

Bài 3: Ngôn ngữ xử lý truy vấn

3.1. Lệnh truy vấn dữ liệu:

Câu lệnh tổng quát:

```
SELECT [predicate] { * | table.* | <Danh sách Field>  
FROM <Danh sách Quan hệ>  
[WHERE <Điều kiện> ]  
[GROUP BY <Danh sách Thuộc tính> ]  
[HAVING <Điều kiện của nhóm> ]  
[ORDER BY Field1 [ACS|DESC],... ]
```

Công dụng: Tạo ra 1 quan hệ mới từ các quan hệ ghi sau từ khóa From

Bài 3(tt)

- **SELECT**
 - Tương đương phép chiếu của ĐSQH
 - Liệt kê các thuộc tính cần hiển thị trong kết quả
- **WHERE**
 - Tương ứng với điều kiện chọn trong ĐSQH
 - Điều kiện liên quan tới thuộc tính, sử dụng các phép nối luận lý **AND, OR, NOT**, các phép toán so sánh, **BETWEEN**
- **FROM**
 - Liệt kê các quan hệ cần thiết, các phép kết

Mệnh đề select

SELECT [ALL | DISTINCT] [TOP n [PERCENT] [WITH TIES]]
 < select_list >

< select_list > ::= { * | fieldName | expression as
 column_alias }

- **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
```

Mệnh đề From

FROM table1 <join type> table2 ON <condition>

Trong đó <join type>= inner join; left join;right join

<condition>: điều kiện kết hai table

- table: *có thể là một* subquery trả về các record trong database
- INNER: lấy ra tất cả các cặp dòng so trùng (join type default)
- FULL : 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 : 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 : 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 { = | < > | ! = | > | > = | ! > | < | <
= | ! < } }

Các toán tử cần lưu ý

- Toán tử so sánh: =, >, <, >=, <=, <>, !=, !<, !>
- Toán tử logic: AND, OR, NOT
- Phép toán: +, -, *, /
- BETWEEN AND
- IS NULL, IS NOT NULL
- LIKE (_ %)
- IN, NOT IN
- EXISTS, NOT EXISTS
- SOME, ALL

Bài 3(tt)

Toán tử LIKE

- So sánh chuỗi tương đối
- Cú pháp: s LIKE p, p có thể chứa % hoặc _
- %: thay thế một chuỗi ký tự bất kỳ
- _: thay thế một ký tự bất kỳ
- Ví dụ: **Select** masp,tensp **from** SANPHAM
where masp like 'B%01'

Bài 3(tt)

- **IS NULL, IS NOT NULL**

Select sohd from HOADON where makh is Null

Select * from HOADON where makh is Not Null

- **Toán tử so sánh, phép toán**

Select gia*1.1 as [gia ban] from SANPHAM where nuocsx <>
'Viet Nam'

Select * from SANPHAM where (gia between 20000 and
30000) OR (nuocsx='Viet Nam')

- **Toán tử IN, NOT IN**

Select * from SANPHAM where masp NOT IN
('BB01','BB02','BB03')

Bài 3(tt)

In hoặc **Exists**

- **Ví dụ:** Tìm các số hóa đơn mua cùng lúc 2 sản phẩm có mã số "BB01" và "BB02".

Select distinct sohd

From CTHD where masp = 'BB01' and sohd IN
(select distinct sohd from CTHD

Where masp = 'BB02')

Select distinct A.sohd

From CTHD A

Where A.masp = 'BB01' and
EXISTS (select * from CTHD B
where B.masp = 'BB02' and
A.sohd = B.sohd)

Bài 3(tt)

- **GROUP BY và HAVING**

Dùng tổng hợp, thống kê dữ liệu theo từng nhóm dựa trên giá trị của các thuộc tính xuất hiện sau từ khóa Group by. Trong đó, phát biểu Having dùng chọn ra những nhóm thỏa điều kiện.

- **Các hàm tổng hợp theo nhóm:**

- **Sum(Biểu thức):** Tính tổng giá trị các biểu thức số trên các bộ thuộc nhóm
- **Max(Biểu thức):** Chọn ra giá trị lớn nhất trên các bộ thuộc nhóm
- **Min(Biểu thức):** Chọn ra giá trị thấp nhất trên các bộ thuộc nhóm
- **Avg(Biểu thức):** Tính trung bình cộng giá trị biểu thức trên các bộ thuộc nhóm
- **Count(* | Biểu thức | Distinct Biểu thức):** Đếm các bộ trong từng nhóm.

Ví dụ về group by

- Với mỗi nhân viên cho biết mã số, số lượng đề án và tổng thời gian mà họ tham gia

MA_NVIEN	SODA	THOIGIAN
123456789	1	32.5
123456789	2	7.5
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
888665555	20	20.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
453453453	1	20.0
453453453	2	20.0

Ví dụ group by có điều kiện having

- Cho biết mã những nhân viên tham gia từ 2 đề án trở lên

MA_NVIENT	SODA	THOIGIAN
123456789	1	32.5
123456789	2	7.5
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
888665555	20	20.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
453453453	1	20.0
453453453	2	20.0

bị loại ra

Bài 3(tt)

- **ORDER BY**

Sắp thứ tự các bộ theo thứ tăng (asc-default) hay giảm(desc) của các giá trị trên các thuộc tính xuất hiện sau từ khóa order by, thứ tự ưu tiên từ trái sang phải.

Ví dụ: Tính tiền bán từng mặt hàng trong CTDDH và sắp thứ tự giảm dần theo SoDDH, và thành tiền bán hàng.

