

--Tuan 03
--BaiTap2:

--câu 1:Dùng T-SQL tạo CSDL Movies

CREATE DATABASE Movies
ON PRIMARY

(
 NAME = 'Movies_data',
 FILENAME = 'C:\Movies\Movies_data.mdf',
 SIZE = 25MB,
 MAXSIZE = 40MB,
 FILEGROWTH =1MB
)

LOG ON

(
 NAME = 'Movies_log',
 FILENAME = 'C:\Movies\Movies_log.ldf',
 SIZE = 6MB,
 MAXSIZE =8MB,
 FILEGROWTH = 1MB
)
GO

USE Movies
Go

sp_helpdb Movies
GO

-- cau 2. Thực hiện và kiểm tra kết quả các yêu cầu sau
--Thêm một Data file thứ 2 có:

```
USE Movies
ALTER DATABASE Movies ADD FILE (
    NAME = 'Movies_data2',
    FILENAME = 'C:\Movies\Movies_data2.ndf',
    SIZE = 10MB
)
GO
```

```
--Lần lượt cấu hình CSDL Movies với chế độ
ALTER DATABASE Movies
SET SINGLE_USER
GO
```

```
ALTER DATABASE Movies
SET RESTRICTED_USER
GO
```

```
ALTER DATABASE Movies
SET MULTI_USER
GO
```

```
--Tăng kích cỡ của data file thứ 2 từ 10 MB lên 15 MB
ALTER DATABASE Movies MODIFY FILE(
    NAME = 'Movies_data2',
    SIZE = 15MB
)
GO
sp_helpDb Movies
GO
```

```
--Cấu hình CSDL về chế độ tự động SHRINK
ALTER DATABASE Movies SET AUTO_SHRINK ON;
```

GO

```
--Xoá CSDL Movies  
DROP DATABASE Movies  
GO
```

```
--CAU 3. Mở tập tin Movies.sql, thực hiện:  
--Bổ sung thêm câu lệnh tạo một filegroup tên là DataGroup  
USE Movies  
ALTER DATABASE Movies ADD FILEGROUP DataGroup
```

```
--Hiệu chỉnh maxsize của tập tin transaction log thành 10 MB  
ALTER DATABASE Movies MODIFY FILE(  
    NAME='Movies_log',  
    MAXSIZE = 10MB  
)  
GO
```

```
--Size của tập tin data file thứ 2 thành 20 MB  
ALTER DATABASE Movies MODIFY FILE(  
    NAME = 'Movies_data2',  
    SIZE = 20MB  
)  
GO
```

```
--Thêm data file thứ 3 thuộc filegroup DataGroup (thông số tùy ý)  
USE Movies  
ALTER DATABASE Movies ADD FILE (  
    NAME = 'Movies_data3',  
    SIZE = 20MB,  
    MAXSIZE = 40MB,  
    FILEGROWTH = 1MB) TO FILEGROUP DataGroup  
GO
```

```
sp_helpDb Movies
GO
```

```
-- CAU 5.Thực hiện định nghĩa các user-defined datatype sau trong CSDL Movies.
```

```
USE Movies
```

```
EXEC sp_addtype Movies_num, 'int', 'NOT NULL';
```

```
EXEC sp_addtype Category_num, 'int', 'NOT NULL';
```

```
EXEC sp_addtype Cust_num, 'int', 'NOT NULL';
```

```
EXEC sp_addtype Invoice_num, 'int', 'NOT NULL';
```

```
--xem lai
```

```
SELECT domain_name, data_type, character_maximum_length
FROM information_schema.domains
```

```
--6. Thực hiện tạo các bảng vào CSDL Movies
```

```
USE Movies
```

```
CREATE TABLE Customer(
    Cust_num cust_num IDENTITY(300,1) NOT NULL,
    Lname varchar(20) NOT NULL,
    Fname varchar(20) NOT NULL,
    Address1 varchar(30),
    Address2 varchar(20),
    City varchar(20),
    State char(2),
    Zip char(10),
    Phone varchar(10) NOT NULL,
    Join_date Smalldatetime NOT NULL
);
```

```
CREATE TABLE Category(  
    Category_num category_num IDENTITY(1,1) NOT NULL,  
    Description varchar(20) NOT NULL  
);
```

```
CREATE TABLE Movies(  
    Movies_num movies_num NOT NULL,  
    Title Cust_num not null,  
    Category_Num category_num not null,  
    Date_purch Smalldatetime,  
    Rental_price Int,  
    Rating Char(5)  
);
```

```
create table Rental(  
    Invoice_num Invoice_num Not null,  
    Cust_num Cust_num Not null,  
    Rental_date Smalldatetime Not null,  
    Due_date Smalldatetime not null  
);
```

```
CREATE TABLE Rental_Detail(  
    Invoice_num Invoice_num Not null,  
    Line_num Int Not null,  
    Rental_price Smallmoney Not null,  
    Movie_num Movies_num Not null,  
);
```

```
exec sp_help  
go
```

--9. Thực hiện định nghĩa các khoá chính (Primary Key Constraint)

use Movies

```
alter table Movies add constraint PK_movie PRIMARY KEY (Movie_num)
```

Go

```
alter table Customer add constraint PK_customer PRIMARY KEY (Cust_num)
```

Go

```
alter table Category add constraint PK_category PRIMARY KEY (Category_num)
```

go

```
alter table Rental add constraint PK_rental PRIMARY KEY(Invoice_num)
```

go

```
exec sp_helpconstraint Movies;
```

```
exec sp_helpconstraint Customer;
```

```
exec sp_helpconstraint Category;
```

```
exec sp_helpconstraint Rental;
```

Go

--10. Thực hiện định nghĩa các khoá ngoại (Foreign Key Constraint)

use Movies

```
alter table Movies add constraint FK_movie foreign key(Category_num) references Category(category_num);
```

go

```
alter table Rental add constraint FK_rental foreign key(Cust_num) references Customer(Cust_num);
```

go

```
alter table Rental_Detail add constraint FK_detail_invoice foreign key(Invoice_num) references  
Rental(Invoice_num) on delete cascade;
```

go

```
alter table Rental_Detail add constraint FK_detail_movie foreign key(Movie_num) references
Movies(Movies_num);
go
```

--kiem tra lai

```
EXEC sp_helpconstraint 'Movies';
EXEC sp_helpconstraint 'Rental';
EXEC sp_helpconstraint 'Rental_Detail';
GO
```

-- CAU 12. Thực hiện định nghĩa các giá trị mặc định (Default Constraint)

```
alter table Movies add constraint DK_movie_date_purch default getdate() for Date_purch;
go
```

```
alter table Customer add constraint DK_customer_join_date default getdate() for join_date;
go
```

```
alter table Rental add constraint DK_rental_rental_date default getdate() for Rental_date;
go
```

```
alter table Rental add constraint DK_rental_due_date default dateadd(day,2,getdate() ) for Due_date;
go
```

--kiemtra lai

```
EXEC sp_helpconstraint 'Movies';
EXEC sp_helpconstraint 'Customer';
EXEC sp_helpconstraint 'Rental';
GO
```

--CAU 13. Thực hiện định nghĩa các ràng buộc miền giá trị (Check Constraint)

```
use Movies
```

```
alter table Movies add constraint CK_movie check (Rating IN ('G', 'PG', 'R', 'NC17', 'NR'));  
go
```

```
alter table Rental add constraint CK_Due_date check (Due_date >= Rental_date);  
go
```

```
EXEC sp_helpconstraint 'Movies';  
EXEC sp_helpconstraint 'Rental';
```