

# CHI TIẾT DATABASE SCHEMA - ENGLISH LEARNING PLATFORM

## MỤC LỤC

1. Tổng quan
  2. Users Module
  3. Vocabulary Module
  4. Grammar Module
  5. Reading Module
  6. Listening Module
  7. Writing Module
  8. Speaking Module
  9. Shared Tables
  10. Indexes
  11. Entity Relationships
- 

## TỔNG QUAN

### Thống kê

- **Tổng số bảng:** 25 bảng
- **Module chính:** 7 modules (Users, Vocabulary, Grammar, Reading, Listening, Writing, Speaking)
- **Bảng dùng chung:** 2 bảng (questions, question\_options)

### Danh sách bảng theo module

Module	Số bảng	Tên bảng
Users	1	users
Vocabulary	4	vocabulary_topics, vocabulary_words, topic_words, user_vocabulary_progress
Grammar	4	grammar_topics, grammar_lessons, user_grammar_progress, + questions (shared)
Reading	2	reading_lessons, user_reading_progress, + questions (shared)
Listening	4	listening_lessons, listening_blanks, user_listening_progress, user_listening_submissions
Writing	4	writing_exercises, writing_segments, user_writing_progress, user_writing_submissions
Speaking	3	speaking_topics, user_speaking_progress, user_speaking_submissions, + questions (shared)
Shared	2	questions, question_options

1. USERS MODULE

◆ Table: `users`

Mục đích: Quản lý tài khoản người dùng và tiến độ học tập tổng thể

Column	Type	Constraints	Mô tả
<code>id</code>	bigserial	PRIMARY KEY	ID tự tăng
<code>username</code>	varchar(255)	NOT NULL, UNIQUE	Tên đăng nhập
<code>email</code>	varchar(255)	NOT NULL, UNIQUE	Email
<code>password</code>	varchar(255)	NOT NULL	Mật khẩu đã mã hóa (BCrypt)
<code>full_name</code>	varchar(100)		Họ tên đầy đủ
<code>role</code>	varchar(20)	DEFAULT 'USER'	USER, ADMIN, TEACHER
<code>english_level</code>	varchar(20)	DEFAULT 'BEGINNER'	BEGINNER, INTERMEDIATE, ADVANCED
<code>total_points</code>	integer	DEFAULT 0	Tổng điểm tích lũy
<code>streak_days</code>	integer	DEFAULT 0	Số ngày học liên tục
<code>last_login_date</code>	timestamp		Lần login cuối (tính streak)
<code>is_active</code>	boolean	DEFAULT true	Tài khoản còn hoạt động?
<code>is_verified</code>	boolean	DEFAULT false	Đã verify email?
<code>created_at</code>	timestamp	DEFAULT NOW()	Ngày tạo tài khoản
<code>updated_at</code>	timestamp	DEFAULT NOW()	Ngày cập nhật cuối

Business Logic:

- `streak_days`: Reset về 0 nếu user không login quá 24h
- `total_points`: Tổng điểm từ tất cả modules
- `english_level`: Dùng để filter lessons phù hợp

Indexes:

```
sql
PRIMARY KEY (id)
UNIQUE (email)
UNIQUE (username)
```

2. VOCABULARY MODULE

◆ Table: `vocabulary_topics`

Mục đích: Phân loại từ vựng theo chủ đề (system hoặc user-created)

Column	Type	Constraints	Mô tả
<code>id</code>	bigserial	PRIMARY KEY	
<code>name</code>	varchar(100)	NOT NULL	Tên chủ đề (VD: "Animals", "Food")
<code>description</code>	text		Mô tả chi tiết
<code>level_required</code>	varchar(20)	DEFAULT 'BEGINNER'	Trình độ yêu cầu
<code>is_system</code>	boolean	DEFAULT true	true: admin tạo, false: user tạo
<code>is_public</code>	boolean	DEFAULT true	Công khai cho user khác?
<code>created_by</code>	bigint	FK → users(id)	User tạo topic này
<code>order_index</code>	integer	DEFAULT 0	Thứ tự hiển thị
<code>is_active</code>	boolean	DEFAULT true	Còn hoạt động?
<code>created_at</code>	timestamp	DEFAULT NOW()	

**Business Logic:**

- System topics (`is_system=true`): Hiển thị cho tất cả users
- User topics (`is_system=false`): Chỉ hiển thị cho creator hoặc public topics

**Use Cases:**

- Admin tạo: "IELTS Vocabulary", "TOEIC Common Words"
- User tạo: "My Personal Flashcards", "Medical Terms"

◆ **Table:** `vocabulary_words`

**Mục đích:** Kho từ vựng chung (shared database)

Column	Type	Constraints	Mô tả
<code>id</code>	bigserial	PRIMARY KEY	
<code>word</code>	varchar(200)	NOT NULL	Từ vựng
<code>pronunciation</code>	varchar(300)		IPA: /'æp.əl/
<code>meaning_vi</code>	text	NOT NULL	Nghĩa tiếng Việt
<code>meaning_en</code>	text		Định nghĩa tiếng Anh
<code>part_of_speech</code>	varchar(50)		noun, verb, adjective, etc.
<code>example_sentence</code>	text		Câu ví dụ
<code>example_translation</code>	text		Dịch câu ví dụ
<code>audio_url</code>	varchar(500)		Link file mp3 phát âm
<code>image_url</code>	varchar(500)		Link hình ảnh minh họa
<code>difficulty_level</code>	varchar(20)	DEFAULT 'BEGINNER'	
<code>created_at</code>	timestamp	DEFAULT NOW()	

