

Bài 4

Tạo bảng (nâng cao)

Phần I – Hướng dẫn thực hành từng bước

1. Mục tiêu

- Nắm vững cú pháp tạo bảng.
- Cài đặt khóa chính.
- Hiểu biết về khóa ngoại và cài đặt lên cột trong bảng.
- Hiểu và viết được mã lệnh ràng buộc cho cột dữ liệu trong bảng.
- Nắm vững cú pháp chỉnh sửa bảng, xóa bảng.

Tạo bảng (nâng cao)

- Sản phẩm phần mềm khi triển khai phía khách hàng thì phần cơ sở dữ liệu (csdl) thường sẽ được chuyển giao dưới dạng file sql. Một csdl thực hiện backup hoặc attach-detach chỉ phù hợp với nhu cầu sao lưu dự phòng.
- File SQL khi chuyển giao sẽ là mã lệnh khởi tạo CSDL, bảng, các ràng buộc, khóa chính, khóa ngoại và dữ liệu mặc định (nếu có).
- Sản phẩm phần mềm trong quá trình sử dụng, bảo trì, nâng cấp có thể sẽ cần sửa đổi, bổ sung hoặc loại bỏ những trường dữ liệu vì thế cần nắm rõ các cú pháp khởi tạo, chỉnh sửa, ràng buộc để có thể đáp ứng các yêu cầu nghiệp vụ.

Các ràng buộc: mục đích để đảm bảo dữ liệu được lưu trữ đúng đắn và phù hợp với quy tắc, quy chuẩn, nghiệp vụ đề ra khi thiết kế. Trong MS SQL có các ràng buộc sau:

1. **Primary Key** – Khóa chính, là duy nhất, mỗi phần tử trong bảng được đánh một khóa và không lặp lại.
2. **Foreign Key** – Khóa ngoại, mối quan hệ giữa các thực thể khi vật lý hóa thành bảng trong csdl thì được thể hiện bằng khóa ngoại.
3. **Unique** – Định danh duy nhất, trong trường hợp muốn thể hiện một

trường dữ liệu mỗi bản ghi không trùng lặp trong tập hợp kiểu như mã nhân viên, email, số điện thoại.

4. **Check** – Kiểm tra, sử dụng trong tình huống muốn xác thực dữ liệu khi thêm vào phải đảm bảo thỏa mãn điều kiện nào đó ví dụ lớn hơn 0, nằm trong khoảng....
5. **NOT NULL** – Không để trống, sử dụng để đảm bảo dữ liệu phải được nhập có giá trị, không được để trống.

Mã định danh duy nhất – uniqueidentifier: là một kiểu dữ liệu đặc biệt trong MS SQL, trường dữ liệu này lưu trữ 32 ký tự gồm số và chữ lẫn lộn, nếu tính tổ hợp thì có đến hàng tỷ tỷ. Khi được cài đặt kiểu dữ liệu này thì mỗi bản ghi thêm vào chỉ cần gọi hàm **NEWID()**.

2. Thực hiện

Vận dụng lý thuyết về khởi tạo, chỉnh sửa, ràng buộc bảng để áp dụng vào bài toán xây dựng csdl cho một phần mềm thương mại điện tử với yêu cầu quản lý được Loại Sản Phẩm, Sản Phẩm, Khách Hàng, Hóa Đơn. Trước hết, khởi tạo csdl

ProjectShoes bằng lệnh sau:

```
CREATE DATABASE ProjectShoes
GO

USE ProjectShoes
GO
```

Bài thực hành 1: Dựa vào mô tả bảng dữ liệu sau, thực hiện viết lệnh SQL để tạo bảng dữ liệu **CATELOG**

Tên cột Field	Kiểu dữ liệu DataType	Null	Ràng buộc ContraINT	Mô tả Content
<u>CatalogId</u>	INT	No	Primary Key Auto_increment	Mã danh mục sản phẩm
CatalogName	NVARCHAR(100)	No		Tên danh mục sản phẩm
Status	TINYINT		Default 1	Trạng thái: 1- Đang hiển thị

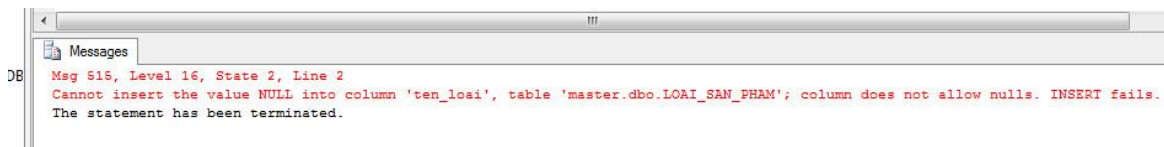
				0- Đang ẩn
--	--	--	--	------------

Bước 1: Mã sql để tạo bảng với các yêu cầu như đặc tả:

```
-- Tạo bảng danh mục sản phẩm
CREATE TABLE CATELOG (
    CatelogId INT IDENTITY(1,1) PRIMARY KEY, -- Khóa chính,
    kiểu INT, tự tăng
    CatelogName NVARCHAR(50) NOT NULL, -- Không được để
    trống
    Status TINYINT DEFAULT(1) -- Mặc định dữ liệu là 1
)
GO
```

Bước 2: Thêm dữ liệu vào bảng:

```
-- Thêm dữ liệu
INSERT INTO CATELOG (CatelogName) VALUES
(NULL)
GO
```



Câu lệnh trên khi thực thi bị lỗi bởi vì cột dữ liệu **CatelogName** đã ràng buộc không được để trống.

