

BÀI THỰC HÀNH SỐ 6

Môn học: Cơ sở dữ liệu

I. Nội dung

- ❖ Ôn tập thi thực hành và thi lý thuyết cuối kỳ phần SQL

II. Hướng dẫn thực hành tại lớp (Database: QlyCungcapPhutung)

Cho lược đồ CSDL “Quản lý đỗ xe” như sau:

KHACHHANG (MAKH, HOTEN, DIACHI, NGSINH, EMAIL)

Thông tin khách hàng gồm: mã khách hàng, họ tên, địa chỉ, ngày sinh, email

HANGXE (MAHX, TENHX)

Thông tin hãng xe gồm: mã hãng xe, tên hãng xe

XE (MAXE, NAMSX, MAHX)

Thông tin xe gồm: mã xe, năm sản xuất, mã hãng xe của xe này.

VEXE (MAVE, MOTA)

Thông tin về vé xe gồm mã vé, mô tả

PHIEUDOXE (MAPHIEU, MAKH, MAXE, MAVE, NGAYGHIPHIEU, NGAYTRATIEN)

Thông tin phiếu đỗ xe gồm: mã phiếu, mã khách hàng, mã xe, mã vé, ngày ghi phiếu, ngày trả tiền.

Bảng	Tên cột	Kiểu dữ liệu	Ghi chú
KHACHHANG	<u>MAKH</u>	CHAR(5)	Mã khách hàng
	HOTEN	VARCHAR(50)	Họ tên khách hàng
	DIACHI	VARCHAR(100)	Địa chỉ
	NGSINH	SMALLDATETIME	Ngày tháng năm sinh của khách hàng
	EMAIL	VARCHAR(70)	Email của khách hàng
HANGXE	<u>MAHX</u>	CHAR(5)	Mã hãng xe
	TENHX	VARCHAR(40)	Tên hãng xe
XE	<u>MAXE</u>	CHAR(5)	Mã xe
	NAMSX	INT	Năm sản xuất
	MAHX	CHAR(5)	Mã hãng xe
VEXE	<u>MAVE</u>	CHAR(5)	Mã vé xe
	MOTA	VARCHAR(60)	Mô tả của vé xe
PHIEUDOXE	<u>MAPHIEU</u>	CHAR(5)	Mã phiếu
	MAKH	CHAR(5)	Mã khách hàng
	MAXE	CHAR(5)	Mã xe

	MAVE	CHAR(5)	Mã vé
	NGAYGHIPHIEU	SMALLDATETIME	Ngày ghi phiếu
	NGAYTRATIEN	SMALLDATETIME	Ngày trả tiền đỗ xe

A. Viết câu lệnh DDL, DML:

1. Tạo database, bảng với kiểu dữ liệu của các thuộc tính được mô tả như trên. (Phải có đầy đủ khóa chính, khóa ngoại).
2.
 - a. Thêm thuộc tính NAMTL có kiểu dữ liệu VARCHAR(10) vào bảng HANGXE.
 - b. Thay đổi thuộc tính NAMTL của HANGXE thành INT.
 - c. Xóa thuộc tính NAMTL của bảng HANGXE.
3.
 - a. Thêm một dòng dữ liệu vào bảng VEXE.
 - b. Thay đổi mô tả (MOTA) của VEXE đã được thêm ở câu a thành 'Lan xe buýt'.
 - c. Xóa dòng dữ liệu vừa mới thêm ở câu a ra khỏi bảng VEXE.