**Constraints:**

```
sql
```

```
UNIQUE (word, part_of_speech) -- "run" có thể là verb và noun
```

#### Data Source:

- Import từ dataset (Oxford 3000, Cambridge, etc.)
- Fetch từ Dictionary API
- Admin/Teacher thêm thủ công

#### ◆ Table: `topic_words`

**Mục đích:** Junction table - Many-to-Many giữa topics và words

Column	Type	Constraints	Mô tả
<code>id</code>	bigserial	PRIMARY KEY	
<code>topic_id</code>	bigint	FK → vocabulary_topics(id)	
<code>word_id</code>	bigint	FK → vocabulary_words(id)	
<code>order_index</code>	integer	DEFAULT 0	Thứ tự từ trong topic
<code>added_at</code>	timestamp	DEFAULT NOW()	

#### Constraints:

```
sql
```

```
UNIQUE (topic_id, word_id) -- 1 từ chỉ xuất hiện 1 lần trong 1 topic
```

#### Ví dụ:

Topic "Animals":    Topic "Farm":

- cat (order: 1)    - cat (order: 3)
- dog (order: 2)    - cow (order: 1)
- bird (order: 3)    - pig (order: 2)

#### ◆ Table: `user_vocabulary_progress`

**Mục đích:** Track tiến độ học từ vựng của từng user (Spaced Repetition System)

Column	Type	Constraints	Mô tả
<code>id</code>	bigserial	PRIMARY KEY	
<code>user_id</code>	bigint	FK → users(id)	
<code>word_id</code>	bigint	FK → vocabulary_words(id)	
<code>correct_count</code>	integer	DEFAULT 0	Số lần trả lời đúng
<code>incorrect_count</code>	integer	DEFAULT 0	Số lần trả lời sai
<code>mastery_level</code>	integer	DEFAULT 0	0-5 (SRS level)
<code>last_reviewed_at</code>	timestamp		Lần học gần nhất
<code>next_review_at</code>	timestamp		Lần ôn tập tiếp theo
<code>is_learning</code>	boolean	DEFAULT false	User có bấm "Learn" chưa?
<code>created_at</code>	timestamp	DEFAULT NOW()	
<code>updated_at</code>	timestamp	DEFAULT NOW()	

### Constraints:

```
sql
UNIQUE (user_id, word_id)
```

### SRS Algorithm:

mastery\_level 0: Review after 1 day  
mastery\_level 1: Review after 3 days  
mastery\_level 2: Review after 7 days  
mastery\_level 3: Review after 14 days  
mastery\_level 4: Review after 30 days  
mastery\_level 5: Mastered (review after 90 days)

If incorrect: mastery\_level -= 1

### Query ví dụ:

```
sql
-- Lấy từ cần ôn hôm nay
SELECT w.*
FROM user_vocabulary_progress uvp
JOIN vocabulary_words w ON uvp.word_id = w.id
WHERE uvp.user_id = 1
AND uvp.is_learning = true
AND uvp.next_review_at <= NOW()
ORDER BY uvp.next_review_at
LIMIT 20;
```

3. GRAMMAR MODULE

◆ Table: `grammar_topics`

Mục đích: Phân loại ngữ pháp theo chủ đề lớn

Column	Type	Constraints	Mô tả
<code>id</code>	bigserial	PRIMARY KEY	
<code>name</code>	varchar(200)	NOT NULL	VD: "12 Tenses", "Articles"
<code>description</code>	text		Mô tả tổng quan
<code>level_required</code>	varchar(20)	DEFAULT 'BEGINNER'	
<code>order_index</code>	integer	NOT NULL	Roadmap học (1,2,3...)
<code>is_active</code>	boolean	DEFAULT true	
<code>created_at</code>	timestamp	DEFAULT NOW()	

Ví dụ topics:

- 1. 12 Tenses
- 2. Articles (a/an/the)
- 3. Prepositions
- 4. Conditional Sentences
- 5. Passive Voice
- 6. Reported Speech

◆ Table: `grammar_lessons`

Mục đích: Bài học lý thuyết và thực hành ngữ pháp

Column	Type	Constraints	Mô tả
<code>id</code>	bigserial	PRIMARY KEY	
<code>topic_id</code>	bigint	FK → <code>grammar_topics(id)</code>	
<code>title</code>	varchar(200)	NOT NULL	Tiêu đề bài học
<code>lesson_type</code>	varchar(50)	NOT NULL	'THEORY' hoặc 'PRACTICE'
<code>content</code>	text		HTML/Markdown (cho THEORY)
<code>order_index</code>	integer	NOT NULL	Thứ tự bài học
<code>estimated_duration</code>	integer	DEFAULT 30	Thời gian đọc (giây)
<code>points_reward</code>	integer	DEFAULT 10	Điểm thưởng
<code>is_active</code>	boolean	DEFAULT true	
<code>created_at</code>	timestamp	DEFAULT NOW()	

Lesson Structure:

Topic: "Present Simple"

- └ Lesson 1: [THEORY] What is Present Simple?
- └ Lesson 2: [PRACTICE] Present Simple - Exercise 1
- └ Lesson 3: [THEORY] When to use Present Simple?
- └ Lesson 4: [PRACTICE] Present Simple - Final Test

### Business Logic:

- THEORY: User phải đọc hết (scroll to bottom) và dành ít nhất `estimated_duration` giây
- PRACTICE: User làm 15-20 câu hỏi, đạt  $\geq 70\%$  để hoàn thành

