**Môn** : Cơ sở dữ liệu

**Họ tên**: Nguyễn Quỳnh Hương Quyên

**MSSV**: 22162036

**ÔN TẬP CHƯƠNG 4**

* **TẠO DATABASE VÀ NHẬP DỮ LIỆU BÀI 1**

CREATE DATABASE QLNV1

GO

USE QLNV1

GO

CREATE TABLE dbo.PhongBan(

MaPB int PRIMARY KEY,

TenPB char(10) NULL,

TrPhong char(4) NULL,

NgNhanChuc date Null

)

GO

CREATE TABLE dbo.NhanVien(

Manv char(4) PRIMARY KEY,

HoNV char(10) NULL,

Tenlot char(20) NULL,

tenNV char(10) NULL,

NgSinh date NULL,

Dchi char(50) NULL,

Phai char(3) NULL,

Luong int NULL,

MaNQL char(4) NULL,

Phong int

)

Go

Insert into Nhanvien Values ('nv01', 'Nguyen', 'Ngọc', 'Hoang', '1972-08-13', '111 Vo Van Ngan', 'Nam', 30000, 'NV02', 1)

Insert into Nhanvien Values ('nv02', 'Nguyen', 'Quang', 'Ngoc', '1975-08-23', '111 Nguyen Van Troi', 'Nam', 25000, 'NV15', 1)

Insert into Nhanvien Values ('nv03', 'Le', 'Thi', 'Ngọc', '1980-08-13', '111 Duong 3/2', 'Nu', 28000, 'NV15', 2)

Insert into Nhanvien Values ('nv04', 'Le', 'Ngọc', 'Hoang', '1982-08-18', '111 Le Van Duyet', 'Nam', 15000, 'NV03', 2)

Insert into Nhanvien Values ('nv05', 'Nguyen', 'Thi', 'Ngọc', '1992-08-13', '108 Vo Van Ngan', 'Nu', 28000, 'NV06', 3)

Insert into Nhanvien Values ('nv06', 'Truong', 'Anh', 'Kiet', '1960-12-13', '11 Hai Ba Trung', 'Nam', 30000, 'NV15', 3)

Insert into Nhanvien Values ('nv07', 'Le', 'Phung', 'Hieu', '1972-08-13', '11 Hau Giang', 'Nam', 10000, 'NV15', 4)

Insert into Nhanvien Values ('nv08', 'Nguyen', 'Bao', 'Hung', '1978-08-23', '10 Tran Hung Dao', 'Nam', 20000, 'NV15', 5)

Insert into Nhanvien Values ('nv09', 'Nguyen', 'Bao', 'Hoang', '1972-08-13', '111 Tran Hung Dao', 'Nam', 10000, 'NV15', 6)

Insert into Nhanvien Values ('nv10', 'Le', 'Hoang', 'Ngọc', '1972-08-13', '11 Lac Long Quan', 'Nam', 10000, 'NV15', 7)

Insert into Nhanvien Values ('nv11', 'Nguyen', 'Bao','Ngọc', '1972-08-13', '111 Minh Phung', 'Nu', 15000, 'NV15', 8)

Insert into Nhanvien Values ('nv12', 'Nguyen', 'Ngọc', 'A', '1982-08-13', '11 Nguyen Duy Trinh', 'Nam', 10000, 'NV11', 8)

Insert into Nhanvien Values ('nv13', 'Le', 'Quang', 'Hoang', '1979-08-23', '10 Ly Chinh Thang', 'Nam', 15000, 'NV15', 9)

Insert into Nhanvien Values ('nv14', 'Nguyen', 'Quang', 'Hoang', '1972-08-13', '11 Le Van Thinh', 'Nam', 10000, 'NV13', 9)

Insert into Nhanvien Values ('nv15', 'Dang', 'Tan', 'Dung', '1960-08-13', '23 Vo Van Ngan', 'Nam', 50000, NULL, 10)

Go

Insert into PhongBan Values (1, 'Nghien cuu', 'nv02', '2013-12-12');

Insert into PhongBan Values (2, 'To chuc', 'nv03', '2013-11-21');

Insert into PhongBan Values (3, 'Vat tu', 'nv06', '2013-1-25');

Insert into PhongBan Values (4, 'Dao tao', 'nv15', '2013-2-19');

Insert into PhongBan Values (5, 'Kinh doanh', 'nv15', '2013-08-18');

Insert into PhongBan Values (6, 'Doi ngoai', 'nv15', '2013-10-17');

Insert into PhongBan Values (7, 'Xay dung', 'nv15', '2013-02-20');

Insert into PhongBan Values (8, 'QL Du an', 'nv11', '2013-12-12');

Insert into PhongBan Values (9, 'Ke toan', 'nv13', '2013-12-12');

Insert into PhongBan Values (10, 'BGD', 'nv15', '2010-11-29');

Go

Alter table NHANVIEN WITH CHECK ADD FOREIGN KEY(MaNQL) REFERENCES NhanVien(Manv)

Go

ALTER TABLE NHANVIEN WITH CHECK ADD FOREIGN KEY(Phong) REFERENCES PhongBan(MaPB)

GO

Create table DIADIEM\_PHONG(

MaPB int references PhongBan(MaPB),

DiaDiem char(30),

Primary key (MaPB, DiaDiem) -- khai báo khóa chính gồm 2 cột

)

Go

Insert into DIADIEM\_PHONG Values(1, 'Tang tret - Khu A')

Insert into DIADIEM\_PHONG Values(2, 'Tang 1 - Khu A')

Insert into DIADIEM\_PHONG Values(3, 'Tang tret - Khu B')

Insert into DIADIEM\_PHONG Values(4, 'Tang tret - Khu C')

Insert into DIADIEM\_PHONG Values(5, 'Tang tret - Khu E')

Insert into DIADIEM\_PHONG Values(6, 'Tang tret - Khu Bát giác')

Insert into DIADIEM\_PHONG Values(7, 'Tang tret - Khu Trung tâm')

Insert into DIADIEM\_PHONG Values(8, 'Tang 1 - Khu Trung tâm')

Insert into DIADIEM\_PHONG Values(9, 'Tang 2 - Khu Trung tâm')

Insert into DIADIEM\_PHONG Values(10, 'Tang 3 - Khu Trung tâm')

Go

Create Table DUAN(

MaDA char(4) primary key,

TenDA Char(30),

DiaDiem char(30),

Phong int references PhongBan(MaPB)

)

Go

Insert into DUAN Values ('DA01', 'Cap thoat nuoc', 'Go Vap', 9)

Insert into DUAN Values ('DA02', 'Ban chung cu', 'Thu Thiem', 5)

Insert into DUAN Values ('DA03', 'Xay biet thu', 'Thu Duc', 7)

Insert into DUAN Values ('DA04', 'Lien ket dao tao', 'Go Vap', 4)

Insert into DUAN Values ('DA05', 'Cung cap vat tu', 'Quan 2', 3)