B. Viết các câu lệnh SQL thực hiện các truy vấn sau:

1. Hiển thị những khách hàng (makh, hoten, diachi) sinh vào tháng 6 năm 2000. Sắp xếp theo hoten khách hàng.
2. Hiển thị những phiếu đỗ xe (maphieu, makh, ngayghiphieu) được ghi trong năm 2001 hoặc năm 2002. Sắp xếp theo mã khách hàng.
3. Xe nào được sản xuất trong khoảng thời gian từ năm 1990 đến năm 2000. Sắp xếp theo năm sản xuất.
4. Khách hàng nào (MAKH) có phiếu đỗ xe của hãng có tên hãng xe là 'Toyota'.
5. Xe nào (MAXE, NAMSX, TENHX) có phiếu đỗ xe vào ngày '10/10/2020'.
6. Khách hàng nào (MAKH, HOTEN) vừa có phiếu đỗ xe vào năm 2020 vừa có phiếu đỗ xe vào năm 2021.
7. Xe nào (MAXE, NAMSX) có phiếu đỗ xe trong cả hai ngày: ngày 1 và ngày 25 tháng tháng 9 năm 2020.
8. Khách hàng nào (MAKH, HOTEN) có phiếu đỗ xe vào tháng 7 năm 2000 nhưng không có phiếu đỗ xe vào tháng 8 năm 2000.
9. Hiển thị thông tin khách hàng (MAKH, HOTEN) và số lượng phiếu đỗ của khách hàng này trong năm 1999. Chỉ hiển thị những khách hàng có số lượng phiếu đỗ xe lớn hơn 9.
10. Hiển thị thông tin xe (MAXE, MAHX) sản xuất năm 2000 và số lượng phiếu đỗ xe của xe này. Chỉ hiển thị những xe có số lượng phiếu đỗ xe nhiều hơn 3.
11. Mỗi năm có bao nhiêu phiếu đỗ xe của mã hãng xe 'TYT'. Thông tin hiển thị gồm NAM, SLXE. Chỉ hiển thị những năm có số lượng phiếu đỗ xe nhiều hơn 20.

12. Mỗi ngày có bao nhiêu phiếu đỗ xe của khách hàng sinh năm 1990. Thông tin hiển thị gồm: NGAYGHIPHIEU, SLPHIEU. Chỉ hiển thị những ngày có phiếu đỗ xe nhiều hơn 10.
 13. Năm 2021, mã xe nào có nhiều phiếu đỗ xe nhất.
 14. Năm 2019, mã khách hàng nào có nhiều phiếu đỗ xe nhất.
 15. Mã khách hàng nào có nhiều phiếu đỗ xe chưa trả tiền nhất (phiếu đỗ xe chưa trả tiền thì có NGAYTRATIEN IS NULL).
 16. Trong năm 2017, tháng nào có nhiều phiếu đỗ xe nhất.
 17. Khách hàng nào (MAKH, HOTEN) có phiếu đỗ xe của tất cả các xe của mã hãng xe là 'TYT'.
 18. Khách hàng nào (MAKH, HOTEN, DIACHI) có phiếu đỗ xe của tất cả các xe sản xuất năm 1990.
 19. Mỗi năm, ứng với mỗi lứa tuổi của khách hàng có bao nhiêu phiếu đỗ xe. Lứa tuổi của khách hàng có ba nhóm như sau: nhỏ hơn 30, từ 30 đến 55, lớn hơn 55.
- C. Ràng buộc toàn vẹn:
1. Ngày sinh của một khách hàng phải nhỏ hơn ngày hiện tại.
 2. Ngày trả tiền phải lớn hơn ngày ghi phiếu đỗ xe.
 3. Năm sản xuất của xe phải nhỏ hơn năm ghi trên phiếu đỗ xe của xe này. Chỉ yêu cầu viết trigger cho sự kiện thêm một phiếu đỗ xe và sửa năm sản xuất của xe.
 4. Ngày sinh của khách hàng phải nhỏ hơn ngày ghi phiếu đỗ xe của khách hàng này. Chỉ yêu cầu viết trigger cho sự kiện **thêm** và **sửa** phiếu đỗ xe.

III. Bài tập thực hành tự học (DATABASE: EMPDB4, Database được cho sẵn bên dưới)

1. Simple

- 1.1 Find the employee whose salary is greater than 31000 and gender is M
- 1.2 List all projects in plocation='Houston'. Order by project name.
- 1.3 List all dependents who has relationship='Daughter' and was born in 1988
- 1.4 Find all employees who were born in December 1955.
- 1.5 Find all employees who were born in 1937 and salary > 20000
- 1.6 Find all employees who work at DNo=5 and have salary>35000
- 1.7 Find projects of department number 5 (Dnum=5) located at location 'Houston'

2. JOIN, distinct

- 2.1 Find all employees (Ssn, Name, Bdate) who work on project named 'Reorganization'. Order the result by Lname.
- 2.2 Find employees of department named 'Research' who work on Project Pno=2
- 2.3 Find Project of department named 'Research' that has employee '333445555' works on
- 2.4 Find employees who have sex='M' and work on project located in 'Houston'. Make sure no duplicates appear in your output.