Bước 3: Thêm dữ liệu vào bảng:

```
-- Thêm dữ liệu
INSERT INTO CATELOG (CatelogName) VALUES
(N'Giày nam'),
(N'Giày cao'),
(N'Ủng cao su'),
(N'Bốt nữ cao'),
(N'Bốt nữ da')
GO

-- Truy vấn bảng
SELECT * FROM CATELOG
GO
```

Results		Messages	
	CatalogId	CatalogName	Status
1	1	Giày nam	1
2	2	Giày cao	1
3	3	Ủng cao su	1
4	4	Bốt nữ cao	1
5	5	Bốt nữ da	1

Dữ liệu tại cột trạng thái mặc định có giá trị là 1 do đã được cài đặt từ lúc tạo bảng.

Bài thực hành 2: Dựa vào mô tả bảng dữ liệu sau, thực hiện viết lệnh SQL để tạo bảng dữ liệu **PRODUCT**

Tên cột Field	Kiểu dữ liệu DataType	Null	Ràng buộc Constraint	Mô tả Content
<u>ProductId</u>	INT	No	Primary Key; Auto_increment	Mã sản phẩm
ProductName	NVARCHAR(200)	No		Tên sản phẩm
Content	NVARCHAR(250)	Yes		Mô tả ngắn gọn
ContentDetail	NTEXT	Yes		Mô tả chi tiết
PriceInput	FLOAT	No	Default:0	Giá nhập
PriceOutput	FLOAT	No	Default: 0	Giá bán
Views	INT	No	Default:0	Lượt xem
Created	DATETIME	Yes	Current, Check (>= ngày hiện tại)	Ngày cập nhập vào hệ thống, >= ngày hiện tại
Status	TINYINT	Yes	Check (0 hoặc 1)	Trạng thái: 1- Đang hiển thị 0- Đang ẩn
CatalogId	INT	Yes	Foreign key	Khóa ngoại, tham chiếu đến CATELOG

Bước 1: Mã sql để tạo bảng với các yêu cầu như đặc tả:

```
-- Tạo bảng SAN_PHAM
CREATE TABLE PRODUCT (
    ProductId INT IDENTITY(1,1) PRIMARY KEY, -- Khóa chính
    ProductName NVARCHAR(200) NOT NULL, -- Không để trống
    Content NVARCHAR(250),
```

```

ContentDetail NTEXT,
PriceInput FLOAT NOT NULL DEFAULT(0), -- Không để
trống, mặc định là 0
PriceOutput FLOAT NOT NULL DEFAULT(0), -- Không để
trống, mặc định là 0
Views INT NOT NULL DEFAULT(0), -- Không để trống, mặc
định là 0
Created DATETIME CHECK(Created >= GETDATE()), -- Kiểm
tra giá trị nhập luôn >= ngày hiện tại
Status TINYINT CHECK(Status = 0 OR Status = 1)
DEFAULT(1), -- Trạng thái chỉ có 2 giá trị 0 hoặc 1
CatalogId INT FOREIGN KEY REFERENCES CATALOG(CatalogId)
)
GO

```

Bước 2: Thêm dữ liệu với tình huống ngày mặc định:

```

-- Thêm dữ liệu
INSERT INTO PRODUCT(ProductName, PriceInput, PriceOutput,
Created, CatalogId) VALUES
('Conserve Chuck 1', 600, 650, GETDATE(), 1),
('Gucci SNN69', 120, 150, GETDATE(), 2),
('Adidas', 1450, 1600, GETDATE(), 1),
('Nike MG21', 690, 750, GETDATE(), 3),
('Tom 300', 55, 65, GETDATE(), 3),
('Lacoste 066', 99, 105, GETDATE(), 5)
GO

-- Truy vấn bảng
SELECT * FROM PRODUCT
GO

```

	ProductId	ProductName	Content	ContentDetail	PriceInput	PriceOutput	Views	Created	Status	CatalogId
1	1	Conserve Chuck 1	NULL	NULL	600	650	0	2017-08-23 11:21:41.520	1	1
2	2	Gucci SNN69	NULL	NULL	120	150	0	2017-08-23 11:21:41.520	1	2
3	3	Adidas	NULL	NULL	1450	1600	0	2017-08-23 11:21:41.520	1	1
4	4	Nike MG21	NULL	NULL	690	750	0	2017-08-23 11:21:41.520	1	3
5	5	Tom 300	NULL	NULL	55	65	0	2017-08-23 11:21:41.520	1	3
6	6	Lacoste 066	NULL	NULL	99	105	0	2017-08-23 11:21:41.520	1	5