Insert into DUAN Values ('DA06', 'Huan luyen', 'Go Vap', 4)

Go

Create table PhanCong(

MaNV char(4) references NhanVien(maNV),

MaDA char(4) references DuAn(MaDA),

ThoiGian int,

Primary key (MaNV, MaDA)

)

Go

Insert into PhanCong Values ('nv01', 'DA01', 3)

Insert into PhanCong Values ('nv01', 'DA02', 3)

Insert into PhanCong Values ('nv02', 'DA01', 3)

Insert into PhanCong Values ('nv02', 'DA03', 3)

Insert into PhanCong Values ('nv03', 'DA04', 3)

Insert into PhanCong Values ('nv04', 'DA03', 3)

Insert into PhanCong Values ('nv09', 'DA04', 3)

Insert into PhanCong Values ('nv01', 'DA05', 3)

Insert into PhanCong Values ('nv05', 'DA05', 3)

Insert into PhanCong Values ('nv07', 'DA06', 3)

Insert into PhanCong Values ('nv08', 'DA06', 3)

Go

Create table ThanNhan(

MaNV char(4) references NhanVien(maNV),

TenTN char(20),

Phai char(3),

NgaySinh date,

QuanHe char(10),

Primary key (MaNV, TenTN)

)

Go

Insert into ThanNhan Values ('nv01', 'Nguyen Ngoc Huy', 'Nam', '2018-3-26','con')

Insert into ThanNhan Values ('nv01', 'Nguyen Ngoc Hoa', 'Nu', '2020-3-26','con')

Insert into ThanNhan Values ('nv02', 'Nguyen Ngoc Huy', 'Nam', '2018-3-26','con')

Insert into ThanNhan Values ('nv02', 'Nguyen Ngoc Hoa', 'Nu', '2020-3-26','con')

Insert into ThanNhan Values ('nv15', 'Dang Ngoc Hoa', 'Nu', '2020-3-26','con')

Insert into ThanNhan Values ('nv01', 'Nguyen Nguyen Hung', 'Nam', '2020-3-26','con')

* **CODE TẠO DATABASE VÀ NHẬP DỮ LIỆU BÀI 2**

CREATE DATABASE Bai2

GO

USE Bai2

GO

CREATE TABLE dbo.ĐầuSách(

MãSách nchar(10) PRIMARY KEY,

Tựa nvarchar(50) NULL,

MãNXB nchar(10) NULL,

)

GO

-- Tạo cấu trúc bảng CuonSach

create table dbo.CuốnSách(

MãSách nchar(10) NULL,

MãCuốn nchar(10) primary key,

VịTrí nvarchar(50) null,

)

GO

CREATE TABLE dbo.TácGiả(

MãSách nchar(10),

TênTácGiả nvarchar(50),

primary key (MãSách, TênTácGiả),

)

GO

CREATE TABLE dbo.NXB(

MãNXB nchar(10) primary key,

TênNXB nvarchar(50) null,

ĐịaChỉ nvarchar(50) null,

SốĐT nchar(10) null,

)

GO

CREATE TABLE dbo.Mượn(

MãCuốn nchar(10),

MãĐG nchar(10),

NgàyMượn date null,

NgàyTrả date null,

primary key (MãCuốn, MãĐG),

)

GO

CREATE TABLE dbo.ĐộcGiả(

MãĐG nchar(10) primary key,

TênĐG nvarchar(50) null,

ĐịaChỉ nvarchar(50) null,

SốĐT nchar(10) null,

)

GO

Alter table ĐầuSách WITH CHECK ADD FOREIGN KEY(MãNXB) REFERENCES NXB(MãNXB)

Go

ALTER TABLE CuốnSách WITH CHECK ADD FOREIGN KEY(MãSách) REFERENCES ĐầuSách(MãSách)

GO

ALTER TABLE TácGiả WITH CHECK ADD FOREIGN KEY(MãSách) REFERENCES ĐầuSách(MãSách)

GO

ALTER TABLE Mượn WITH CHECK ADD FOREIGN KEY(MãCuốn) REFERENCES CuốnSách(MãCuốn)

GO

ALTER TABLE Mượn WITH CHECK ADD FOREIGN KEY(MãĐG) REFERENCES ĐộcGiả(MãĐG)

GO

INSERT INTO dbo.NXB --(MaNXB, TenNXB, DiaChi, SoDT)

VALUES

('XB01', 'Addison Wesley', 'London', '0392768491'),

('XB02', 'Kim Đồng', 'Hà Nội', '0250238196'),

('XB03', 'Tuổi Trẻ', 'Tp.HCM', '0272175158'),

('XB04', 'Thanh Niên', 'Tp.HCM', '0390565658'),

('XB05', 'Văn Hóa', 'Hà Nội', '0378571434');

INSERT INTO dbo.ĐầuSách --(MaSach, Tua, MaNXB)

VALUES

('MS01', 'HARRY POTTER', 'XB01'),

('MS02', 'ĐẮC NHÂN TÂM', 'XB01'),

('MS03', 'DORAEMON', 'XB04'),

('MS04', 'CONAN', 'XB03'),

('MS05', 'ÔNG GIÀ VÀ BIỂN CẢ', 'XB01'),

('MS06', 'TUỔI TRẺ ĐÁNG BAO NHIÊU', 'XB05'),

('MS07', 'CHUÔNG NGUYỆN HỒN AI', 'XB01'),

('MS08', 'TIẾNG GIÀY ĐINH', 'XB02'),

('MS09', 'GIÃ TỪ VŨ KHÍ', 'XB05'),

('MS10', 'THẦN ĐỒNG ĐẤT VIỆT', 'XB03');

-- Insert data into TacGia table

INSERT INTO dbo.TácGiả --(MaSach, TenTacGia)

VALUES

('MS01', 'J.K.ROWLING'),

('MS02', 'DALE CARNERGIE'),

('MS03', 'HIROSHI FUJIMOTO'),

('MS04', 'GOSHO AOYAMA'),

('MS05', 'HEMINGWAY'),

('MS06', 'ROSIE NGUYỄN'),

('MS07', 'HEMINGWAY'),

('MS08', 'SƯU TẦM'),

('MS09', 'HEMINGWAY'),

('MS10', 'LÊ LINH');

-- Insert data into CuonSach table

INSERT INTO dbo.CuốnSách --(MaSach, MaCuon, Vitri)

VALUES

('MS01', 'MC01', 'A1-S2'),

('MS01', 'MC02', 'A1-S2'),

('MS02', 'MC03', 'A1-S1'),

('MS03', 'MC04', 'A3-S4'),

('MS03', 'MC05', 'A2-S3'),

('MS04', 'MC06', 'A2-S2'),

('MS05', 'MC07', 'B1-S2'),

('MS07', 'MC08', 'B1-S2'),

('MS09', 'MC09', 'A4-S3'),

('MS10', 'MC10', 'A1-S3'),

('MS10', 'MC11', 'A1-S4'),

('MS10', 'MC12', 'A1-S5'),

('MS06', 'MC13', 'A5-S2'),

('MS08', 'MC14', 'A5-S3'),

('MS06', 'MC15', 'B2-S3');

