# CHƯƠNG I : CƠ SỞ LÝ THUYẾT

## Cơ sở dữ liệu

### Khái niệm

### Mô hình cơ sở dữ liệu

#### Mô hình thực thể kết hợp

#### Mô hình quan hệ

## Hệ quản trị cơ sở dữ liệu SQL Server

# CHƯƠNG II : THIẾT KẾ CƠ SỞ DỮ LIỆU

## Phát biểu bài toán

Một trung tâm đào tạo cần xây dựng một chương trình thi thử trắc nghiệm các môn học trên máy tính để học viên có thể ôn tập. Chia làm hai phần, phần thứ nhất là phần quản lý do bên phía trung tâm quản lý và phần thứ hai cho phép học sinh được đăng ký và tiến hành thi trên hệ thống. Phần thứ nhất gồm các chức năng chính là quản lý thông tin lớp học, môn học, thông tin học sinh và các câu hỏi trắc nghiệm.Phần thứ hai, học sinh có thể đăng ký tài khoản, tiến hành thi trên hệ thống và có thể tra kết quả.

Trung tâm có nhiều môn học, thông tin về môn học gồm: mã môn học, tên môn học. Mỗi môn học gồm nhiều chương, mỗi chương bao gồm nhiều câu hỏi, một câu hỏi chỉ thuộc một chương, một chương chỉ thuộc một môn học. Thông tin về câu hỏi gồm: mã câu hỏi, nội dung câu hỏi, nội dung các câu a, b, c, d. Một đề thi chỉ dùng cho một lần thi của một môn học. Thông tin về đề thi gồm: mã đề thi, ngày thi, các câu hỏi thuộc môn học của đề thi này. Khi một học viên làm bài cho một đề thi, chương trình cần phải lưu lại bài làm của học viên này bao gồm chọn lựa (a, b, c, d) của các câu hỏi trong đề thi này. Thông tin về học viên gồm: mã học viên, họ tên, địa chỉ, số điện thoại, giới tính. Một học viên chỉ thuộc một lớp học. Thông tin về lớp học gồm: mã lớp, tên lớp. Một lớp học phải học nhiều môn học và một học viên chỉ phải thi các môn học thuộc lớp học của mình.

## Đặc tả chức năng

### Danh sách phân hệ chức năng

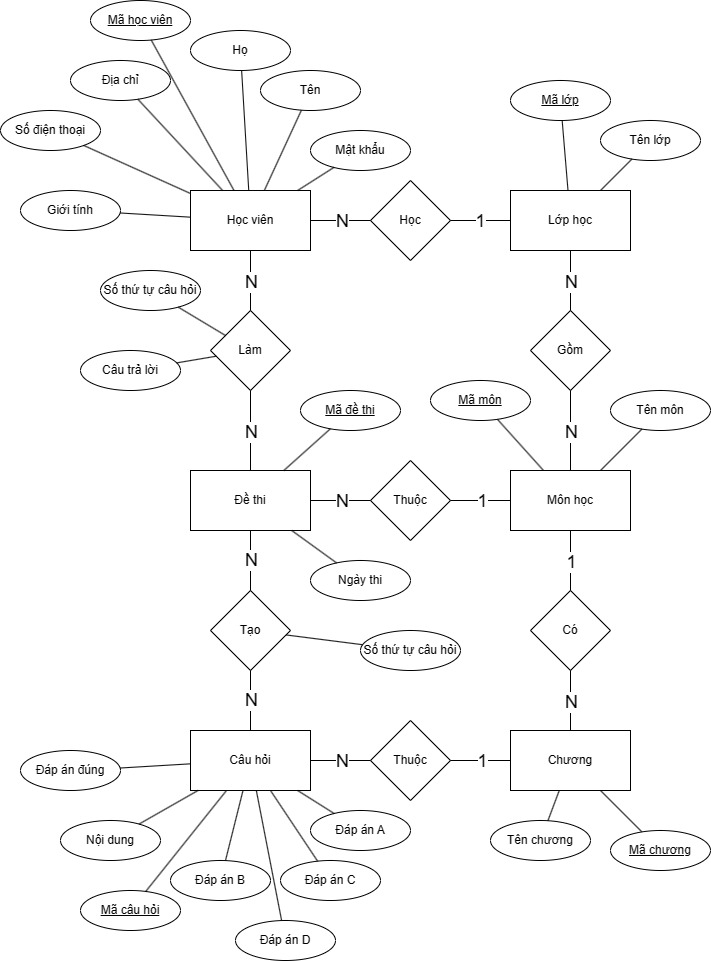
|  |  |  |  |
| --- | --- | --- | --- |
| **Mã nhóm chức năng** | **Nhóm chức năng** | **Mã nhóm chức năng chính** | **Chức năng chính** |
| M-QL | Quản lý | * M-QL-01 * M-QL-02 * M-QL-03 * M-QL-04 * M-QL-05 | * Quản lý thông tin lớp học * Quản lý thông tin môn học * Quản lý thông tin chương học * Quản lý thông tin câu hỏi trắc nghiệm * Quản lý thông tin học viên |
| M-HV | Học viên | * M-HV-01 * M-HV-02 | * Tiến hành thi trên hệ thống * Quản lý thông tin cá nhân học viên |

### Mô tả chi tiết chức năng

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã nhóm chức năng chính** | **Mã nhóm chức năng con** | **Tên nhóm chức năng con** | **Mô tả chi tiết** |
| M-QL-01 | M-QL-01-01 | Thêm thông tin lớp học | Kiểm tra null của dữ liệu đầu vào, kiểm tra sự tồn tại của dữ liệu trong cơ sở dữ liệu.  Nếu dữ liệu hợp lệ cho phép thêm mới dữ liệu. |
| M-QL-01 | M-QL-01-02 | Sửa thông tin lớp học | Kiểm tra null của dữ liệu đầu vào, kiểm tra sự tồn tại của dữ liệu, nếu không tồn tại hãy thông báo cho người dùng.  Nếu đã tồn tại và dữ liệu hợp lệ hãy cho phép cập nhật. |
| M-QL-01 | M-QL-01-03 | Xóa thông tin lớp học | Kiểm tra null của dữ liệu đầu vào, kiểm tra sự tồn tại của dữ liệu và trước khi xóa thông tin dữ liệu hãy lưu lại dữ liệu vào một bảng riêng để khôi phục dữ liệu. |
| M-QL-01 | M-QL-01-04 | Tra cứu thông tin lớp học theo tên hoặc mã lớp | Cho phép người dùng tìm kiếm thông tin lớp học. |
| M-QL-01 | M-QL-01-05 | Tra cứu thông tin các môn học của một lớp | Cho phép người dùng tìm kiếm thông tin môn học thuộc lớp học đó. |
| M-QL-01 | M-QL-01-06 | Thêm môn học vào danh sách môn của lớp | Kiểm tra null của dữ liệu đầu vào, kiểm tra sự tồn tại của dữ liệu trong cơ sở dữ liệu.  Nếu dữ liệu hợp lệ cho phép thêm mới dữ liệu. |
| M-QL-01 | M-QL-01-07 | Xóa môn học trong danh sách môn học của lớp | Kiểm tra null của dữ liệu đầu vào, kiểm tra sự tồn tại của dữ liệu trong cơ sở dữ liệu.  Nếu dữ liệu hợp lệ cho phép xóa dữ liệu. |
| M-QL-02 | M-QL-02-01 | Thêm thông tin môn học | Kiểm tra null của dữ liệu đầu vào, kiểm tra sự tồn tại của dữ liệu trong cơ sở dữ liệu.  Nếu dữ liệu hợp lệ cho phép thêm mới dữ liệu. |
| M-QL-02 | M-QL-02-02 | Sửa thông tin môn học | Kiểm tra null của dữ liệu đầu vào, kiểm tra sự tồn tại của dữ liệu, nếu không tồn tại hãy thông báo cho người dùng.  Nếu đã tồn tại và dữ liệu hợp lệ hãy cho phép cập nhật. |
| M-QL-02 | M-QL-02-03 | Xóa thông tin môn học | Kiểm tra null của dữ liệu đầu vào, kiểm tra sự tồn tại của dữ liệu và trước khi xóa thông tin dữ liệu hãy lưu lại dữ liệu vào một bảng riêng để khôi phục dữ liệu. |
| M-QL-02 | M-QL-02-04 | Tra cứu thông tin môn học theo tên hoặc mã môn học | Cho phép người dùng tìm kiếm thông tin môn học tên hoặc mã môn học. |
| M-QL-03 | M-QL-03-01 | Thêm thông tin chương học | Kiểm tra null của dữ liệu đầu vào, kiểm tra sự tồn tại của dữ liệu trong cơ sở dữ liệu.  Nếu dữ liệu hợp lệ cho phép thêm mới dữ liệu. |
| M-QL-03 | M-QL-03-02 | Sửa thông tin chương học | Kiểm tra null của dữ liệu đầu vào, kiểm tra sự tồn tại của dữ liệu, nếu không tồn tại hãy thông báo cho người dùng.  Nếu đã tồn tại và dữ liệu hợp lệ hãy cho phép cập nhật. |
| M-QL-03 | M-QL-03-03 | Xóa thông tin chương học | Kiểm tra null của dữ liệu đầu vào, kiểm tra sự tồn tại của dữ liệu và trước khi xóa thông tin dữ liệu hãy lưu lại dữ liệu vào một bảng riêng để khôi phục dữ liệu. |
| M-QL-03 | M-QL-03-04 | Tra cứu thông tin chương học theo tên hoặc mã chương học | Cho phép người dùng tìm kiếm thông tin chương học theo tên hoặc mã chương học. |
| M-QL-04 | M-QL-04-01 | Thêm thông tin câu hỏi | Kiểm tra null của dữ liệu đầu vào, kiểm tra sự tồn tại của dữ liệu trong cơ sở dữ liệu.  Nếu dữ liệu hợp lệ cho phép thêm mới dữ liệu. |
| M-QL-04 | M-QL-04-02 | Sửa thông tin câu hỏi | Kiểm tra null của dữ liệu đầu vào, kiểm tra sự tồn tại của dữ liệu, nếu không tồn tại hãy thông báo cho người dùng.  Nếu đã tồn tại và dữ liệu hợp lệ hãy cho phép cập nhật. |
| M-QL-04 | M-QL-04-03 | Xóa thông tin câu hỏi | Kiểm tra null của dữ liệu đầu vào, kiểm tra sự tồn tại của dữ liệu và trước khi xóa thông tin dữ liệu hãy lưu lại dữ liệu vào một bảng riêng để khôi phục dữ liệu. |
| M-QL-04 | M-QL-04-04 | Tra cứu thông tin câu hỏi theo tên hoặc mã câu hỏi | Cho phép người dùng tìm kiếm thông tin câu hỏi theo tên hoặc mã câu hỏi. |
| M-QL-05 | M-QL-05-01 | Thêm thông tin học viên | Kiểm tra null của dữ liệu đầu vào, kiểm tra sự tồn tại của dữ liệu trong cơ sở dữ liệu.  Nếu dữ liệu hợp lệ cho phép thêm mới dữ liệu và tạo tự động mật khẩu cho học viên. |
| M-QL-05 | M-QL-05-02 | Sửa thông tin học viên | Kiểm tra null của dữ liệu đầu vào, kiểm tra sự tồn tại của dữ liệu, nếu không tồn tại hãy thông báo cho người dùng.  Nếu đã tồn tại và dữ liệu hợp lệ hãy cho phép cập nhật. |
| M-QL-05 | M-QL-05-03 | Xóa thông tin học viên | Kiểm tra null của dữ liệu đầu vào, kiểm tra sự tồn tại của dữ liệu và trước khi xóa thông tin dữ liệu hãy lưu lại dữ liệu vào một bảng riêng để khôi phục dữ liệu. |
| M-QL-05 | M-QL-05-04 | Tra cứu thông tin học viên theo tên hoặc mã học viên | Cho phép người dùng tìm kiếm thông tin học viên theo tên hoặc mã học viên. |
| M-HV-01 | M-HV-01-01 | Tạo đề thi tự động | Kiểm tra null của dữ liệu đầu vào, kiểm tra sự tồn tại của dữ liệu trong cơ sở dữ liệu.  Kiểm tra số lượng câu hỏi trong thư viện câu hỏi gồm có ít nhất 40 câu hỏi.  Nếu hợp lệ tiến hành tạo đề thi tự động. Đề thi gồm có 40 câu hỏi được tiến hành lấy bất kì trong thư viện câu hỏi. |
| M-HV-01 | M-HV-01-02 | Lưu kết quả bài làm | Kiểm tra null của dữ liệu đầu vào, kiểm tra sự tồn tại của dữ liệu trong cơ sở dữ liệu.  Tiến hành lưu kết quả bài làm gồm mã học viên, mã câu hỏi và đáp án, nếu học viên không điền đáp án thì đáp án sẽ được lưu là null. |
| M-HV-02 | M-HV-02-01 | Tra cứu kết quả thi | Kiểm tra null của dữ liệu đầu vào, kiểm tra sự tồn tại của dữ liệu trong cơ sở dữ liệu.  Nếu hợp lệ hãy viết một hàm để tính điểm thi của học viên. Sau đó cho phép học viên tra cứu kết quả thi theo mã học viên. |
| M-HV-02 | M-HV-02-02 | Cập nhật mật khẩu | Kiểm tra null của dư liệu đầu vào, kiểm tra sự tồn tại của mã học viên, cho học viên nhập mật khẩu cũ và mới.  Nếu mật khẩu cũ trùng khớp với mật khẩu đang lưu cho phép học viên cập nhật mật khẩu theo mật khẩu mới. |