Bước 3: Thêm ràng buộc PriceInput buộc phải >= 0:

```

-- Thêm ràng buộc giá nhập phải lớn hơn 0
ALTER TABLE PRODUCT
ADD CONSTRAINT ck_PriceInput CHECK(PriceInput >= 0)

-- Câu lệnh sẽ lỗi vì đã có ràng buộc CHECK

```

```
INSERT INTO PRODUCT(ProductName, PriceInput, Created,
CatelogId) VALUES
('Thượng Đình Bata', -69, GETDATE(), 4)
```

100 % <

Messages

Msg 547, Level 16, State 0, Line 54
The INSERT statement conflicted with the CHECK constraint "CK__PRODUCT__Created__276EDEB3".
The conflict occurred in database "DatabaseTH4", table "dbo.PRODUCT", column 'Created'.
The statement has been terminated.

Do đã thêm ràng buộc CHECK nên lệnh thêm dữ liệu với giá sản phẩm nhỏ hơn 0 bị từ chối.

Bước 4: Thêm ràng buộc PriceOut buộc phải > PriceInput:

```
-- Thêm ràng buộc giá nhập phải lớn hơn giá bán
ALTER TABLE PRODUCT
ADD CONSTRAINT ck_PriceOutPut CHECK (PriceOutput >
PriceInput)

-- Câu lệnh sẽ lỗi vì đã có ràng buộc CHECK
INSERT INTO PRODUCT(ProductName, PriceInput,
PriceOutput,Created, CatelogId) VALUES
('Thượng Đình Bata', 45,35, GETDATE(), 4)

-- Câu lệnh sẽ lỗi vì đã có ràng buộc CHECK
INSERT INTO PRODUCT(ProductName, PriceInput, PriceOutput,Created, CatelogId) VALUES
('Thượng Đình Bata X1', 45,35, GETDATE(), 4)
```

0 % <

Messages

Msg 547, Level 16, State 0, Line 61
The INSERT statement conflicted with the CHECK constraint "ck_PriceOutPut".
The conflict occurred in database "DatabaseTH4", table "dbo.PRODUCT".
The statement has been terminated.

Do đã thêm ràng buộc CHECK nên lệnh thêm dữ liệu với giá bán phải lớn hơn giá nhập

Bài thực hành 3: Trong bảng dữ liệu **PRODUCT**, sau một thời gian dữ liệu đã nhập thì nghiệp vụ yêu cầu chuyển đổi kiểu dữ liệu giá sang money, và thêm 3 cột với đặc tả như sau:

Tên cột Field	Kiểu dữ liệu DataType	Null	Ràng buộc ContraINT	Mô tả Content
Bar_code	uniqueidentifier	No	Duy nhất	Mã quản lý sản phẩm phục vụ mục đích quản lý.
PriceUnit	NVARCHAR(16)	Yes		Đơn vị tính cho sản phẩm. VD: cái, chiếc, lít, thùng, tấn....
Img_thumb	NVARCHAR(512)	Yes		Liên kết link ảnh thu nhỏ của sản phẩm.

Bước 1: Sửa kiểu dữ liệu cột giá_nhap từ FLOAT thành money:

```
-- Sửa kiểu dữ liệu cột giá nhập, giá bán thành money
ALTER TABLE PRODUCT
ALTER COLUMN PriceInput money

ALTER TABLE PRODUCT
ALTER COLUMN PriceOutput money
```

...

Và gặp lỗi

Messages

```
Msg 5074, Level 16, State 1, Line 67
The object 'DF__PRODUCT__PriceIn__24927208' is dependent on column 'PriceInput'.
Msg 5074, Level 16, State 1, Line 67
The object 'ck_PriceInput' is dependent on column 'PriceInput'.
Msg 5074, Level 16, State 1, Line 67
The object 'ck_PriceOutPut' is dependent on column 'PriceInput'.
Msg 4922, Level 16, State 9, Line 67
ALTER TABLE ALTER COLUMN PriceInput failed because one or more objects access this column.
```

Nguyên nhân là do cột PriceInput, PriceOutput đã có một số ràng buộc, để thực hiện thay đổi kiểu dữ liệu thì cần phải xóa bỏ ràng buộc > thay đổi kiểu dữ liệu > thêm lại ràng buộc.