-- Insert data into DocGia table

INSERT INTO dbo.ĐộcGiả --(MaDG, TenDG, DiaChi, SoDT)

VALUES

('DG01', 'Nguyễn Văn A', 'Quận 9, Tp.HCM', '0317383676'),

('DG02', 'Trần Tiến Bình', 'Thủ Đức, Tp. HCM', '0383304051'),

('DG03', 'Hoàng Thị Chi', 'Cầu Giấy, Hà Nội', '0274644764'),

('DG04', 'Nguyễn Thanh Duy', 'Bà Rịa, Vũng Tàu', '0246188656'),

('DG05', 'Bùi Minh Hoàng', 'Long Khánh, Đồng Nai', '0394972721'),

('DG06', 'Lê Yến Nhi', 'Quận 1, Tp.HCM', '0306193947'),

('DG07', 'Lý Minh Phát', 'Củ Chi, Tp.HCM', '0256137338'),

('DG08', 'Lê Trúc Quỳnh', 'Dĩ An, Bình Dương', '0255375751');

-- Insert data into Muon table

INSERT INTO dbo.Mượn --(MaCuon, MaDG, NgayMuon, NgayTra)

VALUES

('MC01', 'DG01', '2024-10-01', '2024-10-15'),

('MC05', 'DG04', '2024-10-02', '2024-10-16'),

('MC06', 'DG05', '2024-10-03', '2024-10-17'),

('MC07', 'DG01', '2024-05-04', '2024-05-16'),

('MC08', 'DG01', '2024-07-02', '2024-07-10');

* **CODE TẠO DATABASE VÀ NHẬP DỮ LIỆU BÀI 3**

CREATE DATABASE Bai3

GO

USE Bai3

GO

CREATE TABLE dbo.COQUAN(

MSCOQUAN nchar(10) primary key,

TENCOQUAN nvarchar(50) null,

DIACHI nvarchar(50) null

)

GO

CREATE TABLE dbo.NV(

MSNV Nchar(10) primary key,

TEN nvarchar(50) null,

MSCOQUAN nchar(10) null,

CONGVIEC nvarchar(50) null,

LUONG int null,

)

GO

ALTER TABLE NV WITH CHECK ADD FOREIGN KEY(MSCOQUAN) REFERENCES COQUAN(MSCOQUAN)

GO

INSERT INTO dbo.COQUAN(MSCOQUAN, TENCOQUAN, DIACHI)

VALUES

('10', 'KIDO', 'Hà Nội'),

('15', 'COCACOLA', 'TP.HCM'),

('20', 'VINAMILK', 'BÌNH DƯƠNG'),

('25', '7UP', 'ĐỒ SƠN'),

('50', 'THIÊN LONG', 'BÌNH DƯƠNG');

INSERT INTO dbo.NV(MSNV, Ten, MSCOQUAN, CONGVIEC, Luong)

VALUES

('NV0001', 'TRẦN QUỐC ANH', '10', 'NHÂN VIÊN KỸ THUẬT', 5000),

('NV0002', 'LÊ THANH BÌNH', '50', 'CHĂM SÓC KHÁCH HÀNG', 3500),

('NV0003', 'NGUYỄN THANH DUY', '15', 'KẾ TOÁN', 4000),

('NV0004', 'BÙI DIỄM QUYÊN', '20', 'CHUYÊN VIÊN BẢO MẬT', 7000),

('NV0005', 'NGUYỄN HOÀNG VŨ', '20', 'QUẢN LÝ VẬN CHUYỂN', 6500),

('NV0006', 'NGUYỄN ANH TÚ', '25', 'NHÂN VIÊN BÁN HÀNG', 4000),

('NV0007', 'NGUYỄN HOÀNG VŨ', '20', 'QUẢN LÝ VẬN CHUYỂN', 6500),

('NV0008', 'HỒ MỸ LỆ', '15', 'TRƯỞNG KHO', 5000),

('NV0009', 'NGUYỄN HOÀNG YẾN', '10', 'NHÂN VIÊN BÁN HÀNG', 5000),

('NV0010', 'HOÀNG THẢO VY', '25', 'KẾ TOÁN TRƯỞNG', 6000);

* **CODE TẠO DATABASE VÀ NHẬP DỮ LIỆU BÀI 4**

CREATE DATABASE Bai4;

GO

USE Bai4;

GO

CREATE TABLE dbo.THO (

MaTho nchar(10) PRIMARY KEY,

TenTho nvarchar(50) NULL,

Nhom int NULL,

NhomTruong nchar(10) NULL

);

GO

CREATE TABLE dbo.CONGVIEC (

MaCV nchar(10) PRIMARY KEY,

NoiDungCV nvarchar(MAX) NULL

);

GO

CREATE TABLE dbo.HOPDONG (

SoHD nchar(10) PRIMARY KEY,

NgayHD date NULL,

MaKH nchar(10) NULL,

SoXe nchar(10) NULL,

TriGiaHD int NULL,

NgayGiaoDK date NULL,

NgayNgThu date NULL

);

GO

CREATE TABLE dbo.KHACHHANG (

MaKH nchar(10) PRIMARY KEY,

TenKH nvarchar(50) NULL,

DiaChi nvarchar(MAX) NULL,

DienThoai nchar(10) NULL

);

GO

CREATE TABLE dbo.CHITIET\_HD (

SoHD nchar(10),

MaCV nchar(10),

TriGiaCV int NULL,

MaTho nchar(10) NULL,

KhoanTho int NULL,

PRIMARY KEY (SoHD, MaCV),

FOREIGN KEY(SoHD) REFERENCES HOPDONG(SoHD),

FOREIGN KEY(MaCV) REFERENCES CONGVIEC(MaCV),

FOREIGN KEY(MaTho) REFERENCES THO(MaTho)

);

GO

CREATE TABLE dbo.PHIEUTHU (

SoPT nchar(10) PRIMARY KEY,

NgaylapPT date NULL,

SoHD nchar(10) NULL,

HoTen nvarchar(50) NULL,

SoTienThu int NULL,

FOREIGN KEY(SoHD) REFERENCES HOPDONG(SoHD),

FOREIGN KEY(SoHD) REFERENCES HOPDONG(SoHD),

);

GO

INSERT INTO dbo.THO --(MaTho, TenTho, Nhom, NhomTruong)

VALUES

('TH001', 'LÊ ĐỨC AN', 1, 'TH002'),

('TH002', 'NGUYỄN VĂN CHÍ', 1, NULL),

('TH003', 'BÙI NGỌC DUY', 1, 'TH002'),

('TH004', 'LÊ MINH HOÀNG', 2, NULL),

('TH005', 'TRẦN VĂN MINH', 2, 'TH004'),

('TH006', 'NGUYỄN VĂN NHẬT', 2, 'TH004'),

('TH007', 'TRẦN NGỌC PHÁT', 3, 'TH009'),

('TH008', 'LÊ MINH QUANG', 3, 'TH009'),

('TH009', 'NGUYỄN NGỌC SƠN', 3, NULL),

('TH010', 'LÊ VĂN THÀNH', 4, NULL),

('TH011', 'VƯƠNG NGỌC VINH', 4, 'TH010'),

('TH012', 'HOÀNG VĂN VỸ', 4, 'TH010');

