**DỰ ÁN: XẾP HẠNG ĐẠI HỌC TOÀN CẦU (UNIVERSITY RANKINGS)**

| **Mục** | **Chi tiết** |
| --- | --- |
| **Hệ quản trị CSDL đề xuất** | MySQL 8.0 / PostgreSQL |
| **Ngày tạo** | 22/10/2025 |
| **Tác giả** | Lê Hồng Thái |

1. **Use Data Definition Language (DDL) Scripts :**

Tập lệnh SQL dưới đây định nghĩa cấu trúc lược đồ, bao gồm các Khóa Chính (PK), Khóa Ngoại (FK), và các ràng buộc duy nhất (UNIQUE).

-- 1. Bảng country: Lưu trữ danh sách quốc gia

CREATE TABLE country (

id INT PRIMARY KEY,

country\_name VARCHAR(100) NOT NULL UNIQUE

);

-- 2. Bảng ranking\_system: Lưu trữ tên các hệ thống xếp hạng

CREATE TABLE ranking\_system (

id INT PRIMARY KEY,

system\_name VARCHAR(100) NOT NULL UNIQUE

);

-- 3. Bảng university: Thông tin cơ bản về trường

CREATE TABLE university (

id INT PRIMARY KEY,

country\_id INT,

university\_name VARCHAR(255) NOT NULL,

-- FK: Liên kết với country (1 Quốc gia có N Trường)

FOREIGN KEY (country\_id) REFERENCES country(id)

);

-- 4. Bảng ranking\_criteria: Các tiêu chí con của hệ thống xếp hạng

CREATE TABLE ranking\_criteria (

id INT PRIMARY KEY,

ranking\_system\_id INT,

criteria\_name VARCHAR(255) NOT NULL,

-- FK: Liên kết với ranking\_system (1 Hệ thống có N Tiêu chí)

FOREIGN KEY (ranking\_system\_id) REFERENCES ranking\_system(id)

);

-- 5. Bảng university\_year: Thông số thống kê của trường theo năm

CREATE TABLE university\_year (

university\_id INT,

year INT,

num\_students INT,

student\_staff\_ratio DECIMAL(5, 2),

pct\_international\_students DECIMAL(5, 2),

pct\_female\_students DECIMAL(5, 2),

-- PK Kép: university\_id + year đảm bảo duy nhất theo năm

PRIMARY KEY (university\_id, year),

FOREIGN KEY (university\_id) REFERENCES university(id)

);

-- 6. Bảng university\_ranking\_year: Điểm số chi tiết theo tiêu chí và năm

CREATE TABLE university\_ranking\_year (

university\_id INT,

ranking\_criteria\_id INT,

year INT,

score DECIMAL(5, 2),

-- PK Ba cột: university\_id + ranking\_criteria\_id + year đảm bảo mỗi điểm số là duy nhất

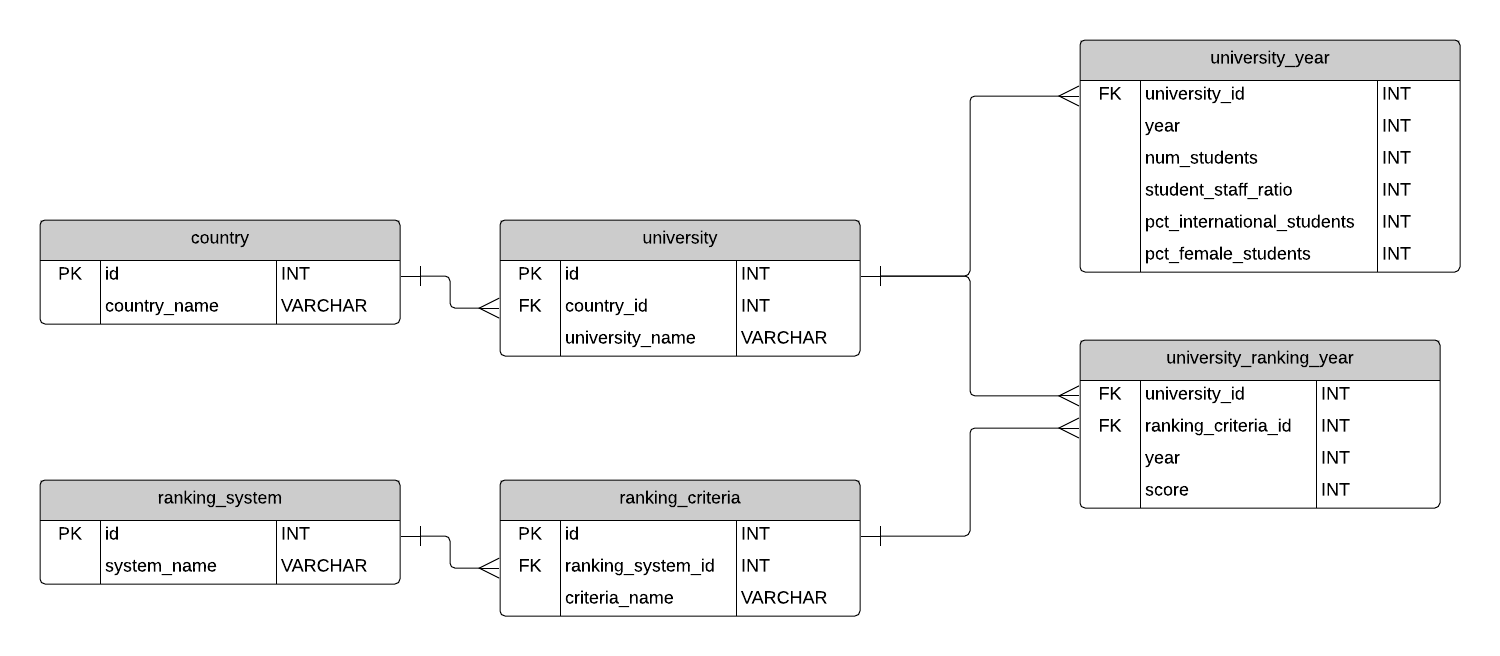
PRIMARY KEY (university\_id, ranking\_criteria\_id, year),

