



00_Ôn lại các câu lệnh SQL

Ngày học @September 9, 2023

I. Tạo database và nhập dữ liệu cho database:

1. Tạo database

Database —> New Database —> Nhập Database Name —> OK

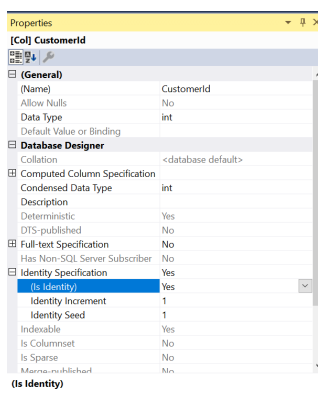
2. Tạo bảng:

Truy cập Database vừa tạo → Table → New table → Nhập dữ liệu → Set Primary Key → Ctrl S → Nhập tên bảng → Save

- Nếu cần **Reset giá trị Identity trong SQL Server về mặc định** (Trong **SQL Server** khi bạn đặt giá trị ID là tự động tăng và sau khi xóa toàn bộ dữ liệu thì nó vẫn cứ tiếp tục tăng giá trị tiếp theo mà không quay lại bắt đầu từ 1. Ví dụ bảng student của mình có 10 giá trị với ID từ 1 đến 10, khi mình xóa toàn bộ dữ liệu 10 bản ghi đó đi và bắt đầu thêm mới, thì ID tiếp theo sẽ là 11. Vậy để đưa giá trị tiếp theo mà bạn muốn về bất kì số nào bạn muốn thì dùng lệnh sau.)

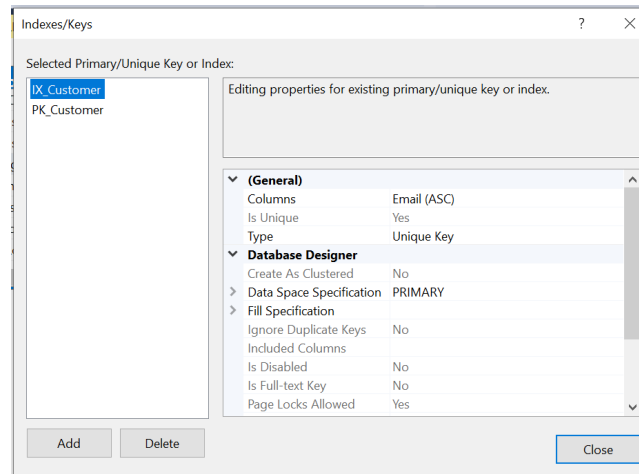
Execute lệnh: DBCC CHECKIDENT('Tên_bảng', RESEED, 0)

- Set Identity:

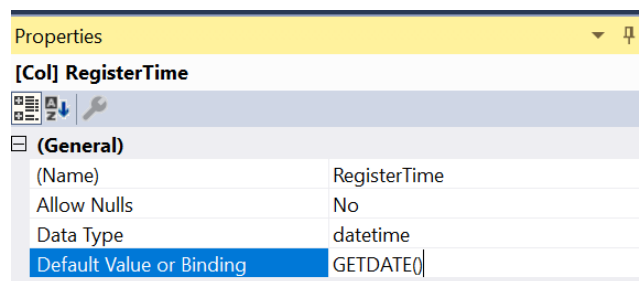


- Set **Unique Key** (các dòng không được trùng email) cho nó

Chuột phải bảng —> **Indexes/Keys** —> tại **Indexes/Keys** dialog box, chọn **Add** —> tại grid dưới **General**, chọn **Type** và chọn **Unique Key** từ drop-down list box, đổi tên key thành UK_Tên_bảng và **Close**.