INSERT INTO dbo.CONGVIEC -- (MaCV, NoiDungCV)

VALUES

('CV001', 'Thay dầu máy'),

('CV002', 'Kiểm tra hệ thống phanh'),

('CV003', 'Thay lọc gió'),

('CV004', 'Kiểm tra và cân bằng lốp xe'),

('CV005', 'Thay nước làm mát'),

('CV006', 'Vệ sinh bộ làm mát'),

('CV007', 'Kiểm tra hệ thống điện'),

('CV008', 'Điều chỉnh hệ thống lái'),

('CV009', 'Bơm gas lạnh'),

('CV010', 'Kiểm tra hệ thống cảm biến'),

('CV011', 'Điều chỉnh van xả'),

('CV012', 'Vệ sinh bộ lọc khí'),

('CV013', 'Kiểm tra hệ thống khí thải'),

('CV014', 'Thay nước cấp đông'),

('CV015', 'Kiểm tra và thay dây đai động cơ'),

('CV016', 'Kiểm tra và thay dây curoa'),

('CV017', 'Kiểm tra hệ thống làm mát hộp số'),

('CV018', 'Kiểm tra và thay dây phanh'),

('CV019', 'Kiểm tra và thay bộ lọc dầu'),

('CV020', 'Thay nhớt');

INSERT INTO dbo.HOPDONG --(SoHD, NgayHD, MaKH, SoXe, TriGiaHD, NgayGiaoDK, NgayNgThu)

VALUES

('HD001', '2002-01-05', 'KH014', 'ABC123', 1500000, '2002-01-07', '2002-01-08'),

('HD002', '2002-02-10', 'KH001', 'XYZ789', 1500000, '2002-02-15', '2002-02-16'),

('HD003', '2002-03-15', 'KH009', 'DEF456', 1000000, '2002-03-19', '2002-03-20'),

('HD004', '2003-01-20', 'KH007', 'GHI789', 2500000, '2003-01-26', '2003-01-27'),

('HD005', '2003-02-25', 'KH001', 'JKL123', 2200000, '2003-03-01', '2003-03-02'),

('HD006', '2002-03-03', 'KH003', 'MNO456', 1300000, '2002-03-09', '2002-03-10'),

('HD007', '2003-01-10', 'KH010', 'PQR789', 1900000, '2003-01-15', '2003-01-16'),

('HD008', '2002-02-15', 'KH011', 'STU123', 1700000, '2002-02-20', '2002-02-21'),

('HD009', '2002-03-20', 'KH003', 'VWX456', 2800000, '2002-03-28', '2002-03-29'),

('HD010', '2003-01-28', 'KH006', 'YZA789', 2100000, '2003-02-05', '2003-02-06'),

('HD011', '2002-02-05', 'KH001', 'ABC456', 1600000, '2002-02-09', '2002-02-10'),

('HD012', '2002-08-10', 'KH013', 'DEF789', 2300000, '2002-08-17', '2002-08-18'),

('HD014', '2002-11-03', 'KH002', 'JKL456', 2000000, '2002-11-15', '2002-11-16'),

('HD015', '2003-01-07', 'KH002', 'XYZ123', 1900000, '2003-01-16', '2003-01-17');

INSERT INTO dbo.KHACHHANG --(MaKH, TenKH, DiaChi, DienThoai)

VALUES

('KH001', 'Nguyễn Văn A', 'Số 10, Đường ABC, TP.HCM', '0901234567'),

('KH002', 'Trần Thị B', 'Số 20, Đường XYZ, Hà Nội', '0987654321'),

('KH003', 'Lê Văn C', 'Số 30, Đường DEF, Đà Nẵng', '0912345678'),

('KH004', 'Phạm Thị D', 'Số 40, Đường GHI, Hải Phòng', '0976543210'),

('KH005', 'Hoàng Văn E', 'Số 50, Đường JKL, Cần Thơ', '0923456789'),

('KH006', 'Nguyễn Thị F', 'Số 60, Đường MNO, Hà Nội', '0912345678'),

('KH007', 'Trần Văn G', 'Số 70, Đường PQR, TP.HCM', '0987654321'),

('KH008', 'Lê Thị H', 'Số 80, Đường STU, Đà Nẵng', '0901234567'),

('KH009', 'Phạm Văn I', 'Số 90, Đường VWX, Hải Phòng', '0976543210'),

('KH010', 'Hoàng Thị K', 'Số 100, Đường YZA, Cần Thơ', '0923456789'),

('KH011', 'Nguyễn Văn L', 'Số 110, Đường ABC, TP.HCM', '0912345678'),

('KH012', 'Trần Thị M', 'Số 120, Đường XYZ, Hà Nội', '0987654321'),

('KH013', 'Lê Văn N', 'Số 130, Đường DEF, Đà Nẵng', '0901234567'),

('KH014', 'Phạm Thị O', 'Số 140, Đường GHI, Hải Phòng', '0976543210'),

('KH015', 'Hoàng Văn P', 'Số 150, Đường JKL, Cần Thơ', '0923456789');

INSERT INTO dbo.CHITIET\_HD --(SoHD, MaCV, TriGiaCV, MaTho, KhoanTho)

VALUES

('HD001', 'CV001', 300000, 'TH001', 150000),

('HD001', 'CV003', 500000, 'TH009', 250000),

('HD001', 'CV004', 200000, 'TH012', 100000),

('HD002', 'CV001', 400000, 'TH003', 200000),

('HD002', 'CV010', 350000, 'TH003', 175000),

('HD003', 'CV007', 450000, 'TH002', 225000),

('HD005', 'CV002', 600000, 'TH003', 300000),

('HD005', 'CV003', 550000, 'TH005', 275000),

('HD006', 'CV005', 250000, 'TH007', 125000),

('HD007', 'CV005', 350000, 'TH012', 175000),

('HD007', 'CV014', 200000, 'TH004', 100000),

('HD008', 'CV020', 400000, 'TH006', 200000),

('HD009', 'CV010', 300000, 'TH007', 150000),

('HD010', 'CV007', 600000, 'TH001', 300000),

('HD012', 'CV009', 450000, 'TH011', 225000);

INSERT INTO dbo.PHIEUTHU -- (SoPT, NgaylapPT, SoHD, HoTen, SoTienThu)

