

Mức logic và vật lý

Hai mức logic và vật lý không tách rời nhau. Ví dụ khi thực hiện mệnh đề tạo một bảng (là gì sẽ bàn phần kế tiếp) mức logic sẽ song song với việc yêu cầu hệ thống phải cấp phát vùng nhớ bao nhiêu, vị trí,...

Mức vật lý là công việc của máy tính và chúng ta không cần đi sâu. Chỉ quan tâm mức logic.

Cơ sở dữ liệu quan hệ mức logic

Mức khái niệm	Mức logic
Entity (Entities)	Table (Tables)
Attribute (Attributes)	Column (Columns)
Instance (Instances)	Row (Rows)
Bussiness Rules	Constraints

* Quá trình ánh xạ (mapping) từ thiết kế mức khái niệm sang thiết kế mức logic gọi là chuẩn hóa (normalization).

Bảng (Table)

Là đơn vị lưu trữ chính trong mô hình quan hệ mức logic. Tương đương với khái niệm Entity mức khái niệm.

Là một cấu trúc hai chiều gồm các cột (columns) và các hàng (rows).

Mỗi bảng có duy nhất một tên là danh từ tương ứng với một đối tượng trong thế giới thực như sinh viên, khoa, môn học,...Chiều dài tối đa cho tên bảng phụ thuộc vào các DBMS, thông thường từ 18 đến 255 ký tự.

Cột (Column) – Kiểu dữ liệu (Data type)

Mỗi cột (column) tương ứng với một thuộc tính (attribute) từ mô hình khái niệm.

Là đơn vị dữ liệu nhỏ nhất có thể được tham chiếu trong một cơ sở dữ liệu quan hệ.

Giống như các biến trong ngôn ngữ lập trình, mỗi cột luôn có một cái tên duy nhất và đi kèm với một kiểu dữ liệu.

Một số kiểu dữ liệu tổng quát bao gồm:

- **Fixed-length character:** Dùng cho dữ liệu là các chuỗi ký tự có chiều dài cố định, ví dụ số điện thoại, mã bưu chính,...
- **Variable-length character:** Dùng cho dữ liệu là các chuỗi ký tự có chiều dài thay đổi, ví dụ: họ tên, email,... Đây là kiểu dữ liệu phổ biến trong các cơ sở dữ liệu.
- **Long text:** Dùng cho dữ liệu dạng văn bản, ví dụ các cột mô tả thông tin.
- **Integer:** Dùng cho dữ liệu kiểu số nguyên, ví dụ tuổi, năm sinh, số ngày,...
- **Decimal:** Dùng cho dữ liệu kiểu số có phần thập phân, ví dụ số điện thoại, mã bưu chính,...
- **Currency:** Dùng cho kiểu tiền tệ.
- **Date / time:** Dùng cho kiểu thời gian.

Tùy theo các DBMS mà các kiểu dữ liệu trên sẽ có những kiểu tương đương. Bảng so sánh các kiểu dữ liệu từ các DBMS phổ biến là Access, SQL Server và Oracle:

Data Type	Microsoft Access	Microsoft SQL Server	Oracle
Fixed-length character	TEXT	CHAR	CHAR
Variable-length character	MEMO	VARCHAR	VARCHAR
Long text	MEMO	TEXT	CLOB or LONG (deprecated)
Integer	INTEGER or LONG INTEGER	INTEGER or SMALLINT or TINYINT	NUMBER
Decimal	NUMBER	DECIMAL or NUMERIC	NUMBER
Currency	CURRENCY	MONEY or SMALLMONEY	None, use NUMBER
Date/time	DATE/TIME	DATETIME or SMALLDATETIME	DATE or TIMESTAMP

Sự tương đương trên chỉ tương đối, ví dụ kiểu VARCHAR từ Oracle cho phép tối đa 4000 ký tự, trong khi kiểu MEMO của Access có thể lên tới 1 tỷ ký tự.

Ví dụ chúng ta có hai bảng **KHOA** và **SINHVIEN** được thiết kế trong SQL Server 2012 (bảng Express) như sau:

KHOA: gồm hai cột **MaKH** (Kiểu **nchar** và chiều dài tối đa là 4 ký tự) và **TenKH** (Kiểu **nchar** và chiều dài tối đa là 10 ký tự):

DESKTOP-O6BPUKE\...hVien - dbo.KHOA			
	Column Name	Data Type	Allow Nulls
🔑	MaKH	nchar(4)	<input type="checkbox"/>
	TenKH	nchar(10)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Kiểu **nchar** tương tự kiểu **char** nhưng có chứa ký tự Unicode.