## Mô hình cơ sở dữ liệu

### Mô hình thực thể



### Mô tả các thuộc tính của tập thực thể

* Học viên ( Mã học viên, họ, tên, địa chỉ, số điện thoại, giới tính, mật khẩu) mã học viên là khóa chính, còn lại là thuộc tính đơn trị.
* Lớp học ( Mã lớp, tên lớp) mã lớp là khóa chính, tên lớp là thuộc tính đơn trị.
* Môn học ( Mã môn, tên môn) mã môn là khóa chính, tên môn là thuộc tính đơn trị.
* Chương ( Mã chương, tên chương) mã chương là khóa chính, tên chương là thuộc tính đơn trị.
* Câu hỏi ( Mã câu hỏi, nội dung, đáp án A, đáp án B, đáp án C, đáp án D, đáp án đúng) mã câu hỏi là khóa chính, còn lại là thuộc tính đơn trị.
* Đề thi ( Mã đề thi, ngày thi) mã đề thi là khóa chính, ngày thi là thuộc tính đơn trị.

### Mối quan hệ giữa các thực thể

* Lớp học có mối quan hệ một nhiều(1-n) với học viên.
* Lớp học có mối quan hệ nhiều nhiều(n-n) với môn học.
* Môn học có mối quan hệ một nhiều(1-n) với chương.
* Môn học có mối quan hệ một nhiều(1-n) với đề thi.
* Chương có mối quan hệ một nhiều(1-n) với câu hỏi.
* Đề thi có mối quan hệ nhiều nhiều(n-n) với câu hỏi sinh ra thuộc tính số thứ tự câu hỏi.
* Đề thi có mối quan hệ nhiều nhiều(n-n) với học viên sinh ra thuộc tính số thứ tự câu hỏi và câu trả lời.

### Chuẩn hóa cơ sở dữ liệu

#### Chuẩn hóa cơ sở dữ liệu đạt chuẩn 1NF

* Ta có thể biểu diễn lược đồ cơ sở dữ liệu dưới dạng Q={A,F} trong đó A là tập các thuộc tính và F là phụ thuộc hàm của cơ sở dữ liệu hệ thống thi thử trắc nghiệm.
* Ta có tập các thuộc tính A={Mã học viên, họ, tên, địa chỉ, số điện thoại, mật khẩu, giới tính, mã lớp học, tên lớp, mã môn, tên môn, mã chương, tên chương, mã câu hỏi, nội dung, đáp án A, đáp án B, đáp án C, đáp án D, đáp án đúng, số thứ tự câu hỏi, mã đề thi, ngày thi, số thứ tự câu hỏi, câu trả lời }.
* Ta có tập các phụ thuộc hàm F={ Mã học viên => họ, tên, địa chỉ, số điện thoại, mật khẩu, giới tính, mã lớp học; Mã lớp học => tên lớp; Mã môn => tên môn; Mã chương => tên chương, mã môn; Mã câu hỏi => nội dung, đáp án A, đáp án B, đáp án C, đáp án D, đáp án đúng, mã chương; Mã đề thi => ngày thi, mã môn học; Mã câu hỏi, mã đề thi => số thứ tự câu hỏi; Mã học viên, mã đề thi, số thứ tự câu hỏi => câu trả lời}.

🡺 Ta nhận thấy các thuộc tính trong tập thuộc tính A đều là thuộc tính đơn trị. Vì vậy cơ sở dữ liệu thi thử trắc nghiệm đã đạt chuẩn 1NF.