2.5 Find employees who have dependent relationship ='Son' and work at department name='Research'

2.6 Find dependent who is the spouse of employee worked at department named 'Administration'

3. SET

3.1 Find employee (Essn, Fname, Lname) who works on both project Pno=10 AND Pno=30.

3.2 Find employee (Esn) who works on project named 'ProductY' but does not work on project named 'Reorganization'

3.3 Find project (Pnumber, Pname) that has BOTH employees Essn='123456789' and Essn='453453453' work on.

3.4 Find project (Pnumber, Pname) that employee Essn='987654321' works on this project but employee Essn='666884444' does not.

4. GROUP BY, HAVING

4.1 How many projects does each employee work on? Show the Ssn, Fname, Lname, Project_count. Order the result by Lname.

4.2 Which employees work more than 45 hours in total. Show the Essn and total_Work.

4.3 Which location has more than 1 project. Show the Plocation and Project_count.

4.4 How many dependences that each employee has. Show Ssn, Fname, Lname and dependence_count

4.5 How many employees gender 'M' that each department has? Show Dnumber, Dname and number of male employee.

4.6 Which employee has more than 2 dependences? Show Ssn, Fname, Lname, NumOfDependences.

5. TOP

5.1 Which employee (Snn, Name, Salary) of gender male has the most salary. Fname and Lname concatenated into Name

5.2 Which gender has the most employers.

5.3 Which Department (Dno) has the most employees.

5.4 Which employee (Snn, FName, LName, Salary) has the smallest salary.

5.5 Which employee (Snn, FName, LName, Salary) has the most dependents.

5.6 Which location has the most Project?

6. CASE

6.1 Show employee (Ssn, Fname, Lname, Sex) and detail of the gender of this employee. The output will look like:

6.2 Show employee (Ssn, Fname, Lname, Salary) and the discription for the salary:

If salary>30000 then 'High salary'

Else 'Low salary'

The output will look like:

7. CHECK

7.1 Write a CHECK constraint to make sure: Data of hours column of table WORKS_ON must be more than 0 (hours >0).

7.2 gender of an employee must be 'M' or 'F'.

7.3 Write a CHECK constraint to make sure: Salary of an employee must be greater than 0.

8. TRIGGER

8.1 With a daughter relationship, the birth date of the daughter must be less than the birth date of this daughter's employee. Write a trigger to insert data into Dependent table.

8.2 With a son relationship, the birth date of the employee must be greater than the birth date of the son of this employee. Write a trigger to update data in employee table.

-----SCRIPT DATABASE-----

```
Create DATABASE EMPDB4
GO
USE EMPDB4
GO
```

```
CREATE TABLE DEPARTMENT(
Dnumber int NOT NULL PRIMARY KEY,
Dname varchar(15) NOT NULL)
```

```
CREATE TABLE DEPENDENT(
Essn char(9) NOT NULL,
Dependent_name varchar(15) NOT NULL,
Sex char(1) NULL,
Bdate date NULL,
Relationship varchar(8) NULL,
PRIMARY KEY (Essn, Dependent_name))
```

```
CREATE TABLE EMPLOYEE(
Ssn char(9) NOT NULL PRIMARY KEY,
Fname varchar(15) NOT NULL,
Lname varchar(15) NOT NULL,
Bdate date NULL,
Address varchar(30) NULL,
Sex char(1) NULL,
Salary decimal(10, 2) NULL,
```