VALUES

('PT001', '2002-01-06', 'HD001', 'Nguyễn Văn A', 1500000),

('PT002', '2002-02-11', 'HD002', 'Trần Thị B', 700000),

('PT003', '2002-03-16', 'HD003', 'Lê Văn C', 400000),

('PT004', '2003-01-21', 'HD004', 'Phạm Thị D', 2500000),

('PT005', '2003-02-26', 'HD005', 'Hoàng Văn E', 600000),

('PT006', '2002-03-04', 'HD006', 'Nguyễn Thị F', 800000),

('PT007', '2003-01-11', 'HD007', 'Trần Văn G', 1900000),

('PT008', '2002-02-16', 'HD008', 'Lê Thị H', 550000),

('PT009', '2002-03-21', 'HD009', 'Phạm Văn I', 100000),

('PT010', '2003-01-29', 'HD010', 'Hoàng Thị K', 2100000);

* **CODE TẠO DATABASE VÀ NHẬP DỮ LIỆU BÀI 5**

CREATE DATABASE Bai5;

GO

USE Bai5;

GO

CREATE TABLE dbo.GV (

MAGV nchar(10) PRIMARY KEY,

TENGV nvarchar(50) NULL,

MAMH nchar(10) NULL,

FOREIGN KEY (MAMH) REFERENCES MHOC(MAMH)

);

GO

CREATE TABLE dbo.MHOC (

MAMH nchar(10) PRIMARY KEY,

TENMH nvarchar(50),

SOTIET int

);

GO

CREATE TABLE dbo.BUOITHI (

HKY int,

NGAY date,

GIO nvarchar(10),

PHG int,

MAMH nchar(10),

TGTHI int,

PRIMARY KEY(HKY, NGAY, GIO, PHG),

FOREIGN KEY (MAMH) REFERENCES MHOC(MAMH)

);

GO

CREATE TABLE dbo.PC\_COI\_THI (

MAGV nchar(10),

HKY int,

NGAY date,

GIO nvarchar(10),

PHG int,

PRIMARY KEY (MAGV, HKY, NGAY, GIO, PHG),

FOREIGN KEY (MAGV) REFERENCES GV(MAGV),

FOREIGN KEY (HKY, NGAY,GIO, PHG) REFERENCES BUOITHI(HKY, NGAY, GIO, PHG)

);

INSERT INTO dbo.GV (MAGV, TENGV, MAMH)

VALUES

('GV001', N'NGUYỄN HỒNG NHUNG', 'MH01'),

('GV002', N'VŨ THÚY NGA', NULL),

('GV003', N'TRẦN KIM TRI', 'MH02'),

('GV004', N'NGUYỄN THỊ THANH THU', NULL),

('GV005', N'TRẦN HOÀI DẠ VŨ', 'MH03'),

('GV006', N'TRẦN XUÂN QUỲNH', NULL),

('GV007', N'LÊ THU HOÀI', 'MH04'),

('GV008', N'NGUYỄN THỊ HỒNG THANH', 'MH08'),

('GV009', N'TRẦN THANH TÙNG', 'MH05'),

('GV010', N'VŨ KIM HOÀNG', NULL);

INSERT INTO dbo.MHOC (MAMH, TENMH, SOTIET)

VALUES

('MH01', N'TOÁN', 45),

('MH02', N'VẬT LÝ', 30),

('MH03', N'HÓA HỌC', 45),

('MH04', N'SINH HỌC', 30),

('MH05', N'TIẾNG ANH', 45),

('MH06', N'LỊCH SỬ', 30),

('MH07', N'ĐỊA LÝ', 45),

('MH08', N'VĂN HỌC', 30),

('MH09', N'MỸ THUẬT', 45),

('MH10', N'ÂM NHẠC', 30),

('MH11', N'THỂ DỤC', 45),

('MH12', N'TIN HỌC', 30);

INSERT INTO dbo.BUOITHI (HKY, NGAY, GIO, PHG, MAMH, TGTHI)

VALUES

(1, '2024-03-15', '08:00', 101, 'MH01', 150),

(1, '2024-03-16', '09:00', 102, 'MH02', 120),

(1, '2024-03-17', '10:00', 103, 'MH03', 150),

(1, '2024-03-18', '11:00', 104, 'MH04', 120),

(1, '2024-03-19', '12:00', 105, 'MH05', 150),

(2, '2024-03-20', '08:00', 106, 'MH06', 120),

(2, '2024-03-21', '09:00', 107, 'MH07', 150),

(2, '2024-03-22', '10:00', 108, 'MH08', 120),

(2, '2024-03-23', '11:00', 109, 'MH09', 150),

(2, '2024-03-24', '12:00', 110, 'MH10', 120),

(3, '2024-03-25', '08:00', 111, 'MH11', 150),

(3, '2024-03-26', '09:00', 112, 'MH12', 120),

(3, '2024-03-27', '10:00', 101, 'MH01', 150),

(3, '2024-03-28', '11:00', 102, 'MH02', 120),

(3, '2024-03-29', '12:00', 103, 'MH03', 150);

INSERT INTO dbo.PC\_COI\_THI (MAGV, HKY, NGAY, GIO, PHG)

VALUES

('GV001',1, '2024-03-15', '08:00', 101),

('GV002',1, '2024-03-16', '09:00', 102),

('GV003',1, '2024-03-17', '10:00', 103),

('GV008',1, '2024-03-18', '11:00', 104),

('GV008',1, '2024-03-19', '12:00', 105),

('GV008',2, '2024-03-20', '08:00', 106),

('GV007',2, '2024-03-21', '09:00', 107),

('GV005',2, '2024-03-22', '10:00', 108),

('GV009',2, '2024-03-23', '11:00', 109),

('GV010',2, '2024-03-24', '12:00', 110),

('GV008',3, '2024-03-25', '08:00', 111),

('GV008',3, '2024-03-26', '09:00', 112),

('GV003',3, '2024-03-27', '10:00', 101),

('GV004',3, '2024-03-28', '11:00', 102),

('GV005',3, '2024-03-29', '12:00', 103);

**CÂU HỎI ÔN TẬP CHƯƠNG 4**

1. **Ngôn ngữ SQL là gì?**

SQL (Structure Query Language) là ngôn ngữ truy vấn có cấu trúc. SQL hỗ trợ việc tạo, xóa, lưu trữ và truy xuất dữ liệu trong CSDL quan hệ. SQL được sử dụng trong các hệ quản trị CSDL thông dụng: MS Access, MySQL, Oracle, SQL Sever, Informix, …

1. **Thứ tự viết các mệnh đề trong câu SQL?**

SELECT 🡪 FROM 🡪 WHERE 🡪 GROUP BY 🡪 HAVING 🡪 ORDER BY

1. **Thứ tự thực hiện các mệnh đề trong câu SQL?**

FROM 🡪 WHERE 🡪 GROUP BY 🡪 HAVING 🡪 SELECT 🡪 ORDER BY 🡪 KẾT QUẢ TRUY VẤN

1. **Cho Q(A, B, C). Viết câu truy vấn tương đương với biểu thức đại số quan hệ πA,B(Q)**

SELECT A,B

FROM Q

1. **Cho Q(A, B, C, D). Dấu \* ở mệnh đề Select trong câu truy vấn SELECT \* FROM Q có nghĩa là gì?**