```
SELECT SoDDH, MSHH, SL, Dg, GiamGia, SL*DG-Giamgia As T_Tien  
FROM CTDDH  
Order By SoDDH, ThanhTien Desc;
```

Ví dụ về sử dụng mệnh đề ORDER BY

```
SELECT MA_NVIEN, SODA  
FROM PHANCONG  
ORDER BY MA_NVIEN DESC, SODA
```

MA_NVIEN	SODA
999887777	10
999887777	30
987987987	10
987987987	30
987654321	10
987654321	20
987654321	30

3.2 Lệnh định nghĩa dữ liệu

- Là ngôn ngữ mô tả
 - Lược đồ cho mỗi quan hệ
 - Miền giá trị tương ứng của từng thuộc tính
 - Ràng buộc toàn vẹn
 - Chỉ mục trên mỗi quan hệ
- Gồm
 - CREATE TABLE (tạo bảng)
 - DROP TABLE (xóa bảng)
 - ALTER TABLE (sửa bảng)
 - CREATE DATABASE
 - ...

Kiểu dữ liệu

- Số
 - SMALLINT
 - INT
 - NUMERIC
 - DECIMAL
 - REAL
 - FLOAT
 -

Kiểu dữ liệu (tt)

- Chuỗi ký tự
 - CHAR, VARCHAR
 - NCHAR, NVARCHAR (gõ dấu tiếng Việt Unicode)
 - ...
- Chuỗi bit
 - BIT
 - BITINT
 - ...
- Ngày giờ
 - DATETIME
 - SMALLDATETIME

Lệnh tạo bảng

- Để định nghĩa một bảng
 - Tên bảng
 - Các thuộc tính
 - Tên thuộc tính
 - Kiểu dữ liệu
 - Các RBTV trên thuộc tính

- Cú pháp

```
CREATE TABLE <Tên_bảng> (  
    <Tên_cột> <Kiểu_dữ_liệu> [<RBTV>],  
    <Tên_cột> <Kiểu_dữ_liệu> [<RBTV>],  
    ...  
    [<RBTV>]  
)
```

Ví dụ - Tạo bảng

```
CREATE TABLE NHANVIEN (  
    HONV NVARCHAR (20),  
    TENLOT NVARCHAR (50),  
    TENNV NVARCHAR (20),  
    MANV NVARCHAR (20),  
    NGSINH SMALLDATETIME,  
    DCHI NVARCHAR (50),  
    PHAI NVARCHAR (10),  
    LUONG INT,  
    MA_NQL NVARCHAR (20),  
    PHG INT  
)
```

Lệnh tạo bảng (tt)

- <RBTV>

- NOT NULL
- UNIQUE
- DEFAULT
- PRIMARY KEY
- FOREIGN KEY / REFERENCES
- CHECK

- Đặt tên cho RBTV

CONSTRAINT <Ten_RBTV> <RBTV>

Ví dụ - Tạo bảng có RBTV