Bước 2: Sửa bảng PRODUCT xóa ràng buộc và sửa đổi kiểu dữ liệu:

```
-- Xóa ràng buộc giá trị mặc định
ALTER TABLE PRODUCT
DROP CONSTRAINT DF__PRODUCT__PriceIn__24927208
GO
-- xóa ràng buộc giá trị mặc định của giá bán
ALTER TABLE PRODUCT
```

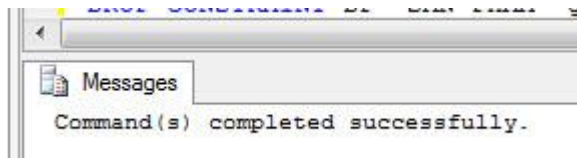
```
DROP CONSTRAINT DF__PRODUCT__PriceOu__25869641
GO

-- Xóa ràng buộc kiểm tra giá nhập lớn hơn 0
ALTER TABLE PRODUCT
DROP CONSTRAINT ck_PriceInput
GO

-- Xóa ràng buộc kiểm tra giá bán lớn hơn giá nhập
ALTER TABLE PRODUCT
DROP CONSTRAINT ck_PriceOutput
GO
```

Bước 3: Sửa bảng SAN PHAM sửa đổi kiểu dữ liệu:

```
-- Sửa kiểu dữ liệu cột giá nhập thành money
ALTER TABLE PRODUCT
ALTER COLUMN PriceInput money
```

**Thêm lại ràng buộc**

```
-- Thêm lại ràng buộc thêm giá trị mặc định
ALTER TABLE PRODUCT
ADD CONSTRAINT df_gia_PriceInput DEFAULT(0) FOR PriceInput
GO

-- Thêm lại ràng buộc kiểm tra lớn hơn 0
ALTER TABLE PRODUCT
ADD CONSTRAINT ck_PriceInput CHECK(PriceInput >= 0)
GO
```

Bước 4: Sửa bảng PRODUCT thêm 3 cột như đặc tả:

```
-- Thêm 3 cột như mô tả
ALTER TABLE PRODUCT
ADD Bar_code uniqueidentifier
GO
ALTER TABLE PRODUCT
ADD PriceUnit NVARCHAR(16)
GO
ALTER TABLE PRODUCT
ADD Img_thumb NVARCHAR(512)
GO
```

Bước 5: Xem bảng PRODUCT sau khi thêm 3 cột như đặc tả:


```
-- Thêm dữ liệu mẫu
INSERT INTO PRODUCT(ProductName, PriceInput, PriceOutput, Created,
CatalogId, PriceUnit, Img_thumb, Bar_code) VALUES
('Conserve Chuck 111', 600, 650, GETDATE(),
1, N'chiếc', '/giaynam/cc12016sdsdf.png', NEWID()),
('Gucci SNN99', 120, 150, '2016-03-12 14:32:15',
2, N'hộp', '/giaynu/cc12016sdsdf.png', NEWID()),
('Adidas 123', 1450, 1500, GETDATE(),
1, N'cái', '/giaycao/cc12016sdsdf.png', NEWID())
GO
SELECT * FROM PRODUCT
GO
```

ProductId	ProductName	Content	ContentDetail	PriceInput	PriceOutput	Views	Created	Status	CatalogId	Bar_code
1	Conserve Chuck 1	NULL	NULL	600.00	650.00	0	2017-08-23 11:21:41.520	1	1	NULL
2	Gucci SNN69	NULL	NULL	120.00	150.00	0	2017-08-23 11:21:41.520	1	2	NULL
3	Adidas	NULL	NULL	1450.00	1600.00	0	2017-08-23 11:21:41.520	1	1	NULL
4	Nike MG21	NULL	NULL	690.00	750.00	0	2017-08-23 11:21:41.520	1	3	NULL
5	Tom 300	NULL	NULL	55.00	65.00	0	2017-08-23 11:21:41.520	1	3	NULL
6	Lacoste 066	NULL	NULL	99.00	105.00	0	2017-08-23 11:21:41.520	1	5	NULL
7	Conserve Chuck 111	NULL	NULL	600.00	650.00	0	2017-08-23 11:58:47.637	1	1	C139BFF6-F1F3-4117-8B90-
8	Gucci SNN99	NULL	NULL	120.00	150.00	0	2017-08-23 11:58:47.637	1	2	422AD760-2126-4456-9E03-
9	Adidas 123	NULL	NULL	1450.00	1500.00	0	2017-08-23 11:58:47.637	1	1	95F38719-BABE-4CEE-A6A0-