SINHVIEN: Gồm các cột **MaSV** (Kiểu **nchar** và chiều dài tối đa là 5 ký tự), **TenSV** (Kiểu **nchar** và chiều dài tối đa là 10 ký tự), **Tuoi** (Kiểu **smallint** hay **tinyint** hay **int**), **KhoaID** (Kiểu **nchar** và chiều dài tối đa 4 ký tự):

DESKTOP-O6BPUKE\...n - dbo.SINHVIEN			
	Column Name	Data Type	Allow Nulls
🔑	MaSV	nchar(5)	<input type="checkbox"/>
	TenSV	nchar(10)	<input checked="" type="checkbox"/>
	Tuoi	smallint	<input checked="" type="checkbox"/>
	KhoaID	nchar(4)	<input type="checkbox"/>

Các ràng buộc (Constraints)

Các ràng buộc (constraints) mức logic tương ứng với các ràng buộc nghiệp vụ (business rules) mức khái niệm. Một số ràng buộc chính:

- **Ràng buộc khóa chính (Primary Key Constraints):** Mỗi bảng cần có một khóa chính để xác định tính duy nhất cho mỗi hàng, nghĩa là, khóa chính đảm bảo trong bảng không có 2 hàng nào giống nhau. Khóa chính có thể là một hay nhiều thuộc tính (cột) kết hợp. Ví dụ các khóa chính MaKH và MaSV từ các bảng KHOA và SINHVIEN. Khi thiết kế trong SQL Server sẽ có biểu tượng chìa khóa.
- **Ràng buộc tham chiếu (Referential Constraints Constraints):** Một ràng buộc tham chiếu giữa hai bảng mức logic tương đương với một quan hệ giữa hai entities trong mức khái niệm. Và mỗi quan hệ được đề cập ở đây là mỗi quan hệ một – nhiều, ví dụ bảng SINHVIEN và bảng KHOA. Trong mỗi quan hệ này, bảng phía một (bảng KHOA) được gọi là bảng cha (parent table) và bảng phía nhiều (SINHVIEN) gọi là bảng con (child table). Bí mật của ràng buộc tham chiếu hay quan hệ một – nhiều là khóa ngoại (foreign key) – là một cột (hay nhóm cột) vừa thuộc bảng cha và vừa thuộc bảng con. Tuy nhiên, có một yêu cầu quan trọng, khóa ngoại là khóa chính

(primary key) của bảng cha. Mặc dù là một, nhưng có thể đặt tên khác nhau khi ở bảng cha (đóng vai trò khóa chính) và khi ở bảng con (vai trò khóa ngoại). Mặc dù cho phép khác tên nhưng các ràng buộc khác như kiểu dữ liệu, NOT NULL,...phải đảm bảo.

Ví dụ: MaKH là khóa chính bảng KHOA nhưng là khóa ngoại bảng SINHVIEN có thể đặt tên là KhoaID.

Mỗi DBMS sẽ có những cách khác nhau khi thiết kế ràng buộc tham chiếu giữa các bảng. Trong SQL Server, chúng ta vào mục Database Diagrams

Tables and Columns ? X

Relationship name:

FK_SINHVIEN_KHOA

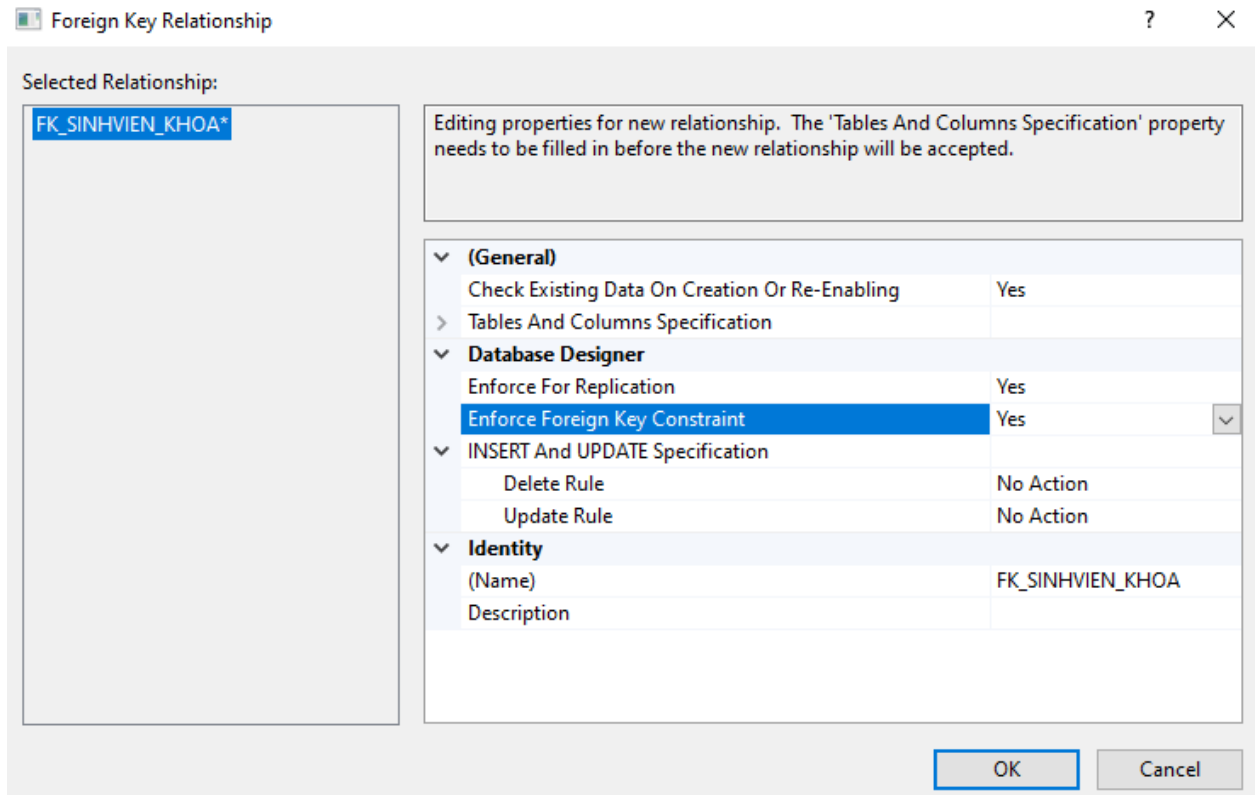
Primary key table: Foreign key table:

KHOA SINHVIEN

MaKH	KhoaID

OK Cancel

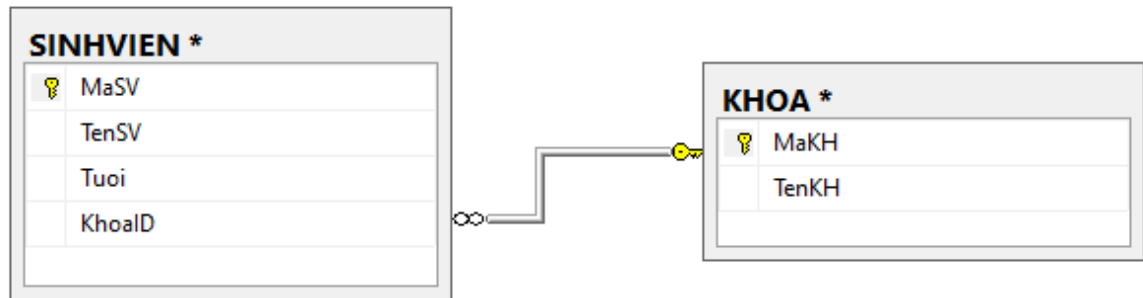
Nhấn OK sẽ xuất hiện:



Trong mục Database Designer chúng ta cần quan tâm một số sự kiện:

- Check Existing Data on Creation or Re-enabling: Nếu mục này chọn Yes thì khi thêm một khóa ngoại mới sẽ không được phép nếu xung đột với dữ liệu tồn tại sẵn. Mặc định là Yes.
- Name: Tên của ràng buộc tham chiếu
- Description: Một vài mô tả cho ràng buộc tham chiếu
- Enforce for Replication: Xác định liệu có áp dụng ràng buộc tham chiếu khi thay đổi dữ liệu trong quá trình sao chép (Replication process) của SQL Server hay không. Mặc định là Yes để áp dụng.
- Enforce Foreign Key Constraint: Thiết lập ràng buộc tham chiếu nếu chọn Yes. Mặc định Yes.
- INSERT and UPDATE Specification: Có hai tùy chọn (options) chính là Delete Rule và Update Rule nhằm xác định các sự kiện sẽ xảy ra khi cập nhật (Update Rule) hay xóa (Delete Rule) thông tin từ bảng cha. Các giá trị cho các tùy chọn bao gồm No Action, Cascade, Set Null và Set Default. Mặc định cho cả hai tùy chọn là No Action.

