



Bài 8: View Manager

CÁC LỆNH SQL

1. Lệnh tạo các record: Insert
2. Lệnh truy vấn dữ liệu: select
3. Lệnh cập nhật dữ liệu: Update
4. Lệnh xóa dữ liệu: Delete

Tạo mới record vào trong table

- ❑ Tạo một record với đầy đủ giá trị cho các fields:

**INSERT INTO table_name VALUES
(value1, value2,...)**

- ❑ Chỉ định các field cụ thể:

**INSERT INTO table_name
(column1, column2,...) VALUES
(value1, value2,...)**

Insert ... select

- ❑ Chèn dữ liệu vào table từ lệnh select của từ một hoặc nhiều table hoặc view

```
INSERT INTO TableName [(column1,  
column2...)] select ....
```

Ví dụ:

```
INSERT INTO MyCustomers SELECT  
CustomerID, CompanyName FROM  
Customers WHERE country= 'USA'
```

Các mệnh đề select

```
SELECT select_list
  [ INTO new_table ]
FROM table_source
  [ WHERE search_condition ]
  [ GROUP BY group_by_expression ]
  [ HAVING search_condition ]
  [ ORDER BY order_expression [ ASC |
DESC ] ]
```

Mệnh đề select

```
SELECT [ ALL | DISTINCT ]  
      [ TOP n [ PERCENT ] [ WITH TIES ] ]  
      < select_list >  
  
< select_list > ::=  
    { * | fieldName | expression as  
      column_alias  
      [,..n]  
  
    }
```

Tham số trong mệnh đề select

- ❑ **ALL**: chỉ định rằng tất cả các dòng trùng nhau có thể xuất hiện trong tập record kết quả. Default là ALL.
- ❑ **DISTINCT**: Khi có nhiều hơn hai dòng trùng nhau, chỉ hiển thị một dòng
- ❑ **TOP n [PERCENT]**: n dòng đầu tiên được xuất ra. n nằm trong khoảng 0 -> 4294967295. Nếu có PERCENT, thì n phải nằm trong khoảng từ 0 đến 100.
- ❑ **WITH TIES**: Chỉ định rằng kết quả các record trả về có thể nhiều hơn n dòng bằng cách thêm vào các dòng có cùng giá trị trong cột order. (bắt buộc phải có order)

Mệnh đề FROM

[FROM { < table_source > } [,...*n*]]

```
< table_source > ::=  
  table_name [ [ AS ] table_alias ]  
  | view_name [ [ AS ] table_alias ]  
    | derived_table [ AS ] table_alias [ ( column_alias [ ,...n ] ) ]  
  | < joined_table >
```

```
< joined_table > ::=  
  < table_source > < join_type > < table_source > ON <  
search_condition >  
  | < table_source > CROSS JOIN < table_source >  
  | < joined_table >  
< join_type > ::=  
  [ INNER | { { LEFT | RIGHT | FULL } [ OUTER ] } ]  
  [ < join_hint > ]  
JOIN
```

FROM table1 <join type> table2

ON <condition>

Trong đó <join type> = inner join

left join

right join

<condition>: điều kiện kết hai table

Tham số

- ❑ *derived_table*: là một subquery trả về các record trong database
- ❑ INNER: lấy ra tất cả các cặp dòng so trùng (join type default)
- ❑ FULL [OUTER]: lấy luôn tất cả các record của bảng hoặc trái hoặc phải mà không tìm thấy trong điều kiện join và giá trị không tìm thấy sẽ được điều vào là null.
- ❑ LEFT [OUTER]: lấy luôn tất cả các record của bảng trái mà không tìm thấy trong điều kiện join và giá trị không tìm thấy sẽ được điều vào là null.
- ❑ RIGHT [OUTER]: lấy luôn tất cả các record của bảng phải mà không tìm thấy trong điều kiện join và giá trị không tìm thấy sẽ được điều vào là null.

