

## BÀI TẬP HƯỚNG DẪN THỰC HÀNH MÔN CƠ SỞ DỮ LIỆU PHẦN CREATE DB, TABLE, Ràng BUỘC, SQL

### I. Bài tập

#### Bài 1. Quản lý điểm sinh viên

Các field in đậm và gạch dưới là khoá chính của bảng. Tạo bảng theo định nghĩa sau

##### KETQUA

<i>Field Name</i>	<i>Field Type</i>	<i>Field Size</i>	<i>Description</i>
<b><u>MASV</u></b>	Char	3	
<b><u>MAMH</u></b>	Char	2	
<b><u>LANTHI</u></b>	Tinyint		
DIEM	Decimal	4,2	

##### DMKHOA

<i>Field Name</i>	<i>Field Type</i>	<i>Field Size</i>	<i>Description</i>
<b><u>MAKHOA</u></b>	Char	2	
TENKHOA	nVarChar	20	

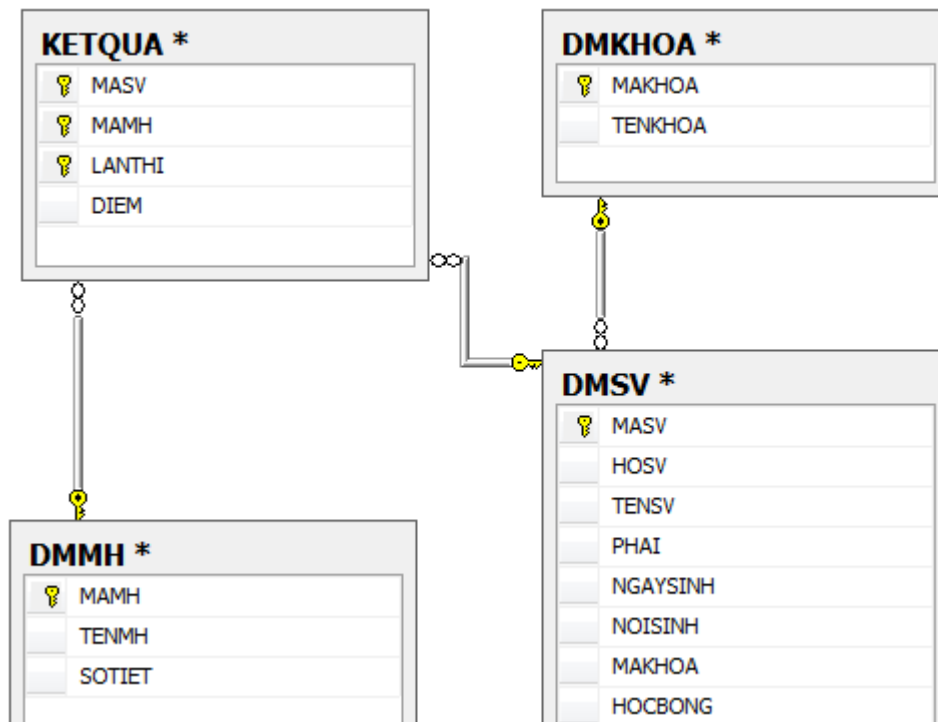
##### DMMH

<i>Field Name</i>	<i>Field Type</i>	<i>Field Size</i>	<i>Description</i>
<b><u>MAMH</u></b>	Char	2	
TENMH	nVarchar	30	
SOTIET	Tinyint		

##### DMSV

<i>Field Name</i>	<i>Field Type</i>	<i>Field Size</i>	<i>Description</i>
<b><u>MASV</u></b>	Char	3	
HOSV	nVarchar	30	
TENSV	nVarChar	10	
PHAI	bit		
NGAYSINH	Date/Time		
NOISINH	nVarchar	25	
MAKHOA	Char	2	
HOCBONG	float		

Kết quả quan hệ thể hiện như sau:



Dữ liệu mẫu

DMMH		
MaMH	TenMH	SoTiet
01	Cơ sở dữ liệu	45
02	Trí tuệ nhân tạo	45
03	Truyền tin	45
04	Đồ hoạ	60
05	Văn phạm	60
06	Kỹ thuật lập trình	45