#### Chuẩn hóa cơ sở dữ liệu đạt chuẩn 2NF

* Dựa trên phụ thuộc hàm F ta có thể chia nhỏ phụ thuộc hàm F thành các phụ thuộc hàm như sau:
  + - F1={ Mã học viên => họ, tên, địa chỉ, số điện thoại, mật khẩu, giới tính, mã lớp học}
    - F2={ Mã lớp học => tên lớp}
    - F3={ Mã môn => tên môn}
    - F4={ Mã chương => tên chương, mã môn}
    - F5={ Mã câu hỏi => nội dung, đáp án A, đáp án B, đáp án C, đáp án D, đáp án đúng, mã chương}
    - F6={ Mã đề thi => ngày thi, mã môn học}
    - F7={ Mã bài làm}
    - F8={ Mã lớp học, mã môn}
    - F9={ Mã câu hỏi, mã đề thi => số thứ tự câu hỏi}
    - F10={ Mã học viên, mã đề thi, số thứ tự câu hỏi => câu trả lời }
* Các thuộc tính xuất hiện bên phải không xuất hiện bên trái trong phụ thuộc hàm F1 là họ, tên, địa chỉ, số điện thoại, mật khẩu, giới tính, mã lớp học và xuất hiện bên trái không xuất hiện bên phải là mã học viên.
* Mã học viên+ = {Mã học viên, họ, tên, địa chỉ, số điện thoại, mật khẩu, giới tính, mã lớp học }. Mã học viên là khóa chính của phụ thuộc hàm F1.
* Thuộc tính xuất hiện bên phải không xuất hiện bên trái trong phụ thuộc hàm F2 là tên lớp học và xuất hiện bên trái không xuất hiện bên phải là mã lớp học.
* Mã lớp học+ ={ Mã lớp học, tên lớp học}. Mã lớp học là khóa chính của phụ thuộc hàm F2.
* Thuộc tính xuất hiện bên phải không xuất hiện bên trái trong phụ thuộc hàm F3 là tên môn và xuất hiện bên trái không xuất hiện bên phải là mã môn.
* Mã môn+ ={ Mã môn, tên môn}. Mã môn là khóa chính của phụ thuộc hàm F3.
* Các thuộc tính xuất hiện bên phải không xuất hiện bên trái trong phụ thuộc hàm F4 là tên chương, mã môn và xuất hiện bên trái không xuất hiện bên phải là mã chương.
* Mã chương+ = { Mã chương, tên chương, mã môn}. Mã chương là khóa chính của phụ thuộc hàm F4.
* Các thuộc tính xuất hiện bên phải không xuất hiện bên trái trong phụ thuộc hàm F5 là nội dung, đáp án A, đáp án B, đáp án C, đáp án D, đáp án đúng, mã chương và xuất hiện bên trái không xuất hiện bên phải là mã câu hỏi.
* Mã câu hỏi+ ={ Mã câu hỏi, nội dung, đáp án A, đáp án B, đáp án C, đáp án D, đáp án đúng, mã chương}. Mã câu hỏi là khóa chính của phụ thuộc hàm F5.
* Các thuộc tính xuất hiện bên phải không xuất hiện bên trái trong phụ thuộc hàm F6 là ngày thi, mã môn học và xuất hiện bên trái không xuất hiện bên phải là mã đề thi.
* Mã đề thi+ ={ Mã đề thi, ngày thi, mã môn học}. Mã đề thi là khóa chính của phụ thuộc hàm F6.
* Phụ thuộc hàm F7 chỉ là một thuộc tính đơn trị là mã bài làm.
* Mã bài làm là khóa chính của phụ thuộc hàm F7.
* Trong phụ thuộc hàm F8 chỉ có các thuộc tính xuất hiện bên trái không xuất hiện bên phải là mã lớp học và mã môn.
* Mã lớp học và mã môn là khóa chính của phụ thuộc hàm F8.
* Thuộc tính xuất hiện bên phải không xuất hiện bên trái trong phụ thuộc hàm F9 là số thứ tự câu hỏi và xuất hiện bên trái không xuất hiện bên phải là mã câu hỏi và mã đề thi.
* Mã câu hỏi, mã đề thi+ ={ Mã câu hỏi, mã đề thi, số thứ tự câu hỏi}. Mã câu hỏi và mã đề thi là khóa chính của phụ thuộc hàm F9.
* Các thuộc tính xuất hiện bên phải không xuất hiện bên trái trong phụ thuộc hàm F10 là câu trả lời và xuất hiện bên trái không xuất hiện bên phải là mã học viên, mã đề thi, số thứ tự câu hỏi.
* Mã học viên, mã đề thi, số thứ tự câu hỏi + ={ Mã học viên, mã đề thi, số thứ tự câu hỏi, câu trả lời}. Mã học viên, mã đề thi, số thứ tự câu hỏi là khóa chính của phụ thuộc hàm F10.

🡺 Ta nhận thấy các thuộc tính trong phụ thuộc hàm F là phụ thuộc toàn phần và cơ sở dữ liệu thi thử trắc nghiệm đã đạt chuẩn 1NF. Vì vậy cơ sở dữ liệu thi thử trắc nghiệm đã đạt chuẩn 2NF.

#### Chuẩn hóa cơ sở dữ liệu đạt chuẩn 3NF

* Phân rã lược đồ quan hệ cơ sở dữ liệu thi thử trắc nghiệm Q dựa trên các phụ thuộc hàm F ta có thể phân rã Q như sau:
* Q1={ Mã học viên, họ, tên, địa chỉ, số điện thoại, mật khẩu, giới tính, mã lớp học}
  + - Q2={ Mã lớp học, tên lớp}
    - Q3={ Mã môn, tên môn}
    - Q4={ Mã chương, tên chương, mã môn}
    - Q5={ Mã câu hỏi, nội dung, đáp án A, đáp án B, đáp án C, đáp án D, đáp án đúng, mã chương}
    - Q6={ Mã đề thi, ngày thi, mã môn học}
    - Q7={ Mã bài làm}
    - Q8={ Mã lớp học, mã môn}
    - Q9={ Mã câu hỏi, mã đề thi, số thứ tự câu hỏi}
* Q10={ Mã học viên, mã đề thi, số thứ tự câu hỏi , câu trả lời}

🡺 Ta nhận thấy trong các phân rã của Q, không tồn tại các phân rã chứa thuộc tính bắc cầu và cơ sở dữ liệu thi thử trắc nghiệm đã đạt chuẩn 3NF. Vì vậy cơ sở dữ liệu thi thử trắc nghiệm đã đạt chuẩn 3NF.

### Lược đồ quan hệ

**Bảng học viên**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Mô tả** | **Null/Not null** |
| 01 | Mã học viên | Int | Là khóa chính, sử dụng constraint identity dùng để định danh học viên. | Not null |
| 02 | Họ | Nvarchar(20) | Là thuộc tính đơn trị, dùng để lưu trữ học viên. | Not null |
| 03 | Tên | Nvarchar(20) | Là thuộc tính đơn trị, dùng để lưu trữ tên học viên. | Not null |
| 04 | Địa chỉ | Nvarchar(100) | Là thuộc tính đơn trị, dùng để lưu trữ địa chỉ thường trú của học viên. | Null |
| 05 | Số điện thoại | Numeric(12,0) | Là thuộc tính đơn trị, dùng để lưu trữ số điện thoại học viên. | Not null |
| 06 | Giới tính | Nvarchar(3) | Là thuộc tính đơn trị, dùng để lưu trữ giới tính học sinh và có ràng buộc chỉ được là “Nam” hoặc “Nữ”. | Not null |
| 07 | Mật khẩu | Nvarchar(16) | Là thuộc tính đơn trị, cùng với mã học viên tạo thành tài khoản đăng nhập vào thi. | Not null |
| 08 | Mã lớp học | Int | Là khóa phụ liên kết đến (Mã lớp học) của bảng lớp học. Dùng để xác định học viên trực thuộc lớp nào. | Null |

**Bảng lớp học**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Mô tả** | **Null/Not null** |
| 01 | Mã lớp học | Int | Là khóa chính, sử dụng constraint identity để định danh lớp học. | Not null |
| 02 | Tên lớp | Nvarchar(20) | Là thuộc tính đơn trị, sử dụng constraint unique để tránh việc trùng lặp và dùng để lưu trữ tên lớp học. | Not null |

**Bảng môn học**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Mô tả** | **Null/Not null** |
| 01 | Mã môn học | Int | Là khóa chính, sử dụng constraint indentity. Dùng để định danh môn học. | Not null |
| 02 | Tên môn học | Nvarchar(20) | Là thuộc tính đơn trị, sử dụng constraint unique để tránh việc trùng lặp và dùng để lưu trữ tên môn học. | Not null |

**Bảng chương**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Mô tả** | **Null/Not null** |
| 01 | Mã chương | Int | Là khóa chính, sử dụng constraint identity. Dùng để định danh chương trong môn học. | Not null |
| 02 | Tên chương | Nvarchar(20) | Là thuộc tính đơn trị, dùng để lưu tên chương học. | Not null |
| 03 | Mã môn học | Int | Là khóa phụ, liên kết đến (Mã môn học) của bảng môn học. Dùng để xác định chương học đó thuộc môn học nào. | Null |

**Bảng câu hỏi**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Mô tả** | **Null/Not null** |
| 01 | Mã câu hỏi | Int | Là khóa chính, sử dụng constraint indentity. Dùng để định danh câu hỏi. | Not null |
| 02 | Nội dung | Text | Là thuộc tính đơn trị, dùng để lưu trữ nội dung câu hỏi. | Not null |
| 03 | Đáp án A | Text | Là thuộc tính đơn trị, dùng để lưu trữ nội dung đáp án câu A. | Not null |
| 04 | Đáp án B | Text | Là thuộc tính đơn trị, dùng để lưu trữ nội dung đáp án câu B. | Not null |
| 05 | Đáp án C | Text | Là thuộc tính đơn trị,  dùng để lưu trữ nội dung đáp án câu C. | Not null |
| 06 | Đáp án D | Text | Là thuộc tính đơn trị, dùng để lưu trữ nội dung đáp án câu D. | Not null |
| 07 | Đáp án đúng | Varchar(1) | Là thuộc tính đơn trị, dùng để lưu trữ câu trả lời đúng (A,B,C,D). | Not null |
| 08 | Mã chương | Int | Là khóa phụ, liên kết đến (Mã chương) bảng chương. Dùng để xác định câu hỏi thuộc chương nào. | Null |

**Bảng đề thi**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Mô tả** | **Null/Not null** |
| 01 | Mã đề thi | Int | Là khóa chính, sử dụng constraint identity. Dùng để định danh đề thi. | Not null |
| 02 | Ngày thi | Date | Là thuộc tính đơn trị, dùng để lưu trữ ngày thi. | Not null |
| 03 | Mã môn học | Int | Là khóa phụ, liên kết (Mã môn học) của bảng môn học. Dùng để xác định đề thi thuộc môn học nào. | Null |

**Bảng bài làm**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Mô tả** | **Null/Not null** |
| 01 | Mã bài làm | Int | Là khóa chính, sử dụng constraint identity. Dùng để định danh bài làm. | Not null |

**Bảng danh sách môn học**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Mô tả** | **Null/Not null** |
| 01 | Mã lớp học | Int | Là 1 trong các trường dữ liệu của khóa chính. Là khóa phụ liên kết đến (Mã lớp học) bảng lớp học. | Not null |
| 02 | Mã môn học | Int | Là 1 trong các trường dữ liệu của khóa chính. Là khóa phụ liên kết đến (Mã môn học) bảng môn học. | Not null |

**Bảng chi tiết đề thi**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Mô tả** | **Null/Not null** |
| 01 | Mã câu hỏi | Int | Là 1 trong các trường dữ liệu của khóa chính. Là khóa phụ liên kết đến (Mã câu hỏi) bảng câu hỏi. | Null |
| 02 | Mã đề thi | Int | Là 1 trong các trường dữ liệu của khóa chính. Là khóa phụ liên kết đến (Mã đề thi) bảng đề thi. | Not null |
| 03 | Số thứ tự câu hỏi | Int | Là 1 trong các trường dữ liệu của khóa chính. Là thuộc tính đơn trị, dùng để lưu trữ số thứ tự của từng câu hỏi. | Not null |