- Set tính Default:



3. Tạo Diagram

Database Diagram → New Database Diagram → Chọn các bảng cần thiết lập → Add → Close cửa sổ
→ Kéo từ khóa chính qua khóa ngoại → Nhập tên diagram → OK

4. Nhập data cho các bảng

Bảng cần nhập → Edit top 200 rows → Nhập

II. Các hàm thường dùng



MONTH(tên cột), YEAR(tên cột): lấy tháng, năm của data

- **getdate():** hàm lấy ngày giờ hiện tại: thường được dùng khi với hàm datediff khi tính tuổi (VD: *where DATEDIFF(yy, GETDATE(), YEAR(NAM_SINH)) < 30*)
- **getday() / getmonth() / getyear():** hàm lấy ngày / tháng / năm hiện tại
- **isnull():** hàm kiểm tra giá trị có null hay không
Dùng ISNULL (1 GIÁ TRỊ, GIÁ TRỊ THAY THẾ NẾU GIÁ TRỊ KIA NULL)
VD: ISNULL(t2.DoanhThu, 0) as DoanhThu
- **datediff(dd | mm | yy, ngày 1, ngày 2):** hàm dùng để lấy số ngày | tháng | năm từ ngày 1 đến ngày 2)
Set hiển thị với dạng bit:

Cấu trúc CASE được sử dụng trong danh sách chọn nhằm thay đổi kết quả của truy vấn tùy thuộc vào các trường hợp khác nhau. Cấu trúc này có cú pháp như sau:

CASE biểu thức

```
WHEN biểu_thức_kiểm_tra THEN kết_quả
[ ... ]
[ELSE kết_quả_của_else]
```

END

hoặc:

CASE

```
WHEN điều_kiện THEN kết_quả
[ ... ]
```

[ELSE kết_quả_của_else]

END

- VD: Lấy ra các tác giả có tuổi dưới 35 và giới tính là nữ.

```
select *,
case
    when GIOI_TINH=1 then 'Nam'
    else N'Nữ'
end as 'GIOI_TINH'
from tbl_TAC_GIA
where GIOI_TINH = '0'
and datediff(yy, YEAR(NAM_SINH), getdate())<35
```

Một số lưu ý:

- Khi muốn sử dụng CSDL khác: **use <tên database>**
- Khi lấy data từ nhiều bảng, mà có sự trùng tên cột giữa các bảng: **<tên bảng>.<tên cột>**
- SQL không phân biệt chữ hoa & chữ thường

III. Các lệnh SQL thường dùng

1. Select:

Cú pháp:



select [top n / distinct] <*> / list các cột cần hiển thị data>
 [into <tên bảng cần đưa data vào>]
 from <list các bảng tham gia vào truy vấn>
 [where <các điều kiện lọc data>]
 [group by <ds các cột cần gom nhóm>]
 [having <các điều kiện lọc data để gom nhóm>]
 [order by <list các cần sắp xếp data> [asc | desc]]

Note:

select top n: lấy n dòng data đầu tiên

select distinct: loại bỏ các dòng data trùng nhau, chỉ lấy 1 dòng data trong số đó

các điều kiện lọc data: >, <, >=, <=, =, <>...

where:

- thường dùng thêm and, or
- nếu điều kiện lọc: dữ liệu ... có **chứa** từ abc:



where <dữ liệu> like N'%abc%' (N: được chèn vào khi dữ liệu có dấu, nếu không có dấu thì bỏ N)
 % và abc vị trí tùy vào yêu cầu:
 Ví dụ: có họ Nguyễn: like N'Nguyễn%'

- nếu điều kiện lọc theo kiểu so sánh tuyệt đối (tại, là...abc):