DMKHOA	
MaKhoa	TenKH
AV	Anh Văn
TH	Tin Học
TR	Triết
VL	Vật Lý

DMSV							
MaSV	HoSV	TenSV	Phai	NgaySinh	NoiSinh	MaKhoa	HocBong
A01	Nguyễn thị	Hải	1	23/02/1993	Hà Nội	TH	130000
A02	Trần văn	Chính	0	24/12/1992	Bình Định	VL	150000
A03	Lê thu bạch	Yến	1	21/02/1993	Tp HCM	TH	170000
A04	Trần anh	Tuấn	0	20/12/1994	Hà Nội	AV	80000
B01	Trần thanh	Mai	1	12/08/1993	Hải Phòng	TR	0
B02	Trần thị thu	Thủy	1	02/01/1994	Tp HCM	AV	0

(Ghi chú: Phái: 1- nữ; 0- nam)

KETQUA			
MaSV	MaMH	LanThi	Diem
A01	01	1	3
A01	01	2	6

A01	02	2	6
A01	03	1	5
A02	01	1	4.5
A02	01	2	7
A02	03	1	10
A02	05	1	9
A03	01	1	2
A03	01	2	5
A03	03	1	2.5
A03	03	2	4
A04	05	2	10
B01	01	1	7
B01	03	1	2.5
B01	03	2	5
B02	02	1	6
B02	04	1	10

### Tham khảo:

```
alter table ketqua add constraint kq_pk primary key (masv, mamh, lanthi)
alter table ketqua add constraint kq_masv_fk foreign key(masv) references dmsv(masv)
alter table ketqua add constraint kq_mamh_fk foreign key(mamh) references dmmh(mamh)
```

```
set dateformat DMY insert into .....values (,....., '23/02/2000',...)
```

## 2. Dùng Query Analyzer

### 2.1. Tạo CSDL bằng ngôn ngữ DDL, ràng buộc khóa chính, khóa ngoại.

- Cú pháp tạo cơ sở dữ liệu

**Create Database <Ten\_database>**

- Cú pháp mở CSDL

**USE <tên DataBase>**

- Cú pháp tạo bảng

**Create Table <Tên\_bảng>**

(

<Tên\_cột><Kiểu\_dữ\_liệu> [RBTv],

<Tên\_cột><Kiểu\_dữ\_liệu> [RBTv],

...

[<RBTv>]

)

- Cú pháp thêm ràng buộc

```
ALTER TABLE <tên_bảng> ADD  
    Constraint <Ten_RBTV> <RBTV>,  
    Constraint <Ten_RBTV> <RBTV>,  
    ...
```

## 2.2. Sử dụng sp hệ thống kiểm tra cấu trúc dữ liệu:

- Kiểm tra các bảng: **sp\_tables**
- Kiểm tra cấu trúc bảng: **sp\_columns** <tên\_bảng>
- Kiểm tra khóa chính: **sp\_pkeys** <tên\_bảng>
- Kiểm tra các ràng buộc trong bảng: **sp\_helpconstraint** <tên\_bảng>

## 2.3. Thay đổi cấu trúc dữ liệu dùng ngôn ngữ DDL

- Thêm thuộc tính (cột)

```
ALTER TABLE <Tên_bảng> ADD <Tên_cột>  
<Kiểu_dữ_liệu> [<RBTV>]
```

- Xóa thuộc tính (cột)

```
ALTER TABLE <Tên_bảng> DROP COLUMN <Tên_cột>
```

- Mở rộng thuộc tính (cột)

```
ALTER TABLE <Tên_bảng> ALTER COLUMN  
    <Tên_cột> <Kiểu_dữ_liệu_mới>
```

- Xóa ràng buộc

```
ALTER TABLE <Tên_bảng> DROP <Tên_RBTV>
```

- Đổi tên

```
EXEC sp_rename 'Tên_bảng.tên_cột',  
    'Tên_mới' [, 'COLUMN']
```

- 3.1 Thêm vào DMKHOA thuộc tính NamTL(năm thành lập) có kiểu dữ liệu là int.

```
ALTER TABLE DMKHOA ADD NamTL int
```

- 3.2 Thay đổi kiểu dữ liệu NamTL thành smallInt

```
ALTER TABLE DMKHOA
```

```
ALTER COLUMN NamTL smallint
```

