Phiếu 2 ( làm bài 1,2)( vì một bài làm theo QA, một bài làm theo EM)

use master

go

create database QLB\_Hang

on primary(

name='qlbanhang\_dat', filename='D:\qlbanhang.mdf',

size=5MB, maxsize=50MB, filegrowth=10%

)

log on(

name='qlbanhang\_log', filename='D:\qlbanhang.ldf',

size=1MB, maxsize=5MB, filegrowth=10%

)

go

use QLB\_Hang

go

create table HangSX(

MaHangSX Nvarchar(12) not null primary key,

TenHang Nvarchar(25) not null,

SoDT Nvarchar(12),

DiaChi Nvarchar(30),

Email Nvarchar(40),

)

create table SanPham(

MaSP Nvarchar(12) not null primary key,

MaHangSX Nvarchar(12),

TenSP Nvarchar(25) not null,

SoLuong int,

MauSac Nvarchar(12),

GiaBan float,

DonViTinh Nvarchar(12),

MoTa text,

constraint FK\_SanPham\_HangSX foreign key(MaHangSX)

references HangSX(MaHangSX)

)

create table NhanVien(

MaNV Nvarchar(12) not null primary key,

TenNV Nvarchar(30),

GioiTinh Nvarchar(25),

SoDT Nvarchar(12),

DiaChi Nvarchar(30),

Email Nvarchar(40),

TenPhong Nvarchar(20),

)

create table PNhap(

SoHDN Nvarchar(12) not null primary key,

MaNV Nvarchar(12),

Dates Datetime,

constraint FK\_PNhap\_NhanVien foreign key(MaNV)

references NhanVien(MaNV),

)

create table Nhap(

SoHDN Nvarchar(12) not null,

MaSP Nvarchar(12) not null,

SoLuongN int,

DonGiaN float,

constraint PK\_Nhap primary key(SoHDN,MaSP),

constraint fk\_Nhap\_SoHDN foreign key(SoHDN)

references PNhap(SoHDN),

constraint fk\_Nhap\_SanPham foreign key(MaSP)

references SanPham(MaSP)

)

create table PXuat(

SoHDX Nvarchar(12) not null primary key,

MaNV Nvarchar(12),

NgayXuat datetime,

constraint FK\_PXuat\_NhanVien foreign key(MaNV)

references NhanVien(MaNV)

)

create table Xuat(

SoHDX Nvarchar(12) not null,

MaSP Nvarchar(12) not null,

SoLuongN int,

constraint PK\_Xuat primary key(SoHDX,MaSP),

constraint fk\_Xuat\_SoHDX foreign key(SoHDX)

references PXuat(SoHDX),

constraint fk\_Xuat\_SanPham foreign key(MaSP)

references SanPham(MaSP)

)

Nhập vào thì nhập bằng bảng rồi

Phiếu 3 bài 1:

--- tao database

use master

go

create database DeptEmp

on primary(

name='DeptEmp', filename='D:\DeptEmp.mdf',

size=5MB, maxsize=50MB, filegrowth=2MB

)

log on(

name='DeptEmp', filename='D:\DeptEmp.ldf',

size=5MB, maxsize=50MB, filegrowth=10%

)

go

use DeptEmp

go

--- tao bang

create table Department(

DepartmentNo integer not null primary key,

DepartmentName char(25) not null,

Location char(25) not null,

)

create table Employee(

EmpNo integer not null primary key,

Fname varchar(15) not null,

Lname varchar(15) not null,

Job varchar(25) not null,

HireDate datetime not null,

Salary numeric not null,

Commision numeric,

DepartmentNo integer,

constraint fk\_Employee\_Department foreign key(DepartmentNo) references Department(DepartmentNo )

on delete cascade on update cascade

)

----------

insert into Department values(10, 'Accounting', 'Melbourne')

insert into Department values(20, 'Research', 'Adealide')

insert into Department values(30,'Sales' ,'Sydney')

insert into Department values(40, 'Operations', 'Perth')