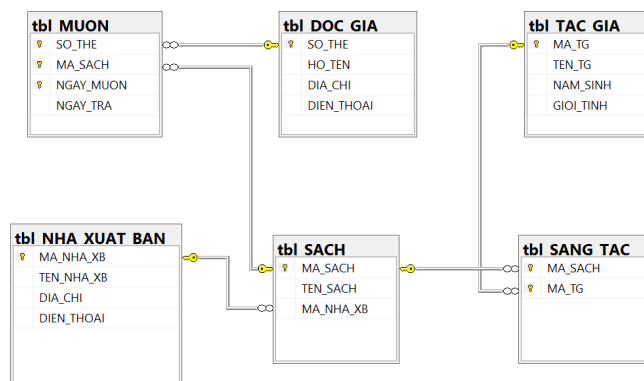


where <dữ liệu> = N'abc' : nếu data là chữ
where <dữ liệu> = abc : nếu data là số

group by: dùng trong các trường hợp cần thống kê

asc: tăng dần > desc: giảm dần

Ví dụ:



```

-- a. Đưa ra danh sách các đầu sách có trong thư viện.
-- Mỗi đầu sách có những thông tin (Tên sách, Tác giả, Nhà xuất bản).
-- Danh sách được sắp xếp theo tên sách.
select TEN_SACH, TEN_TG, TEN_NHA_XB
from tb1_SACH, tb1_TAC_GIA, tb1_NHA_XUAT_BAN, tb1_SANG_TAC
where tb1_NHA_XUAT_BAN.MA_NHA_XB = tb1_SACH.MA_NHA_XB
```

```

and tbl_SACH.MA_SACH = tbl_SANG_TAC.MA_SACH
and tbl_SANG_TAC.MA_TG = tbl_TAC_GIA.MA_TG
(where
order by TEN_SACH asc

```

```

C2:
select TEN_SACH, TEN_TG, TEN_NHA_XB
from ((tbl_TAC_GIA inner join tbl_SANG_TAC on tbl_SANG_TAC.MA_TG = tbl_TAC_GIA.MA_TG)
inner join tbl_SACH on tbl_SACH.MA_SACH = tbl_SANG_TAC.MA_SACH)
inner join tbl_NHA_XUAT_BAN on tbl_NHA_XUAT_BAN.MA_NHA_XB = tbl_SACH.MA_NHA_XB
order by TEN_SACH asc

```

```

-- b. Lấy ra tất cả các quyển sách có tên có chứa từ là 'Thể giới'.
select *
from tbl_SACH
where TEN_SACH like N'%Thể giới%'

```

```

-- c. Hiển thị các quyển sách được xuất bản tại Hà Nội.
select tbl_SACH.*
from tbl_SACH, tbl_NHA_XUAT_BAN
where tbl_SACH.MA_NHA_XB = tbl_NHA_XUAT_BAN.MA_NHA_XB
and DIA_CHI = N'Hà Nội'

```

```

C2:
select tbl_SACH.*
from tbl_SACH inner join tbl_NHA_XUAT_BAN
on tbl_SACH.MA_NHA_XB = tbl_NHA_XUAT_BAN.MA_NHA_XB
and DIA_CHI = N'Hà Nội'

```

```

-- d. Lấy ra danh sách các độc giả có mượn sách mà chưa trả.
select *
from tbl_DOC_GIA, tbl_MUON
where tbl_DOC_GIA.SO_THE = tbl_MUON.SO_THE
and NGAY_TRA is null

```

```

-- e. Lấy ra các độc giả tới mượn sách trong tháng 4/2008.
select *
from tbl_DOC_GIA, tbl_MUON
where tbl_DOC_GIA.SO_THE = tbl_MUON.SO_THE
and MONTH(NGAY_MUON) = 4
and YEAR(NGAY_MUON) = 2008

```

```

-- g. Lấy ra các tác giả có tuổi dưới 35 và giới tính là nữ.
select *,
case
    when GIOI_TINH = 1 then 'Nam'
    else N'Nữ'
end as 'GIOI_TINH'
from tbl_TAC_GIA
where GIOI_TINH = '0'
and datediff(yy, YEAR(NAM_SINH), getdate())<35

```

```

-- i. Hiển thị các tác giả có họ là Nguyễn.
select *
from tbl_TAC_GIA
where TEN_TG like N'Nguyễn%'