### ◆ Table: `user_grammar_progress`

**Mục đích:** Track tiến độ học ngữ pháp

Column	Type	Constraints	Mô tả
<code>id</code>	bigserial	PRIMARY KEY	
<code>user_id</code>	bigint	FK → users(id)	
<code>lesson_id</code>	bigint	FK → grammar_lessons(id)	
<code>is_completed</code>	boolean	DEFAULT false	Hoàn thành chưa?
<code>score_percentage</code>	decimal(5,2)	DEFAULT 0	Điểm % (cho PRACTICE)
<code>reading_time</code>	integer	DEFAULT 0	Thời gian đọc (giây, THEORY)
<code>has_scrolled_to_end</code>	boolean	DEFAULT false	Đã scroll hết? (THEORY)
<code>attempts</code>	integer	DEFAULT 0	Số lần làm bài
<code>completed_at</code>	timestamp		Thời điểm hoàn thành
<code>created_at</code>	timestamp	DEFAULT NOW()	
<code>updated_at</code>	timestamp	DEFAULT NOW()	

### Constraints:

sql

**UNIQUE** (user\_id, lesson\_id)

### Completion Criteria:

- **THEORY:** `reading_time >= estimated_duration` AND `has_scrolled_to_end = true`
- **PRACTICE:** `score_percentage >= 70`

## 4. READING MODULE

### ◆ Table: `reading_lessons`

**Mục đích:** Bài đọc hiểu (passage + questions)

Column	Type	Constraints	Mô tả
<code>id</code>	bigserial	PRIMARY KEY	
<code>title</code>	varchar(200)	NOT NULL	Tiêu đề bài đọc
<code>content</code>	text	NOT NULL	Đoạn văn tiếng Anh
<code>content_translation</code>	text		Bản dịch tiếng Việt
<code>level_required</code>	varchar(20)	DEFAULT 'BEGINNER'	
<code>order_index</code>	integer	NOT NULL	
<code>points_reward</code>	integer	DEFAULT 25	
<code>is_active</code>	boolean	DEFAULT true	
<code>created_at</code>	timestamp	DEFAULT NOW()	

**Question Structure:**

- 5 câu Multiple Choice
- 5 câu True/False
- 5 câu Short Answer
- **Total: 15 questions per lesson**

◆ **Table:** `user_reading_progress`

**Mục đích:** Track kết quả làm bài reading

Column	Type	Constraints	Mô tả
<code>id</code>	bigserial	PRIMARY KEY	
<code>user_id</code>	bigint	FK → users(id)	
<code>lesson_id</code>	bigint	FK → reading_lessons(id)	
<code>is_completed</code>	boolean	DEFAULT false	
<code>score_percentage</code>	decimal(5,2)	DEFAULT 0	12/15 = 80%
<code>attempts</code>	integer	DEFAULT 0	
<code>completed_at</code>	timestamp		
<code>created_at</code>	timestamp	DEFAULT NOW()	
<code>updated_at</code>	timestamp	DEFAULT NOW()	

**Constraints:**

```
sql
UNIQUE (user_id, lesson_id)
```



5. LISTENING MODULE

◆ Table: listening\_lessons

Mục đích: Bài tập nghe điền từ (fill-in-the-blank)

Column	Type	Constraints	Mô tả
id	bigserial	PRIMARY KEY	
title	varchar(200)	NOT NULL	
description	text		
audio_url	varchar(500)	NOT NULL	Link file mp3/wav
transcript	text	NOT NULL	Transcript đầy đủ
transcript_with_blanks	text	NOT NULL	Có chỗ trống: {blank_1}
level_required	varchar(20)	DEFAULT 'BEGINNER'	
duration_seconds	integer		Độ dài audio
order_index	integer	NOT NULL	
points_reward	integer	DEFAULT 20	
is_active	boolean	DEFAULT true	
created_at	timestamp	DEFAULT NOW()	

Ví dụ:

transcript:

"Good morning! I'd like to order a pizza with extra cheese and mushrooms."

transcript\_with\_blanks:

"Good morning! I'd like to {blank\_1} a pizza with {blank\_2} cheese and {blank\_3}."

◆ Table: listening\_blanks

Mục đích: Định nghĩa các chỗ trống và đáp án

Column	Type	Constraints	Mô tả
<code>id</code>	bigserial	PRIMARY KEY	
<code>lesson_id</code>	bigint	FK → listening_lessons(id)	
<code>blank_number</code>	integer	NOT NULL	1, 2, 3...
<code>correct_word</code>	varchar(200)	NOT NULL	Từ đúng
<code>word_meaning_vi</code>	text		Nghĩa tiếng Việt
<code>word_pronunciation</code>	varchar(300)		IPA
<code>example_sentence</code>	text		Câu ví dụ khác
<code>example_translation</code>	text		
<code>start_time_seconds</code>	integer		Từ xuất hiện lúc nào (3.5s)
<code>end_time_seconds</code>	integer		Kết thúc lúc nào (4.2s)

#### Constraints:

```
sql
UNIQUE (lesson_id, blank_number)
```

#### Use Case:

- User click vào blank → highlight audio timestamp
- User có thể nghe lại đoạn từ `start_time` đến `end_time`

