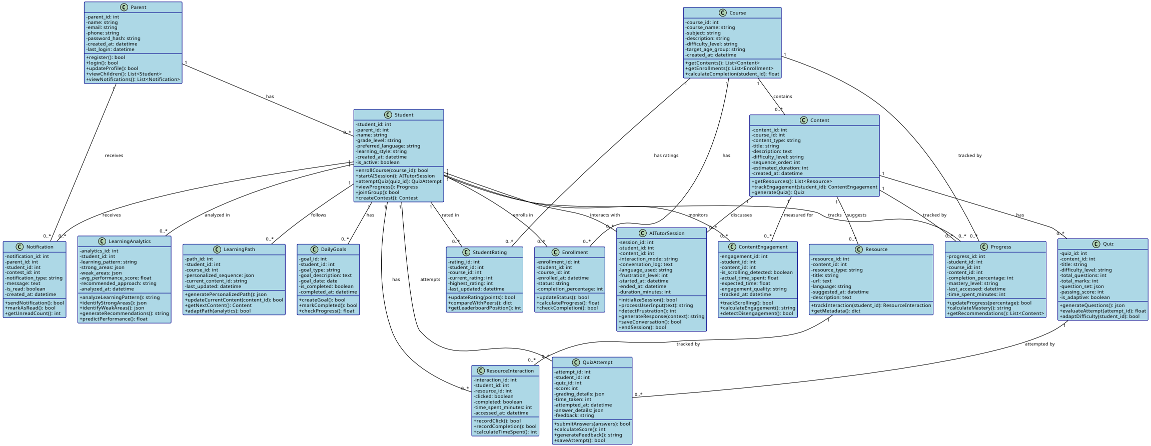
Click "Start Quiz"
GET /api/v1/assessments/quiz/
{content_id=20, student_id=5}
objects.filter(content_id=20, is_adaptive=True).first()
SELECT * FROM quizzes
WHERE content_id=20 LIMIT 1
No adaptive quiz found
None
generate_adaptive_quiz(student_id, content_id)
objects.get(student_id=5)
SELECT mastery_level FROM progress
mastery_level = "Intermediate"
difficulty_level
generate_questions(topic, difficulty)
LLaMA-3 question generation
for "Multiplication" (Medium level)
{questions: [{q, options, answer}]}
create(content_id, questions, difficulty)
INSERT INTO quizzes
(content_id, question_set, is_adaptive=True)
quiz_id = 10
quiz_instance
quiz_object
HTTP 200 OK
{quiz_id, questions[]}
Display quiz

Submit Quiz

Submit answers
POST /api/v1/assessments/attempt/
{quiz_id=10, student_id=5, answers[]}
objects.get(id=10)
SELECT * FROM quizzes WHERE id=10
quiz_data with answer key
quiz_instance
grade_answers(student_answers, correct_answers)
score = 8/10
generate_feedback(answers, score)
"Great! Review Q3 (factors) and Q7 (commutative property)"
create(student, quiz, score, answers)
INSERT INTO quiz_attempts
(student_id, quiz_id, score=8, feedback)
attempt_id = 1
attempt_instance
update_progress(student_id, content_id)
Calculate new completion
completion = 80%
UPDATE progress
SET completion_percentage=80
Updated
Updated
add_points(student_id, score)
points = score * 10 = 80
UPDATE student_ratings
SET current_rating = current_rating + 80
new_rating = 1280
rating_update
HTTP 201 Created
{score: 8, feedback, rating: 1280}
"Score: 8/10 ⭐
Rating: 1280 (+80)"

Student
Flutter App
QuizView
:Quiz
QuizGeneratorService
:QuizAttempt
:Progress
:StudentRating
LlamaService
PostgreSQL

Class Diagram



Parent
-parent_id: int
-name: string
-email: string
-phone: string
-password_hash: string
-created_at: datetime
-last_login: datetime
+register(): bool
+login(): bool
+updateProfile(): bool
+viewChildren(): List<Student>
+viewNotifications(): List<Notification>