Dno int NOT NULL)

```
CREATE TABLE PROJECT(  
Pnumber int NOT NULL PRIMARY KEY,  
Pname varchar(15) NOT NULL,  
Plocation varchar(15) NULL,  
Dnum int NOT NULL)
```

```
CREATE TABLE WORKS_ON(  
Essn char(9) NOT NULL,  
Pno int NOT NULL,  
Hours decimal(3, 1) NULL,  
PRIMARY KEY (Essn, Pno))  
GO
```

```
INSERT DEPARTMENT (Dnumber, Dname) VALUES (4, N'Administration')  
INSERT DEPARTMENT (Dnumber, Dname) VALUES (1, N'Headquarters')  
INSERT DEPARTMENT (Dnumber, Dname) VALUES (5, N'Research')  
GO
```

```
INSERT DEPENDENT (Essn, Dependent_name, Sex, Bdate, Relationship) VALUES  
(N'123456789', N'Alice', N'F', CONVERT(varchar, '1988-12-30', 23), N'Daughter')  
INSERT DEPENDENT (Essn, Dependent_name, Sex, Bdate, Relationship) VALUES  
(N'123456789', N'Elizabeth', N'F', CONVERT(varchar, '1967-05-05', 23), N'Spouse')  
INSERT DEPENDENT (Essn, Dependent_name, Sex, Bdate, Relationship) VALUES  
(N'123456789', N'Michael', N'M', CONVERT(varchar, '1988-01-04', 23), N'Son')  
INSERT DEPENDENT (Essn, Dependent_name, Sex, Bdate, Relationship) VALUES  
(N'333445555', N'Alice', N'F', CONVERT(varchar, '1986-04-05', 23), N'Daughter')  
INSERT DEPENDENT (Essn, Dependent_name, Sex, Bdate, Relationship) VALUES  
(N'333445555', N'Joy', N'F', CONVERT(varchar, '1958-05-03', 23), N'Spouse')  
INSERT DEPENDENT (Essn, Dependent_name, Sex, Bdate, Relationship) VALUES  
(N'333445555', N'Theodore', N'M', CONVERT(varchar, '1983-10-25', 23), N'Son')  
INSERT DEPENDENT (Essn, Dependent_name, Sex, Bdate, Relationship) VALUES  
(N'987654321', N'Abner', N'M', CONVERT(varchar, '1942-02-28', 23), N'Spouse')  
GO
```

```
INSERT EMPLOYEE (Ssn, Fname, Lname, Bdate, Address, Sex, Salary, Dno) VALUES  
(N'123456789', N'John', N'Smith', CONVERT(varchar, '1965-01-09', 23), N'731 Fondren,  
Houston,TX', N'M', 30000.00, 5)  
INSERT EMPLOYEE (Ssn, Fname, Lname, Bdate, Address, Sex, Salary, Dno) VALUES  
(N'333445555', N'Franklin', N'Wong', CONVERT(varchar, '1955-12-08', 23), N'638 Boss,  
Houston, TX', N'M', 55000.00, 5)
```

```
INSERT EMPLOYEE (Ssn, Fname, Lname, Bdate, Address, Sex, Salary, Dno) VALUES
(N'453453453', N'Joyce', N'English', CONVERT(varchar, '1972-07-31', 23), N'5631 Rice,
Houston, TX', N'F', 25000.00, 5)
```

```
INSERT EMPLOYEE (Ssn, Fname, Lname, Bdate, Address, Sex, Salary, Dno) VALUES
(N'666884444', N'Ramesh', N'Narayan', CONVERT(varchar, '1962-09-15', 23), N'975 Fire
Oak, Humble, TX', N'M', 38000.00, 5)
```

```
INSERT EMPLOYEE (Ssn, Fname, Lname, Bdate, Address, Sex, Salary, Dno) VALUES
(N'888665555', N'James', N'Borg', CONVERT(varchar, '1937-11-10', 23), N'450 Stone,
Houston, TX', N'M', 55000.00, 1)
```

```
INSERT EMPLOYEE (Ssn, Fname, Lname, Bdate, Address, Sex, Salary, Dno) VALUES
(N'987654321', N'Jennifer', N'Wallace', CONVERT(varchar, '1941-06-20', 23), N'291 Berry,
Bellaire, TX', N'F', 43000.00, 4)
```

```
INSERT EMPLOYEE (Ssn, Fname, Lname, Bdate, Address, Sex, Salary, Dno) VALUES
(N'987987987', N'Ahmad', N'Jabbar', CONVERT(varchar, '1969-03-29', 23), N'980 Dallas,
Houston, TX', N'M', 25000.00, 4)
```

```
INSERT EMPLOYEE (Ssn, Fname, Lname, Bdate, Address, Sex, Salary, Dno) VALUES
(N'999887777', N'Alicia', N'Zelaya', CONVERT(varchar, '1968-01-19', 23), N'3321 Castle,
Spring, TX', N'F', 25000.00, 4)
```

```
INSERT EMPLOYEE (Ssn, Fname, Lname, Bdate, Address, Sex, Salary, Dno) VALUES
(N'999887772', N'Nika', N'Zelaya', CONVERT(varchar, '1968-01-19', 23), N'3321 Castle,
Spring, TX', N'F', 25000.00, 4)
```

