

## Bài 1: Cài đặt SQL Server 2005

Đọc file: *Huong dan cai dat SQL Server 2005.doc*

## Bài 2: Tạo và xóa database từ giao diện đồ họa

## Bài 3: Xem thông tin về file \*.mdf và \*.ldf

## Bài 4: Backup và Restore 1 database

## Bài 5: Attach và detach 1 database

## Bài 6: Tạo user và login

### Câu lệnh: CREATE, INSERT, UPDATE,DELETE

#### Bài 1:

Tạo CSDL quản lý điểm học viên. CSDL có tên là **DIEM** và gồm 3 bảng

#### Bảng Student:

*Student(RN,Name,Age,Gender)*

#### Bảng Subject(môn học):

*Subject(sID, sName)*

#### Bảng StudentSubject

*StudentSubject(RN,SID,Mark,Date).*

##### 1. Tạo các bảng trên

##### a. Đặt khóa chính cho các bảng:

Student      khóa là RN    //Roll Number: mã sinh viên

Subject      khóa là sID    //Subject Identify

StudentSubject    khóa là (RN,sID)

##### b. Đặt ràng buộc để trường Mark chỉ nhận các giá trị trong khoảng [0->10]

##### c. Đặt ràng buộc khóa chính khóa ngoại giữa 2 bảng Student(RN) và StudentSubject(RN).

##### 2. Nhập dữ liệu vào bảng để thể hiện thông tin sau

- a. Học viên Mỹ Linh đạt điểm 8 môn SQL vào ngày 7/28/2005
  - b. Học viên Đàm Vĩnh Hưng đạt điểm 3 môn LGC vào ngày 7/29/2005
  - c. Học viên Kim Tử Long đạt điểm 9 môn HTML vào ngày 7/31/2005
  - d. Học viên Tài Linh đạt điểm 5 môn SQL vào ngày 7/30/2005
  - e. Học viên Mỹ Lệ đạt điểm 10 môn CF vào ngày 7/19/2005
  - f. Học viên Ngọc Oanh đạt điểm 9 môn SQL vào ngày 7/25/2005
3. Cập nhật giới tính cho các học viên
    - a. Mỹ Linh, Tài Linh, Mỹ Lệ là 0
    - b. Kim Tử Long là 1
    - c. Ngọc Oanh vẫn để Null
  4. Nhập thêm các môn học sau vào bảng Subject
    - a. Core Java, VB.Net
  5. Hiển thị tất cả các môn học mà chưa có học viên nào nhận điểm
  6. Hiển thị danh sách tất cả các môn học, với điểm cao nhất mà học viên đạt được với môn học đó, môn nào chưa có điểm thì để trống (Null) phần điểm
  7. Hiển thị tên môn học mà có nhiều hơn một điểm