```
CREATE TABLE NHANVIEN (  
    HONV NVARCHAR (20) NOT NULL,  
    TENLOT NVARCHAR (50) NOT NULL,  
    TENNV NVARCHAR (20) NOT NULL,  
    MANV NVARCHAR (20) PRIMARY KEY,  
    NGSINH SMALLDATETIME,  
    DCHI NVARCHAR(50),  
    PHAI NVARCHAR(10) CHECK (PHAI IN ('Nam', 'Nu')),  
    LUONG INT DEFAULT (1000000),  
    MA_NQL NVARCHAR(20),  
    PHG INT  
)
```

Ví dụ - Tạo bảng có RBTV (tt)

```
CREATE TABLE PHONGBAN (  
    TENPHG NVARCHAR(40) UNIQUE,  
    MAPHG INT PRIMARY KEY,  
    TRPHG NVARCHAR(20),  
    NG_NHANCHUC SMALLDATETIME DEFAULT (GETDATE())  
)
```

```
CREATE TABLE DIADIEM_PHG(  
    MAPHG INT NOT NULL,  
    DIADIEM NVARCHAR(50) NOT NULL,  
    CONSTRAINT PK_DIADIEM_PHG PRIMARY KEY (MAPHG , DIADIEM)  
)
```

Ví dụ - Tạo bảng có RBTV (tt)

```
CREATE TABLE DEAN (  
    TENDA NVARCHAR(40) UNIQUE,  
    MADA INT PRIMARY KEY,  
    DDIEM_DA NVARCHAR(50),  
    PHONG INT  
)
```

```
CREATE TABLE PHANCONG (  
    MA_NVIEN NVARCHAR(20) FOREIGN KEY (MA_NVIEN)  
        REFERENCES NHANVIEN(MANV),  
    SODA INT FOREIGN KEY (SODA)  
        REFERENCES DEAN(MADA),  
    THOIGIAN NUMERIC(3,1)  
)
```


Ví dụ - Đặt tên cho RBTV

```
CREATE TABLE NHANVIEN (  
    HONV NVARCHAR(20) CONSTRAINT NV_HONV_NN NOT NULL,  
    TENLOT NVARCHAR(50) NOT NULL,  
    TENNV NVARCHAR(20) NOT NULL,  
    MANV NVARCHAR(20) CONSTRAINT NV_MANV_PK PRIMARY KEY,  
    NGSINH SMALLDATETIME,  
    DCHI NVARCHAR(50),  
    PHAI NVARCHAR(10) CONSTRAINT NV_PHAICHK  
        CHECK (PHAI IN ('Nam', 'Nu')),  
    LUONG INT CONSTRAINT NV_LUONG_DF DEFAULT (10000),  
    MA_NQL NVARCHAR(20),  
    PHG INT  
)
```

Ví dụ - Đặt tên cho RBTv (tt)

```
CREATE TABLE PHANCONG (  
    MA_NVIEN NVARCHAR(20),  
    SODA INT,  
    THOIGIAN NUMERIC(3,1),  
    CONSTRAINT PC_MANVIEN_SODA_PK PRIMARY KEY (MA_NVIEN, SODA),  
    CONSTRAINT PC_MANVIEN_FK FOREIGN KEY (MA_NVIEN)  
        REFERENCES NHANVIEN(MANV),  
    CONSTRAINT PC_SODA_FK FOREIGN KEY (SODA)  
        REFERENCES DEAN(MADA)  
)
```

Lệnh sửa bảng

- Dùng để: thay đổi cấu trúc bảng, thay đổi RBTV
- Thêm cột

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

– Ví dụ

```
ALTER TABLE NHANVIEN ADD NGHENGHIEP NVARCHAR(50)
```

- Xóa cột

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

– Ví dụ

```
ALTER TABLE NHANVIEN DROP COLUMN NGHENGHIEP
```

Lệnh sửa bảng (tt)

- Hiệu chỉnh cột

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

– Ví dụ

