ỦY BAN NHÂN DÂN TP.HCM

**TRƯỜNG CAO ĐẲNG CÔNG NGHỆ THỦ ĐỨC**

**BÁO CÁO ĐỒ ÁN CHUYÊN ĐỀ WEB 1 – CNTT**

**NĂM HỌC 2019 - 2020**

**Tên đề tài: Phân tích và tìm hiểu những vấn đề liên quan đến tạo cơ sở dữ liệu với MySQL .**

**Danh Sách Nhóm:**

**Giáo viên hướng dẫn ( GVHD ) :** Phan Thanh Nhuần

**Nhóm trưởng:** Lê Văn Huy (16211TT1794)

**Nhóm phó :** Trần Hưu Phước (16211TT1735)

**Thành viên 1 :** Phạm Thị Huỳnh Thi (16211TT1712)

**Thành viên 2 :** Nguyễn Văn Thái (16211TT1352)

**Thành viên 3 :** Phạm Minh Hiển (15211TT2203)

**TP. Hồ Chí Minh, năm 2019**

**Danh mục chữ viết tắt**

|  |  |
| --- | --- |
| **Chữ thu gọn** | **Chữ đầy đủ** |
| CNTT | Công nghệ thông tin |
| GV | Giảng viên |
| SV | Sinh viên |
| HTGD&HT | Hỗ trợ giảng dạy và học tập |
|  |  |
|  |  |
|  |  |

# 

# Phần mở đầu

## **1. Lý do chọn đề tài**

+ Nhóm em muốn tìm hiểu và nắm rõ về việc tạo và lựa chọn những thành phần cần thiết cho cơ sở dữ liệu , giúp nắm vững hơn về những tính năng , những thành phần riêng biệt , liên quan đến phân tích lựa chọn kiểu dữ liệu MySQL. Cũng như chuẩn bị cho môi trường làm việc khi ra trường , và các yêu cầu cơ bản của nhà tuyển dụng về kiến thức cơ sở dữ liệu .

+ Tính cấp thiết đáp ứng nhu cầu phát triển kinh tế - xã hội: Đối với chuyên ngành về máy tính khi đi ứng tuyển vào các vị trí lập trình viên, tester, QA, BA,... thì việc nắm vững được ngôn ngữ truy vấn cấu trúc SQL luôn là một kỹ năng quan trọng mà nhà tuyển dụng muốn ứng viên có được. MySQL là ngôn ngữ hỗ trợ việc lưu các thông tin phức tạp bằng cách đưa các thông tin này về một dàng cấu trúc thống nhất, rõ ràng và dễ hiểu. Tuy nhiên , có rất nhiều lựa chọn cho kiểu dữ liệu , và nhiều vấn đề quan trọng cần biết, để hiểu và làm việc tốt hơn với dữ liệu , cũng như thành thạo về dữ liệu , cần có hiểu biết rõ ràng và đúng đắn về kiểu dữ liệu trong MySQL. Đó cũng là lý do nhóm chọn chủ đề phân tích và thiết kế kiểu dữ liệu .

## **2. Mục đích nghiên cứu**

+ Nâng cao hiểu biết về các kiểu dữ liệu và vấn đề trong tổ chức kiểu dữ liệu MySQL.

+ Nâng cao hiểu biết về những tính năng MySQL như ( PRIMARY, UNIQUE, INDEX, FULLTEXT, SPATIAL).

+ Nhằm phục vụ cho mục đích công việc ra trường sau này .

## **3. Đối tượng và khách thể nghiên cứu**

+ Đối tượng là tất cả mọi người có kiến thức cơ bản về MySQL.

## **4. Giả thuyết nghiên cứu**

+ Index – chỉ mục có thể dùng được ký tự chữ hay không .

+ Unique so với Primary có điểm gì khác biệt .

## **5. Nhiệm vụ nghiên cứu**

+ Nghiên cứu về các dữ liệu , những vấn đề đáng chú ý trong tạo mới và chỉnh sửa cơ sở dữ liệu với MySQL.

+ Nhiệm vụ nghiên cứu và cách sử dụng về những tính năng PRIMARY, UNIQUE, INDEX, FULLTEXT, SPATIAL.

## **6. Phương pháp nghiên cứu**

+ Tìm kiếm thông tin trên một số trang web trên internet.

+ Điều tra trên các diễn đàn công nghệ , về những câu hỏi có lỗi tương tự .

+ Thực hành các quy trình kiểm tra lỗi trên phpMyAdmin.

+ Check các lỗi và ghép nối chúng với các câu lệnh khác.

## **7. Phạm vi ảnh hưởng**

+ Ảnh hưởng tới kiểu dữ liệu hệ thống, có thể lưu trữ tất cả các dữ liệu của một trang web , hoặc các ứng dụng có liên quan đến dữ liệu

+ Ảnh hưởng đến những thao tác có liên quan đến ngôn ngữ SQL .

# Phần nội dung

1. **Đánh giá các bài test**

**Test 1: Phạm Thị Huỳnh Thi**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **THANG ĐIỂM ĐÁNH GIÁ BÀI TEST 1** | | | | |
| **STT** | **Tiêu chí** | **Điểm** | **Ghi chú** | **SV đạt** |
| Câu 1 | - Thiết kế CSDL đủ các table & field cần thiết | 15 | Mỗi lỗi trừ 1đ | 15 |
| - Kiểu dữ liệu của các field đặt hợp lý | 5 | Mỗi lỗi trừ 0.5đ | 5 |
| - Tạo relation | 10 | Mỗi lỗi trừ 1đ | 5 |
| - Thêm dữ liệu và export đúng theo yêu cầu | 5 | Mỗi lỗi trừ 0.5đ | 5 |
| Câu 2 | a) Trang chủ | 10 | Đổ dữ liệu danh sách thành phố 6đ Kiểm tra dữ liệu trước khi submit 4đ Mỗi lỗi trừ 1đ | 10 |
| b) Đăng ký | 15 | Validate dữ liệu: 5đ Insert được dữ liệu: 5đ Thông báo kết quả: 5đ Mỗi lỗi trừ 1đ | 14 |
| c) Đăng nhập, Đăng xuất | 10 | Đăng nhập 8đ, Đăng xuất 2đ.  Mỗi lỗi trừ 1đ | 10 |
| d) Cập nhật thông tin | 15 | Lấy thông tin 5đ Validate thông tin nhập: 5đ Cập nhật thông tin: 5đ Mỗi lỗi trừ 1đ | 15 |
| e) Tính năng sai password 3 lần liên tiếp | 5 | Mỗi lỗi trừ 0.5đ | 5 |
| Câu 3 | - Bảo mật CSRF cho trang đăng ký | 10 | Tạo token 3đ, verify 2đ | 5 |
| **Tổng điểm:** | | **100** |  | **90** |