Course
-course_id: int
-course_name: string
-subject: string
-description: string
-difficulty_level: string
-target_age_group: string
-created_at: datetime
+getContents(): List<Content>
+getEnrollments(): List<Enrollment>
+calculateCompletion(student_id): float

Student
-student_id: int
-parent_id: int
-name: string
-grade_level: string
-preferred_language: string
-learning_style: string
-created_at: datetime
-is_active: boolean
+enrollCourse(course_id): bool
+startAISession(): AITutorSession
+attemptQuiz(quiz_id): QuizAttempt
+viewProgress(): Progress
+joinGroup(): bool
+createContest(): Contest

Content
-content_id: int
-course_id: int
-content_type: string
-title: string
-description: text
-difficulty_level: string
-sequence_order: int
-estimated_duration: int
-created_at: datetime
+getResources(): List<Resource>
+trackEngagement(student_id): ContentEngagement
+generateQuiz(): Quiz

Notification
-notification_id: int
-parent_id: int
-student_id: int
-contest_id: int
-notification_type: string
-message: text
-is_read: boolean
-created_at: datetime
+sendNotification(): bool
+markAsRead(): bool
+getUnreadCount(): int

LearningAnalytics
-analytics_id: int
-student_id: int
-learning_pattern: string
-strong_areas: json
-weak_areas: json
-avg_performance_score: float
-recommended_approach: string
-analyzed_at: datetime
+analyzeLearningPattern(): string
+identifyStrongAreas(): json
+identifyWeakAreas(): json
+generateRecommendations(): string
+predictPerformance(): float

LearningPath
-path_id: int
-student_id: int
-course_id: int
-personalized_sequence: json
-current_content_id: string
-last_updated: datetime
+generatePersonalizedPath(): json
+updateCurrentContent(content_id): bool
+getNextContent(): Content
+adaptPath(analytics): bool

DailyGoals
-goal_id: int
-student_id: int
-goal_type: string
-goal_description: text
-goal_date: date
-is_completed: boolean
-completed_at: datetime
+createGoal(): bool
+markCompleted(): bool
+checkProgress(): float

StudentRating
-rating_id: int
-student_id: int
-course_id: int
-current_rating: int
-highest_rating: int
-last_updated: datetime
+updateRating(points): bool
+compareWithPeers(): dict
+getLeaderboardPosition(): int

Enrollment
-enrollment_id: int
-student_id: int
-course_id: int
-enrolled_at: datetime
-status: string
-completion_percentage: int
+updateStatus(): bool
+calculateProgress(): float
+checkCompletion(): bool

AITutorSession
-session_id: int
-student_id: int
-content_id: int
-interaction_mode: string
-conversation_log: text
-language_used: string
-frustration_level: int
-started_at: datetime
-ended_at: datetime
-duration_minutes: int
+initializeSession(): bool
+processUserInput(text): string
+detectFrustration(): int
+generateResponse(context): string
+saveConversation(): bool
+endSession(): bool

ContentEngagement
-engagement_id: int
-student_id: int
-content_id: int
-is_scrolling_detected: boolean
-actual_time_spent: float
-expected_time: float
-engagement_quality: string
-tracked_at: datetime
+trackScrolling(): bool
+calculateEngagement(): string
+detectDisengagement(): bool

Resource
-resource_id: int
-content_id: int
-resource_type: string
-title: string
-url: text
-language: string
-suggested_at: datetime
-description: text
+trackInteraction(student_id): ResourceInteraction
+getMetadata(): dict