**Bảng chi tiết bài làm**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Mô tả** | **Null/Not null** |
| 01 | Mã bài làm | Int | Là 1 trong các trường dữ liệu của khóa chính. Là khóa phụ liên kết đến (Mã bài làm) bảng bài làm. | Not null |
| 02 | Mã đề thi | Int | Là 1 trong các trường dữ liệu của khóa chính. Là khóa phụ liên kết đến (Mã đề thi) bảng đề thi. | Null |
| 03 | Số thứ tự câu hỏi | Int | Là 1 trong các trường dữ liệu của khóa chính. Là thuộc tính đơn trị, dùng để lưu trữ số thứ tự của từng câu hỏi. | Not null |
| 04 | Câu trả lời | Varchar(1) | Là thuộc tính đơn trị dùng để lưu trữ câu trả lời của học viên(A,B,C,D). | Null |

# CHƯƠNG III : SỬ SỤNG SQL SEVER XÂY DỰNG CSDL

## Tạo CSDL

USE master

GO

CREATE DATABASE THITHUTRACNGHIEM

ON PRIMARY (

NAME = 'THITHUTRACNGHIEM\_DATA',

FILENAME = 'E:\HQTCSDL\THITHUTRACNGHIEM\_DATA.mdf',

SIZE = 50MB,

MAXSIZE = UNLIMITED,

FILEGROWTH = 5MB

)

LOG ON (

NAME = 'THITHUTRACNGHIEM\_LOG',

FILENAME = 'E:\HQTCSDL\THITHUTRACNGHIEM\_LOG.ldf',

SIZE = 20MB,

MAXSIZE = UNLIMITED,

FILEGROWTH = 5MB

)

COLLATE Vietnamese\_CI\_AS;

GO

## Tạo Table (tương ứng với lược đồ quan hệ)

* Bảng học viên

USE THITHUTRACNGHIEM

GO

CREATE TABLE HOCVIEN (

MAHV INT IDENTITY PRIMARY KEY,

HOHV NVARCHAR(20) NOT NULL,

TENHV NVARCHAR(20) NOT NULL,

DIACHI NVARCHAR(100) NULL,

GIOITINH NVARCHAR(3) NOT NULL CHECK (GIOITINH IN (N'Nam',N'Nữ')),

SDT NUMERIC(12,0) NOT NULL CHECK (SDT > 0 AND LEN(SDT)<12),

MATKHAU NVARCHAR(16) NOT NULL CHECK (LEN(MATKHAU) > 5),

MALH INT NULL

)

GO

* Bảng lớp học

CREATE TABLE LOPHOC (

MALH INT IDENTITY PRIMARY KEY,

TENLH NVARCHAR(20) UNIQUE NOT NULL

)

GO

* Bảng môn học

CREATE TABLE MONHOC (

MAMH INT IDENTITY PRIMARY KEY,

TENMH NVARCHAR(20) UNIQUE NOT NULL

)

GO

* Bảng chương

CREATE TABLE CHUONG (

MACHUONG INT IDENTITY PRIMARY KEY,

TENCHUONG NVARCHAR(20) NOT NULL,

MAMH INT NULL

)

GO

* Bảng câu hỏi

CREATE TABLE CAUHOI (

MACH INT IDENTITY PRIMARY KEY,

NOIDUNG TEXT NOT NULL,

DAPANA TEXT NOT NULL,

DAPANB TEXT NOT NULL,

DAPANC TEXT NOT NULL,

DAPAND TEXT NOT NULL,

DAPANDUNG VARCHAR(1) NOT NULL,

MACHUONG INT NULL

)

GO

* Bảng đề thi

CREATE TABLE DETHI (

MADT INT IDENTITY PRIMARY KEY,

NGAYTHI DATE NOT NULL,

MAMH INT NULL

)

GO

* Bảng danh sách môn học

CREATE TABLE DANHSACHMH (

MALH INT NOT NULL,

MAMH INT NOT NULL,

CONSTRAINT PK\_DSMH PRIMARY KEY (MALH,MAMH)

)

GO

* Bảng chi tiết đề thi

CREATE TABLE CHITIETDETHI(

MACH INT NOT NULL,

MADT INT NOT NULL,

SOTT INT NOT NULL,

CONSTRAINT PK\_CTDT PRIMARY KEY (MACH, MADT, SOTT)

)

GO

* Bảng chi tiết bài làm

CREATE TABLE CHITIETBAILAM (

MAHV INT NOT NULL,

MADT INT NOT NULL,

SOTT INT NOT NULL,

CAUTL VARCHAR(1) NULL,

CONSTRAINT PK\_CTBL PRIMARY KEY (MAHV, MADT, SOTT),

)

GO

* Các ràng buộc

ALTER TABLE HOCVIEN ADD CONSTRAINT FK\_HVLH FOREIGN KEY (MALH) REFERENCES LOPHOC(MALH)

GO

ALTER TABLE CHUONG ADD CONSTRAINT FK\_CMH FOREIGN KEY (MAMH) REFERENCES MONHOC(MAMH)

GO

ALTER TABLE CAUHOI ADD CONSTRAINT FK\_CHC FOREIGN KEY (MACHUONG) REFERENCES CHUONG(MACHUONG)

GO

ALTER TABLE DANHSACHMH ADD CONSTRAINT FK\_DSMH\_MH FOREIGN KEY (MAMH) REFERENCES MONHOC(MAMH)

GO

ALTER TABLE DANHSACHMH ADD CONSTRAINT FK\_DSMH\_LH FOREIGN KEY (MALH) REFERENCES LOPHOC(MALH)

GO

ALTER TABLE CHITIETDETHI ADD CONSTRAINT FK\_CTDT\_DT FOREIGN KEY (MADT) REFERENCES DETHI(MADT)

GO

ALTER TABLE CHITIETDETHI ADD CONSTRAINT FK\_CTDT\_CH FOREIGN KEY (MACH) REFERENCES CAUHOI(MACH)

GO

ALTER TABLE CHITIETDETHI ADD CONSTRAINT CHECK\_RANGE CHECK (SOTT >0 AND SOTT <41)

GO

ALTER TABLE CHITIETBAILAM ADD CONSTRAINT FK\_CTBL\_HV FOREIGN KEY (MAHV) REFERENCES HOCVIEN(MAHV)

GO

ALTER TABLE CHITIETBAILAM ADD CONSTRAINT FK\_CTBL\_DT FOREIGN KEY (MADT) REFERENCES DETHI(MADT)

GO

ALTER TABLE CHITIETBAILAM ADD CONSTRAINT CHECK\_RANGE\_BL CHECK (SOTT >0 AND SOTT <41)