**Test 2: Phạm Thị Huỳnh Thi**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **THANG ĐIỂM ĐÁNH GIÁ BÀI TEST 2** | | | | |
| **STT** | **Tiêu chí** | **Điểm** | **Ghi chú** | **SV đạt** |
| 1 | - Lưu trữ: Quản lý danh sách các sân bay | 5 | Mỗi lỗi trừ 1đ | 5 |
| 2 | - Hiển thị: Hiển thị danh sách các sân bay theo tỉnh | 5 | Mỗi lỗi trừ 1đ | 5 |
| 3 | - Lưu trữ: Quản lý danh sách các hãng bay | 5 | Mỗi lỗi trừ 1đ | 5 |
| 4 | - Hiển thị: Hiển thị danh sách các hãng bay | 5 | Mỗi lỗi trừ 1đ | 5 |
| 5 | - Hiển thị: Tạo tuyến bay nội địa | 10 | Mỗi lỗi trừ 2đ Chú ý: Kiểm tra hãng bay nội địa khai thác tuyến nội địa | 0 |
| 6 | - Hiển thị: Tạo tuyến bay xuyên quốc gia | 10 | Mỗi lỗi trừ 2đ | 0 |
| 7 | - Lưu trữ: Tạo mối quan hệ các Quốc gia có kết nối với nhau để bay trực tiếp | 15 | Mỗi lỗi trừ 2đ | 0 |
| 8 | - Thiết kế: tính giá vé máy bay | 10 | Mỗi lỗi trừ 2đ. Chú ý: có thể lưu trữ trong db hoặc source code file về công thức tính | 0 |
| 9 | - Tạo chuyến bay phù hợp về thời gian bay | 5 | Mỗi lỗi trừ 2đ Chú ý trước/sau giờ bay 3 tháng | 0 |
| 10 | - Quản lý đặt vé: đặt thành công cho nhiều người | 5 | Tạo token 3đ, verify 2đ | 0 |
| 11 | - Quản lý đặt vé: hủy đặt vé | 5 |  | 0 |
| 12 | - Quản lý đặt vé: các vé đã đặt | 5 |  | 0 |
| 13 | - Quản lý đặt vé: sửa thông tin khách đã đặt (lúc chưa bay) | 5 |  | 0 |
| 14 | - Thống kê doanh thu theo hãng, có phân trang | 5 |  | 0 |
| 15 | - Thống kê sân bay có lưu lượng máy bay đến-đi nhiều nhất | 5 |  | 0 |
| **Tổng điểm:** | | 100 |  | 20 |

**Test 1: Lê Văn Huy**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **THANG ĐIỂM ĐÁNH GIÁ BÀI TEST 1** | | | | |
| **STT** | **Tiêu chí** | **Điểm** | **Ghi chú** | **SV đạt** |
| Câu 1 | - Thiết kế CSDL đủ các table & field cần thiết | 15 | Mỗi lỗi trừ 1đ | 15 |
| - Kiểu dữ liệu của các field đặt hợp lý | 5 | Mỗi lỗi trừ 0.5đ | 5 |
| - Tạo relation | 10 | Mỗi lỗi trừ 1đ | 10 |
| - Thêm dữ liệu và export đúng theo yêu cầu | 5 | Mỗi lỗi trừ 0.5đ | 5 |
| Câu 2 | a) Trang chủ | 10 | Đổ dữ liệu danh sách thành phố 6đ Kiểm tra dữ liệu trước khi submit 4đ Mỗi lỗi trừ 1đ | 10 |
| b) Đăng ký | 15 | Validate dữ liệu: 5đ Insert được dữ liệu: 5đ Thông báo kết quả: 5đ Mỗi lỗi trừ 1đ | 15 |
| c) Đăng nhập, Đăng xuất | 10 | Đăng nhập 8đ, Đăng xuất 2đ.  Mỗi lỗi trừ 1đ | 10 |
| d) Cập nhật thông tin | 15 | Lấy thông tin 5đ Validate thông tin nhập: 5đ Cập nhật thông tin: 5đ Mỗi lỗi trừ 1đ | 15 |
| e) Tính năng sai password 3 lần liên tiếp | 5 | Mỗi lỗi trừ 0.5đ | 5 |
| Câu 3 | - Bảo mật CSRF cho trang đăng ký | 10 | Tạo token 3đ, verify 2đ | 5 |
| **Tổng điểm:** | | **100** |  | **95** |

**Test 2: Lê Văn Huy**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **THANG ĐIỂM ĐÁNH GIÁ BÀI TEST 2** | | | | |
| **STT** | **Tiêu chí** | **Điểm** | **Ghi chú** | **SV đạt** |
| 1 | - Lưu trữ: Quản lý danh sách các sân bay | 5 | Mỗi lỗi trừ 1đ | 5 |
| 2 | - Hiển thị: Hiển thị danh sách các sân bay theo tỉnh | 5 | Mỗi lỗi trừ 1đ | 5 |
| 3 | - Lưu trữ: Quản lý danh sách các hãng bay | 5 | Mỗi lỗi trừ 1đ | 5 |
| 4 | - Hiển thị: Hiển thị danh sách các hãng bay | 5 | Mỗi lỗi trừ 1đ | 5 |
| 5 | - Hiển thị: Tạo tuyến bay nội địa | 10 | Mỗi lỗi trừ 2đ Chú ý: Kiểm tra hãng bay nội địa khai thác tuyến nội địa | 5 |
| 6 | - Hiển thị: Tạo tuyến bay xuyên quốc gia | 10 | Mỗi lỗi trừ 2đ | 5 |
| 7 | - Lưu trữ: Tạo mối quan hệ các Quốc gia có kết nối với nhau để bay trực tiếp | 15 | Mỗi lỗi trừ 2đ | 0 |
| 8 | - Thiết kế: tính giá vé máy bay | 10 | Mỗi lỗi trừ 2đ. Chú ý: có thể lưu trữ trong db hoặc source code file về công thức tính | 10 |
| 9 | - Tạo chuyến bay phù hợp về thời gian bay | 5 | Mỗi lỗi trừ 2đ Chú ý trước/sau giờ bay 3 tháng | 5 |
| 10 | - Quản lý đặt vé: đặt thành công cho nhiều người | 5 | Tạo token 3đ, verify 2đ | 0 |
| 11 | - Quản lý đặt vé: hủy đặt vé | 5 |  | 0 |
| 12 | - Quản lý đặt vé: các vé đã đặt | 5 |  | 5 |
| 13 | - Quản lý đặt vé: sửa thông tin khách đã đặt (lúc chưa bay) | 5 |  | 0 |
| 14 | - Thống kê doanh thu theo hãng, có phân trang | 5 |  | 0 |
| 15 | - Thống kê sân bay có lưu lượng máy bay đến-đi nhiều nhất | 5 |  | 0 |
| **Tổng điểm:** | | 100 |  | 50 |