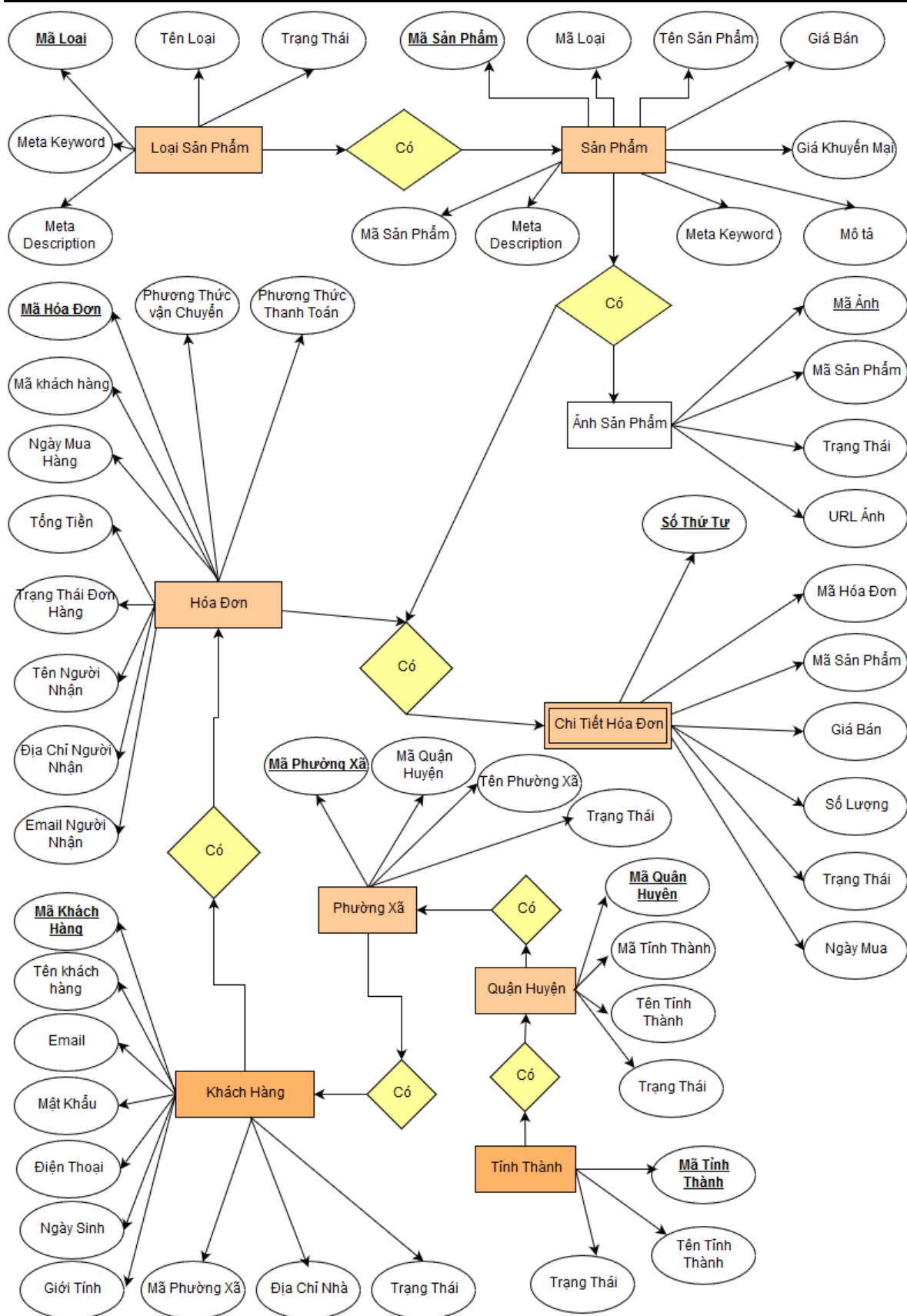
Phần II - Bài tập tổng hợp

❖ Xây dựng cơ sở dữ liệu cơ bản cho trang web bán hàng online (tiếp tục phát triển từ bài lab 03)

- Xác định các đối tượng cần quản lý trong một trang bán hàng trực tuyến bao gồm danh sách loại sản phẩm, sản phẩm thuộc loại nào, bán sản phẩm cho người mua có thông tin cá nhân ra sao, mỗi lần người mua hàng sẽ có đơn đặt hàng gì. mỗi đơn đặt hàng sẽ có danh sách sản phẩm trong đơn hàng đó là gì
- Liệt kê chi tiết các đối tượng như hình