- 3.3 Đổi tên NamTL thành NamThanhLap

```
EXEC sp_rename 'DMKHOA.NamTL', 'NamThanhLap'
```

- 3.4 Xóa thuộc tính NamThanhLap

```
ALTER TABLE DMKHOA DROP COLUMN NamThanhLap
```

3.5 Xóa ràng buộc khóa ngoại giữa sinh viên và khoa

```
ALTER TABLE DMSV DROP fk_dmsv_khoa
```

(chú ý: fk\_dmsv\_khoa là tên khóa ngoại do người dùng đặt)

3.6 Tạo ràng buộc khóa ngoại giữa sinh viên và khoa

```
ALTER TABLE DMSV ADD CONSTRAINT fk_dmsv_khoa FOREIGN KEY(MaKH)  
REFERENCES DMKHOA (MaKH)
```

## 2.4. Thực hiện thao tác dữ liệu (DML): insert, update, delete đơn giản

- Cú pháp thêm một dòng

**INSERT INTO** <tên bảng>(<danh sách các thuộc tính>)  
<câu truy vấn con>

- Cú pháp xóa

**DELETE FROM** <tên bảng>  
**[WHERE** <điều kiện>]

- Cú pháp sửa

**UPDATE** <tên bảng>  
**SET** <tên thuộc tính>=<giá trị mới>,  
    <tên thuộc tính>=<giá trị mới>,  
    ...  
**[WHERE** <điều kiện>]

4.1. Thêm tất cả dữ liệu bằng lệnh insert.

Câu lệnh:     INSERT INTO DMMH  
                  VALUES ('01', N'Cơ sở dữ liệu', 45)

4.2. Cập nhật số tiết của môn Văn phạm thành 45 tiết.

Câu lệnh:     UPDATE DMMH  
                  SET SoTiet = 45  
                  WHERE TenMH = N'Văn phạm'

4.3. Cập nhật tên của sinh viên Trần Thanh Mai thành Trần Thanh Kỳ.

4.4. Cập nhật phái của sinh viên Trần Thanh Kỳ thành phái Nam.

4.5. Cập nhật ngày sinh của sinh viên Trần thị thu Thủy thành 05/07/1990.

4.6. Tăng học bổng cho tất cả những sinh viên có mã khoa "AV" thêm 100,000.

Câu lệnh:     UPDATE DMSV  
                  SET HocBong = HocBong + 100000  
                  WHERE MaKH = 'AV'

4.7. Xóa tất cả những dòng có điểm thi lần 2 nhỏ hơn 5 trong bảng KETQUA.

Câu lệnh:     DELETE FROM KETQUA  
                  WHERE LANTHI = 2 AND DIEM < 5

4.8. Xóa những sinh viên không có học bổng. (Xóa được không? lý do).

## C. Truy vấn SQL

### 1. Truy vấn đơn giản

- 1.1. Danh sách các môn học có tên bắt đầu bằng chữ T, gồm các thông tin: Mã môn, Tên môn, Số tiết.

Câu lệnh:     SELECT MaMH, TenMH, SoTiet  
                  FROM DMMH  
                  WHERE TenMH like N'T%'