Có nghĩa là lấy hết tất cả các cột A,B,C,D của quan hệ Q

1. **Cho Q(A: int, B: int, C: int). Viết 2 câu truy vấn tương đương với biểu thức đại số quan hệ πA,B,C(σA>B(Q))**

SELECT A,B,C

FROM Q

WHERE A > B

SELECT \*

FROM Q

WHERE A > B

1. **Cho Q(A: int, B: int, C: int). Tìm các chỗ sai trong câu truy vấn**

SELECT A.Q FROM Q WHERE A.Q > ‘5’

* Chọn tên cột sai (A.Q) 🡪 sửa thành A
* Sai ở điều kiện (WHERE A.Q > ‘5’), cột A có miền giá trị là số nguyên nên không cần dùng dấu ‘ ’ 🡪 sửa thành WHERE A > 5

1. **Cho SV(MaSV: string, Hoten: string, MaKh:string, DiemTB:real). Câu truy vấn nào dưới đây thực hiện lập danh sách SV thuộc Khoa có MaKh là ‘50’ và có điểm tb > 8:**
2. SELECT MaSV, Hoten FROM SV WHERE MaKh = 50, DiemTB >8
3. SELECT MaSV, Hoten FROM SV WHERE MaKh = ‘50’, DiemTB >8
4. SELECT MaSV, Hoten FROM SV WHERE MaKh = ‘50’∧ DiemTB >8
5. SELECT MaSV, Hoten FROM SV WHERE MaKh = ‘50’ and DiemTB >8
6. **Cho SV(MaSV: string, Hoten: string, MaKh:string, DiemTB:real). Câu truy vấn nào dưới đây sai và tại sao?**
7. SELECT MaSV, Hoten FROM SV WHERE MaKh = 50, DiemTB >8

🡪 Sai vì MaKh là kiểu string nên phải dùng cặp dấu ‘ ’ và kết hợp nhiều điều kiện không dùng dấu phẩy.

1. SELECT MaSV.SV, Hoten.SV FROM SV WHERE DiemTB >8

🡪 Sai cú pháp khi chọn cột

1. SELECT SV.MaSV, SV.Hoten FROM SV WHERE DiemTB >8
2. **Cho NV(MaNV: string, Hoten: string, Luong: Real, MaPB:string). Câu SQL nào dưới đây đúng:**
3. SELECT MaNV, Hoten, Luong \* 2 From NV WHERE MAPB = ‘50’
4. SELECT MaNV, Hoten, Luong \* 2 Thuong From NV WHERE MAPB = ‘50’
5. SELECT MaNV, Hoten, Luong \* 2 as Thuong Form NV WHERE MAPB = ‘50’
6. **Cho NV(MaNV: string, Hoten: string, Luong: Real, MaPB:string). Hãy cho biết kết quả của câu truy vấn: SELECT MaNV, Hoten, Luong \* 2 as Thuong FROM NV**

Kết quả của truy vấn này là quan hệ gồm các cột MaNV, Hoten và Thuong (trong đó Thuong là kết quả của Luong \* 2) được lấy từ quan hệ input là NV.

1. **Từ khóa AS trong câu truy vấn SELECT MaNV, Hoten, Luong \* 2 as Thuong FROM NV dùng để làm gì?**

Dùng để đặt bí danh cho thuộc tính.

1. **Cho NhanVien(MaNV: string, Hoten: string, Luong: Real). câu truy vấn dưới thực hiện việc gì:**

SELECT NV.MaNV, NV.Hoten, NV.Luong FROM NhanVien as NV WHERE Luong >1000

Thực hiện lấy MaNV, Hoten và Luong đối với các nhân viên có Luong lớn hơn 1000

1. **Từ khóa AS trong câu truy vấn SELECT MaNV, Hoten, Luong FROM NhanVien as NV dùng để làm gì?**

Đặt bí danh cho quan hệ NhanVien.

1. **Bí danh của thuộc tính được đặt ở mệnh đề SELECT có được dùng ở các mệnh đề khác không? Tại sao?**

Thứ tự thực hiện các mệnh đề là

FROM 🡪 WHERE 🡪 GROUP BY 🡪 HAVING 🡪 SELECT 🡪 ORDER BY 🡪 KẾT QUẢ TRUY VẤN

Do đó các mệnh đề trước SELECT sẽ không dùng được bí danh đó mà chỉ mệnh đề đứng sau SELECT dùng được

1. **Phát biểu nào dưới đây đúng:**
2. Các biểu thức điều kiện ở mệnh đề WHERE phải được liên kết bằng AND hoặc OR
3. Các biểu thức điều kiện ở mệnh đề WHERE phải được phân cách bằng dấu phẩy
4. **LIKE hoặc NOT LIKE dùng để làm gì?**

LIKE và NOT LIKE: dùng trong so sánh chuỗi.

1. Ý nghĩa của ký tự đại diện ‘%’ hoặc ‘\_’ trong biểu thức điều kiện so sánh chuỗi.

Các ký tự đại diện: ‘%’ (đại diện một nhóm ký tự bất kỳ), ‘\_’ (đại diện 1 ký tự bất kỳ).

1. **Cho Q(A: nvarchar(4), B: int). câu SQL SELECT \* FROM Q WHERE A like ‘%a% thực  
    hiện việc gì?**

Thực hiện việc lấy tất cả các dòng từ quan hệ Q trong đó giá trị của cột A chứa ký tự 'a' ở bất kỳ vị trí nào trong chuỗi.

1. **Cho Q(A: varchar(4), B: int). Viết câu SQL tương đương với câu SELECT \* FROM Q WHERE B Between 10 and 20**

SELECT \*

FROM Q

WHERE B >=10 AND B <=20

1. **Hàm gộp là gì và gồm những hàm nào?**

Hàm gộp nhận vào một tập các giá trị của 1 cột trong một quan hệ và cho kết quả là một giá trị.Các hàm gộp: Min, Max, Sum, Avg, Count

1. **Cho NV(MaNV: varchar(10), Luong: int, MaPB: varchar(4)).**   
    **Câu SQL SELECT count(\*), avg(luong) FROM NV thực hiện việc gì?**

Thực hiện đếm số lượng dòng trong bảng và tính lương trung bình của các nhân viên.

1. **Cho NV(MaNV: varchar(10), Luong: int, MaPB: varchar(4)).   
    Trong SQL Server, kết quả của câu SQL: SELECT count(\*), avg(luong) FROM NV là gì?**

Quan hệ kết quả dồm các cột MaNV, Luong,MaPB và giá trị các cột này là số lượng, còncột avg(Luong) là lương trung bình của nhân viên.

1. **Cho NV(MaNV: varchar(10), Luong: int, MaPB: varchar(4)).   
    Trong SQL Server, kết quả của câu SQL: SELECT count(\*) Sonv, avg(luong) as Luongtb FROM NV là gì?**

Kết quả là một bảng gồm 2 cột, cột Sonv thể hiện số lượng nhân viên, cột Luongtb là cột chứa giá trị lương trung bình của các nhân viên.

