**1. Bảng Course:**

* **Chức năng**: Lưu thông tin về các khóa học do người dùng tạo ra.
* **Các cột**:
  + course\_id: ID duy nhất cho mỗi khóa học (khóa chính, tự động tăng).
  + course\_name: Tên khóa học.
  + description: Mô tả chi tiết về khóa học (nếu có).
  + creator\_id: ID của người tạo khóa học, tham chiếu đến bảng User.
  + created\_at: Thời gian khóa học được tạo (mặc định là thời gian hiện tại).
  + updated\_at: Thời gian khóa học được cập nhật lần cuối.
* **Mục đích**: Quản lý các khóa học do người dùng tạo, bao gồm cả thông tin về người tạo khóa học.

**2. Bảng Submission\_Status:**

* **Chức năng**: Lưu trữ các trạng thái của bài nộp.
* **Các cột**:
  + status\_id: ID duy nhất cho mỗi trạng thái (khóa chính, tự động tăng).
  + status: Trạng thái của bài nộp, ví dụ: "Accepted", "Rejected".
* **Mục đích**: Xác định trạng thái của các bài nộp, giúp theo dõi kết quả của từng lần nộp bài.

**3. Bảng Admin\_Activity:**

* **Chức năng**: Ghi lại các hoạt động của quản trị viên (admin).
* **Các cột**:
  + activity\_id: ID duy nhất cho mỗi hoạt động (khóa chính, tự động tăng).
  + admin\_id: ID của quản trị viên, tham chiếu đến bảng User.
  + action: Hoạt động mà admin đã thực hiện (ví dụ: "delete", "create").
  + target\_type: Kiểu đối tượng mà admin tác động (có thể là "Course", "Problem", hoặc "User").
  + target\_id: ID của đối tượng bị tác động (có thể là course\_id, problem\_id, hoặc user\_id).
  + timestamp: Thời gian mà hoạt động được thực hiện.
* **Mục đích**: Ghi lại lịch sử hoạt động của admin để theo dõi quản trị.

**4. Bảng PasswordReset:**

* **Chức năng**: Lưu thông tin về các yêu cầu đặt lại mật khẩu.
* **Các cột**:
  + reset\_id: ID duy nhất cho mỗi yêu cầu reset (khóa chính, tự động tăng).
  + user\_id: ID của người dùng yêu cầu reset mật khẩu, tham chiếu đến bảng User.
  + reset\_token: Token duy nhất để xác nhận yêu cầu reset mật khẩu.
  + created\_at: Thời gian yêu cầu reset được tạo.
* **Mục đích**: Hỗ trợ tính năng đặt lại mật khẩu cho người dùng.

**5. Bảng Problem:**

* **Chức năng**: Lưu trữ thông tin về các bài toán mà người dùng có thể giải quyết.
* **Các cột**:
  + problem\_id: ID duy nhất cho mỗi bài toán (khóa chính, tự động tăng).
  + title: Tiêu đề của bài toán.
  + description: Mô tả chi tiết về bài toán.
  + input\_format: Định dạng đầu vào của bài toán.
  + output\_format: Định dạng đầu ra của bài toán.
  + constraints: Các ràng buộc mà bài toán yêu cầu.
  + sample\_input: Ví dụ về đầu vào cho bài toán.
  + sample\_output: Ví dụ về đầu ra tương ứng.
  + time\_limit: Giới hạn thời gian để giải bài toán.
  + memory\_limit: Giới hạn bộ nhớ cho bài toán.
  + category\_id: ID danh mục của bài toán, tham chiếu đến bảng Category.
  + created\_at: Thời gian bài toán được tạo.
* **Mục đích**: Quản lý các bài toán để người dùng có thể tham gia giải và nộp bài.

**6. Bảng Match:**

* **Chức năng**: Lưu thông tin về các trận đấu (giải đấu) liên quan đến việc giải các bài toán.
* **Các cột**:
  + match\_id: ID duy nhất cho mỗi trận đấu (khóa chính, tự động tăng).
  + match\_date: Ngày diễn ra trận đấu.
  + status: Trạng thái của trận đấu (ví dụ: "Completed", "Ongoing", "Pending").
  + created\_at: Thời gian thông tin về trận đấu được tạo.
  + updated\_at: Thời gian thông tin về trận đấu được cập nhật lần cuối.
* **Mục đích**: Quản lý các trận đấu hoặc giải đấu mà người dùng tham gia để giải bài toán.