GO

## Câu lệnh SQL

|  |  |  |
| --- | --- | --- |
| **Mã chức năng** | **Tên chức năng** | **Câu lệnh SQL** |
| M-QL-01-01 | Thêm thông tin lớp học | CREATE PROC sp\_THEMLOPHOC  @TENLH NVARCHAR(20)  AS  BEGIN  BEGIN TRANSACTION  IF @TENLH IS NULL  BEGIN  PRINT N'Tên lớp học không được để trống.'  ROLLBACK TRANSACTION  RETURN  END  IF EXISTS (SELECT 1 FROM LOPHOC WHERE TENLH = @TENLH)  BEGIN  PRINT N'Tên lớp học đã tồn tại.'  ROLLBACK  RETURN  END  INSERT INTO LOPHOC (TENLH) VALUES (@TENLH)  COMMIT TRANSACTION  END  GO |
| M-QL-01-02 | Sửa thông tin lớp học | CREATE PROC sp\_CAPNHATLOPHOC  @MALH INT,  @TENLH NVARCHAR(20)  AS  BEGIN  BEGIN TRANSACTION  IF @MALH IS NULL OR @TENLH IS NULL  BEGIN  PRINT N'Thông tin lớp học không được để trống.'  ROLLBACK TRANSACTION  RETURN  END  IF NOT EXISTS (SELECT 1 FROM LOPHOC WHERE MALH = @MALH)  BEGIN  PRINT N'Lớp học không tồn tại.'  ROLLBACK TRANSACTION  RETURN  END  ELSE IF EXISTS (SELECT 1 FROM LOPHOC WHERE TENLH = @TENLH)  BEGIN  PRINT N'Tên lớp học đã tồn tại.'  ROLLBACK TRANSACTION  RETURN  END  UPDATE LOPHOC SET TENLH = @TENLH WHERE MALH = @MALH  COMMIT TRANSACTION  END  GO |
| M-QL-01-03 | Xóa thông tin lớp học | --TẠO BẢNG BACKUP DỮ LIỆU  CREATE TABLE LOPHOC\_BACKUP (  MALH INT NOT NULL,  TENLH NVARCHAR(20) NOT NULL,  DELETEDDATE DATE DEFAULT GETDATE()  )  GO  --TẠO TRIGGER XỬ LÝ DỮ LIỆU BACKUP  CREATE TRIGGER Trg\_BACKUPLOPHOC  ON LOPHOC  AFTER DELETE  AS  BEGIN  BEGIN  SET IDENTITY\_INSERT LOPHOC ON;  END  INSERT INTO LOPHOC\_BACKUP(MALH,TENLH) SELECT MALH,TENLH FROM deleted;  BEGIN  SET IDENTITY\_INSERT LOPHOC OFF;  END  END  GO  --THỦ TỤC XÓA LỚP HỌC THEO MÃ  CREATE PROC sp\_XOALOPHOC  @MALH INT  AS  BEGIN  BEGIN TRANSACTION  IF @MALH IS NULL  BEGIN  PRINT N'Mã lớp học không được để trống.'  ROLLBACK TRANSACTION  RETURN  END  IF NOT EXISTS (SELECT 1 FROM LOPHOC WHERE MALH = @MALH)  BEGIN  PRINT N'Lớp học không tồn tại.'  ROLLBACK TRANSACTION  RETURN  END  UPDATE HOCVIEN SET MALH = NULL WHERE MALH = @MALH  DELETE FROM DANHSACHMH WHERE MALH = @MALH  DELETE FROM LOPHOC WHERE MALH = @MALH  COMMIT TRANSACTION  END  GO  --THỦ TỤC KHÔI PHỤC DỮ LIỆU  CREATE PROC sp\_KHOIPHUCLOPHOC  @MALH INT  AS  BEGIN  BEGIN TRANSACTION  IF @MALH IS NULL  BEGIN  PRINT N'Mã lớp học không được để trống.'  ROLLBACK TRANSACTION  RETURN  END  IF NOT EXISTS (SELECT 1 FROM LOPHOC\_BACKUP WHERE MALH = @MALH)  BEGIN  PRINT N'Lớp học không tồn tại.'  ROLLBACK TRANSACTION  RETURN  END  BEGIN  SET IDENTITY\_INSERT LOPHOC ON;  END  INSERT INTO LOPHOC (MALH,TENLH) SELECT MALH,TENLH FROM LOPHOC\_BACKUP BK WHERE BK.MALH = @MALH  DELETE FROM LOPHOC\_BACKUP WHERE MALH = @MALH  BEGIN  SET IDENTITY\_INSERT LOPHOC OFF;  END  COMMIT TRANSACTION  END  GO |
| M-QL-01-04 | Tra cứu thông tin lớp học theo tên hoặc mã lớp | CREATE PROCEDURE sp\_TIMKIEMLOPHOC  @SearchText NVARCHAR(20)  AS  BEGIN  SELECT LH.MALH, LH.TENLH, COUNT(HV.MAHV) AS SISO  FROM LOPHOC LH JOIN HOCVIEN HV ON HV.MALH = LH.MALH  WHERE TENLH LIKE '%' + @SearchText + '%'  OR CAST(LH.MALH AS NVARCHAR(10)) = @SearchText  GROUP BY LH.MALH,LH.TENLH  ORDER BY LH.MALH  END  GO |
| M-QL-01-05 | Tra cứu thông tin các môn học của một lớp | CREATE PROCEDURE sp\_TIMKIEMDSMH  @SearchText NVARCHAR(20)  AS  BEGIN  SELECT LH.MALH, LH.TENLH, MH.MAMH, MH.TENMH  FROM DANHSACHMH DS JOIN LOPHOC LH ON DS.MALH = DS.MALH JOIN MONHOC MH ON MH.MAMH = DS.MAMH  WHERE TENLH LIKE '%' + @SearchText + '%'  OR CAST(LH.MALH AS NVARCHAR(10)) = @SearchText  ORDER BY LH.MALH  END  GO |
| M-QL-01-06 | Thêm môn học vào danh sách môn của lớp | CREATE PROC sp\_THEMMOIMHVAODSMH  @MALH INT,  @MAMH INT  AS  BEGIN  BEGIN TRANSACTION  IF @MALH IS NULL OR @MALH IS NULL  BEGIN  PRINT N'Thông tin không được để trống.'  ROLLBACK TRANSACTION  RETURN  END  ELSE IF NOT EXISTS (SELECT 1 FROM MONHOC WHERE MAMH = @MAMH)  BEGIN  PRINT N'Môn học không tồn tại.'  ROLLBACK TRANSACTION  RETURN  END  ELSE IF NOT EXISTS (SELECT 1 FROM LOPHOC WHERE MALH = @MALH)  BEGIN  PRINT N'Lớp học không tồn tại.'  ROLLBACK TRANSACTION  RETURN  END  ELSE IF EXISTS (SELECT 1 FROM DANHSACHMH WHERE MALH = @MALH AND MAMH = @MAMH)  BEGIN  PRINT N'Môn học đã tồn tại trong danh sách'  ROLLBACK TRANSACTION  RETURN  END  INSERT INTO DANHSACHMH VALUES (@MALH,@MAMH)  COMMIT TRANSACTION  END  GO |
| M-QL-01-07 | Xóa môn học trong danh sách môn học của lớp | CREATE PROC sp\_XOAMHKHOIDSMH  @MALH INT,  @MAMH INT  AS  BEGIN  BEGIN TRANSACTION  IF @MALH IS NULL OR @MALH IS NULL  BEGIN  PRINT N'Thông tin không được để trống.'  ROLLBACK TRANSACTION  RETURN  END  ELSE IF NOT EXISTS (SELECT 1 FROM MONHOC WHERE MAMH = @MAMH)  BEGIN  PRINT N'Môn học không tồn tại.'  ROLLBACK TRANSACTION  RETURN  END  ELSE IF NOT EXISTS (SELECT 1 FROM LOPHOC WHERE MALH = @MALH)  BEGIN  PRINT N'Lớp học không tồn tại.'  ROLLBACK TRANSACTION  RETURN  END  DELETE FROM DANHSACHMH WHERE MALH = @MALH AND MAMH = @MAMH  COMMIT TRANSACTION  END |
| M-QL-02-01 | Thêm thông tin môn học | CREATE PROC sp\_THEMMH  @TENMH NVARCHAR(20)  AS  BEGIN  BEGIN TRANSACTION  IF @TENMH IS NULL  BEGIN  PRINT N'Tên môn học không được để trống.'  ROLLBACK TRANSACTION  RETURN  END  IF EXISTS (SELECT 1 FROM MONHOC WHERE TENMH = @TENMH)  BEGIN  PRINT N'Tên môn học đã tồn tại.'  ROLLBACK  RETURN  END  INSERT INTO MONHOC(TENMH) VALUES (@TENMH)  COMMIT TRANSACTION  END  GO |
| M-QL-02-02 | Sửa thông tin môn học | CREATE PROC sp\_CAPNHATMH  @MAMH INT,  @TENMH NVARCHAR(20)  AS  BEGIN  BEGIN TRANSACTION  IF @MAMH IS NULL OR @TENMH IS NULL  BEGIN  PRINT N'Thông tin môn học không được để trống.'  ROLLBACK TRANSACTION  RETURN  END  IF NOT EXISTS (SELECT 1 FROM MONHOC WHERE MAMH = @MAMH)  BEGIN  PRINT N'Môn học không tồn tại.'  ROLLBACK TRANSACTION  RETURN  END  ELSE IF EXISTS (SELECT 1 FROM MONHOC WHERE TENMH = @TENMH)  BEGIN  PRINT N'Tên môn học đã tồn tại.'  ROLLBACK TRANSACTION  RETURN  END  UPDATE MONHOC SET TENMH = @TENMH WHERE MAMH = @MAMH  COMMIT TRANSACTION  END  GO |