```

```

-- j. Hiển thị 2 tác giả đầu tiên. Danh sách gồm có
-- mã tác giả, tên tác giả, ngày sinh. Danh sách được
-- sắp xếp tăng dần theo tên và giảm dần theo ngày sinh.
select top 2 MA_TG, TEN_TG, NAM_SINH

```

```
from tbl_TAC_GIA
order by TEN_TG asc, NAM_SINH desc
```

2. Left join, Right join: lấy data từ các bảng, nếu trong đề có “không có thì để null”

VD: Lấy ra danh sách các KH với các hóa đơn mà họ đã tới mua, nếu chưa mua lần nào thì thông tin hóa đơn để null

```
select *
from tbl_KH left join tbl_HD
on tbl_KH.MaKH = tbl_HD.MaKH

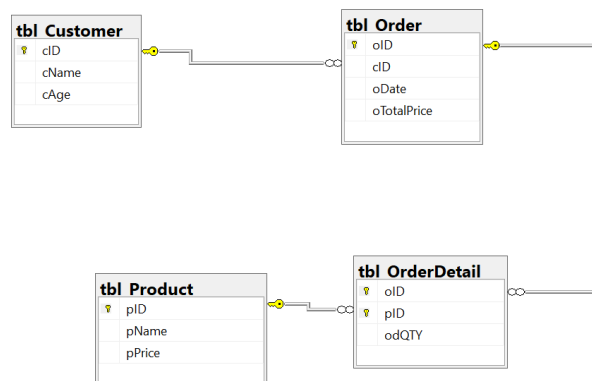
C2:
select *
from tbl_HD right join tbl_KH
on tbl_KH.MaKH = tbl_HD.MaKH
```

3. Hàm gộp - Dùng để thống kê



Thực hiện như bình thường với các data đơn → Excute → Quan sát xem các hàng trong 1 cột, nếu có trùng thì thử dùng **Group by** để gộp các hàng trùng data → Group by đi kèm với việc dùng **HÀM GỘP** (sum(), count(), avg(), max(), min()...)

Having: được dùng khi điều kiện lọc data có sử dụng HÀM GỘP (VD: Having SUM(odQTY * pPrice) > 30)



```
-- a. Hiển thị 2 hóa đơn đầu tiên có giá cao nhất trong tháng 3 năm 2006. (liên hệ câu i)
select top 2 tbl_Order.oID, SUM(odQTY * pPrice) as TongTien
from tbl_Order, tbl_OrderDetail, tbl_Product
where tbl_Order.oID = tbl_OrderDetail.oID
and tbl_OrderDetail.pID = tbl_Product.pID
and MONTH(odate) = 3 and YEAR(odate) = 2006
group by tbl_Order.oID
order by TongTien desc
```

```
-- i. Hiển thị mã hóa đơn, ngày bán và giá tiền của từng hóa đơn
-- (giá một hóa đơn được tính bằng tổng giá bán của từng loại mặt hàng xuất hiện trong hóa đơn.
-- Giá bán của từng loại được tính = odQTY*pPrice) như sau:
select tbl_Order.oID, oDate, SUM(odQTY * pPrice) as Total
from tbl_Order, tbl_OrderDetail, tbl_Product
where tbl_Order.oID = tbl_OrderDetail.oID
and tbl_OrderDetail.pID = tbl_Product.pID
group by tbl_Order.oID, oDate
```