**Test 1: Trần Hữu Phước**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **THANG ĐIỂM ĐÁNH GIÁ BÀI TEST 1** | | | | |
| **STT** | **Tiêu chí** | **Điểm** | **Ghi chú** | **SV đạt** |
| Câu 1 | - Thiết kế CSDL đủ các table & field cần thiết | 15 | Mỗi lỗi trừ 1đ | 15 |
| - Kiểu dữ liệu của các field đặt hợp lý | 5 | Mỗi lỗi trừ 0.5đ | 5 |
| - Tạo relation | 10 | Mỗi lỗi trừ 1đ | 10 |
| - Thêm dữ liệu và export đúng theo yêu cầu | 5 | Mỗi lỗi trừ 0.5đ | 5 |
| Câu 2 | a) Trang chủ | 10 | Đổ dữ liệu danh sách thành phố 6đ Kiểm tra dữ liệu trước khi submit 4đ Mỗi lỗi trừ 1đ | 10 |
| b) Đăng ký | 15 | Validate dữ liệu: 5đ Insert được dữ liệu: 5đ Thông báo kết quả: 5đ Mỗi lỗi trừ 1đ | 15 |
| c) Đăng nhập, Đăng xuất | 10 | Đăng nhập 8đ, Đăng xuất 2đ.  Mỗi lỗi trừ 1đ | 10 |
| d) Cập nhật thông tin | 15 | Lấy thông tin 5đ Validate thông tin nhập: 5đ Cập nhật thông tin: 5đ Mỗi lỗi trừ 1đ | 5 |
| e) Tính năng sai password 3 lần liên tiếp | 5 | Mỗi lỗi trừ 0.5đ | 0 |
| Câu 3 | - Bảo mật CSRF cho trang đăng ký | 10 | Tạo token 3đ, verify 2đ | 10 |
| **Tổng điểm:** | | **100** |  | **70** |

**Test 2: Trần Hữu Phước**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **THANG ĐIỂM ĐÁNH GIÁ BÀI TEST 2** | | | | |
| **STT** | **Tiêu chí** | **Điểm** | **Ghi chú** | **SV đạt** |
| 1 | - Lưu trữ: Quản lý danh sách các sân bay | 5 | Mỗi lỗi trừ 1đ | 5 |
| 2 | - Hiển thị: Hiển thị danh sách các sân bay theo tỉnh | 5 | Mỗi lỗi trừ 1đ | 5 |
| 3 | - Lưu trữ: Quản lý danh sách các hãng bay | 5 | Mỗi lỗi trừ 1đ | 5 |
| 4 | - Hiển thị: Hiển thị danh sách các hãng bay | 5 | Mỗi lỗi trừ 1đ | 5 |
| 5 | - Hiển thị: Tạo tuyến bay nội địa | 10 | Mỗi lỗi trừ 2đ Chú ý: Kiểm tra hãng bay nội địa khai thác tuyến nội địa | 5 |
| 6 | - Hiển thị: Tạo tuyến bay xuyên quốc gia | 10 | Mỗi lỗi trừ 2đ | 5 |
| 7 | - Lưu trữ: Tạo mối quan hệ các Quốc gia có kết nối với nhau để bay trực tiếp | 15 | Mỗi lỗi trừ 2đ | 0 |
| 8 | - Thiết kế: tính giá vé máy bay | 10 | Mỗi lỗi trừ 2đ. Chú ý: có thể lưu trữ trong db hoặc source code file về công thức tính | 0 |
| 9 | - Tạo chuyến bay phù hợp về thời gian bay | 5 | Mỗi lỗi trừ 2đ Chú ý trước/sau giờ bay 3 tháng | 0 |
| 10 | - Quản lý đặt vé: đặt thành công cho nhiều người | 5 | Tạo token 3đ, verify 2đ | 0 |
| 11 | - Quản lý đặt vé: hủy đặt vé | 5 |  | 5 |
| 12 | - Quản lý đặt vé: các vé đã đặt | 5 |  | 5 |
| 13 | - Quản lý đặt vé: sửa thông tin khách đã đặt (lúc chưa bay) | 5 |  | 0 |
| 14 | - Thống kê doanh thu theo hãng, có phân trang | 5 |  | 0 |
| 15 | - Thống kê sân bay có lưu lượng máy bay đến-đi nhiều nhất | 5 |  | 0 |
| **Tổng điểm:** | | 100 |  | 40 |

**Test 1: Phạm Minh Hiển**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **THANG ĐIỂM ĐÁNH GIÁ BÀI TEST 1** | | | | |
| **STT** | **Tiêu chí** | **Điểm** | **Ghi chú** | **SV đạt** |
| Câu 1 | - Thiết kế CSDL đủ các table & field cần thiết | 15 | Mỗi lỗi trừ 1đ | 15 |
| - Kiểu dữ liệu của các field đặt hợp lý | 5 | Mỗi lỗi trừ 0.5đ | 5 |
| - Tạo relation | 10 | Mỗi lỗi trừ 1đ | 10 |
| - Thêm dữ liệu và export đúng theo yêu cầu | 5 | Mỗi lỗi trừ 0.5đ | 5 |
| Câu 2 | a) Trang chủ | 10 | Đổ dữ liệu danh sách thành phố 6đ Kiểm tra dữ liệu trước khi submit 4đ Mỗi lỗi trừ 1đ | 10 |
| b) Đăng ký | 15 | Validate dữ liệu: 5đ Insert được dữ liệu: 5đ Thông báo kết quả: 5đ Mỗi lỗi trừ 1đ | 15 |
| c) Đăng nhập, Đăng xuất | 10 | Đăng nhập 8đ, Đăng xuất 2đ.  Mỗi lỗi trừ 1đ | 10 |
| d) Cập nhật thông tin | 15 | Lấy thông tin 5đ Validate thông tin nhập: 5đ Cập nhật thông tin: 5đ Mỗi lỗi trừ 1đ | 15 |
| e) Tính năng sai password 3 lần liên tiếp | 5 | Mỗi lỗi trừ 0.5đ | 5 |
| Câu 3 | - Bảo mật CSRF cho trang đăng ký | 10 | Tạo token 3đ, verify 2đ | 5 |
| **Tổng điểm:** | | **100** |  | **95** |