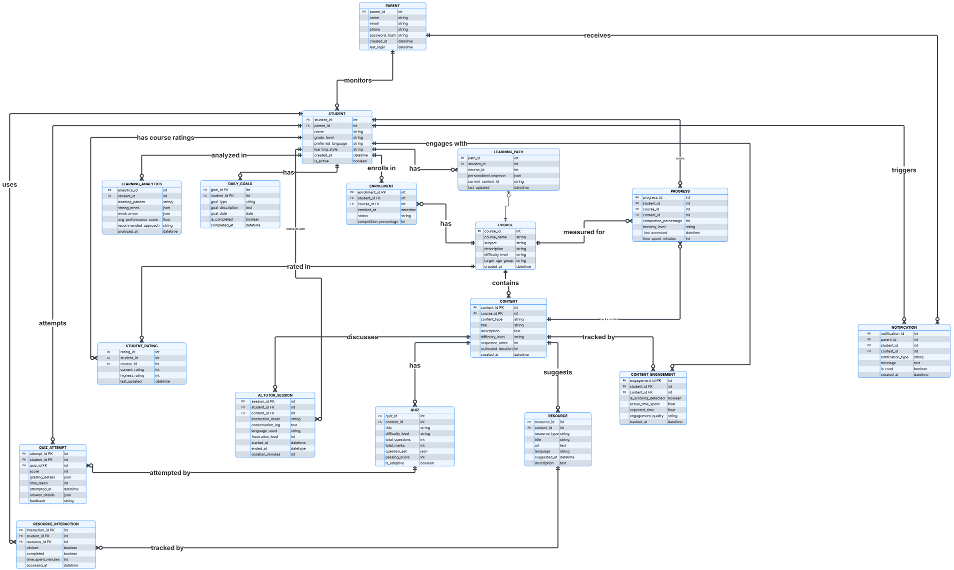
Progress
-progress_id: int
-student_id: int
-course_id: int
-content_id: int
-completion_percentage: int
-mastery_level: string
-last_accessed: datetime
-time_spent_minutes: int
+updateProgress(percentage): bool
+calculateMastery(): string
+getRecommendations(): List<Content>

Quiz
-quiz_id: int
-content_id: int
-title: string
-difficulty_level: string
-total_questions: int
-total_marks: int
-question_set: json
-passing_score: int
-is_adaptive: boolean
+generateQuestions(): json
+evaluateAttempt(attempt_id): float
+adaptDifficulty(student_id): bool

ResourceInteraction
-interaction_id: int
-student_id: int
-resource_id: int
-clicked: boolean
-completed: boolean
-time_spent_minutes: int
-accessed_at: datetime
+recordClick(): bool
+recordCompletion(): bool
+calculateTimeSpent(): int

QuizAttempt
-attempt_id: int
-student_id: int
-quiz_id: int
-score: int
-grading_details: json
-time_taken: int
-attempted_at: datetime
-answer_details: json
-feedback: string
+submitAnswers(answers): bool
+calculateScore(): int
+generateFeedback(): string
+saveAttempt(): bool

ER Diagram



PARENT

PK	parent_id	int
	name	string
	email	string
	phone	string
	password_hash	string
	created_at	datetime
	last_login	datetime

STUDENT

PK	student_id	int
FK	parent_id	int
	name	string
	grade_level	string
	preferred_language	string
	learning_style	string
	created_at	datetime
	is_active	boolean

LEARNING_ANALYTICS

PK	analytics_id	int
FK	student_id	int
	learning_pattern	string
	strong_areas	json
	weak_areas	json
	avg_performance_score	float
	recommended_approach	string
	analyzed_at	datetime

DAILY_GOALS

PK	goal_id PK	int
FK	student_id FK	int
	goal_type	string
	goal_description	text
	goal_date	date
	is_completed	boolean
	completed_at	datetime

ENROLLMENT

PK	enrollment_id PK	int
FK	student_id FK	int
FK	course_id FK	int
	enrolled_at	datetime
	status	string
	competition_percentage	int

LEARNING_PATH

PK	path_id	int
FK	student_id	int
	course_id	int
	personalized_seqence	json
	current_content_id	string
	last_updated	datetime

PROGRESS

PK	progress_id	int
FK	student_id	int
FK	course_id	int
FK	content_id	int
	completion_percentage	int
	mastery_level	string
	last_accessed	datetime
	time_spent_minutes	int

COURSE

PK	course_id	int
	course_name	string
	subject	string
	description	string
	difficulty_level	string
	target_age_group	string
	created_at	datetime