## LỜI GIẢI

```
create database DIEM
use DIEM
--CAU 1
create table Student(rn int,name varchar(25),age int,gender bit)
create table Subject(sID int,sName varchar(10))
create table StudentSubject(rn int,sID int,Mark int,Date datetime)
go
--CAU 1a
alter table Student
alter column rn int not null
go
alter table Student
add constraint PK1 primary key(rn)
go
--
alter table Subject
alter column sID int not null
go
alter table Subject
add constraint PK2 primary key(sID)
go
--
alter table StudentSubject
```

```
alter column sID int not null
go
alter table StudentSubject
alter column RN int not null
go
alter table StudentSubject
add constraint PK3 primary key (rn,sID)
go

--CAU 1b
alter table StudentSubject
add constraint check1 check (mark>=0 and mark <=10)
go

--CAU 1c
alter table studentSubject
add constraint FK1 foreign key (RN) references Student (RN)
go

alter table studentSubject
add constraint FK2 foreign key (sID) references Subject (sID)

--CAU 2
--CAU 2a
Insert student (rn,name) values (1,'My Linh')
Insert subject values (1,'SQL')
go
Insert StudentSubject values (1,1,8,'7/28/2005')

--CAU 2b
Insert student (rn,name) values (2,'Dam Vinh Hung')
Insert subject values (2,'LGC')
go
Insert StudentSubject values (2,2,3,'7/29/2005')

-- CAU 2c
Insert student (rn,name) values (3,'Kim Tu Long')
Insert subject values (3,'HTML')
go
```

```
Insert StudentSubject values (3,3,9, '7/31/2005')
```

```
-- CAU 2d
```

```
Insert student (rn,name) values (4, 'Tai Linh')
```

```
go
```

```
Insert StudentSubject values (4,1,5, '7/30/2005')
```

```
-- CAU 2e
```

```
Insert student (rn,name) values (5, 'My Le')
```

```
Insert subject values (4, 'CF')
```

```
go
```

```
Insert StudentSubject values (5,4,10, '7/19/2005')
```

```
-- CAU 2f
```

```
Insert student (rn,name) values (6, 'Ngoc Oanh')
```

```
go
```

```
Insert StudentSubject values (6,1,9, '7/25/2005')
```

```
go
```

```
-- CAU 3
```

```
-- CAU 3a
```

```
update student
```

```
set gender=0
```

```
where name in ('My Linh', 'Tai Linh', 'My Le')
```

```
go
```

```
-- CAU 3b
```

```
update student
```

```
set gender=1
```

```
where name in ('Kim Tu Long')
```

```
go
```

```
-- CAU 3c
```

```
update student
```

```
set gender=NULL
```

```
where name in ('Ngoc Oanh')
```

```
go
```

```
select * from student
```

```
select * from subject
```

```
select * from studentsubject
```

```
-- CAU 4
-- CAU 4a
Insert subject values (5, 'Core Java')
Insert subject values (6, 'VB.NET')
go

-- CAU 5
select sName, mark from subject s left join studentsubject ss on
s.sID=ss.sID
where ss.sID is null

-- CAU 6
select sName, max(mark) from subject s left join studentsubject ss on
s.sID=ss.sID
group by sName

-- CAU 7
select sName, count(*) from subject s left join studentsubject ss on
s.sID=ss.sID
group by sName
having count(*) > 1

--select sName from subject
--where sID in (select sID from studentsubject ss where subject.sID=ss.sID
group by sID having count(*) > 1)
```

## Câu lệnh: SELECT

### Phần I: QUẢN LÝ SINH VIÊN MƯỢN SÁCH

Đề bài

Tạo cơ sở dữ liệu quản lý sách (QuanLySach) với 2 bảng dữ liệu được cho như sau:

Student
---------

RN	Name	Age
1	Pham Ngoc Hung	28
2	Tran Tuan Duong	22
3	Nguyen Tai Tue	24
4	Tran Tuan Duong	23
5	Pham Van Minh	22
6	Nguyen Thanh Son	27

Và bảng Book chứa thông tin mượn sách của sinh viên:

Book	
RN	BookTitle
1	SQL
2	EPC
1	HTML
1	Java
2	DW
3	Javascript

### Câu 1

Hãy sử dụng các câu lệnh truy vấn T-SQL, thực hiện các việc sau:

1. Hiển thị danh sách học viên đã mượn sách cũng như những sách học viên đó đã mượn.
2. Truy vấn để tìm xem ai đã mượn sách EPC?
3. Kiểm tra xem sinh viên 'Tran Tuan Duong' đã mượn những quyển sách nào?
4. Hiển thị danh sách các sinh viên có cùng tuổi với nhau.
5. Hiển thị danh sách sinh viên chưa từng mượn sách.
6. Sửa tên sinh viên có RN=4 thành 'Tran Tuan Duong B'

7. Tăng tuổi của mỗi sinh viên thêm 1 tuổi.
8. Cập nhật thông tin sinh viên 'Pham Ngoc Hung' trả sách 'EPC'.

#### Câu 2

1. Hiển thị tuổi lớn nhất của các sinh viên.
2. Hiển thị tên học viên có tuổi lớn nhất (học viên đầu tiên tìm thấy).
3. Hiển thị danh sách sinh viên có tuổi bé nhất.
4. Hiển thị danh sách các lứa tuổi học viên và tổng số học viên của từng lứa tuổi
5. Lứa tuổi nào có nhiều học viên nhất?
6. Học viên nào mượn nhiều sách nhất?
7. Hiển thị danh sách học viên và số lượng sách đã mượn của từng học viên theo 2 trường hợp:

- Chỉ hiện những người mượn sách
- Hiện cả những người mượn sách và không mượn sách

#### LỜI GIẢI

```
-- Tao CSDL QuanLySach

create database QuanLySach;
use QuanLySach;
go;

-- Tao bang Student
create table Student
(
    RN int identity(1,1) PRIMARY KEY,
    [name] nvarchar(32)
)

-- Them mot cot vao bang
```



```
alter table Student
    add age int NOT NULL DEFAULT (18)
go

-- Hien thi de xem da tao xong chua
select * from Student;

-- Tao bang Book
create table Book
(
    BookId int identity(100,2) PRIMARY KEY,
    BookTitle nvarchar(256) NOT NULL,
    RN int -- foreign key references Student(RN)
)

-- them rang buoc vao bang da co san
alter table Book
    add constraint fk foreign key (RN) references
Student(RN)
go

-- Them du lieu vao bang Student
insert into Student (Age, [Name]) values (30, 'Pham Ngoc
Hung');
insert into Student (Age, [Name]) values (22, 'Tran Tuan
Duong');
insert into Student (Age, [Name]) values (24, 'Nguyen Tai
Tue');
insert into Student (Age, [Name]) values (23, 'Tran Tuan
Duong');
```

```
insert into Student (Age, [Name]) values (22, 'Pham Van
Minh');
insert into Student (Age, [Name]) values (27, 'Nguyen
Thanh Son');

-- Liet ke du lieu vua nhap
select * from Student;

-- Nhap du lieu Book
insert into Book ([RN], [BookTitle]) values (1, 'SQL');
insert into Book ([RN], [BookTitle]) values (2, 'EPC');
insert into Book ([RN], [BookTitle]) values (1, 'HTML');
insert into Book ([RN], [BookTitle]) values (1, 'Java');
insert into Book ([RN], [BookTitle]) values (2, 'DW');
insert into Book ([RN], [BookTitle]) values (3,
'Javascript');

-- hien thi du lieu vua nhap vao bang Book
select * from Book;

-- cau 1.1
---- Cach 1
select * from Student
select DISTINCT RN from Book;
select RN, [Name], Age from Student where RN in (select
DISTINCT RN from Book);
---- Cach 2
----- Noi 2 bang ma khong co dieu kien gi ca!!!
select Student.RN, Student.[Name], Student.Age, Book.RN,
Book.BookTitle
    from Student, Book
```

```
----- Noi 2 bang, nhuwng chi chon Student.ID trung voi
Book.ID (chi chon
----- nhung sinh vien muon sach)
select Student.RN, Student.[Name], Student.Age, Book.RN,
Book.BookTitle
    from Student, Book where Student.RN = Book.RN;
----- Loai bo truong Book.RN vi thua
select Student.RN, Student.[Name], Student.Age,
Book.BookTitle
    from Student, Book where Student.RN = Book.RN;

-- Cau 1.2
---- De dang tim duoc RN cua Sinh vien muon sach EPC
select Book.RN from Book where Book.BookTitle = 'EPC' --
ta thu duoc RN = 2
select RN, [Name] from Student where RN = 2

---- Hop nhat 2 thao tac de chi lam 1 lan.
----- Cach 1: dung truy van con
select RN, [Name] from Student
    where RN in (select Book.RN from Book where
Book.BookTitle = 'EPC')
----- Cach 2: noi bang voi nhau
select Student.RN, Student.[Name], Student.Age, Book.RN,
Book.BookTitle
    from Student, Book
select Student.RN, [Name] from Student, Book
    where (Student.RN = Book.RN) and (Book.BookTitle =
'EPC')

-- Cau 1.3
```

```
---- đi tu việc nói 2 bảng
select Student.RN, Student.[Name], Student.Age, Book.RN,
Book.BookTitle
    from Student, Book
---- Chọn ra những bộ thỏa mãn điều kiện
select Student.RN, Student.[Name], Student.Age, Book.RN,
Book.BookTitle
    from Student, Book
    where (Student.RN = Book.RN) and (Student.[Name] =
'Tran Tuan Duong');
---- Loại bỏ các trường không cần thiết
select Student.[Name], Book.BookTitle
    from Student, Book
    where (Student.RN = Book.RN) and (Student.[Name] =
'Tran Tuan Duong');
---- Câu hỏi: Tran Tuan Duong muốn bao nhiêu quyển sách?
select COUNT(*) AS [Số quyển sách muốn]
    from Student, Book
    where (Student.RN = Book.RN) and (Student.[Name] =
'Tran Tuan Duong');

-- Câu 1.4
---- Cách 1
Select * from Student s1, Student s2;
Select s1.RN, s1.[Name], s1.Age from Student s1, Student
s2
    where (s1.[RN] != s2.[RN]) and (s1.Age = s2.Age)
---- Cách 2
----- Dấu tiên là cho ra những tuổi có người trùng nhau
Select Age from Student Group by Age Having Count (RN) >=
2;
```