- 1.2. Liệt kê danh sách những sinh viên có chữ cái cuối cùng trong tên là I, gồm các thông tin: Họ tên sinh viên, Ngày sinh, Giới.
- 1.3. Danh sách những khoa có ký tự thứ hai của tên khoa có chứa chữ N, gồm các thông tin: Mã khoa, Tên khoa.
- 1.4. Liệt kê những sinh viên mà họ có chứa chữ Thị.
- 1.5. Cho biết danh sách những sinh viên có ký tự đầu tiên của tên nằm trong khoảng từ a đến m, gồm các thông tin: Mã sinh viên, Họ tên sinh viên, Giới, Học bổng.
- 1.6. Liệt kê các sinh viên có học bổng từ 150,000 trở lên và sinh ở Hà Nội, gồm các thông tin: Họ tên sinh viên, Mã khoa, Nơi sinh, Học bổng.
- 1.7. Danh sách các sinh viên của khoa AV văn và khoa VL, gồm các thông tin: Mã sinh viên, Mã khoa, Giới.
- 1.8. Cho biết những sinh viên có ngày sinh từ ngày 01/01/1992 đến ngày 05/06/1993 gồm các thông tin: Mã sinh viên, Ngày sinh, Nơi sinh, Học bổng.
- 1.9. Danh sách những sinh viên có học bổng từ 80.000 đến 150.000, gồm các thông tin: Mã sinh viên, Ngày sinh, Giới, Mã khoa.
- 1.10. Cho biết những môn học có số tiết lớn hơn 30 và nhỏ hơn 45, gồm các thông tin: Mã môn học, Tên môn học, Số tiết.
- 1.11. Liệt kê những sinh viên nam của khoa Anh văn và khoa tin học, gồm các thông tin: Mã sinh viên, Họ tên sinh viên, tên khoa, Giới.
- 1.12. Liệt kê những sinh viên có điểm thi môn sơ sở dữ liệu nhỏ hơn 5, gồm thông tin: Mã sinh viên, Họ tên, giới, điểm
- 1.13. Liệt kê những sinh viên học khoa Anh văn mà không có học bổng, gồm thông tin: Mã sinh viên, Họ và tên, tên khoa, Nơi sinh, Học bổng.

### 1. Sắp xếp (Order By)

<p><b>SELECT &lt;danh sách thuộc tính&gt;</b> <b>FROM     &lt;danh sách các bảng&gt;</b> <b>WHERE &lt;điều kiện&gt;</b> <b>ORDER BY &lt;thuộc tính&gt; ACS DESC, &lt;thuộc tính&gt; ACS DESC</b></p>
--

*ASC : sắp xếp tăng, DESC : sắp xếp giảm*

- 2.1. Cho biết danh sách những sinh viên mà tên có chứa ký tự nằm trong khoảng từ a đến m, gồm các thông tin: Họ tên sinh viên, Ngày sinh, Nơi sinh, Học bổng. Danh sách được sắp xếp tăng dần theo tên sinh viên.

Câu lệnh: SELECT HoSV+ ‘ ‘+TenSV as HoTenSV, NgaySinh, NoiSinh, HocBong  
FROM DMSV  
WHERE TenSV like ‘%[a-m]’  
ORDER BY TenSV ASC

- 2.2. Liệt kê danh sách sinh viên, gồm các thông tin sau: Mã sinh viên, Họ sinh viên, Tên sinh viên, Học bổng. Danh sách sẽ được sắp xếp theo thứ tự Mã sinh viên tăng dần.
- 2.3. Danh sách các sinh viên gồm thông tin sau: Mã sinh viên, họ tên sinh viên, Phái, Ngày sinh. Danh sách sẽ được sắp xếp theo thứ tự Nam/Nữ.
- 2.4. Thông tin các sinh viên gồm: Họ tên sinh viên, Ngày sinh, Học bổng. Thông tin sẽ được sắp xếp theo thứ tự Ngày sinh tăng dần và Học bổng giảm dần.
- 2.5. Cho biết danh sách các sinh viên có học bổng lớn hơn 100,000, gồm các thông tin: Mã sinh viên, Họ tên sinh viên, Mã khoa, Học bổng. Danh sách sẽ được sắp xếp theo thứ tự Mã khoa giảm dần.

### 3. Truy vấn sử dụng hàm: year, month, day, getdate, case, ....

- 3.1. Danh sách sinh viên có nơi sinh ở Hà Nội và sinh vào tháng 02, gồm các thông tin: Họ sinh viên, Tên sinh viên, Nơi sinh, Ngày sinh.

Câu lệnh: SELECT HoSV, TenSV, NoiSinh, NgaySinh  
FROM DMSV  
WHERE NoiSinh like N’Hà Nội’ AND MONTH(NgaySinh) = 2

- 3.2. Cho biết những sinh viên có tuổi lớn hơn 20, thông tin gồm: Họ tên sinh viên, Tuổi, Học bổng.