Mệnh đề where

```
[ WHERE < search_condition > | < old_outer_join > ]  
< old_outer_join > ::=  
    column_name { * = | = * } column_name
```

```
< search_condition > ::=  
    { [ NOT ] < predicate > | ( < search_condition > )  
    }  
    [ { AND | OR } [ NOT ] { < predicate > | ( <  
search_condition > ) } ] } [ ,...n ]
```

```
< predicate > ::=  
    { expression { = | < > | != | > | > = | ! > | < | <  
= | ! < }
```

Tham số

```
expression
| string_expression [ NOT ] LIKE string_expression
| expression [ NOT ] BETWEEN expression AND
expression
| expression IS [ NOT ] NULL
| CONTAINS
  ( { column | * } , '< contains_search_condition >'
)
| FREETEXT ( { column | * } , 'freetext_string' )
| expression [ NOT ] IN ( subquery | expression [ ,...n
] )
| expression { = | < > | ! = | > | > = | ! > | < | < =
| ! < }
| { ALL | SOME | ANY } ( subquery )
| EXISTS ( subquery )
}
```

Mệnh đề group by

```
[ GROUP BY [ ALL ] group_by_expression  
[ ,...n ] [ WITH { CUBE | ROLLUP } ]]
```

- ❑ All: bao gồm tất cả các group và result set. Không dùng chung với CUBE và ROLLUP
- ❑ CUBE: thêm vào dòng tổng sau mỗi nhóm, số lượng dòng tổng thêm vào tùy thuộc vào số lượng cột gom nhóm với giá trị là null
- ❑ Rollup: thêm vào dòng tổng sau mỗi nhóm của cột nhóm được chỉ định với giá trị là null

Ví dụ

```
select customerid, orders.orderid, sum(quantity)
from [order details] inner join orders on
      (orders.orderid=[order details].orderid)
group by customerid, orders.orderid with cube
```

	customerid	orderid	(No column name)
1	ALFKI	10643	38
2	ALFKI	10692	20
3	ALFKI	10702	21
4	ALFKI	10835	17
5	ALFKI	10952	18
6	ALFKI	11011	60
7	ALFKI	NULL	174
8	ANATR	10308	6
9	ANATR	10625	18
10	ANATR	10759	10
11	ANATR	10926	29
12	ANATR	NULL	63
13	ANTON	10365	24
14	ANTON	10507	30
15	ANTON	10507	30

Áp dụng CSDL QLVT

- ❑ Lấy ra thông tin MAHD, NGAYHD, MAKH, TENKH của hoá đơn tháng 6 năm 2005
- ❑ Đếm xem mỗi khách hàng có bao nhiêu hoá đơn
- ❑ Đếm xem trong tháng 6 năm 2005, mỗi mặt hàng bán được số lượng bao nhiêu

Update Query

- ❑ Dùng để thay đổi nội dung của một hoặc nhiều dòng dữ liệu trong một table

UPDATE

{ table_name | view_name }

SET *{ column_name = { expression |*
DEFAULT | NULL } [,...n]

[WHERE < search_condition >]

Ví dụ 1

- ❑ Trong database Pubs, tăng 10% giá bán cho sách có pud_ID là 0766:

```
UPDATE titles SET price = price * 1.1 WHERE  
(pub_id = '0766')
```

- ❑ Trong database NorthWind, khuyến mãi 10% số tiền cho các order details có số lượng ≥ 100 :

```
UPDATE [Order Details] SET discount=  
quantity*price * 0.1 WHERE (quantity>=  
100)
```

Delete query

```
DELETE [FROM] { table_name [ ...n ] ) |  
              view_name }  
  [ WHERE { < search_condition > } ]
```

Ví dụ dùng database pubs

- ❑ Xóa tất cả các authors

USE pubs

DELETE authors

- ❑ Xóa các authors có điều kiện

USE pubs

DELETE FROM authors WHERE au_lname = 'McBadden'

- ❑ Xóa các titleauthor có điều kiện

USE pubs

DELETE FROM titleauthor

WHERE title_id IN

(SELECT title_id FROM titles

WHERE title LIKE '%computers%')

Áp dụng

- ☐ Xoá các hoá đơn của các khách hàng có địa chỉ là TAN BINH

-
- ❑ Lấy ra danh sách các môn học (mã môn, tên môn) trong học kỳ 1

```
Select mh.mamh, tenmh  
From monhoc mh inner join monhoc_hocky  
    mhhk on mh.mamh = mhhk.mamh  
Where mahk=1
```

```
Select mh.mamh, tenmh  
From monhoc mh, monhoc_hocky mhhk  
Where mh.mamh=mhhk.mamh and  
    mahk =1
```

-
- ❑ Lấy ra danh sách các học kỳ (mã học kỳ, tên HK) mà có học môn MH01

Select hk.mahk, tenhk

From monhoc_hocky mhhk inner join
hocky hk on mhhk.mahk = hk.mahk

Where mamh='mh01'