**Test 2: Phạm Minh Hiển**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **THANG ĐIỂM ĐÁNH GIÁ BÀI TEST 2** | | | | |
| **STT** | **Tiêu chí** | **Điểm** | **Ghi chú** | **SV đạt** |
| 1 | - Lưu trữ: Quản lý danh sách các sân bay | 5 | Mỗi lỗi trừ 1đ | 5 |
| 2 | - Hiển thị: Hiển thị danh sách các sân bay theo tỉnh | 5 | Mỗi lỗi trừ 1đ | 5 |
| 3 | - Lưu trữ: Quản lý danh sách các hãng bay | 5 | Mỗi lỗi trừ 1đ | 5 |
| 4 | - Hiển thị: Hiển thị danh sách các hãng bay | 5 | Mỗi lỗi trừ 1đ | 5 |
| 5 | - Hiển thị: Tạo tuyến bay nội địa | 10 | Mỗi lỗi trừ 2đ Chú ý: Kiểm tra hãng bay nội địa khai thác tuyến nội địa | 5 |
| 6 | - Hiển thị: Tạo tuyến bay xuyên quốc gia | 10 | Mỗi lỗi trừ 2đ | 5 |
| 7 | - Lưu trữ: Tạo mối quan hệ các Quốc gia có kết nối với nhau để bay trực tiếp | 15 | Mỗi lỗi trừ 2đ | 0 |
| 8 | - Thiết kế: tính giá vé máy bay | 10 | Mỗi lỗi trừ 2đ. Chú ý: có thể lưu trữ trong db hoặc source code file về công thức tính | 10 |
| 9 | - Tạo chuyến bay phù hợp về thời gian bay | 5 | Mỗi lỗi trừ 2đ Chú ý trước/sau giờ bay 3 tháng | 5 |
| 10 | - Quản lý đặt vé: đặt thành công cho nhiều người | 5 | Tạo token 3đ, verify 2đ | 0 |
| 11 | - Quản lý đặt vé: hủy đặt vé | 5 |  | 0 |
| 12 | - Quản lý đặt vé: các vé đã đặt | 5 |  | 5 |
| 13 | - Quản lý đặt vé: sửa thông tin khách đã đặt (lúc chưa bay) | 5 |  | 0 |
| 14 | - Thống kê doanh thu theo hãng, có phân trang | 5 |  | 0 |
| 15 | - Thống kê sân bay có lưu lượng máy bay đến-đi nhiều nhất | 5 |  | 0 |
| **Tổng điểm:** | | 100 |  | 50 |

**Test 1: Nguyễn Văn Thái**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **THANG ĐIỂM ĐÁNH GIÁ BÀI TEST 1** | | | | |
| **STT** | **Tiêu chí** | **Điểm** | **Ghi chú** | **SV đạt** |
| Câu 1 | - Thiết kế CSDL đủ các table & field cần thiết | 15 | Mỗi lỗi trừ 1đ | 15 |
| - Kiểu dữ liệu của các field đặt hợp lý | 5 | Mỗi lỗi trừ 0.5đ | 5 |
| - Tạo relation | 10 | Mỗi lỗi trừ 1đ | 10 |
| - Thêm dữ liệu và export đúng theo yêu cầu | 5 | Mỗi lỗi trừ 0.5đ | 5 |
| Câu 2 | a) Trang chủ | 10 | Đổ dữ liệu danh sách thành phố 6đ Kiểm tra dữ liệu trước khi submit 4đ Mỗi lỗi trừ 1đ | 10 |
| b) Đăng ký | 15 | Validate dữ liệu: 5đ Insert được dữ liệu: 5đ Thông báo kết quả: 5đ Mỗi lỗi trừ 1đ | 15 |
| c) Đăng nhập, Đăng xuất | 10 | Đăng nhập 8đ, Đăng xuất 2đ.  Mỗi lỗi trừ 1đ | 10 |
| d) Cập nhật thông tin | 15 | Lấy thông tin 5đ Validate thông tin nhập: 5đ Cập nhật thông tin: 5đ Mỗi lỗi trừ 1đ | 10 |
| e) Tính năng sai password 3 lần liên tiếp | 5 | Mỗi lỗi trừ 0.5đ | 0 |
| Câu 3 | - Bảo mật CSRF cho trang đăng ký | 10 | Tạo token 3đ, verify 2đ | 0 |
| **Tổng điểm:** | | **100** |  | **80** |

**Test 2: Nguyễn Văn Thái**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **THANG ĐIỂM ĐÁNH GIÁ BÀI TEST 2** | | | | |
| **STT** | **Tiêu chí** | **Điểm** | **Ghi chú** | **SV đạt** |
| 1 | - Lưu trữ: Quản lý danh sách các sân bay | 5 | Mỗi lỗi trừ 1đ | 5 |
| 2 | - Hiển thị: Hiển thị danh sách các sân bay theo tỉnh | 5 | Mỗi lỗi trừ 1đ | 5 |
| 3 | - Lưu trữ: Quản lý danh sách các hãng bay | 5 | Mỗi lỗi trừ 1đ | 5 |
| 4 | - Hiển thị: Hiển thị danh sách các hãng bay | 5 | Mỗi lỗi trừ 1đ | 5 |
| 5 | - Hiển thị: Tạo tuyến bay nội địa | 10 | Mỗi lỗi trừ 2đ Chú ý: Kiểm tra hãng bay nội địa khai thác tuyến nội địa | 5 |
| 6 | - Hiển thị: Tạo tuyến bay xuyên quốc gia | 10 | Mỗi lỗi trừ 2đ | 0 |
| 7 | - Lưu trữ: Tạo mối quan hệ các Quốc gia có kết nối với nhau để bay trực tiếp | 15 | Mỗi lỗi trừ 2đ | 0 |
| 8 | - Thiết kế: tính giá vé máy bay | 10 | Mỗi lỗi trừ 2đ. Chú ý: có thể lưu trữ trong db hoặc source code file về công thức tính | 10 |
| 9 | - Tạo chuyến bay phù hợp về thời gian bay | 5 | Mỗi lỗi trừ 2đ Chú ý trước/sau giờ bay 3 tháng | 0 |
| 10 | - Quản lý đặt vé: đặt thành công cho nhiều người | 5 | Tạo token 3đ, verify 2đ | 0 |
| 11 | - Quản lý đặt vé: hủy đặt vé | 5 |  | 0 |
| 12 | - Quản lý đặt vé: các vé đã đặt | 5 |  | 0 |
| 13 | - Quản lý đặt vé: sửa thông tin khách đã đặt (lúc chưa bay) | 5 |  | 0 |
| 14 | - Thống kê doanh thu theo hãng, có phân trang | 5 |  | 0 |
| 15 | - Thống kê sân bay có lưu lượng máy bay đến-đi nhiều nhất | 5 |  | 0 |
| **Tổng điểm:** | | 100 |  | 30 |

1. **Nghiên cứu**

### **Chương 1: Cơ sở lý luận**

#### Tổng quan về đề tài

Tổng quan về các nhóm dữ liệu :

* Kiểu dữ liệu số ( Number ) :
  + Số nguyên ( Integer ) : INT / TINYINT / SMALLINT / MEDIUMINT / BIGINT .
  + Số có dấu phẩy động ( Float ) : DECIMAL / FLOAT / DOUBLE .
  + Kiểu bitmap : BIT / BOOLEAN /REAL
* Kiểu thời gian ( Datetime ) : DATE / TIME / TIMESTAMP / DATETIME / YEAR .
* Kiểu chuỗi ký tự :
  + CHAR
  + VARCHAR
  + TEXT
  + BLOB

MySQL có những tính năng thiết lập khi tạo bảng như : PRIMARY , UNIQUE , INDEX , FULLTEXT , SPATIAL .