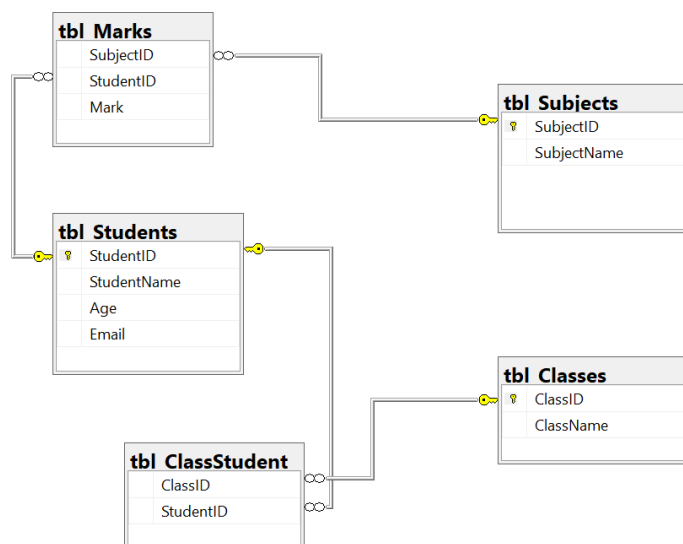
```
-- e. Hiển thị doanh thu của mỗi tháng trong năm 2006
select MONTH(oDate) as 'Thang', sum(tbl_OrderDetail.odQTY * tbl_Product.pPrice) as DoanhThu
from tbl_Order, tbl_OrderDetail, tbl_Product
where tbl_Order.oID = tbl_OrderDetail.oID
and tbl_OrderDetail.pID = tbl_Product.pID
and YEAR(oDate) = 2006
group by MONTH(oDate)
```

```
-- f. Mỗi khách hàng đã đến mua bao nhiêu lần.
select tbl_Customer.cID, cName, COUNT(tbl_Order.oID) as SoLanMua
from tbl_Customer, tbl_Order
where tbl_Customer.cID = tbl_Order.cID
group by cName, tbl_Customer.cID
```

```
-- g. Mỗi khách hàng đã mua hàng bao nhiêu tiền trong năm 2006.
select tbl_Customer.cID, cName, SUM(odQTY * pPrice) as TongTien
from tbl_Customer, tbl_Product, tbl_Order, tbl_OrderDetail
where tbl_Customer.cID = tbl_Order.cID
and tbl_Order.oID = tbl_OrderDetail.oID
and tbl_OrderDetail.pID = tbl_Product.pID
and YEAR(oDate) = 2006
group by tbl_Customer.cID, cName
```

```
-- j. Hiển thị những hóa đơn có tổng thành tiền trên 30. (Liên hệ với câu i)
select tbl_Order.oID, oDate, SUM(odQTY * pPrice) as 'Thanh tien'
from tbl_Order, tbl_OrderDetail, tbl_Product
where tbl_Order.oID = tbl_OrderDetail.oID
and tbl_OrderDetail.pID = tbl_Product.pID
group by tbl_Order.oID, oDate
HAVING SUM(odQTY * pPrice)>30
```

```
-- k. Hiển thị những khách hàng đã mua hàng trên 3 lần. (Liên hệ với câu f)
select tbl_Customer.cID, cName, COUNT(tbl_Order.oID) as 'So lan mua'
from tbl_Customer, tbl_Order
where tbl_Customer.cID = tbl_Order.cID
group by cName, tbl_Customer.cID
HAVING COUNT(tbl_Customer.cID)>3
```



```
-- f. Hiển thị danh sách học viên chưa thi môn nào (chưa có điểm)
select *
from tbl_Students
where StudentID not in (select distinct StudentID
                        from tbl_Marks)
```

```
-- g. Hiển thị môn học chưa đc học viên nào thi
select *
from tbl_Subjects
where SubjectID not in (select distinct SubjectID
                       from tbl_Marks)
```

```
-- h. Tính điểm trung bình cho các học viên
select tbl_Students.StudentID, tbl_Students.StudentName, AVG(Mark) as 'Diem Trung Binh'
from tbl_Marks, tbl_Students
where tbl_Students.StudentID = tbl_Marks.StudentID
group by tbl_Students.StudentID, tbl_Students.StudentName
```

```
-- i. Hiển thị môn học nào có học sinh thi đc điểm cao nhất
select tbl_Subjects.*
from tbl_Marks, tbl_Subjects
where tbl_Marks.SubjectID = tbl_Subjects.SubjectID
and Mark = (select MAX(Mark) from tbl_Marks)

-- C2:
select tbl_Subjects.*
from tbl_Marks, tbl_Subjects
where tbl_Marks.SubjectID = tbl_Subjects.SubjectID
and Mark = (select top 1 Mark from tbl_Marks
            order by Mark desc)
```

```
-- j. Hiển thị môn học nào có nhiều điểm dưới điểm trung bình nhất (<5)
select tbl_Subjects.*, COUNT(Mark) as 'So SV Duoi Diem Trung Binh'
from tbl_Marks, tbl_Subjects
where tbl_Marks.SubjectID = tbl_Subjects.SubjectID and Mark < 5
group by tbl_Subjects.SubjectID, SubjectName
having COUNT(Mark) = (select top 1 COUNT(Mark) as 'So SV Duoi Diem Trung Binh'
                     from tbl_Marks, tbl_Subjects
                     where tbl_Marks.SubjectID = tbl_Subjects.SubjectID
                     and Mark < 5
                     group by tbl_Subjects.SubjectID, SubjectName
                     order by 'So SV Duoi Diem Trung Binh' desc)
```

```
-- l. Hiển thị danh sách các học viên ứng với số lần thi của học viên đó.
-- Nếu học viên nào chưa thi lần nào thì ghi là 0.
select tbl_Students.StudentID, StudentName, COUNT(Mark) as 'So lan thi'
from tbl_Students, tbl_Marks
where tbl_Students.StudentID = tbl_Marks.StudentID
group by tbl_Students.StudentID, StudentName
UNION
select tbl_Students.StudentID, StudentName, 0
from tbl_Students
where StudentID not in (select distinct StudentID
                       from tbl_Marks)
```