----- Loc ra ten nhung nguoi co tuoi nam trong nhom nay

```
Select RN, [Name], Age from Student
      where Age in (Select Age from Student Group by Age
Having Count (RN) >= 2)
```

-- Cau 1.6

```
Update Student set [Name] = [Name] + ' B' where [RN] = 4;
Select * From Student;
```

-- Cau 1.7

```
update Student set Age = Age + 1;
```

-- Cau 1.8: Tran Tuan Duong tra sach EPC, xoa ban ghi

---- trong Bang Book ma co RN=2 va BookTitle = EPC

```
delete from Book
where (Book.RN in (select Student.RN from Student where
Student.[Name] = 'Tran Tuan Duong'))
      and (BookTitle = 'EPC')
```

```
Select * from Book;
```

-- Cau them

```
Select [Name] from Student where [Name] like 'Pham%[gh]';
```

-- Chinh sua RN cua Student se lam cho RN trong Book thay doi

```
update Student set RN= RN + 100 where RN = 1;
```

-- Cau 2.1 Tuoì lon nhat cua Student

-- Cau 2.2

```
Select [Name] from Student
      where Age = (Select MAX(Age) FROM Student);
```

```
Select [Name] from Student
      where Age = (Select MIN(Age) FROM Student);
```

-- cau 2.4

```
select Age, count(RN) from student Group By Age;
```

-- cau 2.5 ???

```
select Age, count(RN) from student Group By Age;
```

-- Cau 2.6 Hoc vien nao muon nhieu sach nhat???

```
DECLARE @TEMP
```

```
Select MAX([SoLuong]) From (Select COUNT(*) As [SoLuong]
From Book Group By RN);
```

## Phần II: QUẢN LÝ BÁN HÀNG

### 1. Sử dụng các câu lệnh T-SQL

Tạo CSDL để quản lý sản phẩm đã bán của một cửa hàng, cơ sở dữ liệu có tên là QLHANG, tạo 3 bảng để lưu trữ các thông tin sau:

**Bảng Customers:** lưu trữ thông tin về khách hàng: *customerid, customername, tei*.

**Bảng Items:** lưu trữ thông tin về hàng hóa: *itemid, itemname, quantity, price*

**Bảng CustomerItem:** lưu trữ thông tin về những sản phẩm đã được bán mua: *customerid, itemid, quantity*.

### 2. Nhập các thông tin sau vào CSDL(Sử dụng câu lệnh T-SQL)

#### a. Nhập dữ liệu bảng Customers như sau

customerid	customername	tel
1	dinh truong son	1234567
2	mai thanh minh	1357777
3	nguyen hong ha	2468888

**b. Nhập dữ liệu bảng Items như sau**

itemid	itemname	quantity	price
1	tu lanh	5	3500
2	tivi	2	3000
3	dieu hoa	1	8000
4	quat da	5	1700
5	may giat	3	5000