CONTENT

PK	content_id PK	int
FK	course_id PK	int
	content_type	string
	title	string
	description	text
	difficulty_level	string
	sequence_order	int
	estimated_duration	int
	created_at	datetime

NOTIFICATION

PK	notification_id	int
FK	parent_id	int
FK	student_id	int
FK	contest_id	int
	notification_type	string
	message	text
	is_read	boolean
	created_at	datetime

STUDENT_RATING

PK	rating_id	int
FK	student_id	int
FK	course_id	int
	current_rating	int
	highest_rating	int
	last_updated	datetime

AI_TUTOR_SESSION

PK	session_id PK	int
FK	student_id FK	int
FK	content_id FK	int
	interaction_mode	string
	conversation_log	text
	language_used	string
	frustration_level	int
	started_at	datetime
	ended_at	datetype
	duration_minutes	int

QUIZ

PK	quiz_id	int
FK	content_id	int
	title	string
	difficulty_level	string
	total_questions	int
	total_marks	int
	question_set	json
	passing_score	int
	is_adaptive	boolean

RESOURCE

PK	resource_id	int
FK	content_id	int
	resource_type	string
	title	string
	url	text
	language	string
	suggested_at	datetime
	description	text

CONTENT_ENGAGEMENT

PK	engagement_id PK	int
FK	student_id FK	int
FK	content_id FK	int
	is_scrolling_detected	boolean
	actual_time_spent	float
	expected_time	float
	engagement_quality	string
	tracked_at	datetime

QUIZ_ATTEMPT

PK	attempt_id PK	int
FK	student_id FK	int
FK	quiz_id FK	int
	score	int
	grading_details	json
	time_taken	int
	attempted_at	datetime
	answer_details	json
	feedback	string

RESOURCE_INTERACTION

PK	interaction_id PK	int
FK	student_id FK	int
FK	resource_id FK	int
	clicked	boolean
	completed	boolean
	time_spent_minutes	int
	accessed_at	datetime

Database Schema Overview

- **Total Tables:** 17
- **Database:** PostgreSQL
- **Categories:**
  - **Core User Tables:** Parent, Student
  - **Course & Content:** Course, Content
  - **Learning:** Enrollment, Progress
  - **Assessment:** Quiz, Quiz Attempt
  - **AI:** AI Tutor Session
  - **Resources:** Resource, Resource Interaction
  - **Analytics:** Content Engagement, Learning Analytics
  - **Gamification:** Student Rating, Daily Goals
  - **Others:** Learning Path, Notification

Core User Table - Parent

```
CREATE TABLE parent (
    parent_id SERIAL PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    email VARCHAR(255) UNIQUE NOT NULL,
    phone VARCHAR(20),
    password_hash VARCHAR(255) NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    last_login TIMESTAMP,
    CONSTRAINT email_format CHECK (
        email ~* ’^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}$’
    )
);
```

Core User Table - Student

```
CREATE TABLE student (
    student_id SERIAL PRIMARY KEY,
    parent_id INTEGER NOT NULL,
    name VARCHAR(255) NOT NULL,
    grade_level VARCHAR(50),
    preferred_language VARCHAR(50) DEFAULT ’Bangla’,
    learning_style VARCHAR(100),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    is_active BOOLEAN DEFAULT TRUE,
    FOREIGN KEY (parent_id)
        REFERENCES parent(parent_id)
        ON DELETE CASCADE,
    CONSTRAINT valid_language CHECK (
        preferred_language IN (’Bangla’, ’English’, ’Both’)
    )
);
```

Course & Content Table - Course

```
CREATE TABLE course (
    course_id SERIAL PRIMARY KEY,
    course_name VARCHAR(255) NOT NULL,
    subject VARCHAR(100) NOT NULL,
    description TEXT,
    difficulty_level VARCHAR(50),
    target_age_group VARCHAR(50),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    CONSTRAINT valid_difficulty CHECK (
        difficulty_level IN (’Easy’, ’Medium’, ’Hard’)
    )
);
```