**7. Bảng Blog:**

* **Chức năng**: Lưu các bài viết (blog) của người dùng.
* **Các cột**:
  + blog\_id: ID duy nhất cho mỗi bài viết (khóa chính, tự động tăng).
  + author\_id: ID của tác giả bài viết, tham chiếu đến bảng User.
  + title: Tiêu đề của bài viết.
  + content: Nội dung của bài viết.
  + created\_at: Thời gian bài viết được tạo.
* **Mục đích**: Cho phép người dùng viết và chia sẻ bài viết về các chủ đề liên quan đến giải bài toán hoặc học thuật.

**8. Bảng Comment:**

* **Chức năng**: Lưu các bình luận của người dùng về các bài toán.
* **Các cột**:
  + comment\_id: ID duy nhất cho mỗi bình luận (khóa chính, tự động tăng).
  + user\_id: ID của người dùng đăng bình luận, tham chiếu đến bảng User.
  + problem\_id: ID của bài toán mà bình luận thuộc về, tham chiếu đến bảng Problem.
  + content: Nội dung bình luận.
  + created\_at: Thời gian bình luận được tạo.
* **Mục đích**: Cho phép người dùng thảo luận và chia sẻ ý kiến về các bài toán.

**9. Bảng Ratings:**

* **Chức năng**: Lưu các đánh giá (ratings) của người dùng về các bài toán.
* **Các cột**:
  + rating\_id: ID duy nhất cho mỗi đánh giá (khóa chính, tự động tăng).
  + user\_id: ID của người dùng đánh giá bài toán, tham chiếu đến bảng User.
  + problem\_id: ID của bài toán được đánh giá, tham chiếu đến bảng Problem.
  + rating: Điểm đánh giá (1-5).
  + created\_at: Thời gian đánh giá được tạo.
* **Mục đích**: Cho phép người dùng đánh giá chất lượng của các bài toán, từ đó có thể phân loại hoặc sắp xếp bài toán theo độ yêu thích.

**10. Bảng Submission:**

* **Chức năng**: Lưu các bài nộp của người dùng cho các bài toán.
* **Các cột**:
  + submission\_id: ID duy nhất cho mỗi bài nộp (khóa chính, tự động tăng).
  + problem\_id: ID của bài toán mà người dùng nộp bài, tham chiếu đến bảng Problem.
  + user\_id: ID của người dùng nộp bài, tham chiếu đến bảng User.
  + code: Mã nguồn của bài nộp.
  + submitted\_at: Thời gian bài nộp được gửi.
  + status\_id: Trạng thái của bài nộp, tham chiếu đến bảng Submission\_Status.
* **Mục đích**: Theo dõi các bài nộp của người dùng cho các bài toán, bao gồm thông tin về mã nguồn và kết quả.

**11. [Class\_Student]**

* **Mục đích**: Quản lý thông tin sinh viên tham gia lớp học nào và thời gian tham gia lớp học đó.
* **Các cột**:
  + class\_id: ID của lớp học.
  + student\_id: ID của sinh viên.
  + joined\_at: Thời gian sinh viên tham gia lớp học.
* **Khóa chính**: (class\_id, student\_id) là khóa chính kép.
* **Khóa ngoại**: Liên kết tới bảng Class và User.

**12. [Admin]**

* **Mục đích**: Quản lý các tài khoản quản trị viên.
* **Các cột**:
  + admin\_id: ID quản trị viên.
  + user\_id: ID người dùng tương ứng với quản trị viên.
  + access\_level: Mức quyền truy cập của quản trị viên (từ 1 đến 10).
  + created\_at, updated\_at: Thời gian tạo và cập nhật.
* **Khóa chính**: admin\_id.
* **Khóa ngoại**: Liên kết với bảng User để lấy thông tin người dùng.

**13. [Submission\_TestCase\_Result]**

* **Mục đích**: Lưu kết quả bài nộp của sinh viên khi chạy qua các bộ test case.
* **Các cột**:
  + result\_id: ID kết quả.
  + submission\_id: ID bài nộp.
  + testcase\_id: ID bộ test case.
  + status: Trạng thái bài nộp (Accepted, Wrong Answer, Compilation Error, v.v.).
  + runtime, memory\_used: Thời gian chạy và bộ nhớ sử dụng.
  + actual\_output, error\_message: Đầu ra thực tế và thông báo lỗi nếu có.
  + graded\_at: Thời gian chấm điểm.
* **Khóa chính**: result\_id.
* **Khóa ngoại**: Liên kết tới Submission và Problem\_TestCase.