--------

insert into Employee(EmpNo,Fname, Lname, Job, HireDate, Salary, DepartmentNo) values(1, 'John', 'Smith', 'Clerk', N'17-Dec-1980' ,800, 20)

insert into Employee values(2,'Peter','Allen','Salesman', '20-Feb-1981', 100, 300 ,30)

insert into Employee values(3, 'Kate', 'Ward', 'Salesman', '22-Feb-1981', 1250 ,500, 30)

insert into Employee(EmpNo,Fname, Lname, Job, HireDate, Salary, DepartmentNo) values(4,'Jack' ,'Jones', 'Manager', '02-Apr-1981' ,2975, 20)

insert into Employee values(5,'Joe', 'Martin', 'Salesman', '28-Sep-1981', 1250, 1400, 30)

---------hien bang

select \* from Department

select \* from Employee

1. Hiển thị nội dung bảng Department

select \* from Department;

2. Hiển thị nội dung bảng Employee

select \* from Employee;

3. Hiển thị employee number, employee first name và employee last name từ bảng Employee mà employee first name có tên là ‘Kate’

select EmpNo, Fname, Lname

from Employee

where Fname='Kate'

4. Hiển thị ghép 2 trường Fname và Lname thành Full Name, Salary,

10%Salary (tăng 10% so với lương ban đầu).

select Fname+' '+Lname as'Ten',Salary, Salary\*0.1 as 'Tangluong'

from Employee

5. Hiển thị Fname, Lname, HireDate cho tất cả các Employee có HireDate là

năm 1981 và sắp xếp theo thứ tự tăng dần của Lname.

select Fname, Lname, HireDate

from Employee

where year(HireDate)='1981'

order by Lname asc

6. Hiển thị trung bình(average), lớn nhất (max) và nhỏ nhất(min) của

lương(salary) cho từng phòng ban trong bảng Employee.

select AVG( Salary) as'TrungBinh', max( Salary) as 'CaoNhat', min( Salary) as 'ThapNhat'

from Employee

group by DepartmentNo

7. Hiển thị DepartmentNo và số người có trong từng phòng ban có trong bảng

Employee.

select DepartmentNo, count(\*) as 'SoNguoi'

from Employee

group by DepartmentNo

8. Hiển thị DepartmentNo, DepartmentName, FullName (Fname và Lname),

Job, Salary trong bảng Department và bảng Employee.

select Department.DepartmentNo , Department.DepartmentName, Employee.Fname +'' +Employee.Lname as 'FullName',Employee.Job, Employee.Salary

from Department inner join Employee on Department.DepartmentNo=Employee.DepartmentNo

9. Hiển thị DepartmentNo, DepartmentName, Location và số người có trong

từng phòng ban của bảng Department và bảng Employee.

select Department.DepartmentNo,Department.DepartmentName,Department.Location, count(Department.DepartmentNo) as 'SoNguoi'

from Department inner join Employee on Department.DepartmentNo=Employee.DepartmentNo

group by Department.DepartmentNo,Department.DepartmentName,Department.Location

10. Hiển thị tất cả DepartmentNo, DepartmentName, Location và số người có

trong từng phòng ban của bảng Department và bảng Employee

select Department.DepartmentNo,Department.DepartmentName,Department.Location, count(Department.DepartmentNo) as 'SoNguoi'

from Department inner join Employee on Department.DepartmentNo=Employee.DepartmentNo

group by Department.DepartmentNo,Department.DepartmentName,Department.Location

Phiếu 3, bài 2

use master

go

create database MarkManagement

on primary(

name='MarkManagement', filename='D:\MarkManagement.mdf',

size=5MB, maxsize=50MB, filegrowth=2MB

)

log on(

name='MarkManagement', filename='D:\MarkManagement.ldf',

size=5MB, maxsize=50MB, filegrowth=10%

)

go

use MarkManagement

go

create table Students(

StudentID Nvarchar(12) not null primary key,

StudentName Nvarchar(25) not null,

DateofBirth Datetime not null,

Email Nvarchar(40),

Phone Nvarchar(12),

Class Nvarchar(10),

)

create table Subjects(

SubjectID Nvarchar(10) not null primary key,

SubjectName Nvarchar(25) not null,

)

create table Mark(

StudentID Nvarchar(12) not null,

SubjectID Nvarchar(10) not null,

Theory Tinyint,

Practical Tinyint,

Dates Datetime,

constraint PK\_Mark primary key(StudentID, SubjectID),

constraint fk\_Mark\_Students foreign key(StudentID) references Students(StudentID)

on delete cascade on update cascade,

constraint fk\_Mark\_Subjects foreign key(SubjectID) references Subjects(SubjectID)

on delete cascade on update cascade,

)