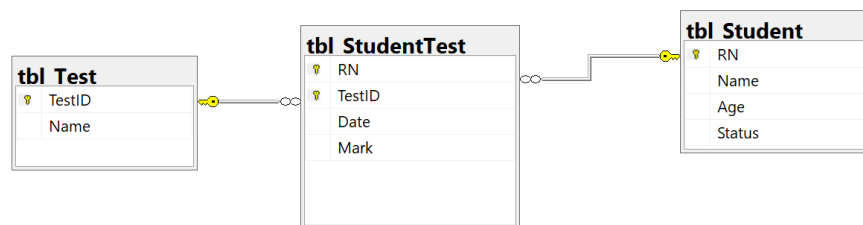
```
-- n. Hiển thị các học viên có điểm trung bình chung là cao nhất.
select tbl_Students.StudentID, StudentName, AVG(Mark) as 'Diem trung binh'
from tbl_Students, tbl_Marks
where tbl_Students.StudentID = tbl_Marks.StudentID
group by tbl_Students.StudentID, tbl_Students.StudentName
having AVG(Mark) = (select top 1 AVG(Mark) as 'Diem trung binh'
                  from tbl_Marks, tbl_Students
                  where tbl_Students.StudentID = tbl_Marks.StudentID)
```



```
group by tbl_Students.StudentID, tbl_Students.StudentName
order by 'Diem trung binh' desc)
```

```
-- o. Hiển thị các môn học mà có số sinh viên đăng ký học nhiều nhất. (liên hệ câu k)
select tbl_Subjects.SubjectID, SubjectName,
COUNT(tbl_Marks.StudentID) as 'Số SV đang ký học'
from tbl_Students, tbl_Marks, tbl_Subjects
where tbl_Marks.SubjectID = tbl_Subjects.SubjectID
and tbl_Students.StudentID = tbl_Marks.StudentID
group by tbl_Subjects.SubjectID, SubjectName
having COUNT(tbl_Marks.StudentID) = (select top 1 COUNT(tbl_Marks.StudentID) as 'Số SV đang ký học'
from tbl_Students, tbl_Marks, tbl_Subjects
where tbl_Marks.SubjectID = tbl_Subjects.SubjectID
and tbl_Students.StudentID = tbl_Marks.StudentID
group by tbl_Subjects.SubjectID, SubjectName
order by 'Số SV đang ký học' desc)
```