**14. [Category]**

* **Mục đích**: Quản lý các danh mục liên quan đến bài tập hoặc các nội dung khác.
* **Các cột**:
  + category\_id: ID danh mục.
  + title: Tên danh mục.
  + about: Mô tả danh mục.
  + created\_at: Thời gian tạo danh mục.
* **Khóa chính**: category\_id.

**15. [Blog\_Comment]**

* **Mục đích**: Quản lý bình luận của người dùng trên các bài blog.
* **Các cột**:
  + comment\_id: ID bình luận.
  + blog\_id: ID bài blog.
  + user\_id: ID người dùng bình luận.
  + content: Nội dung bình luận.
  + rank\_flag: Cờ đánh giá.
  + created\_at: Thời gian bình luận.
* **Khóa chính**: comment\_id.
* **Khóa ngoại**: Liên kết với bảng Blog và User.

**16. [Match\_User]**

* **Mục đích**: Quản lý điểm số và xếp hạng của người dùng trong các trận đấu.
* **Các cột**:
  + match\_user\_id: ID quan hệ giữa người dùng và trận đấu.
  + match\_id: ID trận đấu.
  + user\_id: ID người dùng.
  + score: Điểm số người dùng đạt được.
  + rank\_position: Vị trí xếp hạng.
* **Khóa chính**: match\_user\_id.
* **Khóa ngoại**: Liên kết tới bảng Match và User.

**17. [User\_Badge]**

* **Mục đích**: Lưu trữ thông tin về huy hiệu mà người dùng nhận được.
* **Các cột**:
  + user\_badge\_id: ID của quan hệ người dùng và huy hiệu.
  + user\_id: ID người dùng.
  + badge\_id: ID huy hiệu.
  + awarded\_at: Thời gian được cấp huy hiệu.
* **Khóa chính**: user\_badge\_id.
* **Khóa ngoại**: Liên kết tới bảng User và Badge.

**18. [Problem\_Language]**

* **Mục đích**: Lưu trữ thông tin về các ngôn ngữ lập trình có thể được sử dụng cho một bài tập.
* **Các cột**:
  + problem\_id: ID bài tập.
  + language: Ngôn ngữ lập trình được hỗ trợ (như Python, Java, C++,...).
* **Khóa chính**: (problem\_id, language) là khóa chính kép.
* **Khóa ngoại**: Liên kết tới bảng Problem.

**19. [Badge]**

* **Mục đích**: Lưu thông tin về các huy hiệu có thể được cấp cho người dùng.
* **Các cột**:
  + badge\_id: ID huy hiệu.
  + title: Tên huy hiệu.
  + description: Mô tả huy hiệu.
  + created\_at: Thời gian tạo huy hiệu.
* **Khóa chính**: badge\_id.

**20. [QnA]**

* **Mục đích**: Lưu trữ các câu hỏi và câu trả lời của người dùng.
* **Các cột**:
  + question\_id: ID câu hỏi.
  + user\_id: ID người dùng đặt câu hỏi.
  + question: Nội dung câu hỏi.
  + answer: Câu trả lời (nếu có).
  + created\_at: Thời gian tạo câu hỏi.
* **Khóa chính**: question\_id.
* **Khóa ngoại**: Liên kết với bảng User.

**21. [User\_Role]**

* **Mục đích**: Xác định vai trò của người dùng (sinh viên, giáo viên, hoặc quản trị viên).
* **Các cột**:
  + user\_role\_id: ID vai trò.
  + user\_id: ID người dùng.
  + role: Vai trò người dùng (student, teacher, admin).
* **Khóa chính**: user\_role\_id.