Course & Content Table - Content

```
CREATE TABLE content (
    content_id SERIAL PRIMARY KEY,
    course_id INTEGER NOT NULL,
    content_type VARCHAR(50) NOT NULL,
    title VARCHAR(255) NOT NULL,
    description TEXT,
    difficulty_level VARCHAR(50),
    sequence_order INTEGER NOT NULL,
    estimated_duration INTEGER,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (course_id)
        REFERENCES course(course_id) ON DELETE CASCADE,
    CONSTRAINT valid_content_type CHECK (
        content_type IN ('Video', 'Text', 'Interactive',
                         'Quiz', 'Exercise')
    )
);
```

Learning Table - Enrollment

```
CREATE TABLE enrollment (
    enrollment_id SERIAL PRIMARY KEY,
    student_id INTEGER NOT NULL,
    course_id INTEGER NOT NULL,
    enrolled_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    status VARCHAR(50) DEFAULT ’Active’,
    completion_percentage INTEGER DEFAULT 0,
    FOREIGN KEY (student_id)
        REFERENCES student(student_id) ON DELETE CASCADE,
    FOREIGN KEY (course_id)
        REFERENCES course(course_id) ON DELETE CASCADE,
    CONSTRAINT unique_enrollment UNIQUE(student_id, course_id),
    CONSTRAINT valid_status CHECK (
        status IN (’Active’, ’Completed’, ’Paused’, ’Dropped’)
    )
);
```

Learning Table - Progress

```
CREATE TABLE progress (
    progress_id SERIAL PRIMARY KEY,
    student_id INTEGER NOT NULL,
    course_id INTEGER NOT NULL,
    content_id INTEGER NOT NULL,
    completion_percentage INTEGER DEFAULT 0,
    mastery_level VARCHAR(50),
    last_accessed TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    time_spent_minutes INTEGER DEFAULT 0,
    FOREIGN KEY (student_id)
        REFERENCES student(student_id) ON DELETE CASCADE,
    CONSTRAINT unique_progress UNIQUE(student_id, content_id),
    CONSTRAINT valid_mastery CHECK (
        mastery_level IN (’Beginner’, ’Intermediate’,
                          ’Advanced’, ’Mastered’)
    )
);
```

Assessment Table - Quiz

```
CREATE TABLE quiz (
    quiz_id SERIAL PRIMARY KEY,
    content_id INTEGER NOT NULL,
    title VARCHAR(255) NOT NULL,
    difficulty_level VARCHAR(50),
    total_questions INTEGER NOT NULL,
    total_marks INTEGER NOT NULL,
    question_set JSONB NOT NULL,
    passing_score INTEGER NOT NULL,
    is_adaptive BOOLEAN DEFAULT FALSE,
    FOREIGN KEY (content_id)
        REFERENCES content(content_id) ON DELETE CASCADE,
    CONSTRAINT valid_questions CHECK (total_questions > 0),
    CONSTRAINT valid_marks CHECK (total_marks > 0)
);
```

Assessment Table - Quiz Attempt

```
CREATE TABLE quiz_attempt (
    attempt_id SERIAL PRIMARY KEY,
    student_id INTEGER NOT NULL,
    quiz_id INTEGER NOT NULL,
    score INTEGER NOT NULL,
    grading_details JSONB,
    time_taken INTEGER,
    attempted_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    answer_details JSONB,
    feedback TEXT,
    FOREIGN KEY (student_id)
        REFERENCES student(student_id) ON DELETE CASCADE,
    FOREIGN KEY (quiz_id)
        REFERENCES quiz(quiz_id) ON DELETE CASCADE,
    CONSTRAINT valid_score CHECK (score >= 0)
);
```