```
ALTER TABLE NHANVIEN ALTER COLUMN NGHENGHIEP NVARCHAR(70)
```

Lệnh sửa bảng (tt)

- Thêm RBTV

```
ALTER TABLE <Tên_bảng> ADD  
    CONSTRAINT <Ten_RBTV> <RBTV>,  
    CONSTRAINT <Ten_RBTV> <RBTV>, ...
```

– Ví dụ

```
ALTER TABLE PHONGBAN ADD  
    CONSTRAINT PB_MAPHG_PK PRIMARY KEY (MAPHG),  
    CONSTRAINT PB_TRPHG FOREIGN KEY (TRPHG)  
        REFERENCES NHANVIEN(MANV),  
    CONSTRAINT PB_NGNHANCHUC_DF DEFAULT (GETDATE())  
        FOR NG_NHANCHUC,  
    CONSTRAINT PB_TENPB_UNI UNIQUE (TENPB)
```

Lệnh sửa bảng (tt)

- Xem các RBTV

SP_HELPCONSTRAINT <Tên_bảng>

– Ví dụ **SP_HELPCONSTRAINT** PHONGBAN

- Xóa RBTV

ALTER TABLE <Tên_bảng> **DROP** <Tên_RBTV>

– Ví dụ

ALTER TABLE PHONGBAN **DROP** PB_MAPHG_PK

ALTER TABLE PHONGBAN **DROP** PB_TRPHG

ALTER TABLE PHONGBAN **DROP** PB_NGNHANCHUC_DF

ALTER TABLE PHONGBAN **DROP** PB_TENPB_UNI

Lệnh xóa bảng

- Được dùng để xóa cấu trúc bảng
 - Tất cả dữ liệu của bảng cũng bị xóa

- Cú pháp

DROP TABLE <Tên_bảng>

- Ví dụ DROP TABLE NHANVIEN

 DROP TABLE PHONGBAN

 DROP TABLE PHANCONG

Lệnh xóa bảng (tt)

NHANVIEN

HONV	TENLOT	TENNV	<u>MANV</u>	NGSINH	DCHI	PHAI	LUONG	MA_NQL	PHG
------	--------	-------	-------------	-------------------	-----------------	------	-------	--------	-----

PHONGBAN

TENPHG	MAPHG	TRPHG	NG_NHANCHUC
--------	------------------	-------	-------------

Lệnh tạo kiểu dữ liệu mới

- Tạo ra một kiểu dữ liệu mới kế thừa những kiểu dữ liệu có sẵn
- Cú pháp

```
CREATE TYPE <Tên_kdl_mới> FROM <KDL_Có_sẵn>
```

– Ví dụ **CREATE TYPE** Kieu_Ten **FROM** **NVARCHAR**(30)

- Xóa kiểu dữ liệu tự định nghĩa

```
DROP TYPE <Tên_kdl_mới>
```

– Ví dụ

```
DROP TYPE Kieu_Ten
```

Sử dụng DEFAULT và RULE

- Default và Rule là đối tượng có thể kết với một hoặc nhiều cột

- Tạo Default

```
CREATE DEFAULT <Tên_Default> AS <BT_giá_trị>
```

- Ràng buộc Default vào cột

```
sp_bindefault <Tên_Default>, '<Tên_bảng.Tên_cột>'
```

- Gỡ bỏ Default khỏi cột

```
sp_unbindefault '<Tên_bảng.Tên_cột>'
```

- Ví dụ **CREATE DEFAULT** NV_LUONG_DF **AS** 10000
sp_bindefault NV_LUONG_DF, 'NHANVIEN.LUONG'
sp_unbindefault 'NHANVIEN.LUONG'

Sử dụng DEFAULT và RULE (tt)

- Tạo Rule

```
CREATE RULE <Tên_rule> AS <BT_điều_kiện>
```

- Ràng buộc Rule vào cột

```
sp_bindrule < Tên_rule >, '<Tên_bảng.Tên_cột>'
```

- Gỡ bỏ Rule khỏi cột

```
sp_unbindrule '<Tên_bảng.Tên_cột>'
```

- Ví dụ

```
CREATE RULE R_LUONG AS @LUONG>=10000  
sp_bindrule R_LUONG , 'NHANVIEN.LUONG'  
sp_unbindrule 'NHANVIEN.LUONG'
```

- Xóa Defaule, Rule