Dưới dạng a chữ số, b chữ số sau dấu phẩy: CAST (data cần đổi as numeric(a,b)) as <đặt tên cột>

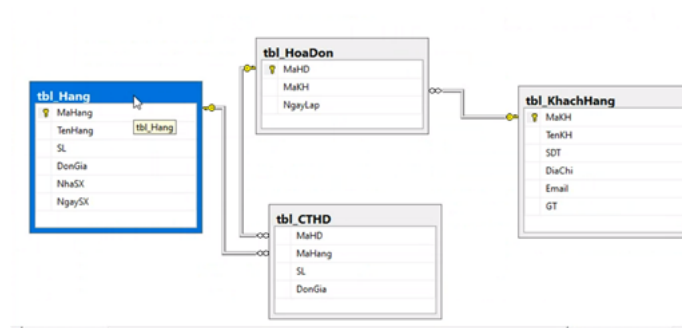


```
-- j. Hiển thị danh sách tất cả các học viên và môn học mà các học viên đó đã thi nếu học viên chưa thi môn nào thì phần tên môn học để Null.
-- C1:
select tbl_Student.RN, tbl_Student.Name as 'Students Name', tbl_Test.Name as 'Test Name'
from tbl_Student, tbl_StudentTest, tbl_Test
where tbl_Student.RN = tbl_StudentTest.RN
and tbl_StudentTest.TestID = tbl_Test.TestID
UNION
(select distinct tbl_Student.RN, tbl_Student.Name as 'Students Name', NULL
from tbl_Student, tbl_StudentTest
where tbl_Student.RN not in (select distinct RN
from tbl_StudentTest))

--C2:
select tbl_Student.RN, tbl_Student.Name, tbl_Test.Name
from tbl_Student left join tbl_StudentTest
on tbl_StudentTest. RN = tbl_Student.RN left join tbl_Test
on tbl_StudentTest.TestID = tbl_Test.TestID
```

```
-- k. Hiển thị danh sách sinh viên và số môn mà họ bị thi lại.
-- Nếu học viên nào không bị thi lại môn nào thì cột số môn ghi là 0. (liên hệ câu c)
select tbl_Student.RN, Name, COUNT(Mark) as 'Số môn thi lại'
from tbl_Student, tbl_StudentTest
where tbl_Student.RN = tbl_StudentTest.RN
and Mark<5
group by tbl_Student.RN, Name
UNION
(select distinct tbl_Student.RN, Name, 0
from tbl_Student, tbl_StudentTest
where tbl_Student.RN = tbl_StudentTest.RN
and Name not in (select distinct Name
from tbl_Student, tbl_StudentTest
where tbl_Student.RN = tbl_StudentTest.RN
and Mark<5))
```

4. Insert



a. Insert 1 dòng data:



insert into <tên bảng> [(ds các cột cần chèn data)] (các cột cách nhau dấu ,)
values (ds các giá trị tương ứng vs các cột)

VD: Thêm mới 1 KH:

C1:

```
insert into tbl_KH
values (4, N'Nguyễn Văn D', 123456, null, null, 1)
```

C2:

```
insert into tbl_KH (MAKH, TenKH, SĐT, GT)
values (5, N'Nguyễn Văn D', 123456, 1)
```

b. Insert nhiều dòng cùng 1 lúc:

VD:

```
insert into tbl_KH (MAKH, TenKH, SĐT, GT)
values (6, N'Nguyễn Văn E, 123456, 1),
(7, N'Nguyễn Thị F, 123456, 0),
(8, N'Nguyễn Văn G, 123456, 1)
```