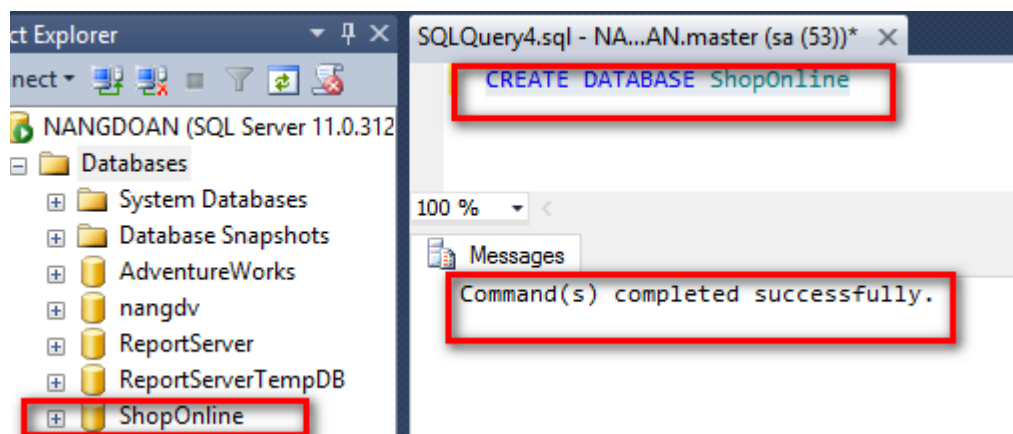
Loại Sản Phẩm	Sản Phẩm	Ảnh Sản Phẩm	Hóa Đơn
Mã Loại	Mã Sản Phẩm	Mã Ảnh	Mã Hóa Đơn
Tên Loại	Mã Loại	Mã Sản Phẩm	Mã Khách hàng
Trạng Thái	Tên Sản Phẩm	Url Ảnh	Ngày Mua hàng
meta keyword	Giá Bán	Trạng thái	Tổng tiền
meta Description	Giá Khuyến Mại		Phương Thức vận Chuyển
	Thông Tin Mô tả		Trạng Thái Hóa Đơn
	meta keyword		Tên Người Nhận
	meta Description		Địa Chỉ Người Nhận
	Trạng thái		Điện Thoại Người Nhận
			Email Người Nhận
Khách hàng	Tỉnh Thành	Quận Huyện	Chi Tiết Hóa Đơn
Mã khách hàng	Mã tỉnh	Mã Quận Huyện	Số Thứ Tự
Tên khách hàng	Tên Tỉnh	Mã Tỉnh Thành	Mã Hóa Đơn
Email	Trạng thái	Tên Quận Huyện	Mã Sản Phẩm
Điện thoại		Trạng Thái	Giá Sản Phẩm
Mật khẩu	Phường Xã		Số Lượng
Ngày Sinh	Mã Phường Xã		Trạng Thái
Giới tính	Mã Quận Huyện		Ngày Mua
Mã Phường Xã	Tên Phường xã		
Địa chỉ nhà	Trạng Thái		
Trạng thái			

➤ Thể hiện các thực thể và các mối quan hệ



➤ Viết script để tạo cơ sở dữ liệu

```
CREATE DATABASE ShopOnline
GO
```



- Tạo ra các bảng theo sơ đồ thực thể cho database

```

USE ShopOnline
GO

CREATE TABLE loaiSanPham(
    Ma_loai INT IDENTITY(1,1) PRIMARY KEY,
    Ten_loai NVARCHAR(50) NOT NULL,
    Trang_thai BIT,
    Meta_keyword NVARCHAR(128),
    Meta_description NTEXT
)
GO

CREATE TABLE SanPham(
    Ma_SP INT IDENTITY(1,1) PRIMARY KEY,
    Ma_loai INT NOT NULL,
    Ten_sp NVARCHAR(128) NOT NULL,
    Gia_ban FLOAT,
    Gia_KM FLOAT,
    Meta_keyword NVARCHAR(128),
    Meta_description NTEXT,
    Mo_ta NTEXT
)
GO

CREATE TABLE HoaDon(
    Ma_HD INT IDENTITY(1,1) PRIMARY KEY,
    Ma_KH INT NOT NULL,
    Ngay_mua_hang DATE NOT NULL,
    Tong_tien FLOAT NOT NULL,
    Trang_thai_don_hang SMALLINT,
    Ten_nguoi_nhan NVARCHAR(100) NOT NULL,
    Dia_chi_nguoi_nhan NVARCHAR(250) NOT NULL,
    Email_nguoi_nhan VARCHAR(50) NOT NULL,

```

```
        Phuong_thuc_thanh_toan SMALLINT NOT NULL,
        Phuong_thuc_van_chuyen SMALLINT NOT NULL
    )
GO
CREATE TABLE ChiTietHoaDon (
    So_TT INT IDENTITY(1,1) PRIMARY KEY,
    Ma_HD INT NOT NULL,
    Ma_SP INT NOT NULL,
    Gia_ban FLOAT NOT NULL,
    So_luong INT NOT NULL,
    Trang_thai BIT,
    Ngay_mua DATE
)
GO
CREATE TABLE KhachHang (
    Ma_KH INT IDENTITY(1,1) PRIMARY KEY,
    Ten_KH NVARCHAR(128) NOT NULL,
    Email_KH VARCHAR(50) NOT NULL,
    Mat_khau VARCHAR(50) NOT NULL,
    Dien_thoai VARCHAR(11) NOT NULL,
    Ngay_sinh DATE,
    Gioi_tinh BIT,
    Ma_phuong_xa INT,
    Dia_chi_nha NVARCHAR(250),
    Trang_thai BIT,
)
GO
CREATE TABLE TinhThanh (
    Ma_tinh_thanh INT IDENTITY(1,1) PRIMARY KEY,
    Trang_thai BIT,
    Ten_tinh_thanh NVARCHAR(50) NOT NULL
)
GO
CREATE TABLE PhuongXa (
    Ma_phuong_xa INT IDENTITY(1,1) PRIMARY KEY,
    Ma_quan_huyen INT NOT NULL,
    Ten_phuong_xa NVARCHAR(50) NOT NULL,
    Trang_thai BIT
)
GO
CREATE TABLE QuanHuyen (
    Ma_quan_huyen INT IDENTITY(1,1) PRIMARY KEY,
    Ma_tinh_thanh INT NOT NULL,
    Ten_quan_huyen NVARCHAR(50) NOT NULL,
    Trang_thai BIT
)
GO
```