| M-QL-02-03 | Xóa thông tin môn học | --CREATE TABLE TO BACKUP DATA  CREATE TABLE MONHOC\_BACKUP (  MAMH INT NOT NULL,  TENMH NVARCHAR(20) NOT NULL,  DELETEDDATE DATE DEFAULT GETDATE()  )  GO  --CREATE TRIGGER TO BACKUP DATA  CREATE TRIGGER Trg\_BACKUPMONHOC  ON MONHOC  AFTER DELETE  AS  BEGIN  BEGIN  SET IDENTITY\_INSERT MONHOC ON;  END  INSERT INTO MONHOC\_BACKUP(MAMH,TENMH) SELECT MAMH,TENMH FROM deleted;  BEGIN  SET IDENTITY\_INSERT MONHOC OFF;  END  END  GO  --CREATE PROC TO DELETE  CREATE PROC sp\_XOAMH  @MAMH INT  AS  BEGIN  BEGIN TRANSACTION  IF @MAMH IS NULL  BEGIN  PRINT N'Mã môn học không được để trống.'  ROLLBACK TRANSACTION  RETURN  END  IF NOT EXISTS (SELECT 1 FROM MONHOC WHERE MAMH = @MAMH)  BEGIN  PRINT N'Môn học không tồn tại.'  ROLLBACK TRANSACTION  RETURN  END  UPDATE DETHI SET MAMH = NULL WHERE MAMH = @MAMH  UPDATE CHUONG SET MAMH = NULL WHERE MAMH = @MAMH  DELETE FROM DANHSACHMH WHERE MAMH = @MAMH  DELETE FROM MONHOC WHERE MAMH = @MAMH  COMMIT TRANSACTION  END  GO  --CREATE PROC RESTORE DATA  CREATE PROC sp\_KHOIPHUCMH  @MAMH INT  AS  BEGIN  BEGIN TRANSACTION  IF @MAMH IS NULL  BEGIN  PRINT N'Mã môn học không được để trống.'  ROLLBACK TRANSACTION  RETURN  END  IF NOT EXISTS (SELECT 1 FROM MONHOC\_BACKUP WHERE MAMH = @MAMH)  BEGIN  PRINT N'Môn học không tồn tại.'  ROLLBACK TRANSACTION  RETURN  END  BEGIN  SET IDENTITY\_INSERT MONHOC ON;  END  INSERT INTO MONHOC(MAMH,TENMH) SELECT MAMH,TENMH FROM MONHOC\_BACKUP BK WHERE BK.MAMH = @MAMH  DELETE FROM MONHOC\_BACKUP WHERE MAMH = @MAMH  BEGIN  SET IDENTITY\_INSERT MONHOC OFF;  END  COMMIT TRANSACTION  END  GO |
| M-QL-02-04 | Tra cứu thông tin môn học theo tên hoặc mã môn học | CREATE PROCEDURE sp\_TIMKIEMMH  @SearchText NVARCHAR(20)  AS  BEGIN  SELECT MH.MAMH, MH.TENMH, COUNT(C.MACHUONG) AS SOCHUONG  FROM MONHOC MH JOIN CHUONG C ON MH.MAMH = C.MAMH  WHERE MH.TENMH LIKE '%' + @SearchText + '%'  OR CAST(MH.MAMH AS NVARCHAR(10)) = @SearchText  GROUP BY MH.MAMH, MH.TENMH  ORDER BY MH.MAMH  END  GO |
| M-QL-03-01 | Thêm thông tin chương học | --TRIGGER TO DEAL WITH DUPLICATE CHAPTER  CREATE TRIGGER Trg\_XULYTRUNGTENCHUONG  ON CHUONG  INSTEAD OF INSERT, UPDATE  AS  BEGIN  IF EXISTS (  SELECT 1  FROM CHUONG c  INNER JOIN inserted i ON c.MAMH = i.MAMH AND c.TENCHUONG = i.TENCHUONG  )  BEGIN  PRINT(N'Tên chương đã tồn tại trong môn học')  RETURN;  END;  INSERT INTO CHUONG(MAMH, TENCHUONG)  SELECT MAMH, TENCHUONG  FROM inserted;  END;  GO  --PROC CREATE CHUONG  CREATE PROC sp\_THEMCHUONG  @TENCHUONG NVARCHAR(20) ,  @MAMH INT  AS  BEGIN  BEGIN TRANSACTION  IF @TENCHUONG IS NULL OR @MAMH IS NULL  BEGIN  PRINT N'Thông tin không được để trống.'  ROLLBACK TRANSACTION  RETURN  END  IF NOT EXISTS (SELECT 1 FROM MONHOC WHERE MAMH = @MAMH)  BEGIN  PRINT N'Mã môn học không tồn tại.'  ROLLBACK TRANSACTION  RETURN  END  INSERT INTO CHUONG (TENCHUONG,MAMH) VALUES (@TENCHUONG,@MAMH)  COMMIT TRANSACTION  END  GO |
| M-QL-03-02 | Sửa thông tin chương học | CREATE PROC sp\_CAPNHATCHUONG  @MACHUONG INT,  @TENCHUONG NVARCHAR(20) ,  @MAMH INT  AS  BEGIN  BEGIN TRANSACTION  IF @TENCHUONG IS NULL OR @MAMH IS NULL OR @MACHUONG IS NULL  BEGIN  PRINT N'Thông tin không được để trống.'  ROLLBACK TRANSACTION  RETURN  END  IF NOT EXISTS (SELECT 1 FROM MONHOC WHERE MAMH = @MAMH)  BEGIN  PRINT N'Mã môn học không tồn tại.'  ROLLBACK TRANSACTION  RETURN  END  IF NOT EXISTS (SELECT 1 FROM CHUONG WHERE MACHUONG = @MACHUONG)  BEGIN  PRINT N'Chương không tồn tại.'  ROLLBACK TRANSACTION  RETURN  END  UPDATE CHUONG SET MAMH = @MAMH, TENCHUONG = @TENCHUONG WHERE MACHUONG = @MACHUONG  COMMIT TRANSACTION  END  GO |
| M-QL-03-03 | Xóa thông tin chương học | --CREATE TABLE TO BACKUP DATA  CREATE TABLE CHUONG\_BACKUP(  MACHUONG INT PRIMARY KEY,  TENCHUONG NVARCHAR(20) NOT NULL,  MAMH INT NULL,  DELETEDDATE DATE DEFAULT GETDATE()  )  GO  --CREATE TRIGGER TO BACKUP  CREATE TRIGGER Trg\_BACKUPCHUONG  ON CHUONG  AFTER DELETE  AS  BEGIN  BEGIN  SET IDENTITY\_INSERT CHUONG ON;  END  INSERT INTO CHUONG\_BACKUP(MACHUONG,TENCHUONG,MAMH) SELECT MACHUONG,TENCHUONG, MAMH FROM deleted;  BEGIN  SET IDENTITY\_INSERT CHUONG OFF;  END  END  GO  --CREATE PROC TO DELETE CHUONG  CREATE PROC sp\_XOACHUONG  @MACHUONG INT  AS  BEGIN  BEGIN TRANSACTION  IF @MACHUONG IS NULL  BEGIN  PRINT N'Mã chương không được để trống.'  ROLLBACK TRANSACTION  RETURN  END  ELSE IF NOT EXISTS (SELECT 1 FROM CHUONG WHERE MACHUONG = @MACHUONG)  BEGIN  PRINT N'Chương học không tồn tại'  ROLLBACK TRANSACTION  RETURN  END  UPDATE CAUHOI SET MACHUONG = NULL WHERE MACHUONG = @MACHUONG  DELETE FROM CHUONG WHERE MACHUONG = @MACHUONG  COMMIT TRANSACTION  END  GO  --CREATE PROC TO RESTORE DATA  CREATE PROC sp\_KHOIPHUCCHUONG  @MACHUONG INT  AS  BEGIN  BEGIN TRANSACTION  IF @MACHUONG IS NULL  BEGIN  PRINT N'Mã chương không được để trống.'  ROLLBACK TRANSACTION  RETURN  END  ELSE IF NOT EXISTS (SELECT 1 FROM CHUONG\_BACKUP WHERE MACHUONG = @MACHUONG)  BEGIN  PRINT N'Mã chương không tồn tại.'  ROLLBACK TRANSACTION  RETURN  END  DECLARE @MAMH INT  SELECT @MAMH = MAMH FROM CHUONG\_BACKUP WHERE MACHUONG = @MACHUONG  IF NOT EXISTS (SELECT 1 FROM MONHOC WHERE MAMH = @MAMH)  BEGIN  PRINT N'Môn học của chương đã không còn tồn tại.'  ROLLBACK TRANSACTION  RETURN  END  BEGIN  SET IDENTITY\_INSERT CHUONG ON;  END  INSERT INTO CHUONG(MACHUONG,TENCHUONG,MAMH) SELECT MACHUONG,TENCHUONG,MAMH FROM CHUONG\_BACKUP WHERE MACHUONG = @MACHUONG  DELETE FROM CHUONG\_BACKUP WHERE MACHUONG = @MACHUONG  BEGIN  SET IDENTITY\_INSERT CHUONG OFF;  END  COMMIT TRANSACTION  END  GO |
| M-QL-03-04 | Tra cứu thông tin chương học theo tên hoặc mã chương học | CREATE PROCEDURE sp\_TIMKIEMCHUONG  @SearchText NVARCHAR(20)  AS  BEGIN  SELECT CH.MACHUONG, CH.TENCHUONG,MH.MAMH, MH.TENMH, COUNT(C.MACH) AS SOCAUHOI  FROM CHUONG CH JOIN MONHOC MH ON MH.MAMH = CH.MAMH JOIN CAUHOI C ON C.MACHUONG = CH.MACHUONG  WHERE CH.TENCHUONG LIKE '%' + @SearchText + '%'  OR CAST(CH.MACHUONG AS NVARCHAR(10)) = @SearchText  GROUP BY CH.MACHUONG, CH.TENCHUONG, MH.MAMH, MH.TENMH  ORDER BY CH.MACHUONG, CH.TENCHUONG, MH.MAMH, MH.TENMH, SOCAUHOI  END  GO |