1. **Cho NV(MaNV: varchar(10), Luong: int, MaPB: varchar(4)).**   
    **Câu SQL SELECT count(MaNV), avg(luong) FROM NV GROUP BY MaPB thực hiện việc gì?**

Đếm số lượng nhân viên, lương trung bình của từng phòng ban

1. **Cho NV(MaNV: varchar(10), Luong: int, MaPB: varchar(4)).   
    Trong SQL Server, kết quả của câu SQL: SELECT MaPB, count(MaNV), avg(luong) FROM NV GROUP BY MaPB là gì?**

Kết quả là một bảng chứa số lượng nhân viên và lương trung bình của nhân viên ở mỗi phòng ban.

1. **Phát biểu nào dưới đây đúng khi câu SQL có sử dụng mệnh đề GROUP BY**
2. Các thuộc tính có trong mệnh đề SELECT phải có trong mệnh đề GROUP BY hoặc trong hàm gộp
3. Thuộc tính có trong mệnh đề GROUP BY có thể không có trong mệnh đề SELECT
4. **Phát biểu nào dưới đây đúng:**
5. Trong câu SQL, mệnh đề HAVING được sử dụng khi điều kiện chọn liên quan tới hàm gộp
6. Trong câu SQL, mệnh đề HAVING (nếu có) phải nằm sau mệnh đề GROUP BY
7. **Phát biểu nào dưới đây đúng:**
8. Trong câu SQL, khi sử dụng mệnh đề HAVING thì phải có mệnh đề GROUP BY
9. Trong câu SQL, mệnh đề WHERE sử dụng với điều kiện không dùng hàm gộp
10. Trong câu SQL, có thể có cả mệnh đề WHERE và mệnh đề HAVING
11. **Cho NV(MaNV: varchar(10), Luong: int, MaPB: varchar(4)).   
    câu SQL: SELECT MaPB, count(MaNV) FROM NV GROUP BY MaPB HAVING count(MaNV) > 2; thực hiện việc gì?**

Liệt kê MaPB và lương trung bình của các phòng ban có nhiều hơn 2 nhân viên.

1. **Cho Q(A: varchar(10), B: int, C: varchar(4)). Chỉ những chỗ sai của câu SQL:   
    SELECT A, count(b) as v FROM Q.A GROUP BY C HAVING v > 2;**

Sai vì FROM Q.A sai vì mệnh đề FROM chỉ có tên quan hệ, gộp C nhưng chọn A là không hợp lý

1. **Cho NV(MaNV: varchar(10), Luong: int, MaPB: varchar(4)). Chỉ những chỗ sai của câu SQL: SELECT MaPB, count(MaNV) as sonv FROM NV n WHERE sonv > 2;**

Sai vì không thể sử dụng mệnh đề WHERE để lọc kết quả dựa trên một hàm tổng hợp như count(). Thay vào đó, sử dụng mệnh đề HAVING sau mệnh đề GROUP BY.

1. **Mệnh đề ORDER BY dùng để làm gì?**

Trong SQL, mệnh đề ORDER BY được sử dụng để sắp xếp dữ liệu theo thứ tự tăng dần hoặc theo thứ tự giảm dần trên một hoặc nhiều cột.

1. **Cho NV (MaNV, Hoten, NgSinh). Câu SQL:   
    SELECT Hoten, NgSinh FROM NV ORDER BY Hoten, NgSinh DESC; thực hiện việc gì?**

Liệt kê họ tên và ngày sinh của những nhân viên. Sắp họ tên theo thứ tự tăng dần, trùng họ tên thì sắp ngày sinh theo thứ tự giảm dần.

1. **Cho Q(A, B, C). Viết câu SQL cho kết quả tương đương với câu SELECT \* FROM Q ORDER BY A DESC,C**

SELECT A,B,C

FROM Q

ORDER BY C, A DESC;

1. **Trình bày phép kết nội và phép kết ngoài**

Phép kết nội:Select <tên cột 1>, <tên cột 2>, …

From < quan hệ1>, <quan hệ 2>

Where <điều kiện kết>

Phép kết ngoài:Select <tên cột 1>, <tên cột 2>, …

From < quan hệ1> Left/Right/Full Outer join <quan hệ 2>

on <điều kiện kết>

1. **Cho Q1(A: int, B: int) và Q2(C: int, D: int) và câu SQL   
   SELECT A, C FROM Q1 INNER JOIN Q2 ON A = C; thực hiện việc gì**

Kết hợp các hàng từ hai bảng Q1 và Q2 trong đó giá trị của cột A từ bảng Q1 phải bằng giá trị của cột C từ bảng Q2.

1. **Cho Q1(A: int, B: int) và Q2(C: int, D: int). Câu SQL nào dưới đây đúng**
2. SELECT A, C FROM Q1 JOIN Q2 ON A = C;
3. SELECT A, C FROM Q1 INNER JOIN Q2 ON A = C;
4. **Cho Q1(A: int, B: int) và Q2(C: int, D: int). Câu SQL nào dưới đây đúng**
5. SELECT A, C FROM Q1 LEFT JOIN Q2 ON A = C;
6. SELECT A, C FROM Q1 LEFT OUTER JOIN Q2 ON A = C;
7. **Cho Q1(A: int, B: int) và Q2(C: int, D: int). Câu SQL   
   SELECT \* FROM Q1 FULL JOIN Q2 ON A = C; thực hiện việc gì?**

Thực hiện việc nối đầy đủ hai bảng Q1 và Q2. Kết quả là tất cả các hàng từ cả hai bảng, Q1 và Q2, bất kể dữ liệu có tương ứng hay không. Nếu có hàng trong Q1 có giá trị cột A khớp với giá trị cột C trong Q2, những hàng này sẽ được kết hợp lại. Đối với những hàng không khớp, câu truy vấn sẽ vẫn trả về hàng đó với các cột từ bảng kia được điền là NULL.

1. **Cho Q1(A: int, B: int) và Q2(C: int, D: int). Câu SQL   
   SELECT \* FROM Q1 FULL OUTER JOIN Q2 ON A = C; tương đương với**
2. SELECT \* FROM Q1 JOIN Q2 ON A = C;
3. SELECT \* FROM Q1 FULL JOIN Q2 ON A = C
4. **Cho Q1(A: int, B: int) và Q2(C: int, D: char(5)). Câu SQL   
   SELECT \* FROM Q1 OUTER JOIN Q2 A = C WHERE D = null; sai ở những chỗ nào?**

Sai vì thiếu mệnh đề ON (trong câu truy vấn phải sử dụng mệnh đề ON để xác định điều kiện nối giữa hai bảng), sử dụng NULL không đúng cách (để kiểm tra một cột có giá trị NULL, phải sử dụng IS NULL thay vì = null) và không rõ loại JOIN (OUTER JOIN cần phải được chỉ định rõ ràng là LEFT, RIGHT, hoặc FULL)

1. **Cho SV(MaSV, Hoten) và Dangky(MaSV, MaMH). Câu SQL  
   (SELECT MaSV FROM SV) EXCEPT (SELECT MaSV FROM Dangky); thực hiện việc gì?**

Liệt kê MaSV của những sinh viên không đăng ký môn học.