❑ Đếm có bao nhiêu môn học trong mỗi học kỳ bao gồm thông tin mahk, tenhk, số môn học

Select hk.mahk, tenhk, count(*) as
soluongmonhoc

From monhoc_hocky mhhk inner join
hocky hk on hk.mahk = mhhk.mahk
Group by hk.mahk, tenhk

□ Tính tổng số tiết của mỗi học kỳ

Select mahk , sum(sotiet)

From monhoc mh inner join

monhoc_hocky mhhk on mhhk.mamh
= mh.mamh

Group by mahk

-
- ❑ Xoá các môn học có số tiết là null
(chưa nhập số tiết)

Delete monhoc

Where sotiet is null

-
- ❑ Thêm một record vào table monhoc có các giá trị MAMH='TINDC', tênMH='Tin hoc dai cuong'
 - ❑ Thêm record vào table hoc kỳ có tên hk là 'Hoc ky 1'
 - ❑ Thêm vào table monhoc_hocky mot record có giá trị mamh='tindc' va mahk=1

View

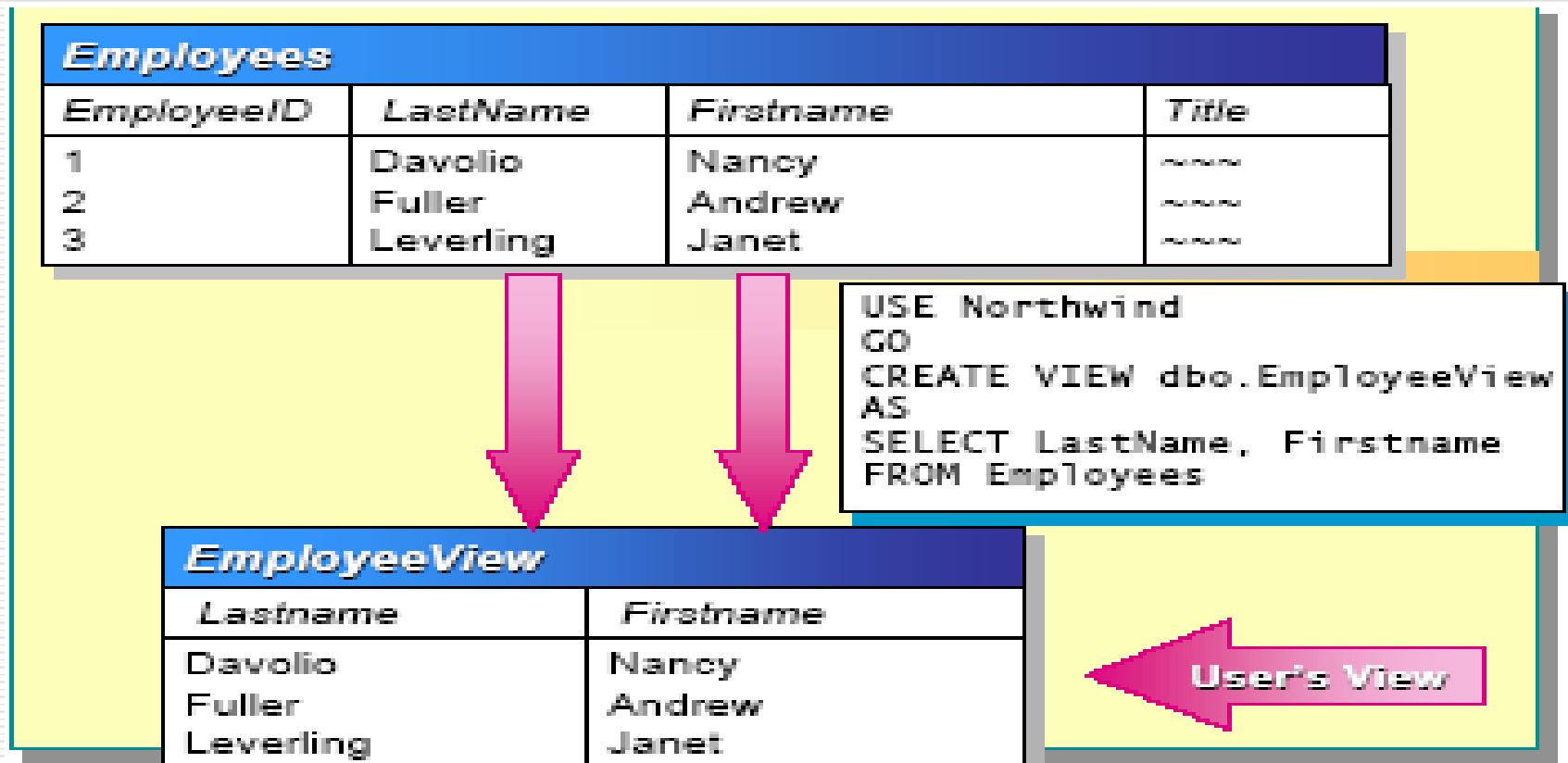
Mục Tiêu

- ❑ Mục tiêu bài học là cung cấp cho sinh viên các kiến thức cần thiết để tạo và sử dụng View
- ❑ Sau khi xong bài học này, sinh viên có thể:
 - Có kiến thức cơ bản về view.
 - Hiểu được các ưu điểm của views.
 - Định nghĩa một view bằng lệnh CREATE VIEW
 - Thao tác dữ liệu thông qua view

Bài học bao các nội dung sau:

1. Giới thiệu View
2. Lợi ích của view
3. Định nghĩa view
4. Sử dụng view
5. Một số lưu ý

Giới thiệu view (1)



Giới thiệu view (2)

- View là một table ảo
- Dùng để lưu trữ một truy vấn được định trước được xem là một đối tượng trong database.
- Các table tham gia truy vấn được gọi là ***base table***:
 - Dữ liệu trong view được lấy từ các bảng
 - Là sự kết hợp từ một hoặc nhiều bảng
 - Có thể chứa các tính toán
 - Có thể kết hợp giữa các bảng và các view khác

Ví dụ

```
USE Northwind
GO
CREATE VIEW dbo.EmployeeView
AS
SELECT LastName, Firstname
FROM Employees

SELECT * from EmployeeView
```

LastName	FirstName
Davolio	Nancy
Fuller	Andrew
Leverling	Janet
.	
.	
.	
(9 row(s) affected)	

Lợi ích của view

- ❑ Tập trung vào dữ liệu đang cần