➤ Thực hiện thêm thông tin vào các bảng đã được tạo

```

INSERT INTO
loaiSanPham(Ten_loai,Trang_thai,Meta_keyword,Meta_descripti
on) VALUES
    ('VGA', '0', 'VGA ASUS', N'VGA hot nhất'),
    ('Mainboard', '1', 'Mainboard ASUS', N'MainBoard hot
nhất'),
    (N'Quạt tản nhiệt', '1', N'Quạt Cooler Master', N'Quạt
tản nhiệt tốt nhất')

GO

INSERT INTO
SanPham(Ma_loai,Ten_sp,Gia_ban,Gia_KM,Meta_keyword,Meta_des
cription,Mô_ta ) VALUES
    (1, 'Vga Asus 601',232.00,215.00,'Asus 601',N'Vga Asus
tốt nhất',N'Bảo hành 3 năm'),
    (2, 'Mainboard Asus 901',899.00,800.00,'Asus
901',N'Mainboard Asus tốt nhất',N'Bảo hành 1 năm'),
    (2, 'Mainboard Asus 100',990.00,818.00,'Asus
100',N'Mainboard Asus tốt nhất',N'Bảo hành 1 năm')

GO

INSERT INTO KháchHang(Ten_KH, Email_KH,
Mat_khau,Dien_thoai, Ngay_sinh, Giới_tinh, Ma_phuong_xa,
Dia_chi_nha, Trang_thai) VALUES
    (N'Hoàng Anh
Tú','anhtu@gmail','123456','0909879877','1990-09-
09','1',3,'3/HQV quận Cầu Giấy HN','0'),
    (N'Mai Hoa','maihoa@gmail','123','0901239877','1987-
09-01','0',4,'8/TTT quận Hoàn Kiếm HNN','1'),
    (N'Nguyễn Lâm','Nlam@gmail','abc','0912379877','1966-
12-12','1',5,'12/Nghĩa Tân quận Cầu Giấy HN','1')

GO

INSERT INTO HoaDon(Ma_KH,Ngay_mua_hang,Tong_tien,
Trang_thai_don_hang, Ten_nguoi_nhan, Dia_chi_nguoi_nhan,
Email_nguoi_nhan,Phuong_thuc_thanh_toan,Phuong_thuc_van_chu
yen) VALUES
    (1, '2016-03-10',800.00,10,N'Hoàng Anh Tú',N'3/HQV quận
Cầu Giấy HN','anhtu@gmail','3','1'),
    (2, '2016-03-09',1200.00,20,N'Mai Hoa',N'8/TTT quận
Hoàn Kiếm HN','maihoa@gmail','4','0'),
    (3, '2016-03-08',989.00,10,N'Nguyễn Lâm',N'12/Nghĩa Tân
quận Cầu Giấy HN','Nlam@gmail','5','1')
GO

```

```
INSERT INTO ChiTietHoaDon(Ma_HD, Ma_SP, Gia_ban, So_luong,
Trang_thai, Ngay_mua ) VALUES
(1,1,800.00,3,'1','2016-03-10'),
(1,2,800.00,1,'1','2016-03-10'),
(1,3,1600.00,2,'1','2016-03-10'),
(2,1,215.00,1,'0','2016-03-08'),
(3,1,215.00,1,'0','2016-03-08')
```

GO

```
INSERT INTO TinhThanh(Trang_thai ,Ten_tinh_thanh) VALUES
('0',N'Hà Nội'),
('0',N'Hà Nam'),
('0',N'Vĩnh Phúc'),
('1',N'Tiền Giang'),
('1',N'Bến Tre')
```

GO

```
INSERT INTO QuanHuyen(Ma_tinh_thanh, Ten_quan_huyen,
Trang_thai ) VALUES
(1,N'Cầu Giấy','0'),
(1,N'Tây Hồ','0'),
(1,N'Hoàn Kiếm','1'),
(1,N'Ba Đình','0')
```

GO

```
INSERT INTO Phường(Xa(Ma_quan_huyen , Ten_phuong_xa ,
Trang_thai ) VALUES
(1,N'Dịch Vọng','0'),
(1,N'Quan Hoa','0'),
(1,N'Nghĩa Tân','0'),
(3,N'Cửa Nam','1')
```

GO

Phần III - Bài tập tự làm

Bài số 1:

Sử dụng câu lệnh T-SQL thực hiện các yêu cầu sau:

Hãy tạo 3 bảng với cấu trúc như sau:

1. [TaxType]

[CodeID] **INT IDENTITY**(1,1) NOT NULL,
[TaxName] **NVARCHAR**(128) NULL,
[Discount] **FLOAT** NULL