1. **Cho SV(MaSV, Hoten) và Dangky(MaSV, MaMH). Câu SQL nào ở dưới tương đương với câu truy vấn (SELECT MaSV FROM SV) EXCEPT (SELECT MaSV FROM Dangky)**
2. SELECT SV.MaSV FROM SV LEFT JOIN Dangky ON SV.MaSV = Dangky.MaSV
3. SELECT SV.MaSV FROM SV LEFT JOIN Dangky ON SV.MaSV = Dangky.MaSV WHERE Dangky.MaMH IS NULL
4. SELECT SV.MaSV FROM SV LEFT JOIN Dangky ON SV.MaSV = Dangky.MaSV WHERE Dangky.MaMH = NULL
5. **Cho SV(MaSV, Hoten) và Dangky(MaSV, MaMH). Câu SQL nào ở dưới tương đương với câu truy vấn (SELECT MaSV FROM SV) EXCEPT (SELECT MaSV FROM Dangky)**
6. SELECT SV.MaSV FROM SV LEFT JOIN Dangky ON SV.MaSV = Dangky.MaSV WHERE Dangky.MaMH IS NULL
7. SELECT s.MaSV FROM SV s LEFT JOIN Dangky d ON s.MaSV = d.MaSV WHERE Dangky.MaMH IS NULL
8. **Cho SV(MaSV, Hoten) và Dangky(MaSV, MaMH). Chỉ những chỗ sai trong câu SQL  
   SELECT MaSV FROM SV OUTER JOIN Dangky ON SV.MaSV = Dangky.MaSV WHERE Dangky.MaMH = NULL;**

Sai vì sử dụng NULL không đúng cách (để kiểm tra một cột có giá trị NULL, phải sử dụng IS NULL thay vì = null) và không rõ loại JOIN (OUTER JOIN cần phải được chỉ định rõ ràng là LEFT, RIGHT, hoặc FULL)

1. **Cho SV(MaSV, Hoten) và Dangky(MaSV, MaMH). Câu SQL nào ở dưới cho kết quả tương đương với câu truy vấn (SELECT MaSV FROM SV) INTERSECT (SELECT MaSV FROM Dangky)**
2. SELECT MaSV FROM Dangky
3. (SELECT MaSV FROM SV) EXCEPT (SELECT MaSV FROM Dangky)
4. SELECT MaSV FROM SV
5. **Cho Q1(A, B) và Q2(A, B). Câu SQL  
   (SELECT A, B FROM Q1) INTERSECT (SELECT A, B FROM Q2); thực hiện việc gì?**

Lấy bộ giá trị cột A,B xuất hiện trong cả hai quan hệ.

1. **Cho SV(MaSV, Hoten) và Dangky(MaSV, MaMH). Câu SQL dưới thực hiện việc gì (SELECT MaSV FROM SV) UNION (SELECT MaSV FROM Dangky)**

Lấy tất cả mã sinh viên trong quan hệ SV, Dangky và loại bỏ các giá trị trùng lặp.

1. **Cho Q1(A, B) và Q2(A, B). Câu SQL  
   (SELECT \* FROM Q1) UNION (SELECT \* FROM Q2); thực hiện việc gì?**

Lấy tất cả bộ giá trị xuất hiện trong quan hệ Q1,Q2 và loại bỏ các giá trị trùng lặp.

1. **Cho SV(MaSV, Hoten) và Dangky(MaSV, MaMH). Câu truy vấn  
    SELECT Hoten FROM SV WHERE MaSV NOT IN (SELECT MaSV FROM Dangky); thực hiện việc gì?**

Liệt kê họ tên của các sinh viên không đăng ký môn học.

1. **Cho SV(MaSV, Hoten) và Dangky(MaSV, MaMH). Câu SQL tương đương với câu:  
    SELECT MaSV FROM SV WHERE MaSV NOT IN (SELECT MaSV FROM Dangky)**

(SELECT MaSV FROM SV) EXCEPT (SELECT MaSV FROM Dangky)

1. **Cho NV(MaNV, Hoten, Luong). Câu SQL   
    SELECT \* FROM NV WHERE Luong > (SELECT AVG(Luong) FROM NV); thực hiện việc gì?**
2. Liệt kê danh sách NV có lương cao nhất
3. Liệt kê danh sách NV có lương lớn hơn mức lương trung bình
4. **Cho NV(MaNV, Hoten, Luong). Câu SQL nào dưới đây tương đương với  
   SELECT \* FROM NV WHERE Luong > (SELECT AVG(Luong) FROM NV);**
5. SELECT \* FROM NV, (SELECT AVG(Luong) as ltb FROM NV) Q WHERE Luong > LTB
6. SELECT \* FROM NV HAVING Luong > (SELECT AVG(Luong) FROM NV)
7. **Cho NV(MaNV, Hoten, Luong). Câu SQL   
    SELECT \* FROM NV WHERE Luong >= all (SELECT Luong FROM NV); thực hiện việc gì?**

Lấy mã nhân viên, họ tên và lương của các nhân viên có lương cao nhất.

1. **Cho NV(MaNV, Hoten, Luong). Câu SQL   
   SELECT \* FROM NV WHERE Luong = (SELECT max(Luong) FROM NV); thực hiện việc gì?**

Lấy mã nhân viên, họ tên và lương của các nhân viên có lương cao nhất.

1. **Cho NV(MaNV, Hoten, Luong). Câu SQL nào dưới đây tương đương với  
   SELECT \* FROM NV WHERE Luong >= all (SELECT Luong FROM NV)**
2. SELECT \* FROM NV WHERE Luong >= any (SELECT Luong FROM NV);
3. SELECT \* FROM NV WHERE Luong = (SELECT max(Luong) FROM NV)
4. **Truy vấn con tương quan là gì?**

Truy vấn con tương quan cho phép so sánh giữa thuộc tính của quan hệ bên trong truy vấn con với thuộc tính của quan hệ trong truy vấn bao bên ngoài

1. **Công dụng của EXISTS**

EXISTS dùng để kiểm tra kết quả của câu truy vấn con tương quan là rỗng hay không rỗng. Nếu kết quả của câu truy vấn là rỗng thì EXISTS trả về False, ngược lại nó trả về True.

1. **Cho NV(MaNV, Hoten) và ThanNhan(MaNV, TenThannhan). Câu SQL nào dưới đây đúng**
2. SELECT Hoten FROM NV WHERE EXISTS (SELECT \* FROM THANNHAN tn WHERE NV.MaNV = tn.MaNV);
3. SELECT Hoten FROM NV WHERE MaNV EXISTS (SELECT \* FROM THANNHAN tn WHERE NV.MaNV = tn.MaNV);