#### ◆ Table: `user_listening_progress`

Column	Type	Constraints	Mô tả
<code>id</code>	bigserial	PRIMARY KEY	
<code>user_id</code>	bigint	FK → users(id)	
<code>lesson_id</code>	bigint	FK → listening_lessons(id)	
<code>is_completed</code>	boolean	DEFAULT false	
<code>score_percentage</code>	decimal(5,2)	DEFAULT 0	
<code>attempts</code>	integer	DEFAULT 0	
<code>completed_at</code>	timestamp		
<code>created_at</code>	timestamp	DEFAULT NOW()	
<code>updated_at</code>	timestamp	DEFAULT NOW()	

#### Constraints:

```
sql
UNIQUE (user_id, lesson_id)
```

◆ **Table:** `user_listening_submissions`

**Mục đích:** Lưu từng lần user điền từ (để phân tích)

Column	Type	Constraints	Mô tả
<code>id</code>	bigserial	PRIMARY KEY	
<code>user_id</code>	bigint	FK → users(id)	
<code>lesson_id</code>	bigint	FK → listening_lessons(id)	
<code>blank_number</code>	integer	NOT NULL	Chỗ trống số mấy
<code>user_answer</code>	varchar(200)	NOT NULL	User điền gì
<code>is_correct</code>	boolean	NOT NULL	Đúng/sai
<code>submitted_at</code>	timestamp	DEFAULT NOW()	

**Analytics:**

```
sql
-- Từ nào user hay sai nhất?
SELECT blank_number, correct_word,
       COUNT(*) as total_attempts,
       SUM(CASE WHEN is_correct THEN 1 ELSE 0 END) as correct_count
FROM user_listening_submissions uls
JOIN listening_blanks lb ON uls.lesson_id = lb.lesson_id
  AND uls.blank_number = lb.blank_number
WHERE uls.user_id = 1
GROUP BY blank_number, correct_word
ORDER BY correct_count ASC;
```

## 6. WRITING MODULE

◆ **Table:** `writing_exercises`

**Mục đích:** Bài tập viết (dịch hoặc tự do)

Column	Type	Constraints	Mô tả
id	bigserial	PRIMARY KEY	
title	varchar(200)	NOT NULL	
exercise_type	varchar(50)	NOT NULL	Loại bài tập
prompt	text	NOT NULL	Yêu cầu đề bài
source_text	text		Văn bản gốc (cho bài dịch)
reference_translation	text		Bản dịch tham khảo
level_required	varchar(20)	DEFAULT 'BEGINNER'	
min_words	integer	DEFAULT 50	Cho FREE_WRITING
max_words	integer	DEFAULT 200	Cho FREE_WRITING
order_index	integer	NOT NULL	
points_reward	integer	DEFAULT 35	
is_active	boolean	DEFAULT true	
created_at	timestamp	DEFAULT NOW()	

Exercise Types:

```
sql
CHECK (exercise_type IN ('TRANSLATION_EN_VI', 'TRANSLATION_VI_EN', 'FREE_WRITING'))
```

3 Loại bài tập:

- 1. TRANSLATION\_EN\_VI: Dịch Anh → Việt
- 2. TRANSLATION\_VI\_EN: Dịch Việt → Anh
- 3. FREE\_WRITING: Viết tự do theo topic

◆ Table: writing\_segments

Mục đích: Chia bài dịch thành các đoạn nhỏ (câu hoặc đoạn văn)

Column	Type	Constraints	Mô tả
id	bigserial	PRIMARY KEY	
exercise_id	bigint	FK → writing_exercises(id)	
segment_number	integer	NOT NULL	Thứ tự: 1, 2, 3...
segment_type	varchar(20)	NOT NULL	'SENTENCE' hoặc 'PARAGRAPH'
source_text	text	NOT NULL	Văn bản cần dịch
reference_translation	text	NOT NULL	Bản dịch tham khảo
sentence_count	integer	DEFAULT 1	Số câu trong segment

Constraints:

sql

**UNIQUE** (exercise\_id, segment\_number)

**CHECK** (segment\_type **IN** ('SENTENCE', 'PARAGRAPH'))

### Ví dụ - Dịch từng câu:

sql

Exercise: "Dịch 5 câu sau sang tiếng Anh"

Segment 1 (SENTENCE):

source: "Tôi thích học tiếng Anh."

reference: "I like learning English."

sentence\_count: 1

Segment 2 (SENTENCE):

source: "Hôm nay trời đẹp."

reference: "The weather is nice today."

sentence\_count: 1

### Ví dụ - Dịch từng đoạn:

sql

Exercise: "Dịch đoạn văn sau"

Segment 1 (PARAGRAPH):

source: "Tôi thích học tiếng Anh. Nó giúp tôi giao tiếp tốt hơn. Tôi học mỗi ngày."

reference: "I like learning English. It helps me communicate better. I study every day."

sentence\_count: 3

Segment 2 (PARAGRAPH):

source: "Học tiếng Anh không khó. Quan trọng là kiên trì."

reference: "Learning English is not difficult. The important thing is perseverance."

sentence\_count: 2

◆ **Table:** `user_writing_progress`

Column	Type	Constraints	Mô tả
<code>id</code>	bigserial	PRIMARY KEY	
<code>user_id</code>	bigint	FK → users(id)	
<code>exercise_id</code>	bigint	FK → writing_exercises(id)	
<code>is_completed</code>	boolean	DEFAULT false	
<code>best_score</code>	integer	DEFAULT 0	Điểm cao nhất (0-100)
<code>attempts</code>	integer	DEFAULT 0	
<code>completed_at</code>	timestamp		

Column	Type	Constraints	Mô tả
<code>created_at</code>	timestamp	DEFAULT NOW()	
<code>updated_at</code>	timestamp	DEFAULT NOW()	