Hướng dẫn: Tuổi = YEAR(GETDATE()) – YEAR(NgaySinh)

- 3.3. Danh sách những sinh viên có tuổi từ 20 đến 25, thông tin gồm: Họ tên sinh viên, Tuổi, Tên khoa.
- 3.4. Danh sách sinh viên sinh vào mùa xuân năm 1990, gồm các thông tin: Họ tên sinh viên, Phái, Ngày sinh. (dùng hàm datepart(“q”,ngaysinh))

#### - Cú pháp case...when

<p><b>CASE</b> &lt;tên cột&gt;</p> <p>    <b>WHEN</b> &lt;giá trị&gt; <b>THEN</b> &lt;biểu thức&gt;</p> <p>    <b>WHEN</b> &lt;giá trị&gt; <b>THEN</b> &lt;biểu thức&gt;</p> <p>    ...</p> <p>    <b>[ELSE</b> &lt;biểu thức&gt; <b>]</b></p> <p><b>END</b></p>
--

- 3.5. Cho biết thông tin về mức học bổng của các sinh viên, gồm: Mã sinh viên, Phái, Mã khoa, Mức học bổng. Trong đó, mức học bổng sẽ hiển thị là “Học bổng cao” nếu giá trị của học bổng lớn hơn 150,000 và ngược lại hiển thị là “Mức trung bình”

Câu lệnh: SELECT MaSV, Phai, MaKH,  
MucHocBong =

CASE

WHEN HocBong > 500000 THEN 'Hoc bong cao'  
ELSE 'Muc trung binh'

END

FROM DMSV

- 3.6. Cho biết kết quả điểm thi của các sinh viên, gồm các thông tin: Họ tên sinh viên, Mã môn học, lần thi, điểm, kết quả (nếu điểm nhỏ hơn 5 thì rút ngược lại đậu).

**4. Truy vấn sử dụng hàm kết hợp: max, min, count, sum, avg và gom nhóm**

**SELECT** <danh sách các cột>  
**FROM** <danh sách các bảng>  
**WHERE** <điều kiện>  
**GROUP BY** <danh sách các cột gom nhóm>

- 4.1. Cho biết tổng số sinh viên của toàn trường.

Câu lệnh:

```
SELECT    Count(*)    AS    SLSV
FROM      DMSV
```

- 4.2. Cho biết tổng sinh viên và tổng sinh viên nữ.

- 4.3. Cho biết tổng số sinh viên của **từng** khoa.

Câu lệnh:     SELECT s.MaKhoa, TenKH, COUNT(MaSV) As SoSV  
                  FROM DMSV s, DMKHOA k  
                  WHERE s.MaKhoa = k.MaKhoa  
                  GROUP BY s.MaKhoa, TenKH

- 4.4. Cho biết số lượng sinh viên học **từng** môn (dùng *Distinct* loại trùng nhau)

Câu lệnh:     SELECT M.MaMH, TenMH, COUNT(Distinct MaSV) As SoMH  
                  FROM DMMH M, KETQUA K  
                  WHERE M.MaMH = K.MaMH  
                  GROUP BY M.MaMH, TenMH

- 4.5. Cho biết số lượng môn học mà **mỗi** sinh viên đã học.

- 4.6. Cho biết học bổng cao nhất của **mỗi** khoa.

- 4.7. Cho biết tổng số sinh viên nam và tổng số sinh viên nữ của **mỗi** khoa.

(*Hướng dẫn: dùng SUM kết hợp với CASE...*)

```
SELECT K.MAKHOA, TENKHOA,
SUM(CASE WHEN PHAI=0 THEN 1 ELSE 0 END ) AS TNAM,
SUM(CASE WHEN PHAI =1 THEN 1 ELSE 0 END ) AS TNU
FROM DMKHOA K, DMSV SV
WHERE K.MAKHOA=SV.MAKHOA
GROUP BY K.MAKHOA, TENKHOA
```

- 4.8. Cho biết số lượng sinh viên theo từng độ tuổi.

- 4.9. Cho biết số lượng sinh viên đậu và số lượng sinh viên rớt của từng môn trong lần thi 1.