#### 2.Giới thiệu về hệ thống tính năng

***# PRIMARY***

1. **Khái niệm**

* PRIMARY KEY là một trường trong một bảng mà nhận diện một cách duy nhất mỗi hàng/bản ghi trong một bảng dữ liệu. Các PRIMARY KEY phải chứa các giá trị duy nhất.

1. **Tổng quan**

* PRIMARY KEY được dùng để xác định mỗi cột trong bảng là duy nhất. Từ đó nó giúp chúng ta phân biệt được các bảng.
* Giá trị của khóa chính là duy nhất cho mỗi record, nghĩa là không thể tồn tại hai record trùng khóa chính
* Khóa chính không thể cho phép NULL .
* Mỗi bảng chỉ tồn tại duy nhất một khóa chính, mỗi khóa chính có thể có nhiều column

***# UNIQUE***

1. **Khái niệm**

* UNIQUE là tính năng ràng buộc nếu bạn xác định một field nào đó là UNIQUE dữ liệu của mỗi record không được trùng lặp nhau .

1. **Tổng quan**

* UNIQUE được dùng khi muốn một column nào đó không được trùng dữ liệu với nhau .
* Giá trị UNIQUE có thể NULL .
* UNIQUE thực chất là một index KEY nên khi bạn thiết lập một field nào đó là UNIQUE thì việc truy vấn dữ liệu trên field đó sẽ nhanh hơn. Nó được dùng trong trường hợp bạn muốn tạo thêm một field có giá trị là duy nhất.

***# INDEX***

1. **Khái niệm**

* Chỉ mục (Index) là bảng tra cứu đặc biệt mà Database Search Engine có thể sử dụng để tăng nhanh thời gian và hiệu suất thu thập dữ liệu. Hiểu đơn giản, một chỉ mục là một con trỏ tới dữ liệu trong một bảng. Một chỉ mục trong một Database là tương tự như một chỉ mục trong Mục lục của cuốn sách.
* Không chỉ với Mysql mà với bất kì cơ sở dữ liệu nào có đánh index nó sẽ cải thiện tốc độ tìm kiếm , sắp xếp truy cập record của một bảng.Tuy nhiên, không phải lúc nào đánh index cũng tốt vì thế chúng ta cần xem xét xem khi nào thì cần đánh index.

1. **Ưu điểm khi dùng INDEX**

* Đẩy nhanh xử lý lọc ,tìm kiếm record theo điều kiện WHERE Không chỉ giới hạn với câu lệnh SELECT mà còn đẩy nhanh tốc độ tìm kiếm record trong các quá trình xử lý UPDATE , DELETE( theo WHERE)。

1. **Nhược điểm khi dùng INDEX**

* Cost của INSERT, UPDATE ,DELETE tăng lên . Hơn nữa, có thể không ảnh hưởng đến tốc độ hoặc tốc độ bị giảm đi bởi index
* UPDATE Khi thay đổi, update column có đánh index thì index cũng được update theo dẫn đến tốc độ bị châm lượng data update và số lượng index sẽ ảnh hưởng đến tốc độ Index mà không update data thì sẽ không bị thay đổi（do đó không ảnh hưởng đến giảm tốc độ bởi index )
* DELETE Vì phải tiến hành update toàn bộ index của table theo record delete nên dẫn đến tốc độ bị chậm thời gian cần để delete tỉ lệ thuận với số lượng index
* INSERT Khi insert record thì đồng thời việc insert index cũng tăng lên dẫn đến tốc độ cũng bị chậm đi

***# FULLTEXT***

1. **Sơ lược MySQL full text search**

* MySQL FTS hiện tại chỉ có trên storage engine MyISAM và mới có trên InnoDB (>=5.6 beta).
* Có 2 chế độ tìm kiếm đó là BOOLEAN MODE và NATURAL LANGUAGE MODE. Trong BOOLEAN MODE thì không có default sorting, và trong chế độ này thì ta có thể qui định từ khóa nào sẽ xuất hiện, và từ khóa nào không xuất hiện trong kết quả trả về. Còn NATURAL LANGUAGE MODE thì tìm kiếm những kết quả thích hợp (relavance) hơn là chính xác keyword được tìm.
* Mặc định thì MySQL có một list các stopwords, nghĩa là các từ mà MySQL sẽ bỏ qua không search nếu gặp phải nó (ví dụ: the, and, or, for v.v).
* Ngoài ra thì mặc định MySQL FTS chỉ tìm những từ có độ dài tối thiểu là 4 ký tự (global variable ft\_min\_word\_len = 4). Từ thực tế, nếu như ta search những chữ có độ dài bé hơn 4 (ví dụ "Hà Nội", "Cà Mau","Y tá","Thư ký" "The way I am") thì sẽ không có kết quả nào trả về, cho nên ta phải lưu ý tới thiết lập này của MySQL.

1. **Khái niệm FullText:**

* MySQL cung cấp tính năng tìm kiếm ‘toàn văn’ – full text. Xét về mặt kĩ thuật thì MySQL đánh chỉ mục các từ của cột được tiến hành tìm kiếm full-text. Như thế việc tìm kiếm trên mục này sẽ hiệu quả hơn. MySQL sử dụng những thuật toán phức tạp để xác định xem dòng nào phù hợp với yêu cầu tìm kiếm nhất.
* Full text search là kĩ thuật tìm kiếm toàn văn cho phép tìm kiếm các mẩu thông tin khớp với một chuỗi trên một hay một số cột nhất định.
* Một chỉ mục toàn văn trong MySQL là một chỉ mục có kiểu FULLTEXT. Các chỉ mục FULLTEXT chỉ được dùng với các bảng có thể được tạo ra từ các cột CHAR, VARCHAR, hay TEXT.
* Sử dụng cơ chế ranking(dựa trên mức độ phù hợp của các tài liệu tìm thấy, tài liệu trả về càng phù hợp thì có số rank càng cao).

***# SPATIAL***

1. **Khái niệm về SPATIAL trong MySql**

* Dữ liệu không gian (SPATIAL) biểu thị thông tin về vị trí và hình dạng vật lý của các đối tượng hình học. Những đối tượng này có thể là vị trí điểm hoặc các đối tượng phức tạp hơn như quốc gia, đường hoặc hồ.
* SQL Server hỗ trợ hai loại dữ liệu không gian: loại dữ liệu hình học và loại dữ liệu địa lý .
  + Kiểu hình học biểu thị dữ liệu trong hệ tọa độ Euclide (phẳng).
  + Kiểu địa lý biểu thị dữ liệu trong hệ tọa độ trái đất tròn.
* Cả hai loại dữ liệu được triển khai dưới dạng kiểu dữ liệu thời gian chạy ngôn ngữ chung (CLR) .NET trong SQL Server.