```
DROP DEFAULT <Tên_Default>  
DROP RULE <Tên_rule>
```

```
DROP DEFAULT NV_LUONG_DF  
DROP RULE R_LUONG
```

Lệnh cập nhật dữ liệu

- **Cập nhật dữ liệu**

- Thêm (insert)
- Xóa (delete)
- Sửa (update)

Lệnh INSERT

- Dùng để thêm 1 hay nhiều dòng vào bảng
- Để thêm dữ liệu
 - Tên bảng
 - Danh sách các thuộc tính cần thêm dữ liệu
 - Danh sách các giá trị tương ứng
- Cú pháp (thêm 1 dòng)

```
INSERT INTO <tên bảng>[(<danh sách các thuộc tính>)]  
VALUES (<danh sách các giá trị>)
```

Ví dụ

```
INSERT INTO NHANVIEN(HONV, TENLOT, TENNV, MANV)  
VALUES (N'Nguyễn', N'Trọng', N'Hòa', '123')
```

```
INSERT INTO NHANVIEN(HONV, TENLOT, TENNV, MANV, DCHI)  
VALUES ( N'Nguyễn', N'Thanh', N'Tùng', '333', NULL)
```

```
INSERT INTO NHANVIEN  
VALUES (N'Trần', N'Thanh', N'Tâm', '453', '7/31/1962', N'Mai Thị Lựu', 'Nam',  
25000,'333',5)
```

Lệnh INSERT (tt)

- Nhận xét
 - Thứ tự các giá trị phải trùng với thứ tự các cột
 - Có thể thêm giá trị NULL ở những thuộc tính không là khóa chính và cho phép NULL
 - Câu lệnh INSERT sẽ gặp lỗi nếu vi phạm RBTV
 - Khóa chính
 - Tham chiếu
 - NOT NULL - các thuộc tính có ràng buộc NOT NULL bắt buộc phải có giá trị

Lệnh INSERT (tt)

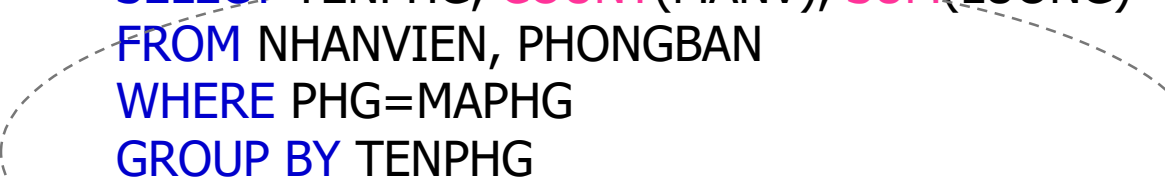
- Cú pháp (thêm nhiều dòng)

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

Ví dụ

```
CREATE TABLE THONGKE_PB (  
    TENPHG NVARCHAR(20),  
    SL_NV INT,  
    LUONG_TC INT  
)
```

```
INSERT INTO THONGKE_PB(TENPHG, SL_NV, LUONG_TC)  
SELECT TENPHG, COUNT(MANV), SUM(LUONG)  
FROM NHANVIEN, PHONGBAN  
WHERE PHG=MAPHG  
GROUP BY TENPHG
```



Lệnh DELETE

- Dùng để xóa các dòng của bảng
- Cú pháp

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

Ví dụ