## 5. Truy vấn theo điều kiện gom nhóm.

- Điều kiện trên nhóm

**SELECT** <danh sách các cột>  
**FROM** <danh sách các bảng>  
**WHERE** <điều kiện>  
**GROUP BY** <danh sách các cột gom nhóm>  
**HAVING** <điều kiện trên nhóm>

5.1. Cho biết những năm sinh nào có 2 sinh viên đang theo học tại trường.

Câu lệnh:     SELECT YEAR(NgaySinh) as NamSinh  
                  FROM DMSV  
                  GROUP BY YEAR(NgaySinh)  
                  HAVING COUNT(MASV) = 2

5.2. Cho biết những nơi nào có hơn 2 sinh viên đang theo học tại trường.

5.3. Cho biết những môn nào có trên 3 sinh viên dự thi.

5.4. Cho biết những sinh viên thi lại trên 2 lần.

5.5. Cho biết những sinh viên nam có điểm trung bình lần 1 trên 7.0

5.6. Cho biết danh sách các sinh viên rớt trên 2 môn ở lần thi 1.

5.7. Cho biết danh sách những khoa có nhiều hơn 2 sinh viên nam

5.8. Cho biết những khoa có 2 sinh đạt học bổng từ 100.000 đến 200.000.

5.9. Cho biết những sinh viên nam học trên từ 3 môn trở lên

## 6. Truy vấn con trỏ về một giá trị

**SELECT** <danh sách các cột>  
**FROM** <danh sách các bảng>  
**WHERE** <so sánh tập hợp> (  
          **SELECT** <danh sách các cột>  
          **FROM** <danh sách các bảng>  
          **WHERE** <điều kiện>)

**So sánh tập hợp:** =, >, >=, <, <=, <>

6.1. Cho biết sinh viên nào có học bổng cao nhất.

*B1. Tìm giá trị học bổng cao nhất (trả về một giá trị duy nhất).*

*B2. Lấy những sinh viên có học bổng bằng học bổng B1*

Câu lệnh:     SELECT \* FROM DMSV

                  WHERE HocBong = (SELECT MAX(HocBong) FROM DMSV)

6.2. Cho biết những sinh viên có điểm thi lần 1 môn cơ sở dữ liệu cao nhất.

- 6.3. Cho biết sinh viên khoa anh văn có tuổi lớn nhất.  
6.4. Cho biết sinh viên khoa anh văn học môn văn phạm có điểm thi lần 1 thấp nhất.  
6.5. Cho biết những sinh viên thi lần 2 môn cơ sở dữ liệu bằng điểm với sinh viên có mã A01 thi ở lần 1 môn cơ sở dữ liệu  
6.6. Cho biết sinh viên không học khoa anh văn có điểm thi môn phạm lớn hơn điểm thi văn phạm của sinh viên học khoa anh văn.  
6.7. Cho biết những sinh viên có học bổng lớn hơn tổng học bổng của sinh viên khoa Triết
- 7. Truy vấn con trả về nhiều giá trị, sử dụng lượng từ IN, ALL, ANY, UNION, TOP.**

**SELECT** <danh sách các cột>  
**FROM** <danh sách các bảng>  
**WHERE** <so sánh tập hợp> (  
    **SELECT** <danh sách các cột>  
    **FROM** <danh sách các bảng>  
    **WHERE** <điều kiện>)

**So sánh tập hợp: IN, ANY, ALL**

7.1 Cho biết sinh viên có nơi sinh cùng với Hải.

*B1. Tìm nơi sinh của Hải (câu con này trả về nhiều giá trị vì có thể nhiều người tên Hải)*

*B2. Tìm những sinh viên có nơi sinh giống với một trong những nơi sinh ở B1*

Câu lệnh:     SELECT \* FROM DMSV  
                  FROM DMSV  
                  WHERE NoiSinh IN (SELECT NoiSinh  
                                      FROM DMSV  
                              WHERE TENSX like N'Hải')  
                  AND TenSX not like N'Hải'

- 7.2 Cho biết những sinh viên có học bổng lớn hơn tất cả học bổng của sinh viên thuộc khoa anh văn  
7.3 Cho biết những sinh viên có học bổng lớn hơn bất kỳ học bổng của sinh viên học khóa anh văn  
7.4 Cho biết sinh viên có điểm thi môn cơ sở dữ liệu lần 2 lớn hơn tất cả điểm thi lần 1 môn cơ sở dữ liệu của những sinh viên khác.  
7.5 Với mỗi sinh viên cho biết điểm thi cao nhất của môn tương ứng.

```
SELECT SV.MASV, HOSV+ ' ' +TENSX AS HOTEN, TENMH, DIEM
FROM KETQUA K1, DMSV SV, DMMH MH
WHERE SV.MASV=K1.MASV
AND K1.MAMH=MH.MAMH
AND DIEM>=ALL(SELECT DIEM
               FROM KETQUA K2
               WHERE K1.MASV=K2.MASV)
```