Nhấn OK. Mối quan hệ hai bảng hình thành như sau:



- Bảng trung gian (Intersection Tables): Quan hệ một – nhiều hay ràng buộc tham chiếu giữa các bảng có thể được xác định bởi khóa ngoại. Xét bảng MONHOC được thiết kế trong SQL Server như sau:

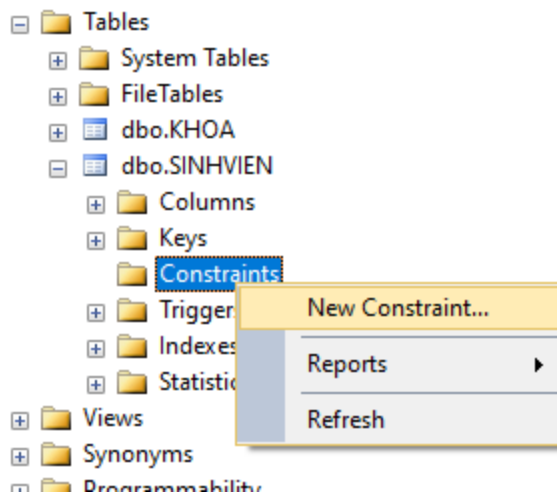
DESKTOP-O6BPUKE...en - dbo.MONHOC			
	Column Name	Data Type	Allow Nulls
	MaMH	nchar(5)	<input type="checkbox"/>
	TenMH	nchar(10)	<input checked="" type="checkbox"/>
	GiangVien	nchar(10)	<input checked="" type="checkbox"/>

- Mối quan hệ giữa hai bảng SINHVIEN và MONHOC là mối quan hệ nhiều nhiều thì ràng buộc được thực hiện thông qua bảng trung gian (ví dụ SV_MH). Nhắc lại rằng, quan hệ nhiều – nhiều có thể được biểu diễn trực tiếp ở mức khái niệm nhưng khi sang mức logic thì các DBMS không cho phép thể hiện trực tiếp quan hệ này.

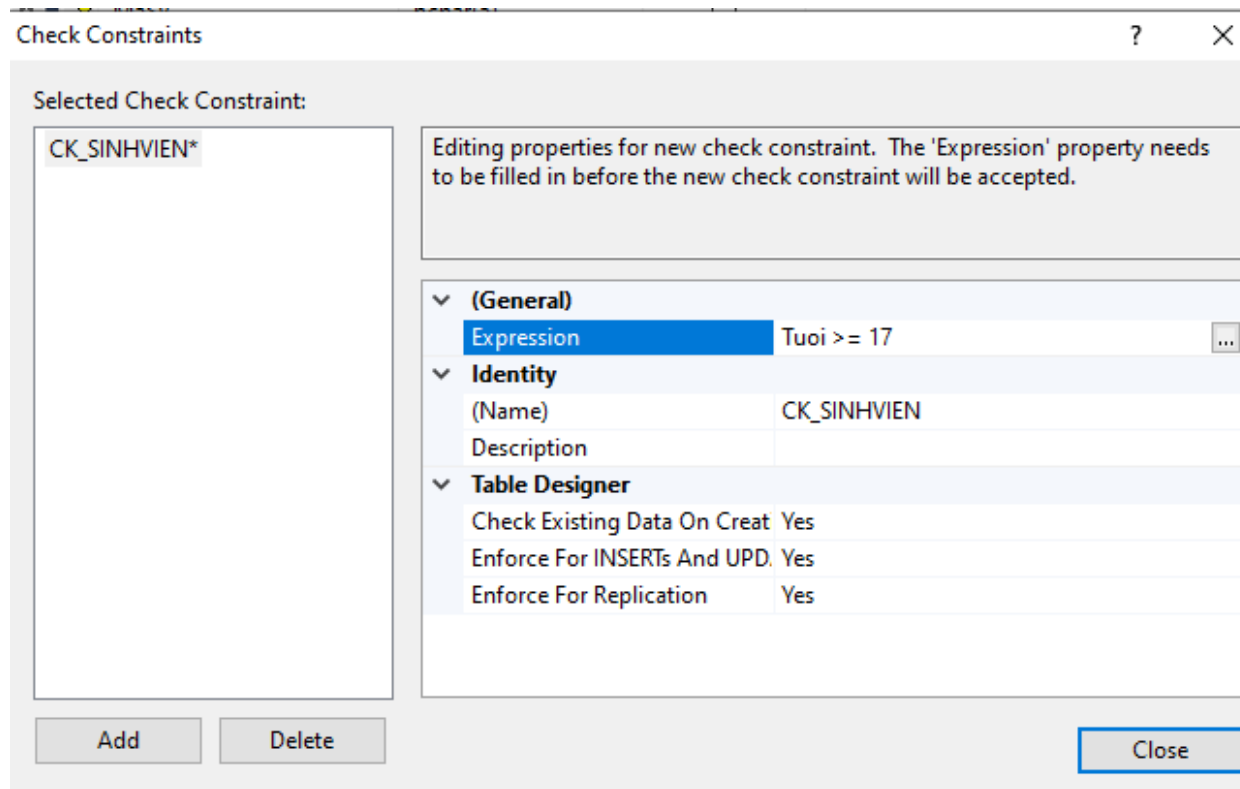
Bảng trung gian có một số đặc điểm:

- Quan hệ giữa bảng trung gian (SV_MH) đến hai bảng chính (SINHVIEN và MONHOC) là quan hệ một – nhiều;
- Hai bảng chính đóng vai trò bảng cha trong khi bảng trung gian đóng vai trò bảng con;
- Khóa chính bảng trung gian có thể là cặp khóa ngoại từ các bảng chính (cặp (MaSV, MaMH)). Khóa chính trong trường hợp này gọi là khóa tự nhiên (natural key). Tuy nhiên, trong thiết kế cơ sở dữ liệu, chúng ta nên tạo một khóa chính (Ví dụ ID) cho bảng trung gian thay vì dùng cặp khóa ngoại. Khóa chính mới được tạo này (ID) gọi là khóa đại diện (surrogate key);
- Nếu xem mỗi hàng của bảng là một tập hợp thì mỗi hàng trong bảng trung gian (SV_MH) là giao của một hàng bảng chính này (SINHVIEN) với một hàng bảng chính khác (MONHOC).

- Ràng buộc toàn vẹn (Integrity Constraints): Là ràng buộc quan trọng để đảm bảo tính chính xác và nhất quán của dữ liệu trong cơ sở dữ liệu quan hệ. Có hai ràng buộc phổ biến thuộc kiểu ràng buộc này là
 - NOT NULL: Cột chọn ràng buộc này bắt buộc phải có giá trị không được để rỗng vì trong SQL Server, một ô nhận giá trị NULL nghĩa là ô rỗng. Khóa chính mỗi bảng luôn phải chọn NOT NULL. Cột Allow Nulls từ các bảng SINHVIEN, KHOA và MONHOC cho phép các cột của bảng nhận giá trị NULL (ô rỗng) nếu chọn, ngược lại là NOT NULL (ví dụ các khóa chính).
 - CHECK: Ràng buộc này dùng một biểu thức luận lý đơn giản để quy định giá trị cho cột. Ví dụ cột Tuoi từ bảng SINHVIEN quy định giá trị ≥ 17 . Trong SQL Server để thực hiện ràng buộc CHECK cho cột Tuoi bảng SINHVIEN chúng ta thực hiện như sau:
 - Mở rộng bảng SINHVIEN trong cửa sổ Object Explorer, nhấn chuột phải mục Constraints chọn New Constraint...



- Trong hộp thoại Check Constraints chọn mục Expression gõ biểu thức $Tuoi \geq 17$



Nhấn Close.

Sau khi tạo các bảng và thiết lập các ràng buộc cần thiết. Công việc còn lại là nhập dữ liệu cho các bảng bằng cách nhấn chuột phải vào tên bảng trong thư mục Tables từ cửa sổ Object Explorer. ***Một lưu ý quan trọng, cần nhập dữ liệu các bảng đóng vai trò bảng cha trước khi nhập các bảng con.***