1. **Tổng quan**

* Trong SQL Server, các chỉ mục không gian được xây dựng bằng cây B, có nghĩa là các chỉ mục phải thể hiện dữ liệu không gian 2 chiều theo thứ tự tuyến tính của cây B. Do đó, trước khi đọc dữ liệu vào một chỉ mục không gian, SQL Server thực hiện phân rã thống nhất phân cấp không gian. Quá trình tạo chỉ mục phân tách không gian thành một hệ thống phân cấp lưới bốn cấp . Các cấp độ này được gọi là cấp 1 (cấp cao nhất), cấp 2 , cấp 3 và cấp 4 .
* Mỗi cấp độ liên tiếp phân rã cấp độ bên trên nó, vì vậy mỗi ô cấp trên chứa một lưới hoàn chỉnh ở cấp độ tiếp theo. Ở một mức độ nhất định, tất cả các lưới có cùng số lượng ô dọc theo cả hai trục (ví dụ: 4 x 4 hoặc 8 x 8) và các ô đều có một kích thước.

### **Chương 2: Thực trạng và giải pháp của vấn đề nghiên cứu**

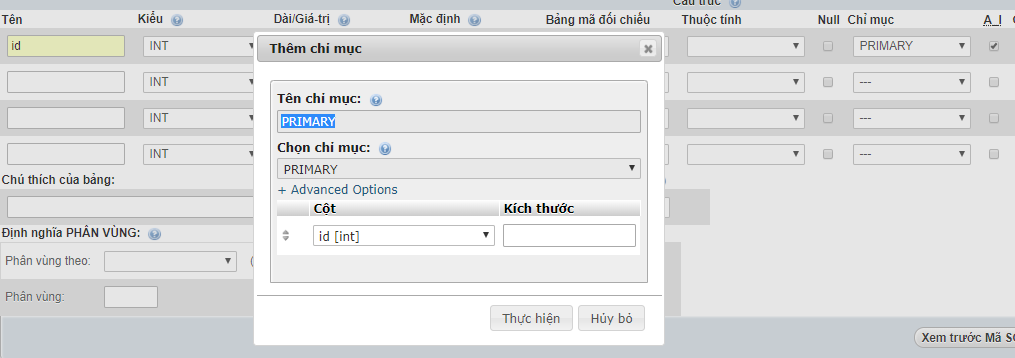
#### Nội dung:

**# PRIMARY KEY :**

**Đặc điểm của khóa chính (Primary key)**

* Giả sử chúng ta có bảng **SINHVIEN** và mỗi sinh viên là một record, lúc này để nhận phân biệt giữa các sinh viên thì ta không thể dựa vào tên của sinh viên được vì tên sinh viên có thể trùng nhau. Chính vì vậy ta chọn giải pháp là ta thêm một field **MaSV** dùng để làm khóa chính và giá trị của nó là duy nhất nên có thể phân biệt giữa các sinh viên với nhau.
* Giá trị của khóa chính là duy nhất cho mỗi record, nghĩa là không thể tồn tại hai record trùng khóa chính
* Khóa chính không thể cho phép NULL .
* Mỗi bảng chỉ tồn tại duy nhất một khóa chính, mỗi khóa chính có thể có nhiều column.
* Bởi vì trong MySQL hoạt động với kiểu số sẽ nhanh hơn kiểu chuỗi nên ta sẽ chọn kiểu INT (INT, BIGINT, ..), sau khi chạy code tạo khóa chính thì nó sẽ tạo một chỉ mục có tên là PRIMARY và đây chính là chỉ mục khóa chính.

1. **Tạo khóa chính trong mysql sever**



**Các kiểu dữ liệu thường dùng làm khóa chính**

* Bởi vì trong MySQL hoạt động với kiểu số sẽ nhanh hơn kiểu chuỗi nên ta sẽ chọn kiểu số như: INT, TINYINT , SMALLINT , MEDIUMINT , BIGINT, FLOAT, DOUBLE, DECIMAL, và các kiểu dữ liệu chuổi như: CHAR, VARCHAR, BLOB , TEXT, TINYBLOB, TINYTEXT,  MEDIUMBLOB, MEDIUMTEXT, LONGTEXT, ENUM.

**# UNIQUE**

1. **UNIQUE có thể làm đặt làm chỉ mục tìm kiếm :**

* UNIQUE có thể tạo bằng từ khóa UNIQUE trong khi tạo .
* Vì giá trị của UNIQUE là không trùng nhau nên có thể đặt làm chỉ mục tìm kiếm nhanh hơn so với những chỉ mục có giá trị trùng nhau . Và UNQUE có thể đặt được ở nhiều cột khác nhau .

1. **Vấn đề duplicate trong MySQL với NULL :**

* Duplicate value là hiện tượng bị lỗi giá trị trùng nhau .
* Null cũng chỉ là 1 giá trị bình thường , như cả giá trị “ ” ( rỗng ) . Nhưng sẽ bị lỗi duplicate value vì có giá trị trùng nhau .
* Vậy nên không nhập giá trị trùng nhau kể cả NULL hay “” .

1. **So sánh PRIMARY KEY và UNIQUE**

* PRIMARY KEY là khóa chính của bảng, đây là index đặc biệt nên bạn không chấp nhận giá trị NULL .
* UNIQUE cũng là một khóa nhưng vì là một column bình thường nên có thể nhận giá trị NULL.
* Chỉ có duy nhất 1 PRIMARY KEY trong table , nhưng có thể có nhiều UNIQUE column .

**# FULLTEXT**

1. **Đặc tính FullText:**
2. **Những đặc tính quan trọng của tìm kiếm full-text trong MySQL:**

* Giao diện thuần túy SQL: Ta sử dụng lệnh LIKE để tìm kiếm full-text.
* Chỉ mục hoàn toàn mang tính động: MySQL tự động lập chỉ mục cột văn bản bất cứ khi nào có diễn ra thay đổi dữ liệu trên cột này.
* Điều tiết dung lượng chỉ mục: không chiếm nhiều dung lượng cho việc lưu trữ chỉ mục.
* Tìm kiếm nhanh chóng ngay cả đối với những câu truy vấn phức tạp.
* MySQL chỉ mới hỗ trợ chức năng tìm kiếm full-text với các bảng dữ liệu kiểu MyISAM.

1. **Giả sử bạn không thể nhớ hết được cả câu cần tìm, mà chỉ nhớ được một số từ có trong câu đó thôi, thì chúng ta sẽ làm thế nào để tìm được các bản ghi đó? Nếu sử dụng câu lệnh như lúc đầu thì nó có chạy được hay không?**

* Câu trả lời của chúng ta là Không. Và còn rất nhiều mặt hạn chế của Search bình thường như:
* Khi không đánh index thì Tốc độ tìm kiếm chậm.
* Không tìm được từ đồng nghĩa.
* Không tìm được các từ viết tắt phổ biến.
* Không tìm kiếm cách từ như ví dụ trên.
* Không search được do lỗi chính tả...
* Gặp vấn đề trong tìm kiếm tiếng việt có dấu và không dấu. Vậy giải pháp của vấn đề trên là MySQL đã hỗ trợ thêm MySQL Fulltext Search.