FOREIGN KEY (university\_id) REFERENCES university(id),

FOREIGN KEY (ranking\_criteria\_id) REFERENCES ranking\_criteria(id)

);

1. **Entity-Relationship Diagrams (ERDs) :**



1. **Table Descriptions :**

****

1. **Index Documentation :**

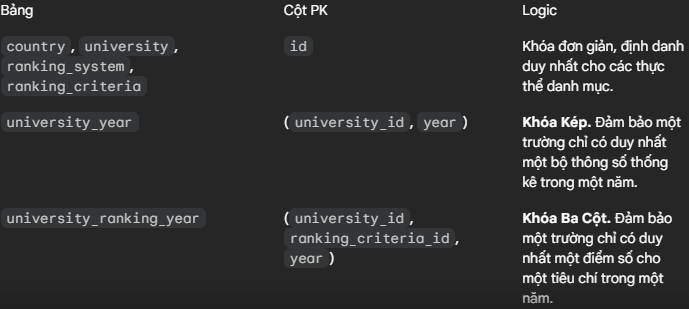
**Chỉ mục Bổ sung (ADDITIONAL INDEXES)**

**-- Chỉ mục này hỗ trợ các truy vấn tìm kiếm nhanh theo tên trường**

**CREATE INDEX idx\_university\_name\_search ON university (university\_name);**

1. **Constraints Documentation :**

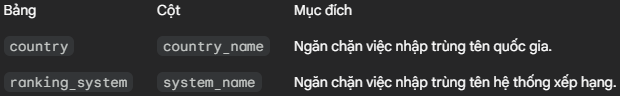
**Ràng buộc Khóa Chính (PRIMARY KEY)**

****

**Ràng buộc Khóa Ngoại (FOREIGN KEY)**

****

**Ràng buộc Duy nhất (UNIQUE)**

****

1. **Version Control**

**Tất cả các tài sản của dự án (DDL Scripts, SP Scripts, và Tài liệu DDD) sẽ được lưu trữ và quản lý phiên bản nghiêm ngặt bằng hệ thống Git, sử dụng nền tảng GitHub/GitLab. Mỗi thay đổi quan trọng đối với lược đồ sẽ được ghi lại bằng một *commit* chi tiết, cho phép theo dõi lịch sử và khôi phục về trạng thái trước đó.**

1. **Regular Updates**

**Tài liệu Thiết kế CSDL cam kết sẽ được xem xét và cập nhật định kỳ và tức thời sau khi có bất kỳ sự thay đổi nào được duyệt trên Lược đồ CSDL. Việc này bao gồm cập nhật mô tả bảng, ràng buộc và logic nghiệp vụ của các Stored Procedure, đảm bảo tài liệu luôn là nguồn thông tin chính xác và tin cậy nhất về cấu trúc CSDL.**

1. **Stored Procedures and Functions Documentation**

**Sored Procedure: GetUniversityScoreByCriteria**

| **Mục** | **Chi tiết** |
| --- | --- |
| **Mục đích** | **Truy xuất điểm số chi tiết (score) của một trường cụ thể, theo một tiêu chí xếp hạng và năm đã cho.** |
| **Tham số IN** | **p\_university\_name, p\_criteria\_name, p\_year.** |
| **Tham số OUT** | **Không có (Trả về tập hợp kết quả SELECT).** |
| **Mô tả Logic** | **Thủ tục sử dụng p\_university\_name, p\_criteria\_name, và p\_year làm điều kiện lọc. Thực hiện JOIN qua 4 bảng (university\_ranking\_year, university, ranking\_criteria, ranking\_system) để phân giải ID thành tên, cung cấp một kết quả trực quan và dễ đọc.** |

**Mã SQL Thủ tục (Ví dụ cho MySQL):**

**DELIMITER //**

**CREATE PROCEDURE GetUniversityScoreByCriteria (**

**IN p\_university\_name VARCHAR(255),**

**IN p\_criteria\_name VARCHAR(255),**

**IN p\_year INT**

**)**

**BEGIN**

**SELECT**

**u.university\_name,**

**rc.criteria\_name,**

**rs.system\_name,**

**ury.score**

**FROM**

**university\_ranking\_year ury**

**JOIN**

**university u ON u.id = ury.university\_id**

**JOIN**

**ranking\_criteria rc ON rc.id = ury.ranking\_criteria\_id**

**JOIN**

**ranking\_system rs ON rs.id = rc.ranking\_system\_id**

**WHERE**

**u.university\_name = p\_university\_name**

**AND rc.criteria\_name = p\_criteria\_name**

**AND ury.year = p\_year;**

**END //**

**DELIMITER ;**