-----

insert into Students values('AV0807005', N'Mai Trung Hiếu', N'10/11/1989', N'trunghieu@yahoo.com', N'0904115116', N'AV1')

insert into Students values('AV0807006', N'Nguyễn Quý Hùng', N'12/2/1988', N'quyhung@yahoo.com', N'090455516', N'AV2')

insert into Students values('AV0807007', N'Đỗ Đắc Huỳnh', N'1/2/1990', N'dachuynh@yahoo.com', N'0904187878', N'AV2')

insert into Students values('AV0807008', N'An Đăng Khuê', N'6/3/1986', N'dangkhue@yahoo.com', N'0904115446', N'AV1')

insert into Students values('AV0807009', N'Nguyễn Thị Tuyết Lan', N'7/12/1989', N'tuyetlan@yahoo.com', N'0904115116', N'AV2')

insert into Students(StudentID, StudentName, DateofBirth, Email, Class) values('AV0807011', N'Đinh Phụng Long', N'12/2/1989', N'phunglong@yahoo.com', N'AV1')

insert into Students(StudentID, StudentName, DateofBirth, Email, Class) values('AV0807012', N'Nguyễn Tuấn Nam', N'3/2/1989', N'tuannam@yahoo.com', N'AV1')

----

insert into Subjects values('S001', N'SQL')

insert into Subjects values('S002', N'Java Simplefiled')

insert into Subjects values('S003', N'Active Server Page')

-----

insert into Mark values('AV0807005', 'S001', 8, 25, N'6/5/2008')

insert into Mark values('AV0807006', 'S002', 16, 30, N'6/5/2008')

insert into Mark values('AV0807007', 'S001', 10, 25, N'6/5/2008')

insert into Mark values('AV0807009','S003', 7, 13,N'6/5/2008')

insert into Mark values('AV0807010','S003',9 ,16 , N'6/5/2008')

insert into Mark values('AV0807011', 'S002', 8, 30, N'6/5/2008')

insert into Mark values('AV0807012', 'S001',7 ,31 , N'6/5/2008')

insert into Mark values('AV0807005', 'S002',12 ,11, N'6/6/2008')

insert into Mark values('AV0807009', 'S003',11 ,20 , N'6/6/2008')

insert into Mark values('AV0807010', 'S001', 7, 6, N'6/6/2008')

1. Hiển thị nội dung bảng Students

select \* from Students

2. Hiển thị nội dung danh sách sinh viên lớp AV1

select \*

from Students

where Class='AV1'

3. Sử dụng lệnh UPDATE để chuyển sinh viên có mã AV0807012 sang lớp

AV2

UPDATE Students

SET Class='AV2'

where StudentID='AV0807012'

4. Tính tổng số sinh viên của từng lớp

select Class, count(\*) as 'SoSinhVien'

from Students

group by Class

5. Hiển thị danh sách sinh viên lớp AV2 được sắp xếp tăng dần theo

StudentName

select \*

from Students

where Class='AV1'

order by StudentName asc

6. Hiển thị danh sách sinh viên không đạt lý thuyết môn S001 (theory <10) thi

ngày 6/5/2008select Students.StudentID,Students.StudentNamefrom Students inner join Mark on Students.StudentID=Mark.StudentIDwhere Mark.Theory<10 and Mark.Dates='6/5/2008'7. Hiển thị tổng số sinh viên không đạt lý thuyết môn S001. (theory <10)select count(\*) as 'SoLuong'from Markwhere Theory<108. Hiển thị Danh sách sinh viên học lớp AV1 và sinh sau ngày 1/1/1980select StudentID, StudentName, DateofBirthfrom Studentswhere Class='AV1' and DateofBirth>'1/1/1980'9. Xoá sinh viên có mã AV0807011

delete

from Students

where StudentID='AV0807011'

10.Hiển thị danh sách sinh viên dự thi môn có mã S001 ngày 6/5/2008 bao gồm các trường sau : StudentID, StudentName, SubjectName, Theory, Practical,Dates

select Students.StudentID, Students.StudentName,Subjects.SubjectName, Mark.Theory, Mark.Practical, Mark.Dates

from ((Mark inner join Students on Students.StudentID=Mark.StudentID)

inner join Subjects on Mark.SubjectID=Subjects.SubjectID)

where Subjects.SubjectID='S001' and Mark.Dates='6/5/2008'