**Constraints:**

```
sql
UNIQUE (user_id, exercise_id)
```

◆ **Table:** `user_writing_submissions`

**Mục đích:** Lưu bài viết của user (cho AI chấm điểm)

Column	Type	Constraints	Mô tả
<code>id</code>	bigserial	PRIMARY KEY	
<code>user_id</code>	bigint	FK → users(id)	
<code>exercise_id</code>	bigint	FK → writing_exercises(id)	
<code>submission_type</code>	varchar(50)	NOT NULL	'FULL_TEXT' hoặc 'SEGMENT'
<code>segment_number</code>	integer		Segment thứ mấy (nếu dịch từng đoạn)
<code>text_content</code>	text	NOT NULL	Bài user viết
<code>ai_score</code>	integer	DEFAULT 0	AI chấm điểm (0-100)
<code>ai_feedback</code>	text		AI feedback chi tiết
<code>word_count</code>	integer		Số từ
<code>submitted_at</code>	timestamp	DEFAULT NOW()	

**Constraints:**

```
sql
CHECK (submission_type IN ('FULL_TEXT', 'SEGMENT'))
```

**Flow:**

```
User dịch segment 1 → Submit
→ Backend gọi AI API (OpenAI/Claude)
→ AI so sánh với reference_translation
→ AI cho điểm + feedback
→ Lưu vào database

User dịch segment 2 → Submit → ...
```

7. SPEAKING MODULE

◆ Table: speaking\_topics

Mục đích: Chủ đề luyện speaking theo tình huống

Column	Type	Constraints	Mô tả
id	bigserial	PRIMARY KEY	
title	varchar(200)	NOT NULL	VD: "Job Interview"
description	text		
scenario	text	NOT NULL	Mô tả tình huống
level_required	varchar(20)	DEFAULT 'BEGINNER'	
suggested_vocabulary	json		Array từ vựng gợi ý
order_index	integer	NOT NULL	
points_reward	integer	DEFAULT 30	
is_active	boolean	DEFAULT true	
created_at	timestamp	DEFAULT NOW()	

Ví dụ:

```
json
{
  "title": "Job Interview",
  "scenario": "You are interviewing for a software engineer position. Answer the interviewer's questions confidently and clearly.",
  "suggested_vocabulary": [
    "experience",
    "skills",
    "qualification",
    "teamwork",
    "problem-solving",
    "deadline"
  ]
}
```

Mỗi topic có 3-5 questions (lưu trong bảng questions)

◆ Table: userSpeakingProgress

Column	Type	Constraints	Mô tả
id	bigserial	PRIMARY KEY	
user_id	bigint	FK → users(id)	
topic_id	bigint	FK → speaking_topics(id)	
is_completed	boolean	DEFAULT false	
best_score	integer	DEFAULT 0	Điểm cao nhất

Column	Type	Constraints	Mô tả
attempts	integer	DEFAULT 0	
completed_at	timestamp		
created_at	timestamp	DEFAULT NOW()	
updated_at	timestamp	DEFAULT NOW()	

**Constraints:**

```
sql
UNIQUE (user_id, topic_id)
```

◆ **Table:** user\_speaking\_submissions

**Mục đích:** Lưu audio recording + AI feedback

Column	Type	Constraints	Mô tả
id	bigserial	PRIMARY KEY	
user_id	bigint	FK → users(id)	
topic_id	bigint	FK → speaking_topics(id)	
question_id	bigint	FK → questions(id)	
audio_url	varchar(500)	NOT NULL	File ghi âm của user
transcription	text		AI chuyển speech → text
pronunciation_score	integer		0-100
grammar_score	integer		0-100
fluency_score	integer		0-100
overall_score	integer		Điểm tổng (0-100)
ai_feedback	text		Nhận xét chi tiết từ AI
submitted_at	timestamp	DEFAULT NOW()	

**AI Scoring Components:**

- **Pronunciation (30%):** Phát âm chuẩn không?
- **Grammar (30%):** Ngữ pháp đúng không?
- **Fluency (20%):** Nói lưu loát, ít ngập ngừng
- **Content (20%):** Nội dung có trả lời đúng câu hỏi không?

**Flow:**



1. User record audio → Upload to S3/Cloudinary
2. Backend gọi AI Speech-to-Text (Whisper API)
3. AI phân tích pronunciation, grammar, fluency
4. Tính overall\_score = weighted average
5. AI generate feedback
6. Lưu vào database

## 8. SHARED TABLES

### ◆ Table: `questions`

**Mục đích:** Bảng câu hỏi dùng chung cho Grammar, Reading, Speaking

Column	Type	Constraints	Mô tả
<code>id</code>	bigserial	PRIMARY KEY	
<code>parent_type</code>	varchar(20)	NOT NULL	'GRAMMAR', 'READING', 'SPEAKING'
<code>parent_id</code>	bigint	NOT NULL	ID của lesson/topic
<code>question_text</code>	text	NOT NULL	Nội dung câu hỏi
<code>question_type</code>	varchar(50)	NOT NULL	Loại câu hỏi
<code>correct_answer</code>	text	NOT NULL	Đáp án đúng
<code>explanation</code>	text		Giải thích đáp án
<code>points</code>	integer	DEFAULT 1	Điểm thưởng
<code>order_index</code>	integer	DEFAULT 0	Thứ tự câu hỏi
<code>suggested_answer</code>	text		Cho SPEAKING
<code>time_limit_seconds</code>	integer		Cho SPEAKING
<code>created_at</code>	timestamp	DEFAULT NOW()	