**c. Nhập thông tin về hàng đã bán được vào bảng CustomerItem như sau**

customerid	itemid	quantity
1	1	4
1	5	1
2	2	1
3	3	1
3	1	1

- Đặt các ràng buộc khóa chính, khóa ngoại sao cho người dùng không thể xóa được thông tin về khách hàng nếu như khách hàng đã mua một sản phẩm bất kỳ
- Hiển thị tổng số tiền mà cửa hàng đã thu được từ các khách hàng trên
- Hiển thị tên, số tiền đã mua của người khách hàng đã trả tiền cho cửa hàng nhiều nhất
- Kiểm tra xem người khách có số điện thoại 2468888 có mua mặt hàng Tủ lạnh không? Nếu có mua hiện ra dòng chữ “Có mua”, ngược lại “Không mua”
- Tính tổng số hàng hóa và tổng tiền còn lại trong kho(Số còn lại bằng tổng số trừ đi số đã bán)
- Hiển thị danh sách 3 mặt hàng bán chạy nhất (số lượng bán nhiều nhất)
- Hiển thị tất cả các mặt hàng mà chưa bán được một cái nào.
- Hiển thị danh sách những người mua nhiều hơn một mặt hàng
- Hiển thị danh sách những người mua hàng có số lượng nhiều hơn một cái

## LỜI GIẢI

### --CAU 1

```
create table customers(  
customerid int not null,  
customername varchar(50),  
tel varchar(10))  
go
```

--tao bang items

```
create table items(  
itemid int not null,  
itemname varchar(50),  
quantity int,  
price float)  
go
```

--tao bang customeritem

```
create table customeritem(  
customerid int,  
itemid int,  
quantity int)  
go
```

--CAU 2a

```
insert customers values(1,'dinh truong son','1234567')  
insert customers values(2,'mai thanh minh','1357777')  
insert customers values(3,'nguyen hong ha','2468888')  
go
```

--CAU 2b

```
insert items values(1,'tu lanh',5,3500)  
insert items values(2,'tivi',2,3000)  
insert items values(3,'dieu hoa',1,8000)
```



```
insert items values (4, 'quat da', 5, 1700)
insert items values (5, 'may giat', 3, 5000)
go
--CAU 2c
```

```
insert customeritem values (1, 1, 4)
insert customeritem values (1, 5, 1)
insert customeritem values (2, 2, 1)
insert customeritem values (3, 3, 1)
insert customeritem values (3, 1, 1)
```

go

--CAU 3

```
alter table customers
add constraint pk_customers primary key (customerid)
go
alter table customeritem
add constraint fk_customeritem foreign key (customerid)
references customers (customerid)
```

go

--CAU 4

```
select sum(customeritem.quantity*price)
from customeritem inner join items on
customeritem.itemid = items.itemid
```

go

--CAU 5

```
select top 1 customername, sum(ci.quantity*price)
from customers c inner join customeritem ci on
c.customerid = ci.customerid
inner join items i on
```

```
i.itemid =ci.itemid
group by customername
order by sum(ci.quantity*price) desc
go
--CAU 6

if exists(select itemname from items inner join
customeritem on items.itemid = customeritem.itemid
inner join customers on customers.customerid =
customeritem.customerid
where itemname = 'tu lanh' and tel = '2468888')
select 'co mua'
else
select 'khong mua'
go
--CAU 7

select sum(items.quantity-customeritem.quantity) as
conlai,
sum(customeritem.quantity*price) as tienlai
from items inner join customeritem on
items.itemid = customeritem.itemid
go
--CAU 8

select top 3 itemname as top3 from
items inner join customeritem on
items.itemid = customeritem.itemid
group by itemname
order by sum(customeritem.quantity) desc
go
--CAU 9

select itemname as chuabanduoc from
items where itemid not in
```

```
(select itemid from customeritem)
go
--CAU 10
select customername
from customers
where customerid in (select customerid from
customeritem
group by customerid
having count(itemid)>1)
go
--CAU 11
select customername from customers
where customerid in(select customerid from
customeritem
group by customerid
having sum(quantity) > 1)
go
```

### Phần III: Tự làm

#### Câu 1

Xây dựng cơ sở dữ liệu có tên là LopHoc, có một bảng dữ liệu tên là DSSV (danh sách sinh viên), trong bảng này có các trường dữ liệu như sau:

Tên dữ liệu	Tên trường	Loại dữ liệu
Mã sinh viên	masv	Chuỗi ký tự có đúng 6 ký tự.
Tên sinh viên	tensv	Chuỗi ký tự không vượt quá 32 ký tự.
Tuổi	tuoi	Số nguyên
Số điện thoại	sodt	Chuỗi ký tự không quá 15 ký tự.
Địa chỉ	diachi	Chuỗi ký tự không vượt quá 256 ký tự.