C. Thêm data từ data của các bảng đã có sẵn



Insert into <tên bảng> [(ds các cột cần chèn data)]
Câu lệnh select

VD: Lưu data các KH đã tới mua hàng trong 2/2022 vào bảng tbl_Hoadon_KH_Backup (đã có)

```
insert into tbl_Hoadon_KH_Backup
select tbl_KH.MaKH, TenKH, MaHD, NgayLap
from tbl_KH, tbl_HoaDon
where tbl_KH.MaKH = tbl_HoaDon.MaKH
and MONTH(NgayLap) = 2
and YEAR(NgayLap) = 2022
```

5. Update

a. Cập nhật data liên quan đến 1 bảng



Update <tên bảng>

Set tên cột 1 = <biểu thức 1>

[, tên cột 2 = <biểu thức 2>]

...

[, tên cột n = <biểu thức n>]

[**where** <điều kiện update data>]

VD: Update địa chỉ và email cho KH có mã là 4:

update tbl_KH

set Diachi = N'Nguyễn Huệ',

Email = abc@gmail.com

where MaKH = 4

b. Cập nhật data liên quan đến nhiều bảng



update <tên bảng cần update data>

set tên cột 1 = <biểu thức 1>

[, tên cột 2 = <biểu thức 2>]

...

[, tên cột n = <biểu thức n>]

[**from** <tên bảng liên quan>]

[**where** <điều kiện update data>]

VD: Update SL ở bảng CTHD thành 0 cho các mặt hàng có tên là USB

update tbl_CTHD

set SL = 0

from tbl_Hang, tbl_CTHD

where tbl_Hang.MaHang = tbl_CTHD.MaHang

and TenHang **like** '%USB%'

6. Delete

a. Xóa data liên quan đến 1 bảng



delete from <tên bảng cần xóa>

[**where** <điều kiện xóa data>]

VD: Xóa những KH có địa chỉ là NULL

delete from tbl_KH

where Diachi is null

b. Xóa data liên quan đến nhiều bảng:



delete from <tên bảng cần xóa>

[**from** <các bảng liên quan>]

[**where** <điều kiện xóa data>]

VD: Xóa data ở bảng CTHD của các mặt hàng có tên là USB

delete from tbl_CTHD

from tbl_Hang, tbl_CTHD

where tbl_Hang.MaHang = tbl_CTHD.MaHang

and TenHang **like** '%USB%'

7. Khung nhìn view

Data của 1 câu lệnh truy vấn được use nhiều lần —> dùng VIEW

Khi tạo view, các cột chưa có Name, bắt buộc phải đặt tên cho chúng

Cú pháp:



CREATE VIEW tên khung nhìn [(danh sách tên cột)]

AS

câu lệnh **SELECT**

VD1: Lấy ra ds các hóa đơn:

select MaHD, tbl_KH.MaKH, TenKH, Ngaylap

from tbl_KH, tbl_HoaDon

where tbl_KH.MaKH = tbl_HoaDon.MaKH

VD2: Lấy ra ds các hóa đơn được lập trong tháng 2/2022

select MaHD, tbl_KH.MaKH, TenKH, Ngaylap

from tbl_KH, tbl_HoaDon

where tbl_KH.MaKH = tbl_HoaDon.MaKH

and MONTH(NgayLap) = 2

and YEAR(NgayLap) = 2022

—> Nhận thấy đoạn **SELECT** trên được use nhiều lần —> tạo chúng thành view để use

create view vw_DanhSach_HoaDon

as

select MaHD, tbl_KH.MaKH, TenKH, Ngaylap

from tbl_KH, tbl_HoaDon

where tbl_KH.MaKH = tbl_HoaDon.MaKH

—>

VD1: **select** * **from** vw_DanhSach_HoaDon

VD2: **select** * **from** vw_DanhSach_HoaDon

where MONTH(NgayLap) = 2

and YEAR(NgayLap) = 2022