### Constraints:

```
sql
CHECK (parent_type IN ('GRAMMAR', 'READING', 'SPEAKING'))
```

### Question Types:

#### Grammar:

- `MULTIPLE_CHOICE`: Chọn đáp án đúng
- `FILL_BLANK`: Điền từ vào chỗ trống
- `VERB_FORM`: Chia động từ
- `TRANSLATE`: Dịch câu

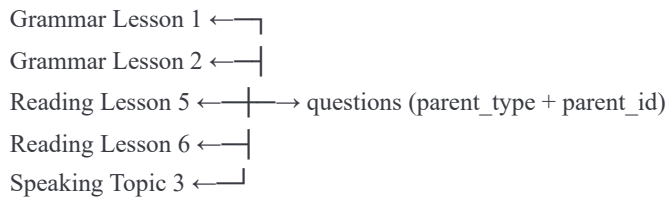
#### Reading:

- **MULTIPLE\_CHOICE**: Chọn A/B/C/D
- **TRUE\_FALSE**: Đúng/Sai
- **SHORT\_ANSWER**: Trả lời ngắn

### Speaking:

- **OPEN\_ENDED**: Câu hỏi mở (user tự trả lời)

### Polymorphic Relationship:



### ◆ Table: **question\_options**

**Mục đích:** Các lựa chọn cho câu hỏi Multiple Choice

Column	Type	Constraints	Mô tả
<b>id</b>	bigserial	PRIMARY KEY	
<b>question_id</b>	bigint	FK → questions(id)	
<b>option_text</b>	varchar(500)	NOT NULL	Nội dung lựa chọn
<b>is_correct</b>	boolean	DEFAULT false	Đáp án đúng?
<b>order_index</b>	integer	NOT NULL	Thứ tự A, B, C, D

### Ví dụ:

```

sql

Question: "She _____ to school every day."

Options:
1. "go"    (is_correct = false, order_index = 1)
2. "goes"  (is_correct = true, order_index = 2)
3. "going" (is_correct = false, order_index = 3)
4. "went"  (is_correct = false, order_index = 4)
  
```

## 9. INDEXES

### Performance Indexes

```

sql
  
```

-- User Progress Indexes

```
CREATE INDEX idx_user_grammar_progress_user
  ON user_grammar_progress(user_id, is_completed);
```

```
CREATE INDEX idx_user_listening_progress_user
  ON user_listening_progress(user_id, is_completed);
```

```
CREATE INDEX idx_user_reading_progress_user
  ON user_reading_progress(user_id, is_completed);
```

```
CREATE INDEX idx_user_speaking_progress_user
  ON user_speaking_progress(user_id, is_completed);
```

```
CREATE INDEX idx_user_writing_progress_user
  ON user_writing_progress(user_id, is_completed);
```

-- Vocabulary Spaced Repetition

```
CREATE INDEX idx_vocabulary_progress_next_review
  ON user_vocabulary_progress(user_id, next_review_at)
  WHERE is_learning = true;
```

-- Lesson Ordering

```
CREATE INDEX idx_grammar_lessons_topic_order
  ON grammar_lessons(topic_id, order_index);
```

```
CREATE INDEX idx_listening_lessons_order
  ON listening_lessons(order_index, level_required);
```

```
CREATE INDEX idx_reading_lessons_order
  ON reading_lessons(order_index, level_required);
```

```
CREATE INDEX idx_speaking_topics_order
  ON speaking_topics(order_index, level_required);
```

```
CREATE INDEX idx_writing_exercises_order
  ON writing_exercises(order_index, level_required);
```

-- Questions (Polymorphic)

```
CREATE INDEX idx_questions_parent
  ON questions(parent_type, parent_id, order_index);
```

```
CREATE INDEX idx_question_options_question
  ON question_options(question_id, order_index);
```

-- Submissions History

```
CREATE INDEX idx_speaking_submissions_user_topic
  ON user_speaking_submissions(user_id, topic_id, submitted_at DESC);
```

```
CREATE INDEX idx_writing_submissions_user_exercise
  ON user_writing_submissions(user_id, exercise_id, submitted_at DESC);
```

```
CREATE INDEX idx_listening_submissions_user_lesson
ON user_listening_submissions(user_id, lesson_id, submitted_at DESC);
```

-- Vocabulary

```
CREATE INDEX idx_vocabulary_topics_creator
ON vocabulary_topics(created_by)
WHERE is_system = false;
```

```
CREATE INDEX idx_topic_words_topic
ON topic_words(topic_id, order_index);
```

## 10. ENTITY RELATIONSHIPS

### ER Diagram (Text Format)

```
users (1) ———< (N) user_vocabulary_progress
users (1) ———< (N) user_grammar_progress
users (1) ———< (N) user_reading_progress
users (1) ———< (N) user_listening_progress
users (1) ———< (N) user_writing_progress
users (1) ———< (N) user_speaking_progress
users (1) ———< (N) vocabulary_topics (created_by)

vocabulary_topics (N) —< topic_words >— (N) vocabulary_words
vocabulary_words (1) ———< (N) user_vocabulary_progress

grammar_topics (1) ———< (N) grammar_lessons
grammar_lessons (1) ———< (N) user_grammar_progress
grammar_lessons (1) ———< (N) questions (polymorphic)

reading_lessons (1) ———< (N) user_reading_progress
reading_lessons (1) ———< (N) questions (polymorphic)

listening_lessons (1) ———< (N) listening_blanks
listening_lessons (1) ———< (N) user_listening_progress
listening_lessons (1) ———< (N) user_listening_submissions

writing_exercises (1) ———< (N) writing_segments
writing_exercises (1) ———< (N) user_writing_progress
writing_exercises (1) ———< (N) user_writing_submissions

speaking_topics (1) ———< (N) questions (polymorphic)
speaking_topics (1) ———< (N) user_speaking_progress
speaking_topics (1) ———< (N) user_speaking_submissions

questions (1) ———< (N) question_options
questions (1) ———< (N) user_speaking_submissions
```