**22. [User]**

* **Mục đích**: Lưu thông tin về người dùng trong hệ thống.
* **Các cột**:
  + user\_id: ID người dùng.
  + username: Tên đăng nhập của người dùng.
  + full\_name: Tên đầy đủ của người dùng (nếu có).
  + address, school, phone\_number: Thông tin cá nhân như địa chỉ, trường học và số điện thoại (nếu có).
  + email: Địa chỉ email của người dùng.
  + password\_hash: Mã hash của mật khẩu.
  + role: Vai trò của người dùng (student, teacher, admin).
  + created\_at, updated\_at: Thời gian tạo và cập nhật thông tin.
* **Khóa chính**: user\_id.
* **Ràng buộc**:
  + Ràng buộc kiểm tra định dạng email (chk\_email\_format).
  + Kiểm tra vai trò người dùng (student, teacher, hoặc admin).
  + Tạo UNIQUE cho username và email để đảm bảo không có trùng lặp.

**23. [Match\_Rank]**

* **Mục đích**: Lưu trữ thông tin về xếp hạng của người dùng trong các trận đấu.
* **Các cột**:
  + rank\_id: ID của xếp hạng.
  + match\_id: ID trận đấu.
  + user\_id: ID người dùng tham gia trận đấu.
  + total\_score: Tổng điểm người dùng trong trận đấu.
  + rank\_position: Vị trí xếp hạng của người dùng.
  + created\_at, updated\_at: Thời gian tạo và cập nhật thông tin.
* **Khóa chính**: rank\_id.
* **Khóa ngoại**: Liên kết tới bảng Match và User.

**24. [Course\_Student]**

* **Mục đích**: Quản lý thông tin sinh viên tham gia khóa học nào và vai trò của họ trong khóa học.
* **Các cột**:
  + course\_id: ID của khóa học.
  + student\_id: ID sinh viên.
  + joined\_at: Thời gian tham gia khóa học.
  + role: Vai trò của người dùng trong khóa học (student hoặc instructor).
* **Khóa chính**: (course\_id, student\_id) là khóa chính kép.
* **Ràng buộc**: Kiểm tra role là student hoặc instructor.
* **Khóa ngoại**: Liên kết với bảng Course và User.

**25. [Notifications]**

* **Mục đích**: Quản lý các thông báo gửi đến người dùng trong khóa học.
* **Các cột**:
  + notification\_id: ID thông báo.
  + course\_id: ID khóa học liên quan.
  + sender\_id: ID người gửi thông báo.
  + message: Nội dung thông báo.
  + created\_at: Thời gian tạo thông báo.
* **Khóa chính**: notification\_id.
* **Khóa ngoại**: Liên kết với bảng Course và User.

**26. [Audit\_Log]**

* **Mục đích**: Ghi lại các thay đổi được thực hiện trên các bảng trong hệ thống để phục vụ kiểm tra và theo dõi.
* **Các cột**:
  + log\_id: ID của bản ghi log.
  + table\_name: Tên bảng bị thay đổi.
  + action: Loại hành động (INSERT, UPDATE, DELETE).
  + user\_id: ID người thực hiện thay đổi.
  + timestamp: Thời gian thay đổi.
  + changes: Thông tin chi tiết về thay đổi (nếu có).
* **Khóa chính**: log\_id.

**27. [Problem\_TestCase]**

* **Mục đích**: Quản lý thông tin các test case của một bài tập lập trình.
* **Các cột**:
  + testcase\_id: ID của test case.
  + problem\_id: ID bài tập liên quan.
  + input\_data: Dữ liệu đầu vào của test case.
  + expected\_output: Kết quả đầu ra mong đợi.
  + created\_at: Thời gian tạo test case.
  + points: Điểm số của test case (mặc định là 1).
* **Khóa chính**: testcase\_id.
* **Khóa ngoại**: Liên kết tới bảng Problem.

**28. [Assignment\_Manager]**

* **Mục đích**: Quản lý thông tin giao bài tập cho sinh viên bởi giáo viên hoặc quản lý.
* **Các cột**:
  + manager\_id: ID của bản ghi quản lý bài tập.
  + manager\_user\_id: ID người quản lý giao bài tập.
  + student\_user\_id: ID sinh viên được giao bài.
  + problem\_id: ID bài tập.
  + assigned\_at: Thời gian giao bài.
* **Khóa chính**: manager\_id.
* **Khóa ngoại**: Liên kết với bảng User và Problem.