| M-QL-04-01 | Thêm thông tin câu hỏi | CREATE PROC sp\_THEMCAUHOI  @NOIDUNG TEXT,  @A TEXT, @B TEXT, @C TEXT, @D TEXT, @DAPAN NVARCHAR(1),  @MACHUONG INT  AS  BEGIN  BEGIN TRANSACTION  IF @NOIDUNG IS NULL OR @A IS NULL OR @B IS NULL OR @C IS NULL OR @D IS NULL OR @DAPAN IS NULL OR @MACHUONG IS NULL  BEGIN  PRINT N'Dữ liệu không được để trống.'  ROLLBACK TRANSACTION  RETURN  END  ELSE IF NOT EXISTS (SELECT 1 FROM CHUONG WHERE MACHUONG = @MACHUONG)  BEGIN  PRINT N'Chương học không tồn tại.'  ROLLBACK TRANSACTION  RETURN  END  INSERT INTO CAUHOI VALUES (@NOIDUNG,@A,@B,@C,@D,@DAPAN,@MACHUONG)  COMMIT TRANSACTION  END  GO |
| M-QL-04-02 | Sửa thông tin câu hỏi | CREATE PROC sp\_CAPNHATCAUHOI  @MACH INT,  @NOIDUNG TEXT,  @A TEXT, @B TEXT, @C TEXT, @D TEXT, @DAPAN NVARCHAR(1),  @MACHUONG INT  AS  BEGIN  BEGIN TRANSACTION  IF @NOIDUNG IS NULL OR @A IS NULL OR @B IS NULL OR @C IS NULL OR @D IS NULL OR @DAPAN IS NULL OR @MACHUONG IS NULL  BEGIN  PRINT N'Dữ liệu không được để trống.'  ROLLBACK TRANSACTION  RETURN  END  ELSE IF NOT EXISTS (SELECT 1 FROM CAUHOI WHERE MACH = @MACH)  BEGIN  PRINT N'Câu hỏi không tồn tại.'  ROLLBACK TRANSACTION  RETURN  END  ELSE IF NOT EXISTS (SELECT 1 FROM CHUONG WHERE MACHUONG = @MACHUONG)  BEGIN  PRINT N'Chương học không tồn tại.'  ROLLBACK TRANSACTION  RETURN  END  UPDATE CAUHOI SET NOIDUNG = @NOIDUNG, DAPANA = @A, DAPANB = @B, DAPANC = @C, DAPAND = @D, DAPANDUNG = @DAPAN , MACHUONG = @MACHUONG  COMMIT TRANSACTION  END  GO |
| M-QL-04-03 | Xóa thông tin câu hỏi | --CREATE TABLE TO BACKUP DATA  CREATE TABLE CAUHOI\_BACKUP (  MACH INT NOT NULL,  NOIDUNG TEXT NOT NULL,  A TEXT NOT NULL,  B TEXT NOT NULL,  C TEXT NOT NULL,  D TEXT NOT NULL,  DAPAN NVARCHAR(1) NOT NULL,  MACHUONG INT NULL)  GO  --CREATE TRIGGER TO BACKUP  CREATE TRIGGER Trg\_BACKUPCAUHOI  ON CAUHOI  INSTEAD OF DELETE  AS  BEGIN  INSERT INTO CAUHOI\_BACKUP(MACH, NOIDUNG, A, B, C, D, DAPAN, MACHUONG)  SELECT MACH, NOIDUNG, DAPANA, DAPANB, DAPANC, DAPAND, DAPANDUNG, MACHUONG  FROM deleted;  DELETE FROM CAUHOI  WHERE MACH IN (SELECT MACH FROM deleted);  END;  GO  --CREATE PROC TO DELETE  CREATE PROC sp\_XOACAUHOI  @MACH INT  AS  BEGIN  BEGIN TRANSACTION  IF @MACH IS NULL  BEGIN  PRINT N'Mã câu hỏi không được để trống.'  ROLLBACK TRANSACTION  RETURN  END  ELSE IF NOT EXISTS (SELECT 1 FROM CAUHOI WHERE MACH = @MACH)  BEGIN  PRINT N'Mã câu hỏi không tồn tại.'  ROLLBACK TRANSACTION  RETURN  END  DELETE FROM CHITIETDETHI WHERE MACH = @MACH  DELETE FROM CAUHOI WHERE MACH = @MACH  COMMIT TRANSACTION  END  GO  --CREATE PROC TO RESTORE DATA  CREATE PROC sp\_KHOIPHUCCAUHOI  @MACH INT  AS  BEGIN  BEGIN TRANSACTION  IF @MACH IS NULL  BEGIN  PRINT N'Mã câu hỏi không được để trống.'  ROLLBACK TRANSACTION  RETURN  END  ELSE IF NOT EXISTS (SELECT 1 FROM CAUHOI\_BACKUP WHERE MACH = @MACH)  BEGIN  PRINT N'Mã câu hỏi không tồn tại.'  ROLLBACK TRANSACTION  RETURN  END  DECLARE @MACHUONG INT  SELECT @MACHUONG = MACHUONG FROM CAUHOI\_BACKUP WHERE MACH = @MACH  IF NOT EXISTS (SELECT 1 FROM CHUONG WHERE MACHUONG = @MACHUONG)  BEGIN  PRINT N'Chương của câu hỏi đã không còn tồn tại.'  ROLLBACK TRANSACTION  RETURN  END  SET IDENTITY\_INSERT CAUHOI ON;  INSERT INTO CAUHOI (MACH,NOIDUNG, DAPANA,DAPANB,DAPANC,DAPAND,DAPANDUNG,MACHUONG) SELECT MACH,NOIDUNG,A,B,C,D,DAPAN,MACHUONG FROM CAUHOI\_BACKUP WHERE MACH = @MACH  DELETE FROM CAUHOI\_BACKUP WHERE MACH = @MACH  SET IDENTITY\_INSERT CAUHOI OFF;  COMMIT TRANSACTION  END  GO |
| M-QL-04-04 | Tra cứu thông tin câu hỏi theo tên hoặc mã câu hỏi | CREATE PROCEDURE sp\_TIMKIEMCAUHOI  @SearchText NVARCHAR(20)  AS  BEGIN  SELECT CH.MACH,CH.NOIDUNG,CH.DAPANA,CH.DAPANB, CH.DAPANC,CH.DAPAND, CH.DAPANDUNG,CH.MACHUONG,C.TENCHUONG  FROM CAUHOI CH JOIN CHUONG C ON C.MACHUONG = CH.MACHUONG  WHERE CAST(CH.NOIDUNG AS NVARCHAR(MAX)) LIKE N'%' + @SearchText + N'%'  OR CAST(CH.MACH AS NVARCHAR(10)) = @SearchText  END  GO |
| M-QL-05-01 | Thêm thông tin học viên | CREATE PROC sp\_THEMHOCVIEN  @HOHV NVARCHAR(20),  @TENHV NVARCHAR(20),  @DIACHI NVARCHAR(100),  @SDT NUMERIC(12,0),  @GIOITINH NVARCHAR(3),  @MALH INT  AS  BEGIN  BEGIN TRANSACTION  IF @HOHV IS NULL OR @TENHV IS NULL OR @DIACHI IS NULL OR @SDT IS NULL OR @GIOITINH IS NULL OR @MALH IS NULL  BEGIN  PRINT N'Thông tin nhập không được để trống.'  ROLLBACK TRANSACTION  RETURN  END  ELSE IF NOT EXISTS (SELECT 1 FROM LOPHOC WHERE MALH = @MALH)  BEGIN  PRINT N'Lớp học không tồn tại.'  ROLLBACK TRANSACTION  RETURN  END  DECLARE @MATKHAU NVARCHAR(16)  SET @MATKHAU = SUBSTRING(CONVERT(NVARCHAR(36), NEWID()), 2, 15)  INSERT INTO HOCVIEN (HOHV,TENHV,DIACHI,SDT,GIOITINH,MATKHAU,MALH) VALUES (@HOHV,@TENHV,@DIACHI,@SDT,@GIOITINH,@MATKHAU,@MALH)  COMMIT TRANSACTION  END  GO |
| M-QL-05-02 | Sửa thông tin học viên | CREATE PROC sp\_CAPNHATHOCVIEN  @MAHV INT,  @HOHV NVARCHAR(20),  @TENHV NVARCHAR(20),  @DIACHI NVARCHAR(100),  @SDT NUMERIC(12,0),  @GIOITINH NVARCHAR(3),  @MALH INT  AS  BEGIN  BEGIN TRANSACTION  IF @MAHV IS NULL OR @HOHV IS NULL OR @TENHV IS NULL OR @DIACHI IS NULL OR @SDT IS NULL OR @GIOITINH IS NULL OR @MALH IS NULL  BEGIN  PRINT N'Thông tin nhập không được để trống.'  ROLLBACK TRANSACTION  RETURN  END  ELSE IF NOT EXISTS (SELECT 1 FROM LOPHOC WHERE MALH = @MALH)  BEGIN  PRINT N'Lớp học không tồn tại.'  ROLLBACK TRANSACTION  RETURN  END  UPDATE HOCVIEN SET HOHV = @HOHV , TENHV = @TENHV, DIACHI = @DIACHI, SDT = @SDT , GIOITINH = @GIOITINH, MALH = @MALH WHERE MAHV = @MAHV  COMMIT TRANSACTION  END  GO |