## COMMON QUERIES

### 1. Lấy từ vựng cần ôn tập hôm nay

```
sql

SELECT
    w.word,
    w.meaning_vi,
    w.pronunciation,
    uvp.mastery_level,
    uvp.next_review_at
FROM user_vocabulary_progress uvp
JOIN vocabulary_words w ON uvp.word_id = w.id
WHERE uvp.user_id = ?
    AND uvp.is_learning = true
    AND uvp.next_review_at <= NOW()
ORDER BY uvp.next_review_at
LIMIT 20;
```

### 2. Dashboard - Tiến độ học tập của user

```
sql

SELECT
    'Grammar' as module,
    COUNT(*) as total_lessons,
    SUM(CASE WHEN is_completed THEN 1 ELSE 0 END) as completed,
    ROUND(AVG(score_percentage), 2) as avg_score
FROM user_grammar_progress
WHERE user_id = ?

UNION ALL

SELECT
    'Reading' as module,
    COUNT(*) as total_lessons,
    SUM(CASE WHEN is_completed THEN 1 ELSE 0 END) as completed,
    ROUND(AVG(score_percentage), 2) as avg_score
FROM user_reading_progress
WHERE user_id = ?

-- Tương tự cho các module khác...
```

### 3. Lấy bài học tiếp theo (Grammar)

```
sql
```

```

SELECT gl.*
FROM grammar_lessons gl
LEFT JOIN user_grammar_progress ugp
  ON gl.id = ugp.lesson_id AND ugp.user_id = ?
WHERE gl.is_active = true
  AND (ugp.is_completed IS NULL OR ugp.is_completed = false)
  AND gl.level_required <= (SELECT english_level FROM users WHERE id = ?)
ORDER BY gl.order_index
LIMIT 1;

```

#### 4. Leaderboard - Top điểm cao nhất

```

sql

SELECT
  u.username,
  u.full_name,
  u.total_points,
  u.streak_days,
  RANK() OVER (ORDER BY u.total_points DESC) as rank
FROM users u
WHERE u.is_active = true
ORDER BY u.total_points DESC
LIMIT 10;

```

#### 5. Phân tích lỗi thường gặp (Listening)

```

sql

SELECT
  lb.correct_word,
  lb.word_meaning_vi,
  COUNT(*) as total_attempts,
  SUM(CASE WHEN uls.is_correct THEN 0 ELSE 1 END) as error_count,
  ROUND(
    SUM(CASE WHEN uls.is_correct THEN 0 ELSE 1 END) * 100.0 / COUNT(*),
    2
  ) as error_rate
FROM user_listening_submissions uls
JOIN listening_blanks lb
  ON uls.lesson_id = lb.lesson_id
  AND uls.blank_number = lb.blank_number
WHERE uls.user_id = ?
GROUP BY lb.correct_word, lb.word_meaning_vi
HAVING error_count > 0
ORDER BY error_rate DESC
LIMIT 20;

```

#### 6. Lịch sử Speaking submissions

```

sql

```

## SELECT

```
st.title as topic,
q.question_text,
uss.overall_score,
uss.pronunciation_score,
uss.grammar_score,
uss.fluency_score,
uss.ai_feedback,
uss.submitted_at
FROM user_speaking_submissions uss
JOIN speaking_topics st ON uss.topic_id = st.id
JOIN questions q ON uss.question_id = q.id
WHERE uss.user_id = ?
ORDER BY uss.submitted_at DESC
LIMIT 20;
```

## SECURITY CONSIDERATIONS

### 1. Password Hashing

```
java
// Spring Boot - BCryptPasswordEncoder
@Bean
public PasswordEncoder passwordEncoder() {
    return new BCryptPasswordEncoder(12);
}
```

### 2. API Authentication

- JWT Token-based authentication
- Refresh token mechanism
- Role-based access control (USER, ADMIN, TEACHER)

### 3. File Upload Security

- Validate file types (audio: mp3/wav, images: jpg/png)
- Limit file size (audio: 10MB, images: 5MB)
- Use cloud storage (S3, Cloudinary) instead of local storage
- Generate unique filenames (UUID)

### 4. SQL Injection Prevention

- Always use JPA/Hibernate (no raw SQL)
- Use PreparedStatement if raw SQL needed
- Validate and sanitize user inputs

## 5. Rate Limiting

- Limit AI API calls (speaking/writing feedback)
  - Implement request throttling per user
- 

## OPTIMIZATION TIPS

### 1. Caching Strategy

```
java

// Cache commonly accessed data
@Cacheable("vocabulary-topics")
public List<VocabularyTopic> getSystemTopics() { ... }

@Cacheable("grammar-topics")
public List<GrammarTopic> getAllTopics() { ... }
```

### 2. Lazy Loading

```
java

@Entity
public class GrammarLesson {
    @ManyToOne(fetch = FetchType.LAZY)
    private GrammarTopic topic;

    @OneToMany(mappedBy = "lesson", fetch = FetchType.LAZY)
    private List<Question> questions;
}
```