**29. [Error\_Report]**

* **Mục đích**: Quản lý các báo cáo lỗi mà người dùng gửi lên hệ thống.
* **Các cột**:
  + report\_id: ID của báo cáo lỗi.
  + user\_id: ID người dùng gửi báo cáo lỗi.
  + error\_message: Nội dung báo lỗi.
  + created\_at: Thời gian báo cáo lỗi.
* **Khóa chính**: report\_id.
* **Khóa ngoại**: Liên kết tới bảng User.

**30. [Class]**

* **Mục đích**: Quản lý thông tin về các lớp học.
* **Các cột**:
  + class\_id: ID của lớp học.
  + class\_name: Tên lớp học.
  + teacher\_id: ID giáo viên phụ trách lớp.
  + created\_at: Thời gian tạo lớp học.
* **Khóa chính**: class\_id.
* **Khóa ngoại**: Liên kết tới bảng User để lấy thông tin giáo viên.

**31. [Rank]**

* **Mục đích**: Quản lý xếp hạng của người dùng trong các danh mục cụ thể.
* **Các cột**:
  + rank\_id: ID của bản ghi xếp hạng.
  + user\_id: ID người dùng.
  + category\_id: ID danh mục liên quan.
  + total\_score: Tổng điểm của người dùng trong danh mục.
  + detailed\_score: Điểm chi tiết (nếu có).
  + rank\_position: Vị trí xếp hạng.
  + updated\_at: Thời gian cập nhật thông tin.
* **Khóa chính**: rank\_id.
* **Ràng buộc**: Kiểm tra rank\_position và total\_score không âm.
* **Khóa ngoại**: Liên kết với bảng User và Category.

**32.[Learning\_Progress]**

* **Mục đích**: Quản lý tiến độ học tập của người dùng trong các khóa học cụ thể. Bảng này cho phép theo dõi trạng thái và tỷ lệ hoàn thành của người học trong từng khóa học hoặc bài tập.
* **Các cột**:
  + progress\_id: ID của bản ghi tiến độ.
  + user\_id: ID người học.
  + course\_id: ID của khóa học liên quan.
  + problem\_id: ID của bài tập (có thể là NULL nếu không có bài tập cụ thể nào liên quan).
  + status: Trạng thái tiến độ của người học (ví dụ: 'Not Started', 'In Progress', 'Completed').
  + progress\_percentage: Phần trăm tiến độ hoàn thành (giá trị từ 0 đến 100).
  + updated\_at: Thời gian cập nhật tiến độ.
* **Khóa chính**: progress\_id.
* **Ràng buộc**:
  + Kiểm tra progress\_percentage phải nằm trong khoảng từ 0 đến 100.
  + Kiểm tra status chỉ có thể là một trong các giá trị đã định nghĩa (ví dụ: 'Not Started', 'In Progress', 'Completed').
* **Khóa ngoại**: Liên kết với bảng User, Course, và Problem.

**1. Tạo chỉ mục (Indexes)**

* **Tạo chỉ mục không phân cụm cho bảng Comment**:

CREATE NONCLUSTERED INDEX [idx\_comment\_user\_problem]

ON [dbo].[Comment]([user\_id] ASC, [problem\_id] ASC);

**Tác dụng**: Tăng tốc độ truy vấn khi tìm kiếm các bình luận theo user\_id và problem\_id, cải thiện hiệu suất truy vấn cho các câu lệnh liên quan đến người dùng và bài tập.

* **Tạo chỉ mục không phân cụm cho bảng Submission**:

CREATE NONCLUSTERED INDEX [idx\_problem\_id]

ON [dbo].[Submission]([problem\_id] ASC);

**Tác dụng**: Tăng tốc độ truy vấn khi tìm kiếm các bài nộp theo problem\_id.

* **Tạo chỉ mục không phân cụm cho bảng Class\_Student**:

CREATE NONCLUSTERED INDEX [idx\_class\_student]

ON [dbo].[Class\_Student]([class\_id] ASC, [student\_id] ASC);

**Tác dụng**: Tăng tốc độ truy vấn để lấy thông tin sinh viên trong một lớp học cụ thể.

* **Tạo chỉ mục không phân cụm cho bảng Match\_User**:

CREATE NONCLUSTERED INDEX [idx\_match\_user]

ON [dbo].[Match\_User]([match\_id] ASC, [user\_id] ASC);

**Tác dụng**: Tăng tốc độ truy vấn khi tìm kiếm thông tin người dùng liên quan đến một trận đấu.

* **Tạo chỉ mục không phân cụm cho bảng Submission** (theo user\_id và problem\_id):