7.6 Cho biết môn nào có nhiều sinh viên học nhất.

```
SELECT MH.MAMH, TENMH, COUNT(DISTINCT MASV) AS SLSV
FROM KETQUA KQ, DMMH MH
WHERE KQ.MAMH=MH.MAMH
GROUP BY MH.MAMH, TENMH
HAVING COUNT(DISTINCT MASV) >= ALL (SELECT COUNT(DISTINCT MASV)
                                     FROM KETQUA
                                     GROUP BY MAMH)
```

7.7 Cho biết những khoa có đông sinh viên nam học nhất.

7.8 Cho biết khoa nào có nhiều sinh viên nhận học bổng nhất.

7.9 Cho biết khoa nào có đông sinh viên nhận học bổng nhất và khoa nào khoa nào có ít sinh viên nhận học bổng nhất

7.10 Cho biết môn nào có nhiều sinh viên rớt lần 1 nhiều nhất.

7.11 Cho biết 3 sinh viên có học nhiều môn nhất.

Câu lệnh:

```
SELECT TOP 3 s.MaSV, HoSV, TenSV, COUNT(DISTINCT(MaMH)) as SoMon
FROM DMSV s, KETQUA k
WHERE s.MASV = k.MASV
GROUP BY s.MaSV, HoSV, TenSV
ORDER BY COUNT(DISTINCT(MaMH)) DESC
```

## 7. Phép trừ

```
SELECT <danh sách các cột>
FROM <danh sách các bảng>
WHERE <so sánh tập hợp> (
    SELECT <danh sách các cột>
    FROM <danh sách các bảng>
    WHERE <điều kiện>)
```

**So sánh tập hợp: NOT IN, NOT EXIST, <>ALL**

7.1. Cho biết sinh viên chưa thi môn cơ sở dữ liệu.

Câu lệnh:

```
SELECT MaSV, HoSV, TenSV
FROM DMSV
WHERE MaSV NOT IN ( SELECT k.MaSV
                    FROM DMMH m, KETQUA k
                    WHERE m.MaMH = k.MaMH AND TenMH
                    = N'Cơ sở dữ liệu')
```

**Ghi chú:** NOT IN tương đương với <>ALL

7.2. Cho biết sinh viên nào không thi lần 1 mà có dự thi lần 2.

7.3. Cho biết môn nào không có sinh viên khoa anh văn học.

7.4. Cho biết những sinh viên khoa anh văn chưa học môn văn phạm.

7.5. Cho biết những môn không có sinh viên khoa anh văn rớt .

7.6. Cho biết những khoa không có sinh viên nữ.

7.7. Cho biết những sinh viên:

- Học khoa anh văn có học bổng hoặc
- Chưa bao giờ rớt.

```
SELECT SV.MASV, HOSV, TENSX, SV.MAKHOA
FROM DMSV SV, DMKHOA K
WHERE SV.MAKHOA=K.MAKHOA
AND TENKHOA LIKE N'ANH VĂN'
AND HOCBONG>0
UNION
SELECT SV.MASV, HOSV, TENSX, MAKHOA
FROM DMSV SV, KETQUA K
WHERE SV.MASV=K.MASV
AND SV.MASV NOT IN (SELECT MASV
                     FROM KETQUA
                     WHERE DIEM<5)
```

7.8. Cho biết những sinh viên:

- Không có học bổng hoặc
- Bị rớt môn học (*sinh viên thi lần 1 bị rớt mà không thi lần 2 và sinh viên thi lần 2 bị rớt*)

## 8. Truy vấn dùng phép chia: $R:S \Leftrightarrow R-(S-R)$

```
SELECT R1.A, R1.B, R1.C
FROM R R1
WHERE NOT EXISTS (
    SELECT *
    FROM S
    WHERE NOT EXISTS (
        SELECT *
        FROM R R2
        WHERE R2.D=S.D AND R2.E=S.E
        AND R1.A=R2.A AND R1.B=R2.B AND R1.C=R2.C ))
```