Sau khi tạo xong, thực hiện các yêu cầu sau:

- Dùng lệnh để chèn danh sách sinh viên của lớp bạn đang học vào bảng DSSV trên.
- Sau khi tạo xong, dùng lệnh SELECT để hiển thị danh sách sinh viên đã chèn vào.
- Dùng lệnh DELETE để xóa toàn bộ danh sách.
- Xóa Bảng DSSV khỏi cơ sở dữ liệu LopHoc.
- Xóa Cơ sở dữ liệu LopHoc.

### Hướng dẫn

\* Để tạo mới cơ sở dữ liệu, ta nhập lệnh sau vào cửa sổ Query:

```
CREATE DATABASE LopHoc;
```

Tô đậm lệnh trên và bấm F5 để chạy.

\* Để tạo một bảng mới, ta chuyển vào làm việc trong cơ sở dữ liệu này (bằng lệnh USE), sau đó tạo bảng bằng các lệnh sau:

```
USE LopHoc;
```

```
GO;
```

```
CREATE TABLE DSSV (
```

```
    masv nvarchar(6) PRIMARY KEY,
```

```
    tensv nvarchar(32),
```

```
    tuoi int,
```

```
    sodt nvarchar(15),
```

```
    diachi nvarchar(256)
```

```
)
```

Nhập các lệnh trên vào cửa sổ Query, tô đậm vùng lệnh và bấm F5 để chạy.

\* Để nhập dữ liệu cho bảng, ta sử dụng câu lệnh INSERT như sau:

```
INSERT INTO DSSV (masv, tensv, tuoi, sodt, diachi)
```

```
VALUES ('A012345', 'Pham Ngoc Hung', 32, '0982872507', 'Hoang Quoc Viet, Cau Giay, Ha Noi');
```

Câu lệnh trên chèn thêm một dòng (bản ghi) dữ liệu vào bảng DSSV. Lưu ý dữ liệu bên trong mệnh đề VALUES phải đúng kiểu và đúng trật tự với danh sách các trường liệt kê ngay sau tên DSSV (tức là 'A012345' là kiểu chuỗi, tương ứng và phù hợp với trường masv có kiểu nvarchar(6), 32 là số nguyên tương ứng với trường tuoi thuộc kiểu int,...).

Chúng ta có thể thêm các bản ghi khác bằng các lệnh INSERT, với lưu ý là giá trị của trường masv trên các bản ghi là không được trùng lặp, vì trường này là trường khóa chính (PRIMARY KEY).

\* Để hiển thị dữ liệu trong bảng DSSV, ta sử dụng lệnh sau:

```
SELECT * FROM DSSV;
```

Nếu chỉ muốn hiển thị tất cả các bản ghi của bảng DSSV trên chỉ với dữ liệu nằm trong các trường masv, tensv, sodt, ta dùng lệnh sau:

```
SELECT masv, tensv, sodt FROM DSSV;
```

Khi đó, kết quả chỉ chứa dữ liệu của tất cả các bản ghi nhưng chỉ với 3 trường masv, tensv và sodt.

\* Để xóa tất cả dữ liệu trong bảng, ta dùng lệnh sau:

```
DELETE FROM DSSV;
```

Sau khi thực hiện lệnh này, tất cả các bản ghi hiện có trong bảng DSSV sẽ bị xóa.

\* Để xóa bảng DSSV khi không còn sử dụng bảng này nữa, ta sử dụng lệnh sau:

```
DROP TABLE DSSV;
```

\* Để xóa cơ sở dữ liệu LopHoc, trước hết ta cần chuyển sang làm việc ở một cơ sở dữ liệu khác rồi mới gọi lệnh xóa cơ sở dữ liệu này (tức là ta chỉ có thể "đập đổ" một ngôi nhà sau khi đã ra khỏi ngôi nhà đó).

```
USE master; -- chuyển sang csdl master
```

```
DROP DATABASE LopHoc;
```

## Câu 2

Xây dựng cơ sở dữ liệu kho sách có tên là KhoSach để chứa sách. Cơ sở dữ liệu này có một bảng có tên là Sach, trong đó có các trường dữ liệu sau:

Tên dữ liệu	Tên trường	Dạng dữ liệu	Kiểu dữ liệu	Ràng buộc
-------------	------------	--------------	--------------	-----------

Mã sách	masach (trường khóa)	Chuỗi ký tự không vượt quá 20 ký tự	nvarchar(20)	Trường khóa
Tiêu đề sách	tieude	Chuỗi ký tự không vượt quá 100 ký tự	nvarchar(100)	Không được bỏ trống (NOT NULL)
Tác giả chính	tacgiachinh	Chuỗi ký tự không vượt quá 32 ký tự	nvarchar(32)	Không được bỏ trống
Đồng tác giả	dongtacgia	Chuỗi ký tự không vượt quá 256 ký tự, chứa danh sách các đồng tác giả, phân cách bằng dấu phẩy.	nvarchar(256)	Được phép bỏ trống
Ngày xuất bản	ngayxuatban	Trường kiểu ngày tháng (DateTime)	datetime	Không được bỏ trống
Nhà xuất bản	nhaxuatban	Chuỗi ký tự không vượt quá 256 ký tự.	nvarchar(256)	Được phép bỏ trống
Giá bán (ghi trên bìa)	dongia	Số thực	float	Không được phép bỏ trống
Số lượng sách	soluong	Số nguyên	int	Không được phép bỏ trống

Sau khi đã có bảng Sach, sử dụng lệnh INSERT để chèn sách vào bảng trên:

<b>masach</b>	<b>tieude</b>	<b>tacgiachinh</b>	<b>dongtacgia</b>	<b>ngayxuatban</b>	<b>nhaxuatban</b>	<b>dongia</b>	<b>soluong</b>
B00001	Phương trình vi phân và tích phân	Phạm Huy Điền	Phạm Cảnh Dương, Đinh Thế Lục	23/11/2001	NXB Khoa học kỹ thuật	80000	1200
B00002	Giải tích hàm	Phạm Kỳ Anh		12/12/2005	NXB Đại học quốc gia Hà	75000	900

					Nội		
B00101	Đất rừng phương nam	Đoàn Giỏi		10/03/2003	NXB văn học	20000	2000
B00102	Ăn mày dĩ vãng	Chu Lai		13/11/2005	NXB văn học	54000	1500
B00104	Một thời để nhớ	Phạm Ngọc Hùng		12/12/2001	NXB văn học	12000	4000

Sau khi tạo và chèn dữ liệu vào bảng Sách trong CSDL KhoSach xong, thực hiện các việc sau:

- Sử dụng lệnh SELECT để liệt kê toàn bộ danh sách Sách có trong cơ sở dữ liệu.
- Sử dụng lệnh SELECT với điều kiện WHERE để lọc ra tất cả sách được xuất bản trước ngày 26/03/2001
- Sử dụng lệnh SELECT với điều kiện WHERE và hàm COUNT để đếm số sách có giá nhỏ hơn 60000.
- Sử dụng lệnh DELETE với điều kiện WHERE để xóa sách của tác giả 'Phạm Ngọc Hùng'.

### Bài 1: Tạo CSDL “QLThuVien”

Tạo lập bảng BANDOC, SACH, MUONTRA có cấu trúc như sau:

BANDOC

STT	MaBD char(2)	Ten varchar(35)	Tuoi(int)	Gioitinh (bit)	Dienthoai number
1	--	--	--	1	--
2	--	--	--	0	--

SACH

STT	MaSach char(4)	Tensach (varchar)	Soluong (number)	MaNXB char(2)	MaLoai char(2)
1	--	--	--	--	TH

2					
---	--	--	--	--	--

MUONTRA

STT	MaBD	MaSach	Ngaymuon (Date)	NgayTra (Date)	Soluong (number)
1	--	--	--	--	--
2					

## Tạo lập các ràng buộc

1. Tạo ràng buộc khoá chính trên các bảng BANDOC, SACH(MaBD, Masach)
2. Tạo ràng buộc khoá ngoại trên bảng MUONTRA
3. Tạo ràng buộc indenty(1,1) trên trường STT của các bảng
4. Cập nhật lại số lượng đầu sách trong bảng SACH theo điều kiện dưới đây:  
Nếu mã loại (MaLoai='TH') thì số lượng sách = 500