GO

```
INSERT PROJECT (Pnumber, Pname, Plocation, Dnum) VALUES (1, N'ProductX',
N'Bellaire', 5)
```

```
INSERT PROJECT (Pnumber, Pname, Plocation, Dnum) VALUES (2, N'ProductY',
N'Sugarland', 5)
```

```
INSERT PROJECT (Pnumber, Pname, Plocation, Dnum) VALUES (3, N'ProductZ',
N'Houston', 5)
```

```
INSERT PROJECT (Pnumber, Pname, Plocation, Dnum) VALUES (10, N'Computerization',
N'Stafford', 4)
```

```
INSERT PROJECT (Pnumber, Pname, Plocation, Dnum) VALUES (20, N'Reorganization',
N'Houston', 1)
```

```
INSERT PROJECT (Pnumber, Pname, Plocation, Dnum) VALUES (30, N'Newbenefits',
N'Stafford', 4)
```

GO

```
INSERT WORKS_ON (Essn, Pno, Hours) VALUES (N'123456789', 1, 52.5)
```

```
INSERT WORKS_ON (Essn, Pno, Hours) VALUES (N'123456789', 2, 7.5)
```

```
INSERT WORKS_ON (Essn, Pno, Hours) VALUES (N'333445555', 2, 10.0)
```

```
INSERT WORKS_ON (Essn, Pno, Hours) VALUES (N'333445555', 3, 10.0)
```



```

INSERT WORKS_ON (Essn, Pno, Hours) VALUES (N'333445555', 10, 10.0)
INSERT WORKS_ON (Essn, Pno, Hours) VALUES (N'333445555', 20, 10.0)
INSERT WORKS_ON (Essn, Pno, Hours) VALUES (N'453453453', 1, 20.0)
INSERT WORKS_ON (Essn, Pno, Hours) VALUES (N'453453453', 2, 20.0)
INSERT WORKS_ON (Essn, Pno, Hours) VALUES (N'666884444', 3, 40.0)
INSERT WORKS_ON (Essn, Pno, Hours) VALUES (N'888665555', 20, NULL)
INSERT WORKS_ON (Essn, Pno, Hours) VALUES (N'987654321', 20, 15.0)
INSERT WORKS_ON (Essn, Pno, Hours) VALUES (N'987654321', 30, 20.0)
INSERT WORKS_ON (Essn, Pno, Hours) VALUES (N'987987987', 10, 35.0)
INSERT WORKS_ON (Essn, Pno, Hours) VALUES (N'987987987', 30, 25.0)
INSERT WORKS_ON (Essn, Pno, Hours) VALUES (N'999887777', 10, 10.0)
INSERT WORKS_ON (Essn, Pno, Hours) VALUES (N'999887777', 30, 30.0)
GO
ALTER TABLE DEPENDENT WITH CHECK ADD CONSTRAINT
DependentRelatedToEmployeeFK FOREIGN KEY(Essn) REFERENCES EMPLOYEE
(Ssn)
GO
ALTER TABLE DEPENDENT CHECK CONSTRAINT DependentRelatedToEmployeeFK
GO
ALTER TABLE EMPLOYEE WITH CHECK ADD CONSTRAINT
EmployeeWorksInDepartmentFK FOREIGN KEY(Dno) REFERENCES DEPARTMENT
(Dnumber)
GO
ALTER TABLE EMPLOYEE CHECK CONSTRAINT EmployeeWorksInDepartmentFK
GO
ALTER TABLE PROJECT WITH CHECK ADD CONSTRAINT
ProjectControlledByDepartmentrFK FOREIGN KEY(Dnum) REFERENCES
DEPARTMENT (Dnumber)
GO
ALTER TABLE PROJECT CHECK CONSTRAINT ProjectControlledByDepartmentrFK
GO
ALTER TABLE WORKS_ON WITH CHECK ADD CONSTRAINT
EmployeeWorksOnProjectFK FOREIGN KEY(Essn) REFERENCES EMPLOYEE (Ssn)
GO
ALTER TABLE WORKS_ON CHECK CONSTRAINT EmployeeWorksOnProjectFK
GO
ALTER TABLE WORKS_ON WITH CHECK ADD CONSTRAINT
ProjectWorkedOnByEmployeeFK FOREIGN KEY(Pno) REFERENCES PROJECT
(Pnumber)

```


GO

ALTER TABLE WORKS_ON CHECK CONSTRAINT ProjectWorkedOnByEmployeeFK

GO