```
DELETE FROM NHANVIEN  
WHERE HONV='Tran'
```

```
DELETE FROM NHANVIEN  
WHERE MANV='333'
```

```
DELETE FROM NHANVIEN
```

Lệnh DELETE (tt)

- Nhận xét
 - Số lượng số dòng bị xóa phụ thuộc vào điều kiện ở mệnh đề WHERE
 - Nếu không chỉ định điều kiện ở mệnh đề WHERE, tất cả các dòng trong bảng sẽ bị xóa
 - Lệnh DELETE có thể gây ra vi phạm RB tham chiếu
 - Không cho xóa
 - Xóa luôn những dòng có giá trị đang tham chiếu đến
 - CASCADE
 - Đặt NULL cho những giá trị tham chiếu

Lệnh DELETE (tt)

MANV	HONV	TENLOT	TENNV	NGSINH	DCHI	PHAI	LUONG	MA_NQL	PHG
333445555	Nguyen	Thanh	Tung	12/08/1955	638 NVC Q5	Nam	40000	888665555	5
987987987	Nguyen	Manh	Hung	09/15/1962	Ba Ria VT	Nam	38000	333445555	5
453453453	Tran	Thanh	Tam	07/31/1972	543 MTL Q1	Nu	25000	333445555	5
999887777	Bui	Ngoc	Hang	07/19/1968	33 NTH Q1	Nu	38000	987654321	4
987654321	Le	Quynh	Nhu	07620/1951	219 TD Q3	Nu	43000	888665555	4
987987987	Tran	Hong	Quang	04/08/1969	980 LHP Q5	Nam	25000	987654321	4
888665555	Pham	Van	Vinh	11/10/1945	450 TV HN	Nam	55000	NULL	1

MA_NVIENT	SODA	THOIGIAN
333445555	10	10.0
888665555	20	20.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
453453453	1	20.0

Lệnh DELETE (tt)

TENPHG	MAPHG	MA_NVIENT	NG_NHANCHUC
Nghien cuu	5	333445555	05/22/1988
Dieu hanh	4	987987987	01/01/1995
Quan ly	1	888665555	06/19/1981

MANV	HONV	TENLOT	TENNV	NGSINH	DCHI	PHAI	LUONG	MA_NQL	PHG
333445555	Nguyen	Thanh	Tung	12/08/1955	638 NVC Q5	Nam	40000	888665555	NULL
987987987	Nguyen	Manh	Hung	09/15/1962	Ba Ria VT	Nam	38000	333445555	NULL
453453453	Tran	Thanh	Tam	07/31/1972	543 MTL Q1	Nu	25000	333445555	NULL
999887777	Bui	Ngoc	Hang	07/19/1968	33 NTH Q1	Nu	38000	987654321	4
987654321	Le	Quynh	Nhu	07620/1951	219 TD Q3	Nu	43000	888665555	4
987987987	Tran	Hong	Quang	04/08/1969	980 LHP Q5	Nam	25000	987654321	4
888665555	Pham	Van	Vinh	11/10/1945	450 TV HN	Nam	55000	NULL	1

Lệnh UPDATE

- Dùng để thay đổi giá trị của thuộc tính cho các dòng của bảng
- Cú pháp

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

Ví dụ

```
UPDATE NHANVIEN  
SET NGSINH='08/12/1965'  
WHERE MANV='333445555'
```

```
UPDATE NHANVIEN  
SET LUONG=LUONG*1.1
```


Ví dụ

- Với đề án có mã số 10, hãy thay đổi nơi thực hiện đề án thành 'Vung Tau' và phòng ban phụ trách là phòng 5

```
UPDATE DEAN  
SET DIADIEM_DA='Vung Tau', PHONG=5  
WHERE MADA=10
```

Lệnh UPDATE

- Nhận xét
 - Những dòng thỏa điều kiện tại mệnh đề WHERE sẽ được cập nhật giá trị mới
 - Nếu không chỉ định điều kiện ở mệnh đề WHERE, tất cả các dòng trong bảng sẽ bị cập nhật
 - Lệnh UPDATE có thể gây ra vi phạm RB tham chiếu
 - Không cho sửa
 - Sửa luôn những dòng có giá trị đang tham chiếu đến
 - CASCADE

Bài 3(tt)

Thêm dữ liệu vào bảng

- **cp1:** INSERT INTO TenQH VALUES (value1[, value2[, ...]])

Ví dụ: Thêm một nhân viên mới

```
INSERT INTO KH VALUES (5, 'Anh', #1/5/78#)
```

- **cp2:** Thêm 1 bộ với các giá trị không đầy đủ (Các Field không liệt kê sẽ có giá trị NULL).