Hình ảnh demo:



1. **Lệnh tạo Full Text:**

ALTER TABLE table\_name ADD FULLTEXT (column\_list);

1. **Lệnh xóa (Drop) Full text:**

ALTER TABLE *table\_name* DROP INDEX *index\_name;*

DROP INDEX index\_name ON table\_name;

1. **Tại sao chúng ta phải dùng Full text search?**

* Bình thường, chúng ta sẽ sử dụng câu truy vấn dạng như sau để tìm kiếm dữ liệu:

VD: SELECT id,title,description FROM book WHERE title LIKE ‘%keyword%’

* Nhưng cách truy vấn này có một số hạn chế như sau:
* Ngay cả full text search của MySQL cũng không giải quyết triệt để các vấn đề này mà phải dùng các search engine ngoài như Solr, Sphinx v.v
  + Không chính xác
  + Tốc độ truy vấn chậm, ‘%keyword%’ không dùng index
  + Vấn đề với tìm kiếm tiếng Việt có dấu và không dấu
  1. **Không chính xác**
  + **Độ nhiễu cao**
    - Giả sử, bạn có câu truy vấn với mệnh đề LIKE như sau:  
      Title LIKE ‘%one%’  
      Thì nó sẽ có thể trả về những kể quả sau: **one**, z**one**, m**one**y, ph**one** nói chung là không chính xác vì dải kết quả trả về sẽ rộng và có thể chứa nhiều kết quả nhiễu không mong muốn.
  + **Từ đồng nghĩa (synonyms)**
    - Như chúng ta đã biết, ngôn ngữ nào cũng có những từ đồng nghĩa, ví dụ như trong tiếng Việt là xe hơi - ôtô, bao thư - phong bì v.v. Tiếng Anh thì đơn giản thì có color-colour, check-cheque, deloper-programmer v.v Nếu như dùng LIKE hay = (thậm chí Full-text search của MySQL) thì tất nhiên không thể giải quyết được vấn từ đồng nghĩa này.
  + **Từ cấu tạo bằng chữ đầu của cụm từ (acronym)**
    - Đôi lúc với những cụm từ dài và phổ biến chúng ta thường viết tắt ví dụ như THPT, CNTT, US, IT. Nhưng khi người dùng tìm kiếm thì họ có thể nhập khác với trong database chúng ta lưu trữ (viết thu gọn - viết đủ và ngược lại) cho nên đây cũng là một khó khăn mà chúng ta gặp phải khi làm chức năng search. Mong muốn của người dùng là họ tìm thấy được kết quả mong muốn cho dù họ viết tắt hay viết đầy đủ).
  1. **Tốc độ truy vấn chậm, ‘%keyword%’ không dùng index**
  + Nếu như ta đặt wildcard ‘%’ ở phía trước thì MySQL sẽ thực hiện câu truy vấn mà không dùng index, MySQL sẽ thực hiện scan toàn bộ dữ liệu của nó từ đầu đến cuối, cho nên câu truy vấn sẽ rất chậm so với search trên index. Giống như ta tìm từng trang trong một cuốn sách thay vì tìm trong trang index đằng sau quyển sách đó vậy.
  1. **Vấn đề với tìm kiếm tiếng Việt có dấu và không dấu**
  + Giả sử ta lưu tiếng Việt có dấu trong database, nhưng người dùng nhập tiếng Việt không dấu thì mệnh đề LIKE chắc chắn sẽ không tìm ra được dữ liệu ta cần. Có một số giải pháp ví dụ như lưu 2 field, một có dấu và một không dấu, nhưng cách này xem ra không tối ưu và không hỗ trợ search gần đúng. Nếu như người dùng nhập “co be mua dong” thì dùng mệnh đề LIKE sẽ không search ra được “Cô bé mùa Đông”, nhưng FTS có thể giải quyết vấn đề này.

**# INDEX**

1. **Đặc điểm của cột nên tạo Index**

* Trong trường hợp search record với số lượng ít , lượng data trong table nhiều
* Hay được sử dụng như điều kiện của mệnh đề ORDER BY , diều kiện kết quả , điều kiện của WHERE Tuy nhiên, sẽ không cần đánh index nếu tất cả các scan là table target
* Vì data có nhiều dữ liệu NULL nên sẽ search ngoài giá trị NULL Do Index không gồm giá trị NULL nên Index sẽ hiệu quả khi search ngoài giá trị NULL

1. **Đặc điểm của cột không nên tạo Index**

* Trong trường hợp search lượng lớn record từ table hay kích thước của table bé
* Không có logic điều kiện where
* Giá trị của cột hay phải insert , update, delete Nếu insert dòng mới vào table , thì giá trị index cũng tự động đưa vào . Do đó tốc độ search data cũng nhanh hơn nhưng khi phát sinh maintain index thông qua xử lý insert , update, delete thì tốc độ sẽ giảm xuống
* Sử dung như điều kiện của mệnh đề WHERE nhưng cột được tham chiếu như một phần của biếu thức

1. **Cú pháp tạo Index**

* **Tạo index cho cột được chỉ rõ trong bảng:**
* Bạn có thể tạo một Unique Index trên một bảng. Một Unique Index nghĩa là hai hàng không thể có cùng giá trị chỉ mục. Dưới đây là cú pháp để tạo một chỉ mục trên một bảng.

CREATE UNIQUE INDEX index\_name  
ON table\_name (column1, column2, ...);

* Bạn có thể sử dụng một hoặc nhiều cột để tạo một chỉ mục. Ví dụ, chúng ta có thể tạo chỉ mục trên users sử dụng email.

CREATE UNIQUE INDEX  user\_email

ON users (email);

* Tạo index trên một bảng:

CREATE UNIQUE INDEX  index\_name

ON table\_name;

Lệnh ALTER để thêm và xóa INDEX trong MySQL

* **Lệnh thêm**
  + Có 4 kiểu lệnh để thêm các chỉ mục cho một bảng:

ALTER TABLE tbl\_name ADD PRIMARY KEY (column\_list)

* + Lệnh này thêm một PRIMARY KEY, nghĩa là các giá trị được lập chỉ mục phải là duy nhất và không thể là NULL.

ALTER TABLE tbl\_name ADD UNIQUE index\_name (column\_list)

* + Lệnh này tạo một chỉ mục cho các giá trị để giá trị đó phải là duy nhất (với giá trị NULL là ngoại lệ, chúng có thể xuất hiện nhiều lần).