- ❑ Che những phần dữ liệu chưa cần dùng trong database
- ❑ Đơn giản hoá việc quản lý quyền user
- ❑ Cải thiện việc thực thi
- ❑ Tổ chức lại dữ liệu để xuất ra cho các ứng dụng bên ngoài

Định nghĩa view

1. Tạo view
2. Ví dụ về tạo view
3. Thay đổi view, Xoá view
4. Các vấn đề khác trên view

Cú pháp tạo view

```
CREATE VIEW view_name
    [(column [,n ])]
[WITH {ENCRYPTION |
      SCHEMABINDING |
      VIEW_METADATA} [,n ]]
AS
select_statement
[WITH CHECK OPTION]
```

Tham số

- ❑ select statement: là câu lệnh select query lấy dữ liệu từ 1 hoặc nhiều table hoặc view khác
- ❑ WITH CHECK OPTION: Bắt buộc các lệnh thay đổi dữ liệu dựa trên view phải thỏa điều kiện định nghĩa view
- ❑ WITH ENCRYPTION: Mã hoá lệnh tạo view.
- ❑ SCHEMABINDING: ràng buộc view vào schema. Khi đó, ta không thể xóa các table hoặc các view là cơ sở của view khi view này còn tồn tại
- ❑ View_METADATA: chỉ định rằng SQL Server sẽ trả về cho DBLIB, ODBC và OLE DB APIs những "metadata information" của view bao gồm các thông tin như tên các base table, các column,

Ví dụ 1

- ❑ Tạo view chứa các thông tin về các customers có country='Mexico'

use Northwind

Create view v1

as

select * from customers where country='Mexico'

- ❑ Xem view

select * from v1

- ❑ Thêm một record vào trong v1

insert into v1 (customerid, companyname, country)
values ('abc', 'Micro system ABC', 'Mexico')

Ví dụ 2

❑ Tạo view với tham số with check option

use Northwind

Create view v2

as

select * from customers where
country='Mexico'

with check option

Ví dụ 2

- Khi đó, việc thêm/sửa dữ liệu trong view phải bảo đảm điều kiện đã định nghĩa

insert into v2 (customerid,
companyname, country) values
(`abc1`,`Micro system ABC`,`Mexico`)

insert into v2 (customerid,
companyname, country) values
(`abc2`,`Micro system ABC`,`USA`)