| M-QL-05-03 | Xóa thông tin học viên | --CREATE TABLE TO BACKUP  CREATE TABLE HOCVIEN\_BACKUP  (  MAHV INT NOT NULL,  HOHV NVARCHAR(20) NOT NULL,  TENHV NVARCHAR(20) NOT NULL,  DIACHI NVARCHAR(100) NULL,  GIOITINH NVARCHAR(3) NOT NULL CHECK (GIOITINH IN (N'Nam',N'Nữ')),  SDT NUMERIC(12,0) NOT NULL CHECK (SDT > 0 AND LEN(SDT)<12),  MATKHAU NVARCHAR(16) NOT NULL CHECK (LEN(MATKHAU) > 5),  MALH INT NULL,  DELETEDDATE DATE DEFAULT GETDATE(),  )  GO  --CREATE TRIGGER TO DELETE  CREATE TRIGGER Trg\_BACKUPHOCVIEN  ON HOCVIEN  AFTER DELETE  AS  BEGIN  INSERT INTO HOCVIEN\_BACKUP(MAHV,HOHV,TENHV,DIACHI,GIOITINH,SDT,MATKHAU,MALH)  SELECT MAHV,HOHV,TENHV,DIACHI,GIOITINH,SDT,MATKHAU,MALH  FROM deleted;  END  GO  --CREATE PROC TO DELETE  CREATE PROC sp\_XOAHOCVIEN  @MAHV INT  AS  BEGIN  BEGIN TRANSACTION  IF @MAHV IS NULL  BEGIN  PRINT N'Mã học viên không được để trống.'  ROLLBACK TRANSACTION  RETURN  END  ELSE IF NOT EXISTS (SELECT 1 FROM HOCVIEN WHERE MAHV = @MAHV)  BEGIN  PRINT N'Mã học viên không tồn tại.'  ROLLBACK TRANSACTION  RETURN  END  DELETE FROM CHITIETBAILAM WHERE MAHV = @MAHV  DELETE FROM HOCVIEN WHERE MAHV = @MAHV  COMMIT TRANSACTION  END  GO  --CREATE RESTORE DATA  CREATE PROC sp\_KHOIPHUCHOCVIEN  @MAHV INT  AS  BEGIN  BEGIN TRANSACTION  IF @MAHV IS NULL  BEGIN  PRINT N'Mã học viên không được để trống.'  ROLLBACK TRANSACTION  RETURN  END  ELSE IF NOT EXISTS (SELECT 1 FROM HOCVIEN\_BACKUP WHERE MAHV = @MAHV)  BEGIN  PRINT N'Mã học viên không tồn tại.'  ROLLBACK TRANSACTION  RETURN  END  DECLARE @MALH INT  SELECT @MALH = MALH FROM HOCVIEN\_BACKUP WHERE MAHV = @MAHV  IF NOT EXISTS (SELECT 1 FROM LOPHOC WHERE MALH = @MALH)  BEGIN  SET IDENTITY\_INSERT HOCVIEN ON;  INSERT INTO HOCVIEN(MAHV,HOHV,TENHV,DIACHI,GIOITINH,SDT,MATKHAU,MALH) SELECT MAHV,HOHV,TENHV,DIACHI,GIOITINH,SDT,MATKHAU,NULL FROM HOCVIEN\_BACKUP  SET IDENTITY\_INSERT HOCVIEN OFF;  DELETE FROM HOCVIEN\_BACKUP WHERE MAHV = @MAHV  COMMIT TRANSACTION  RETURN  END  ELSE  BEGIN  SET IDENTITY\_INSERT HOCVIEN ON;  INSERT INTO HOCVIEN(MAHV,HOHV,TENHV,DIACHI,GIOITINH,SDT,MATKHAU,MALH) SELECT MAHV,HOHV,TENHV,DIACHI,GIOITINH,SDT,MATKHAU,MALH FROM HOCVIEN\_BACKUP  SET IDENTITY\_INSERT HOCVIEN OFF;  DELETE FROM HOCVIEN\_BACKUP WHERE MAHV = @MAHV  COMMIT TRANSACTION  RETURN  END  END  GO |
| M-QL-05-03 | Tra cứu thông tin học viên theo tên hoặc mã học viên | CREATE PROCEDURE sp\_TIMKIEMHOCVIEN  @SearchText NVARCHAR(20)  AS  BEGIN  SELECT HV.MAHV,HV.HOHV+' '+HV.TENHV AS HOTEN, HV.DIACHI,HV.GIOITINH,HV.SDT, LH.MALH, LH.TENLH  FROM HOCVIEN HV JOIN LOPHOC LH ON HV.MALH = LH.MALH  WHERE HV.TENHV LIKE N'%' + @SearchText + N'%'  OR HV.HOHV LIKE N'%' + @SearchText + N'%'  OR CAST(HV.MAHV AS NVARCHAR(10)) = @SearchText  ORDER BY HV.MAHV  END  GO |
| M-HV-01-01 | Tạo đề thi tự động | CREATE PROC sp\_TAODETHITUDONG  @MAHV INT,  @MAMH INT  AS  BEGIN  BEGIN TRANSACTION  IF @MAHV IS NULL OR @MAMH IS NULL  BEGIN  PRINT N'Thông tin không được để trống.'  ROLLBACK TRANSACTION  RETURN  END  ELSE IF NOT EXISTS (SELECT 1 FROM HOCVIEN WHERE MAHV = @MAHV) OR NOT EXISTS (SELECT 1 FROM MONHOC WHERE MAMH = @MAMH)  BEGIN  PRINT N'Mã học viên hoặc mã môn học không tồn tại.'  ROLLBACK TRANSACTION  RETURN  END  DECLARE @SOLUONGCH INT  SELECT @SOLUONGCH = COUNT(CH.MACH) FROM CAUHOI CH JOIN CHUONG C ON C.MACHUONG = CH.MACHUONG JOIN MONHOC MH ON MH.MAMH = C.MAMH WHERE MH.MAMH = @MAMH  IF(@SOLUONGCH <1)  BEGIN  PRINT N'Đề thi chưa sẵn sàng.'  ROLLBACK TRANSACTION  RETURN  END  DECLARE @INSERTEDDETHI TABLE(  MADT INT,  NGAYTHI DATE,  MAMH INT  )  INSERT INTO DETHI (NGAYTHI,MAMH) OUTPUT inserted.MADT,inserted.NGAYTHI,inserted.MAMH INTO @INSERTEDDETHI VALUES (GETDATE(),@MAMH)  DECLARE @MADT INT  DECLARE @NGAYTHI DATE  SELECT @MADT = MADT, @NGAYTHI = NGAYTHI FROM @INSERTEDDETHI  DECLARE @DSCH TABLE(  ID INT IDENTITY,  MACH INT UNIQUE  )  INSERT INTO @DSCH SELECT TOP 1 MACH FROM CAUHOI ORDER BY NEWID()  DECLARE @COUNTER INT = 1  WHILE @COUNTER < (SELECT MAX(ID) FROM @DSCH)  BEGIN  DECLARE @MACAUHOI INT  SELECT @MACAUHOI = MACH FROM @DSCH WHERE ID = @COUNTER  INSERT INTO CHITIETDETHI (MADT, MACH,SOTT)  VALUES (@MADT, @MACAUHOI,@COUNTER)  SET @COUNTER = @COUNTER + 1  END  COMMIT TRANSACTION  END  GO |
| M-HV-01-02 | Lưu kết quả bài làm | CREATE PROC sp\_LUUDAPANBAILAM  @MAHV INT,  @MADT INT,  @STT INT,  @CAUTRALOI VARCHAR(1)  AS  BEGIN  BEGIN TRANSACTION  IF @MAHV IS NULL OR @MADT IS NULL OR @STT IS NULL  BEGIN  PRINT N'Thông tin không được để trống.'  ROLLBACK TRANSACTION  RETURN  END  ELSE IF NOT EXISTS (SELECT 1 FROM HOCVIEN WHERE MAHV = @MAHV) OR NOT EXISTS (SELECT 1 FROM DETHI WHERE MADT = @MADT)  BEGIN  PRINT N'Mã học viên hoặc mã đề thi không còn tồn tại.'  ROLLBACK TRANSACTION  RETURN  END  INSERT INTO CHITIETBAILAM (MAHV,MADT,SOTT,CAUTL) VALUES (@MAHV,@MADT,@STT,@CAUTRALOI)  COMMIT TRANSACTION  END  GO |
| M-HV-02-01 | Tra cứu kết quả thi | --HÀM CHẤM ĐIỂM TỰ ĐỘNG  CREATE FUNCTION dbo.CHAMDIEM (@MAHV INT,@MADT INT)  RETURNS DECIMAL(4,2)  AS  BEGIN  DECLARE @TONG DECIMAL(4,2)  SET @TONG = 0.0  DECLARE @BANGCHAM TABLE (  MACH INT,  STT INT,  CAUTL NVARCHAR(1),  DAPAN NVARCHAR(1)  )  INSERT INTO @BANGCHAM (MACH,STT,CAUTL,DAPAN) SELECT CTDT.MACH,CTBL.SOTT,CTBL.CAUTL,CH.DAPANDUNG FROM CHITIETBAILAM CTBL JOIN CHITIETDETHI CTDT ON CTBL.MADT = CTDT.MADT JOIN CAUHOI CH ON CH.MACH = CTDT.MACH WHERE CTBL.SOTT = CTDT.SOTT AND CTDT.MADT = @MADT  DECLARE @COUNTER DECIMAL(4,2) = 1.0  WHILE @COUNTER <= (SELECT MAX(STT) FROM @BANGCHAM)  BEGIN  DECLARE @TT VARCHAR(1)  DECLARE @DA VARCHAR(1)  SELECT @TT = CAUTL , @DA = DAPAN FROM @BANGCHAM WHERE @COUNTER = STT  IF @TT = @DA  BEGIN  SET @TONG = @TONG + 1  END  SET @COUNTER = @COUNTER + 1  END  SET @TONG = (@TONG / (SELECT MAX(STT) FROM @BANGCHAM)) \* 10  RETURN @TONG  END  GO  --TRA CỨU KẾT QUẢ THI  CREATE FUNCTION dbo.TRACUUKETQUA (@MAHV INT,@MAMH INT)  RETURNS @BANGDIEM TABLE (  MAHV INT,  HOTEN NVARCHAR(41),  MONHOC NVARCHAR(20),  MADETHI INT,  KETQUA DECIMAL(4,2)  )  AS  BEGIN  INSERT INTO @BANGDIEM (MAHV,HOTEN,MONHOC,MADETHI,KETQUA) SELECT TOP 5 HV.MAHV,HV.HOHV+' '+HV.TENHV AS HOTEN,MH.TENMH,DT.MADT, dbo.CHAMDIEM(@MAHV,DT.MADT) AS KETQUA FROM DETHI DT JOIN MONHOC MH ON MH.MAMH = DT.MAMH  JOIN CHITIETBAILAM CTBL ON CTBL.MADT = DT.MADT  JOIN HOCVIEN HV ON HV.MAHV = CTBL.MAHV  WHERE MH.MAMH = @MAMH AND HV.MAHV = @MAHV  GROUP BY HV.MAHV, HV.HOHV+' '+HV.TENHV,MH.TENMH,DT.MADT,dbo.CHAMDIEM(@MAHV,DT.MADT)  RETURN  END  GO |
| M-HV-02-01 | Cập nhật mật khẩu | CREATE PROC sp\_CAPNHATMATKHAU  @MAHV INT,  @MKCU NVARCHAR(16),  @MKMOI NVARCHAR(16)  AS  BEGIN  BEGIN TRANSACTION  IF @MAHV IS NULL OR @MKCU IS NULL OR @MKMOI IS NULL  BEGIN  PRINT N'Thông tin không được để trống.'  ROLLBACK TRANSACTION  RETURN  END  ELSE IF NOT EXISTS (SELECT 1 FROM HOCVIEN WHERE MAHV = @MAHV AND MATKHAU = @MKCU)  BEGIN  PRINT N'Thông tin đăng nhập bị sai.'  ROLLBACK TRANSACTION  RETURN  END  UPDATE HOCVIEN SET MATKHAU = @MKMOI WHERE MAHV = @MAHV  COMMIT TRANSACTION  END  GO |