AI Table - AI Tutor Session

```
CREATE TABLE ai_tutor_session (
    session_id SERIAL PRIMARY KEY,
    student_id INTEGER NOT NULL,
    content_id INTEGER NOT NULL,
    interaction_mode VARCHAR(50),
    conversation_log TEXT,
    language_used VARCHAR(50),
    frustration_level INTEGER DEFAULT 0,
    started_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    ended_at TIMESTAMP,
    duration_minutes INTEGER,
    FOREIGN KEY (student_id)
        REFERENCES student(student_id) ON DELETE CASCADE,
    CONSTRAINT valid_interaction_mode CHECK (
        interaction_mode IN ('Voice', 'Text', 'Both')
    )
);
```

Resource Table - Resource

```
CREATE TABLE resource (
    resource_id SERIAL PRIMARY KEY,
    content_id INTEGER NOT NULL,
    resource_type VARCHAR(50) NOT NULL,
    title VARCHAR(255) NOT NULL,
    url TEXT,
    language VARCHAR(50),
    suggested_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    description TEXT,
    FOREIGN KEY (content_id)
        REFERENCES content(content_id) ON DELETE CASCADE,
    CONSTRAINT valid_resource_type CHECK (
        resource_type IN (’PDF’, ’Video’, ’Article’,
                          ’Exercise’, ’Game’)
    )
);
```

Resource Table - Resource Interaction

```
CREATE TABLE resource_interaction (
    interaction_id SERIAL PRIMARY KEY,
    student_id INTEGER NOT NULL,
    resource_id INTEGER NOT NULL,
    clicked BOOLEAN DEFAULT FALSE,
    completed BOOLEAN DEFAULT FALSE,
    time_spent_minutes INTEGER DEFAULT 0,
    accessed_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (student_id)
        REFERENCES student(student_id) ON DELETE CASCADE,
    FOREIGN KEY (resource_id)
        REFERENCES resource(resource_id) ON DELETE CASCADE
);
```

Analytics Table - Content Engagement

```
CREATE TABLE content_engagement (
    engagement_id SERIAL PRIMARY KEY,
    student_id INTEGER NOT NULL,
    content_id INTEGER NOT NULL,
    is_scrolling_detected BOOLEAN DEFAULT FALSE,
    actual_time_spent FLOAT DEFAULT 0,
    expected_time FLOAT,
    engagement_quality VARCHAR(50),
    tracked_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (student_id)
        REFERENCES student(student_id) ON DELETE CASCADE,
    FOREIGN KEY (content_id)
        REFERENCES content(content_id) ON DELETE CASCADE,
    CONSTRAINT valid_engagement_quality CHECK (
        engagement_quality IN (’High’, ’Medium’, ’Low’,
                               ’Very Low’)
    )
);
```

Analytics Table - Learning Analytics

```
CREATE TABLE learning_analytics (
    analytics_id SERIAL PRIMARY KEY,
    student_id INTEGER NOT NULL,
    learning_pattern VARCHAR(255),
    strong_areas JSONB,
    weak_areas JSONB,
    avg_performance_score FLOAT,
    recommended_approach TEXT,
    analyzed_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (student_id)
        REFERENCES student(student_id) ON DELETE CASCADE,
    CONSTRAINT valid_performance CHECK (
        avg_performance_score >= 0 AND
        avg_performance_score <= 100
    )
);
```

Gamification Table - Student Rating

```
CREATE TABLE student_rating (
    rating_id SERIAL PRIMARY KEY,
    student_id INTEGER NOT NULL,
    course_id INTEGER NOT NULL,
    current_rating INTEGER DEFAULT 0,
    highest_rating INTEGER DEFAULT 0,
    last_updated TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (student_id)
        REFERENCES student(student_id) ON DELETE CASCADE,
    FOREIGN KEY (course_id)
        REFERENCES course(course_id) ON DELETE CASCADE,
    CONSTRAINT unique_student_course_rating
        UNIQUE(student_id, course_id),
    CONSTRAINT valid_rating CHECK (current_rating >= 0)
);
```