2. [AptechEmployees]

[EmployeeID] [INT] NOT NULL,

[CodeID] [INT] NOT NULL,
[EmployeeName] [NVARCHAR](256) NULL,
[DateOfBirth] [DATETIME] NULL

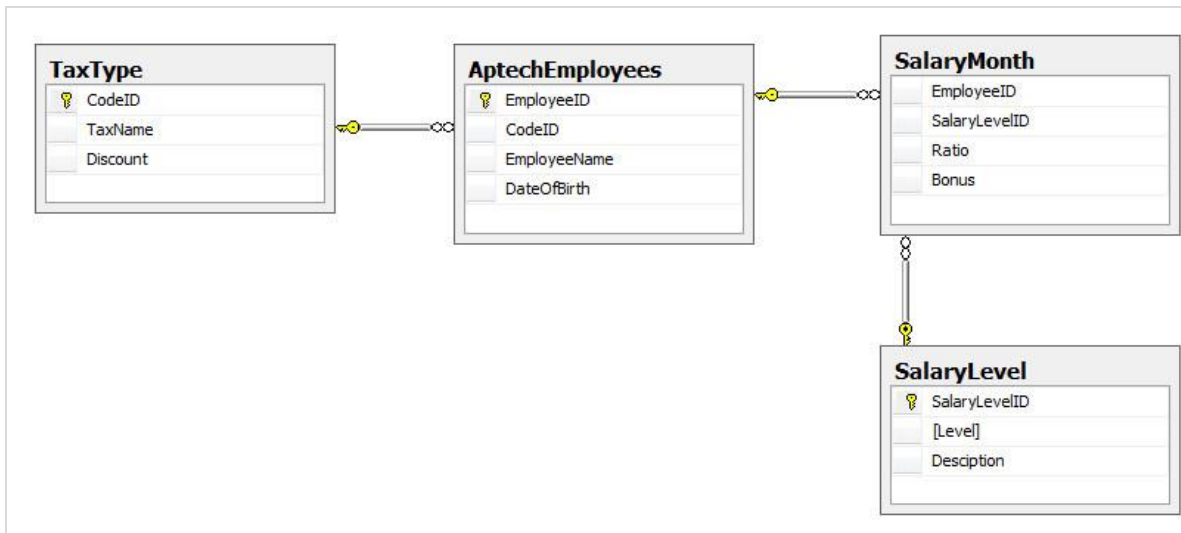
3. [SalaryLevel]

[SalaryLevelID] [INT] NOT NULL,
[Level] [char(1)] NOT NULL,
[Description] [NVARCHAR(128)]

4. [SalaryMonth]

[EmployeeID] [INT] NOT NULL,
[SalaryLevelID] [INT] NOT NULL,
[Ratio] [FLOAT] NOT NULL,
[Bonus] [FLOAT] NULL

1. Hãy tạo **Primary Key** trên cột “**CodeID**” cho bảng [TaxType] .
2. Hãy tạo **Primary Key** trên cột “**EmployeeID**” cho bảng AptechEmployees.
3. Hãy tạo **Primary Key** trên cột “**SalaryLevelID**” cho bảng SalaryLevel.
4. Hãy tạo **Primary Key** trên các cột “**EmployeeID, SalaryLevelID**” cho bảng SalaryMonth.
5. Hãy tạo khóa ngoài **Foreign Key Constraint (CodeID)** cho bảng AptechEmployees tham chiếu đến bảng [TaxType].
6. Hãy tạo khóa ngoài **Foreign Key Constraint (EmployeeID, SalaryLevelID)** cho bảng SalaryMonth tham chiếu đến bảng AptechEmployees và SalaryLevel.



Bài số 2:

Sử dụng câu lệnh T-SQL thực hiện các yêu cầu sau:

1. Hãy tạo một **Check ConstraiNT** trên cột **DateOfBirth** cho bảng **AptechEmployees** nằm trong khoảng từ „01-01-1970“ and „01-01-1985“.
2. Hãy tạo một **Unique ConstraiNT** trên cột **TaxName** cho bảng [TaxType].
3. Hãy tạo một **Default ConstraiNT** trên cột **Ratio** cho bảng **SalaryMonth** với giá trị default bằng 1.

Hãy chèn dữ liệu cho 4 bảng sau:

	SalaryLevelID	Level	Description
1	1	A	Trưởng Phòng
2	2	B	Trưởng Huyện

	CodeID	TaxName	Discount
1	1	Thuế thân	16.8
2	2	Thuế thu nhập cá nhân	40
3	3	Thuế đê con trai	80.5

	EmployeeID	CodeID	EmployeeName	DateOfBirth
1	1	1	Tr?nh Van Chung	1979-03-08 00:00:00.000
2	2	2	L?i Đ?c Chung	1980-11-26 00:00:00.000

Results		Messages		
	EmployeeID	SalaryLevelID	Ratio	Bonus
1	1	2	8.5	888
2	2	1	6.9	999