8.1. Cho biết những môn được tất cả các sinh viên theo học.

Câu lệnh

```
SELECT *
FROM DMMH K1
```

```
WHERE NOT EXISTS ( SELECT * FROM DMSV S
                    WHERE NOT EXISTS(SELECT *
                                      FROM KETQUA K2
                                      WHERE k2.MaSV = s.MaSV
                                      AND k2.MaMH = K1.MaMH))
```

8.2. Cho biết những sinh viên học những môn giống sinh viên có mã số A02 học.

8.3. Cho biết những sinh viên học những môn bằng đúng những môn mà sinh viên A02 học.

## 9. Kết ngoài

```
SELECT <danh sách các cột>
FROM R1 LEFT | RIGHT [OUTER] JOIN R2 ON <biểu thức>
WHERE <điều kiện>
```

9.1. Với mỗi môn học cho biết bao nhiêu sinh viên đã học môn đó.

Câu lệnh:

```
SELECT h.MaMH, TenMH, COUNT(DISTINCT MASV) as SoSV
FROM DMMH h LEFT JOIN KETQUA k ON h.MaMH = k.MaMH
GROUP BY h.MaMH, TenMH
```

9.2. Với mỗi khoa cho biết có bao nhiêu sinh viên.

9.3. Với mỗi sinh viên cho biết đã học bao nhiêu môn.

## 10. Thực hiện insert, update, delete

### - Cú pháp thêm nhiều dòng

```
INSERT INTO <tên bảng>(<danh sách các thuộc tính>)
<câu truy vấn con>
```

10.1. Tạo một bảng mới tên sinhvien-ketqua: gồm: MASV, HoSV, TenSV, SoMonHoc. Sau đó Thêm dữ liệu vào bảng này dựa vào dữ liệu đã có.

Câu lệnh:

```
INSERT INTO sinhvien-ketqua (MaSV, HoSV, TenSV, SoMonHoc)
SELECT s.MaSV, HoSV, TenSV, Count(distinct MaMH)
FROM DMSV s, KETQUA k
WHERE s.MASV = k.MASV
GROUP BY s.MaSV, HoSV, TenSV
```

10.2. Thêm vào bảng khoa cột Siso, lần lượt cập nhật số vào khoa từ dữ liệu sinh viên.

```
UPDATE KHOA
SET SISO=(SELECT COUNT(*)
          FROM DMSV
          WHERE MAKHOA='AV')
WHERE MAKH='AV'
```

- 10.3. Thiết lập học bổng bằng 0 cho những sinh viên thi hai môn rớt ở lần 1
- 10.4. Tăng thêm 1 điểm cho các sinh viên rớt lần 2. Nhưng chỉ tăng tối đa là 5 điểm
- 10.5. Tăng học bổng lên 100000 cho những sinh viên có tất cả điểm thi  $\geq 7$ .
- 10.6. Xóa tất cả những sinh viên chưa dự thi môn nào.

Câu lệnh:

```
DELETE FROM DMSV
```

```
WHERE MaSV NOT IN (SELECT DISTINCT MaSV FROM KETQUA)
```

- 10.7. Xóa những môn mà không có sinh viên học.

## 11. Tạo view

**CREATE VIEW** <tên khung nhìn> **AS**  
<câu truy vấn>

- 11.1. Danh sách sinh viên không bị rớt môn nào

Câu lệnh:

```
CREATE VIEW v_dskhongrot AS
```

```
SELECT MASV, HOSV, TENSX, PHAI
```

```
FROM DMSV
```

```
WHERE MASV IN (SELECT MASV FROM KETQUA
```

```
GROUP BY MASV
```

```
HAVING MIN(DIEM)  $\geq$  5)
```

- 11.2. Danh sách sinh viên vừa học môn văn phạm vừa môn cơ sở dữ liệu
- 11.3. Trong mỗi sinh viên cho biết môn có điểm thi lớn nhất. Thông tin gồm: mã sinh viên, họ tên sinh viên, tên môn, điểm.
- 11.4. Danh sách những khoa có 2 sinh viên nữ trở lên.
- 11.5. Danh sách sinh viên:
  - Không rớt lần 1 hoặc
  - Không học môn văn phạm