```
INSERT INTO TenQH [(field1[, field2[, ...]])]
```

```
VALUES (value1[, value2[, ...]]);
```

Ví dụ: Thêm 1 nhân viên mới chỉ biết Msnv, Tên, Giới tính

```
INSERT INTO KH (Msnv, Ten, GT) VALUES (5, 'Anh', 'Nam')
```

- **cp3:** Thêm vào QH các bộ là kết quả của câu lệnh Select.

```
INSERT INTO TenQH [(field1[, field2[, ...]])]
```

```
SELECT [source.]field1[, field2[, ...]] FROM tableexpression...;
```

Bài 3(tt)

Xóa dữ liệu:

- DELETE [tênQH.*] FROM TênQH [WHERE <điều kiện>];

Ví dụ: Xóa các DDH trước ngày 1/1/2000

```
DELETE FROM DDH WHERE NgayLHD <= '1/1/2000'
```

Cập nhật dữ liệu:

- UPDATE tênQH SET <Field1>=<BT1>,<Field2>=<BT2>...

WHERE <ĐiềuKiện>;

Ví dụ: Cập nhật đơn giá của sản phẩm 2,4,6 lên 5%

```
UPDATE HH SET DonGia = [DonGia]*1.05 WHERE Mshh In (2,4,6);
```

Thêm thuộc tính cho bảng:

- ALTER TABLE TenQH ADD COLUMN fieldname type[(size)]
[NOT NULL] [CONSTRAINT <index>];

Ví dụ: Thêm thuộc tính TổngTien cho quan hệ HoaDon

```
Alter table Hoadon add column TổngTien bigint not null
```

Bài tập bài 3

- Cho CSDL sau:

KH(MaKH, TenKH, ĐC, ĐT)

HĐ(SoHĐ, MaKH, NgayHĐ)

CTHĐ(SoHĐ, MaHG, SLM, ĐGB)

HH(MaHG, TenHG, SLT, ĐGM)

- a- Cho biết tên khách hàng đã đặt mua tất cả các mặt hàng?
- b- Cho biết tên các mặt hàng mà chưa có ai đặt mua?
- c- Cho biết tên các khách hàng đã đặt hàng trong ngày 20/11/2011?
- d- Cho biết số lượng các mặt hàng đã được mua trong ngày 12/05/2013?

Bài tập bài 3

- Congty (mact, tencongy, thanhpho)
 - Nhanvien(manv,tennv, diachi ,bacluong ,
chuyenmon, mact)
 - Duan(mada, tenduan, mact, thoigianthuchien)
 - Thamgia(mada, manv, vai tro)
-
- 1) Tênnv tham gia nhiều dự án nhất?
 - 2) Tên công ty có ít nhân viên nhất?
 - 3) Tên dự án có thời gian thực hiện lớn thứ 2?
 - 4) Liệt kê những dự án mà công ty 'Cổ phần xây dựng 1' đã có nhân viên tham gia với vai trò 'chủ dự án'

- **Độc Giả**(Mãđg, tênđg, đc, đt)
- **Sách**(MãSách, tênsách, ngônngữ, sốlượng)
- **Mượn Sách**(Mãđg, MãSách, ngàymượn, ngàytrả)
- Hãy trả lời các câu hỏi sau bằng ngôn ngữ SQL.
- c_1 - Cho biết tên độc giả có mượn sách và chưa trả?
- c_2 - Cho biết tên các cuốn sách đã được mượn trong ngày 20/10/2018?
- c_3 - Cho biết tên cuốn sách có nhiều độc giả mượn nhất?

- **Sinh Viên**(MãSV, TênSV, NS, Phái, ĐịaChỉ)
- **Môn Học**(MãMH, TênMH, SốTiếtLT, SốTiếtTH)
- **Kết Quả**(MãSV, MãMH, Điểm)
- Hãy trả lời các câu hỏi sau bằng ngôn ngữ SQL.
- c_1 - Cho biết tên sinh viên đã thi môn ‘Kỹ thuật lập trình’?
- c_2 - Cho biết tên các môn học có số tiết lý thuyết bằng số tiết thực hành?
- c_3 - Cho biết tên các sinh viên có điểm trung bình cao nhất?