CREATE NONCLUSTERED INDEX [idx\_user\_problem]

ON [dbo].[Submission]([user\_id] ASC, [problem\_id] ASC);

**Tác dụng**: Tăng tốc độ truy vấn khi tìm kiếm bài nộp theo người dùng và bài tập cụ thể.

* **Tạo chỉ mục không phân cụm cho bảng Ratings**:

CREATE NONCLUSTERED INDEX [idx\_user\_course]

ON [dbo].[Ratings]([user\_id] ASC, [course\_id] ASC);

**Tác dụng**: Tăng tốc độ truy vấn khi lấy thông tin xếp hạng của người dùng trong các khóa học.

* **Tạo chỉ mục không phân cụm cho bảng Error\_Report**:

CREATE NONCLUSTERED INDEX [idx\_user\_id]

ON [dbo].[Error\_Report]([user\_id] ASC);

**Tác dụng**: Tăng tốc độ truy vấn để tìm kiếm các báo cáo lỗi của người dùng.

* **Tạo chỉ mục không phân cụm cho bảng Problem\_TestCase**:

CREATE NONCLUSTERED INDEX [idx\_problem\_id]

ON [dbo].[Problem\_TestCase]([problem\_id] ASC);

**Tác dụng**: Tăng tốc độ truy vấn khi tìm kiếm các test case liên quan đến bài tập cụ thể.

* **Tạo chỉ mục không phân cụm cho bảng Submission** (theo user\_id):

CREATE NONCLUSTERED INDEX [idx\_user\_id]

ON [dbo].[Submission]([user\_id] ASC);

**Tác dụng**: Tăng tốc độ truy vấn khi tìm kiếm bài nộp theo người dùng.

* **Tạo chỉ mục không phân cụm cho bảng Class**:

CREATE NONCLUSTERED INDEX [idx\_class\_teacher]

ON [dbo].[Class]([teacher\_id] ASC);

**Tác dụng**: Tăng tốc độ truy vấn để lấy thông tin về lớp học theo giáo viên.

* **Tạo chỉ mục Columnstore cho bảng Submission**:

CREATE COLUMNSTORE INDEX [idx\_submission\_columnstore]

ON [dbo].[Submission]([score], [runtime], [memory\_used]);

**Tác dụng**: Tối ưu hóa hiệu suất truy vấn cho các truy vấn phân tích dữ liệu, đặc biệt là khi xử lý các số liệu lớn liên quan đến điểm số, thời gian chạy và bộ nhớ đã sử dụng.

**2. Tạo thủ tục lưu trữ (Stored Procedures)**

* **Thủ tục lưu trữ GetUserSubmissions**:

CREATE PROCEDURE [dbo].[GetUserSubmissions]

@user\_id INT

AS

BEGIN

SELECT submission\_id, problem\_id, status, score, submitted\_at

FROM [dbo].[Submission]

WHERE user\_id = @user\_id;

END;

**Tác dụng**: Lấy danh sách các bài nộp của một người dùng cụ thể, giúp quản lý và theo dõi tiến trình học tập của người dùng.

* **Thủ tục lưu trữ UpdateUserRole**:

CREATE PROCEDURE [dbo].[UpdateUserRole]

@admin\_id INT,

@user\_id INT,

@new\_role VARCHAR(50)

AS

BEGIN

-- Kiểm tra xem admin có quyền không

...

END;

**Tác dụng**: Cung cấp chức năng để cập nhật vai trò của người dùng, với kiểm tra quyền hạn để đảm bảo chỉ những người quản trị mới có thể thay đổi vai trò của người dùng.

**3. Tạo trigger (Triggers)**

* **Trigger trg\_user\_deletion**:

CREATE TRIGGER trg\_user\_deletion

ON [dbo].[User]

FOR DELETE

AS

BEGIN

INSERT INTO [dbo].[Audit\_Log] (table\_name, action, user\_id, changes)

SELECT 'User', 'DELETE', deleted.user\_id, CONCAT('Deleted user ', deleted.username)

FROM deleted;

END;

**Tác dụng**: Ghi lại thông tin về các hành động xóa người dùng vào bảng Audit\_Log, giúp theo dõi lịch sử thay đổi và đảm bảo tính toàn vẹn dữ liệu.