Báo
lỗi

Một số hạn chế trong select statement (của view)

- ❑ Không thể chứa mệnh đề COMPUTE hoặc COMPUTE BY
- ❑ Không thể chứa mệnh đề ORDER BY trừ khi có TOP n
- ❑ Không thể chứa mệnh đề INTO
- ❑ Không thể chứa các table tạm
- ❑ Có tối đa 1024 cột

Từ CSDL quản lý vật tư hãy tạo 1 view thực hiện công việc sau:

- ❑ Lấy ra danh sách khách hàng không mua hàng trong tháng 6 năm 2000

Lưu ý

- ❑ Tên view lưu trữ trong sysobjects table, tên cột của view được lưu trữ trong syscolumns, text dùng để tạo ra view lưu trong syscomments table
- ❑ Để tạo view, ta phải có quyền create view và quyền select table trong các table cơ sở.

Thay đổi view

Sửa cấu trúc view:

```
ALTER VIEW <view_name>  
    [(column [,...n])]  
[WITH <view_attribute>[,...n]]  
AS  
    select_statement  
[WITH CHECK OPTION]
```

Xoá view:

```
DROP VIEW <view_name>
```

-
- ❑ Loai(MaLoai, TenLoai)
 - ❑ SanPham(MaSP, TenSP, MaLoai)
 - ❑ NhanVien(MaNV, HoTen, NgaySinh, Phai)
 - ❑ PhieuXuat(MaPX, NgayLap, MaNV)
 - ❑ ChiTietXuat(MaPX, MaSP, SoLuong)

Câu Hỏi:

1- Tạo View cho biết: “mã sản phẩm, tên sản phẩm, tổng số lượng xuất của từng sản phẩm trong năm 2005”. Lấy dữ liệu từ View này sắp xếp tăng dần theo tên sản phẩm.

Create view v2

As

```
selete sp.masp, tensp, sum(soluong)
from (sanpham sp inner join
chitietphieuxuat ctpx on
sp.masp=ctpx.masp) inner join phieuxuat
px on px.mapx = ctpx.mapx
where year(ngaylap) = 2005
group by sp.masp, tensp
```

Go

```
Select * from v2 order by tensp
```

HOADON

MAHD
NGAY
MAKH
TONGTG

CHITIETHOADON

MAHD
MAVT
SL
KHUYENMAI
GIABAN

KHACHHANG

MAKH
TENKH
DIACHI
DT
EMAIL

VATTU

MAVT
TENVT
DVT
GIAMUA
SLTON

Bài tập (database QLVT)

1. Tạo view V1 để lấy ra danh sách các khách hàng có địa chỉ là Tân Bình
2. Tạo view V2 để lấy ra danh sách các khách hàng có địa chỉ không ở Tân Bình
3. Tạo ra view v3 để đếm xem mỗi quận có bao nhiêu khách hàng.
4. Tạo ra view v4 để đếm xem mỗi khách hàng có bao nhiêu hoá đơn (makh, tenkh, soluonghd) tính cả kh không có hóa đơn

Bài tập (tt)

1. Tạo view V5 để xem có bao nhiêu mặt hàng trong một hoá đơn (MAHD, NGAY, MAKH, SLMH)
2. Tạo view V6 lấy ra hoá đơn có số lượng mặt hàng lớn nhất (MAHD, NGAY, MAKH, SLMH)
3. Tạo view V7: Lấy ra danh sách các khách hàng có hoá đơn trong năm 1996

Bài tập (tt)

1. Tạo view V8: Lấy ra danh sách các khách hàng không mua hàng trong năm 1996
2. Tạo view V9: Lấy ra các thông tin hoá đơn bao gồm MAHD, NGÀY, TENKH, TONGTRIGIAHOADON.
3. Tạo view V10: Lấy ra các thông tin gồm MAKH, TENKH, SOLUONGHOADON, TONGTRIGIA.

Trong database QLDIEM

- Lấy ra danh sách sinh viên gồm masv, hoten, diem thi, mon với diem thi sẽ được tính như sau:
 - sẽ lấy điểm L_1 nếu điểm $L_1 \geq 5$
 - sẽ lấy điểm L_2 nếu điểm $L_1 < 5$ và diem $L_2 \neq \text{null}$
 - sẽ là 'chua Thi lai' nếu điểm $L_1 < 5$ và diemL2=null
 - sẽ là 'Hoc lai ' nếu điểm L_1 và điểm L_2 bằng null