ALTER TABLE tbl\_name ADD INDEX index\_name (column\_list)

* + Lệnh này thêm một chỉ mục thông thường, trong đó bất kỳ giá trị nào có thể xuất hiện nhiều hơn một lần.

ALTER TABLE tbl\_name ADD FULLTEXT index\_name (column\_list)

* + Lệnh này tạo một chỉ mục FULLTEXT đặc biệt, được sử dụng cho mục đích tìm kiếm văn bản.
* **Lệnh xóa**
  + - * Bạn có thể xóa bất kỳ INDEX nào bởi sử dụng mệnh đề DROP cùng với lệnh ALTER. Bạn xét ví dụ sau để xóa chỉ mục đã được tạo ở trên.

**DROP INDEX** **index\_name;**

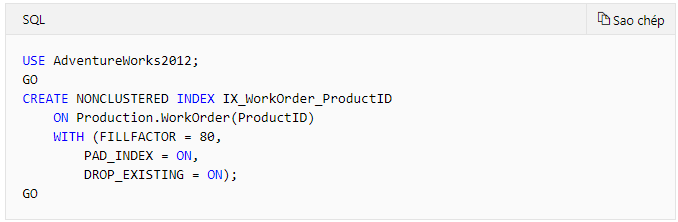
1. **Khi nào cần sử dụng Index**

* **Khi nên sử dụng**
* Khi so sánh giá trị của field với số chỉ định VD: WHERE name = "hogehoge"
* Khi JOIN toàn bộ giá trị của filed VD:WHERE [a.name](http://a.name/) = [b.name](http://b.name/)
* Khi cần giới hạn phạm vi giá trị của field bằng các toán tử so sánh[=、>、>=、<、<=]
* Khi cố định giá trị đầu tiên của chuỗi theo LIKE
* MIN(), MAX()
* Có thể ORDER BY, GROUP BY dựa theo prefix của chuỗi
* Trường hợp nếu tất cả các field của của WHERE có một phần index (không tham chiếu toàn bộ DB)
* Khi áp dụng mệnh đề BETWEEN or IN
* **Khi nên sử dụng**
* Khi bắt đầu LIKE bằng wild card
* Khi Mysql được xác định rằng việc đọc toàn bộ DB sẽ nhanh hơn
* Thông thường index sẽ không sử dụng ORDER BY
* Khi field ORDER BY và WHERE khác nhau nhưng chỉ đánh index trên một filed
* Trong trường hợp data dưới 1000 record, mặc dù đánh index nhưng có thể không được sử dụng

**# SPATIAL**

1. **Để tạo một chỉ mục không gian**
   * + Trong Object Explorer, kết nối với một phiên bản của Công cụ cơ sở dữ liệu SQL Server và sau đó mở rộng phiên bản đó.
     + Mở rộng cơ sở dữ liệu , mở rộng cơ sở dữ liệu chứa bảng với chỉ mục được chỉ định và sau đó mở rộng Bảng .
     + Mở rộng bảng mà bạn muốn tạo chỉ mục.
     + Nhấp chuột phải vào Chỉ mục và chọn Chỉ mục mới .
     + Trong trường Tên chỉ mục , nhập tên cho chỉ mục.
     + Trong danh sách thả xuống Loại chỉ mục, chọn Không gian .
     + Để chỉ định cột không gian mà bạn muốn lập chỉ mục, bấm Thêm .
     + Trong hộp thoại Chọn Cột từ <tên bảng> , chọn một cột có dạng hình học hoặc địa lý bằng cách chọn hộp kiểm tương ứng. Bất kỳ cột không gian khác sau đó trở thành không thể di chuyển. Nếu bạn muốn chọn một cột không gian khác nhau, trước tiên bạn phải xóa cột hiện được chọn. Khi hoàn tất, nhấp OK .
     + Xác nhận lựa chọn cột của bạn trong lưới cột Chỉ mục .
     + Trong ngăn Chọn một trang của hộp thoại Thuộc tính Chỉ mục , bấm Không gian
     + Trên trang Spatial , chỉ định các giá trị bạn muốn sử dụng cho các thuộc tính không gian của chỉ mục.
     + Khi tạo chỉ mục trên cột loại hình học , bạn phải chỉ định tọa độ ( X-min , Y-min ) và ( X-max , Y-max ) của hộp giới hạn. Đối với một chỉ mục trên một vị trí địa lý cột loại, các lĩnh vực bounding-box trở thành read-only sau khi bạn xác định lưới Địa lý chương trình tessellation, vì lưới địa lý tessellation không sử dụng một hộp bounding.
     + Theo tùy chọn, bạn có thể chỉ định các giá trị không phá hủy cho trường Ô trên mỗi đối tượng và cho mật độ lưới ở bất kỳ mức nào của sơ đồ tessname. Số lượng ô mặc định trên mỗi đối tượng là 16 cho SQL Server 2008 hoặc 8 cho SQL Server 2012 (11.x) trở lên và mật độ lưới mặc định là Trung bình cho SQL Server 2008.
     + Bạn có thể chọn GEOMETRY\_AUTO\_GRID hoặc GEOGRAPHY\_AUTO\_GRID cho sơ đồ tessname trong SQL Server. Khi GEOMETRY\_AUTO\_GRID hoặc GEOGRAPHY\_AUTO\_GRID được chọn, thì các tùy chọn mật độ lưới cấp 1, cấp 2, cấp 3 và cấp 4 bị vô hiệu hóa.
     + Để biết thêm thông tin về các thuộc tính này, hãy xem Trợ giúp Chỉ số Thuộc tính F1 .
     + Nhấn OK .
2. **Để thay đổi một chỉ số không gian**

* Kết nối với Cơ sở dữ liệu.
* Từ thanh Tiêu chuẩn, nhấp vào Truy vấn mới .
* Sao chép và dán ví dụ sau vào cửa sổ truy vấn và nhấp vào Thực thi . Ví dụ này giảm và tạo lại một chỉ mục hiện có trên ProductIDcột của Production.WorkOrderbảng bằng cách sử dụng DROP\_EXISTINGtùy chọn. Các tùy chọn FILLFACTORvà PAD\_INDEXcũng được thiết lập.



1. **Để xóa một chỉ mục**

* Trong Object Explorer, mở rộng cơ sở dữ liệu chứa bảng mà bạn muốn xóa chỉ mục.
* Mở rộng thư mục Bảng .
* Mở rộng bảng chứa chỉ mục bạn muốn xóa.
* Mở rộng thư mục Indexes .
* Nhấp chuột phải vào chỉ mục bạn muốn xóa và chọn Xóa .
* Trong hộp thoại Xóa đối tượng , xác minh rằng chỉ mục chính xác nằm trong lưới Đối tượng cần xóa và bấm OK .

# Danh mục tài liệu tham khảo

<https://viblo.asia/p/mot-so-chu-y-tao-index-trong-mysql-djeZ1BYglWz>