Gamification Table - Daily Goals

```
CREATE TABLE daily_goals (
    goal_id SERIAL PRIMARY KEY,
    student_id INTEGER NOT NULL,
    goal_type VARCHAR(100),
    goal_description TEXT,
    goal_date DATE NOT NULL,
    is_completed BOOLEAN DEFAULT FALSE,
    completed_at TIMESTAMP,
    FOREIGN KEY (student_id)
        REFERENCES student(student_id) ON DELETE CASCADE,
    CONSTRAINT valid_goal_type CHECK (
        goal_type IN (’Watch Lesson’, ’Complete Quiz’,
                      ’AI Session’, ’Exercise’,
                      ’Read Resource’)
    )
);
```

Learning Path Table - Learning Path

```
CREATE TABLE learning_path (
    path_id SERIAL PRIMARY KEY,
    student_id INTEGER NOT NULL,
    course_id INTEGER NOT NULL,
    personalized_sequence JSONB NOT NULL,
    current_content_id INTEGER,
    last_updated TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (student_id)
        REFERENCES student(student_id) ON DELETE CASCADE,
    FOREIGN KEY (course_id)
        REFERENCES course(course_id) ON DELETE CASCADE,
    CONSTRAINT unique_learning_path
        UNIQUE(student_id, course_id)
);
```

Notification Table - Notification

```
CREATE TABLE notification (
    notification_id SERIAL PRIMARY KEY,
    parent_id INTEGER,
    student_id INTEGER,
    contest_id INTEGER,
    notification_type VARCHAR(50) NOT NULL,
    message TEXT NOT NULL,
    is_read BOOLEAN DEFAULT FALSE,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (parent_id)
        REFERENCES parent(parent_id) ON DELETE CASCADE,
    CONSTRAINT valid_notification_type CHECK (
        notification_type IN (’Progress’, ’Achievement’,
                              ’Reminder’, ’Contest’, ’Alert’)
    )
);
```

Database Schema - Performance Indexes

– Parent Indexes

```
CREATE INDEX idx_parent_email ON parent(email);
```

– Student Indexes

```
CREATE INDEX idx_student_parent ON student(parent_id);
CREATE INDEX idx_student_active ON student(is_active);
```

– Content Indexes

```
CREATE INDEX idx_content_course ON content(course_id);
CREATE INDEX idx_content_sequence ON content(course_id, sequence_order);
```

Database Schema - More Indexes

– Quiz Attempt Indexes – Enrollment Indexes

```
CREATE INDEX idx_enrollment_student ON enrollment(student_id);
CREATE INDEX idx_enrollment_course ON enrollment(course_id);
CREATE INDEX idx_enrollment_status ON enrollment(status);
```

– Progress Indexes

```
CREATE INDEX idx_progress_student ON progress(student_id);
CREATE INDEX idx_progress_content ON progress(content_id);
```

Database Schema - More Indexes

– Quiz Attempt Indexes

```
CREATE INDEX idx_quiz_attempt_student ON quiz_attempt(student_id);
CREATE INDEX idx_quiz_attempt_quiz ON quiz_attempt(quiz_id);
CREATE INDEX idx_quiz_attempt_date ON quiz_attempt(attempted_at);
```

– AI Tutor Session Indexes

```
CREATE INDEX idx_ai_session_student ON ai_tutor_session(student_id);
CREATE INDEX idx_ai_session_content ON ai_tutor_session(content_id);
CREATE INDEX idx_ai_session_date ON ai_tutor_session(started_at);
```

– Notification Indexes

```
CREATE INDEX idx_notification_parent ON notification(parent_id);
CREATE INDEX idx_notification_student ON notification(student_id);
CREATE INDEX idx_notification_read ON notification(is_read);
```

Database Schema - Key Features

- **JSONB Data Type:** Flexible data storage
  - Quiz `question_set`
  - Learning Analytics `strong_areas`, `weak_areas`
  - Learning Path `personalized_sequence`
- **Constraints & Validations:**
  - CHECK constraints for data validation
  - UNIQUE constraints to prevent duplicates
  - Foreign keys with CASCADE delete
- **Performance Optimization:**
  - 15+ indexes on foreign keys
  - Composite indexes for complex queries