### 3. Pagination

```
java

// Always paginate large result sets
public Page<ReadingLesson> getLessons(Pageable pageable) {
    return readingLessonRepository.findAll(pageable);
}
```

### 4. Database Connection Pool

```
properties

# application.properties
spring.datasource.hikari.maximum-pool-size=20
spring.datasource.hikari.minimum-idle=5
spring.datasource.hikari.connection-timeout=30000
```

## 5. N+1 Query Problem



```
java
```

```
// BAD - N+1 queries
```

```
List<Lesson> lessons = lessonRepo.findAll();
for (Lesson lesson : lessons) {
    lesson.getQuestions().size(); // Lazy load causes N queries
}
```

```
// GOOD - Use JOIN FETCH
```

```
@Query("SELECT l FROM Lesson l LEFT JOIN FETCH l.questions WHERE l.id = :id")
Lesson findByIdWithQuestions(@Param("id") Long id);
```



## MIGRATION SCRIPT

### Initial Setup

```
sql
```

```
-- Run in order:
```

1. Create all tables (users first)
2. Add foreign keys
3. Create indexes
4. Insert seed data

```
-- Seed data examples:
```

```
INSERT INTO grammar_topics (name, description, level_required, order_index) VALUES
('Present Simple', 'Learn about present simple tense', 'BEGINNER', 1),
('Present Continuous', 'Learn about present continuous tense', 'BEGINNER', 2),
('Past Simple', 'Learn about past simple tense', 'BEGINNER', 3);
```

```
INSERT INTO vocabulary_topics (name, description, is_system, order_index) VALUES
('Animals', 'Common animal names', true, 1),
('Food & Drinks', 'Food and beverage vocabulary', true, 2),
('Daily Activities', 'Common daily activities', true, 3);
```



## TESTING RECOMMENDATIONS

### 1. Unit Tests

- Test business logic (SRS algorithm, scoring)
- Test validations
- Mock external services (AI APIs)

### 2. Integration Tests

```
java
```

```
@SpringBootTest
@AutoConfigureTestDatabase
public class VocabularyServiceTest {
    @Test
    void testSpacedRepetitionScheduling() {
        // Test SRS algorithm
    }
}
```

### 3. Performance Tests

- Test with large datasets (10k users, 100k words)
  - Measure query performance
  - Test concurrent user submissions
- 



## FUTURE ENHANCEMENTS

### Phase 2 Features

#### 1. Social Learning

- User can share custom vocabulary topics
- Comments and ratings on lessons
- Study groups

#### 2. Gamification

- Achievements/Badges table
- Daily challenges
- Competitions/Tournaments

#### 3. Advanced Analytics

- Learning curve analysis
- Personalized recommendations
- Weak points detection

#### 4. Mobile App Support

- Offline mode (sync when online)
  - Push notifications for reviews
  - Speech recognition on mobile
-

### Backup Strategy

- Daily automated backups
- Point-in-time recovery enabled
- Test restore quarterly

### Monitoring

- Database performance metrics
- Slow query logs
- Connection pool monitoring
- Storage usage alerts

### Regular Tasks

- Weekly: Review slow queries
  - Monthly: Update statistics, vacuum
  - Quarterly: Review indexes, optimize
- 

## CHECKLIST

### Before Going to Production

- ☐ All tables created with correct constraints
  - ☐ Foreign keys properly set up
  - ☐ Indexes created for all frequently queried columns
  - ☐ Seed data inserted (topics, levels, etc.)
  - ☐ JPA entities match database schema
  - ☐ Password encryption enabled
  - ☐ JWT authentication configured
  - ☐ File upload security implemented
  - ☐ Rate limiting configured
  - ☐ Error logging set up
  - ☐ Database backups automated
  - ☐ Performance testing completed
  - ☐ Security audit done
- 

## REFERENCES

### Technologies Used

- **Database:** PostgreSQL 14+
- **Backend:** Spring Boot 3.x, JPA/Hibernate

- **Authentication:** Spring Security + JWT
- **File Storage:** AWS S3 / Cloudinary
- **AI Services:** OpenAI API / Anthropic Claude API
- **Cache:** Redis (optional)

## Documentation Links


- Spring Data JPA: <https://spring.io/projects/spring-data-jpa>
- PostgreSQL Docs: <https://www.postgresql.org/docs/>
- Hibernate: <https://hibernate.org/orm/documentation/>

---

**Version:** 1.0

**Last Updated:** October 11, 2025

**Author:** Database Schema Design Team

**Status:**  Ready for Implementation







---

## SUMMARY

Hệ thống bao gồm **25 bảng** được tổ chức thành **7 modules chính**:

1. **Users** - Quản lý tài khoản
2. **Vocabulary** - Học từ vựng với SRS
3. **Grammar** - Lý thuyết + Bài tập ngữ pháp
4. **Reading** - Đọc hiểu + Câu hỏi
5. **Listening** - Nghe điền từ
6. **Writing** - Dịch và viết tự do
7. **Speaking** - Luyện nói với AI feedback

### Key Features:

-  Spaced Repetition System cho từ vựng
-  AI-powered feedback cho Writing & Speaking
-  Progressive learning path với level system
-  Comprehensive progress tracking
-  Flexible question system (polymorphic)
-  Gamification (points, streaks)

### Database Size Estimation:

- Small (1K users): ~500 MB

- Medium (10K users): ~5 GB
- Large (100K users): ~50 GB

Chúc bạn triển khai thành công! 🚀