Bảng:

* **User**: Lưu trữ thông tin của người dùng như tên đăng nhập, tên đầy đủ, địa chỉ, trường học, email, số điện thoại, mật khẩu và vai trò (học sinh, giáo viên, quản trị viên).
* **Course**: Quản lý các khóa học, bao gồm tên khóa học, mô tả, người tạo và thông tin thời gian tạo/cập nhật.
* **Submission\_Status**: Lưu trữ các trạng thái cho bài nộp (ví dụ: đang chờ xử lý, đã chấp nhận, lỗi biên dịch, v.v.).
* **Admin\_Activity**: Ghi lại hoạt động của quản trị viên, bao gồm ID quản trị viên, hành động, loại mục tiêu và thời gian thực hiện.
* **PasswordReset**: Quản lý các yêu cầu đặt lại mật khẩu, bao gồm mã token và thời gian tạo.
* **Problem**: Lưu trữ thông tin về các bài toán, bao gồm tiêu đề, mô tả, định dạng đầu vào/đầu ra, hạn chế, mẫu và thời gian/bộ nhớ giới hạn.
* **Match**: Quản lý thông tin về các trận đấu, bao gồm ngày diễn ra và trạng thái.
* **Blog**: Lưu trữ các bài viết blog, bao gồm ID tác giả, tiêu đề, nội dung và thời gian tạo.
* **Comment**: Quản lý các bình luận cho bài toán, bao gồm ID người dùng, ID bài toán, nội dung và thời gian tạo.
* **Ratings**: Ghi lại các đánh giá cho khóa học từ người dùng, bao gồm đánh giá, bình luận và thời gian đánh giá.
* **Submission**: Lưu trữ thông tin về bài nộp của người dùng cho các bài toán, bao gồm trạng thái, thời gian thực thi, bộ nhớ sử dụng và mã nguồn.
* **Class\_Student**: Quản lý mối quan hệ giữa lớp học và học sinh, bao gồm thời gian tham gia.
* **Admin**: Lưu trữ thông tin quản trị viên, bao gồm ID người dùng và mức độ truy cập.
* **Submission\_TestCase\_Result**: Lưu trữ kết quả kiểm tra các bài nộp, bao gồm trạng thái, thời gian thực thi và thông điệp lỗi.
* **Category**: Quản lý các danh mục bài toán, bao gồm tiêu đề, thông tin mô tả và thời gian tạo.
* **Blog\_Comment**: Quản lý các bình luận cho bài viết blog, bao gồm ID bài viết, ID người dùng và nội dung bình luận.
* **Match\_User**: Quản lý điểm số và vị trí xếp hạng của người dùng trong trận đấu.
* **User\_Badge**: Ghi nhận các huy hiệu đã trao cho người dùng.
* **Problem\_Language**: Quản lý các ngôn ngữ lập trình có thể sử dụng cho một bài toán.
* **Badge**: Lưu trữ thông tin về các huy hiệu, bao gồm tiêu đề và mô tả.
* **QnA**: Lưu trữ câu hỏi và câu trả lời từ người dùng.
* **User\_Role**: Quản lý vai trò của người dùng trong hệ thống.
* **Match\_Rank**: Ghi lại xếp hạng của người dùng trong các trận đấu, bao gồm điểm số và vị trí.
* **Course\_Student**: Quản lý mối quan hệ giữa khóa học và học sinh, bao gồm vai trò trong khóa học.
* **Notifications**: Ghi nhận thông báo liên quan đến khóa học từ người gửi đến người nhận.
* **Audit\_Log**: Ghi lại các hành động truy cập hoặc thay đổi trong cơ sở dữ liệu.
* **Problem\_TestCase**: Lưu trữ các trường hợp kiểm tra cho một bài toán, bao gồm dữ liệu đầu vào và đầu ra mong đợi.
* **Assignment\_Manager**: Quản lý thông tin về việc giao bài tập từ quản trị viên cho học sinh.
* **Error\_Report**: Ghi nhận các báo cáo lỗi từ người dùng.
* **Class**: Quản lý các lớp học, bao gồm tên lớp và ID giáo viên.
* **Rank**: Lưu trữ xếp hạng của người dùng trong các danh mục bài toán, bao gồm điểm số tổng và vị trí.