

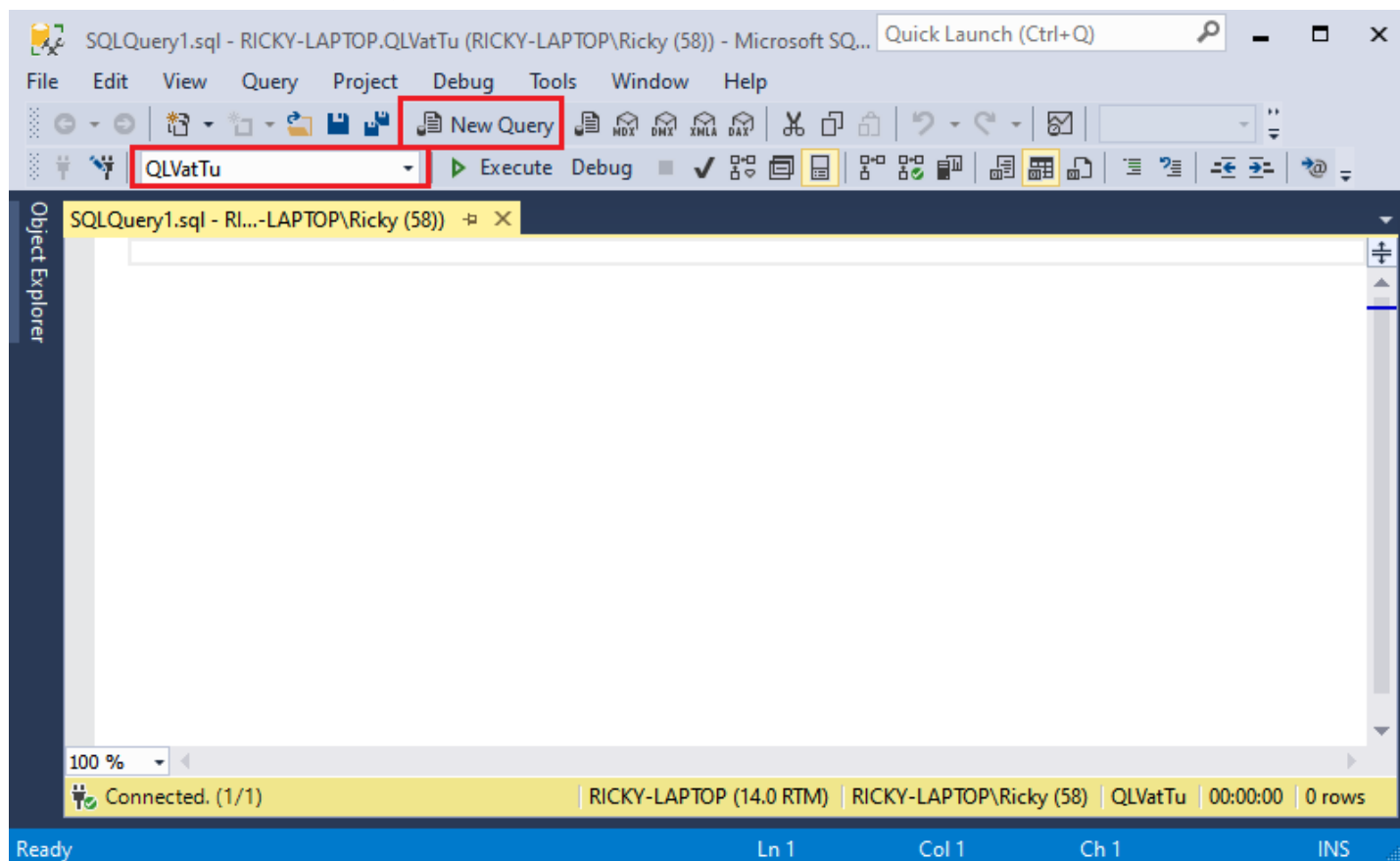
## Buổi 2

# Tạo và quản lý CSDL bằng ngôn ngữ SQL

## I. Sử dụng ngôn ngữ SQL với SQL Server

Ngôn ngữ SQL dùng để thực thi các thao tác với CSDL như tạo CSDL, tạo bảng, thêm, sửa, xóa dữ liệu... mà không cần thông qua giao diện của HQT CSDL. Các câu lệnh trong ngôn ngữ SQL còn được gọi là các câu truy vấn (*query*).

Với SQL Server, để viết truy vấn SQL, nhấn vào nút *New Query* (phím tắt *Ctrl+ N*), sau đó chọn CSDL cần thao tác ở menu phía dưới:



Để lưu lại CSDL với phần mở rộng là .sql, chọn *File* → *Save* hoặc *Save As...* (phím tắt *Ctrl + S*).

Để thực thi 1 đoạn lệnh SQL, bôi đen đoạn lệnh đó và nhấn nút **Execute** (phím tắt *Ctrl + E*).

**Lưu ý:** Ngôn ngữ SQL không phân biệt chữ hoa và chữ thường, kể cả trong các từ khóa và trong tên CSDL, tên bảng, tên thuộc tính, giá trị thuộc tính.

## II. Tạo cấu trúc CSDL

Để tạo CSDL, chúng ta sử dụng cú pháp:

```
CREATE DATABASE <Tên CSDL>;  
GO;
```

Để tạo bảng, đầu tiên phải chuyển sang CSDL cần thao tác như hướng dẫn ở phần I, hoặc sử dụng cú pháp:

```
USE <Tên CSDL>;  
GO;
```

Câu lệnh GO sẽ dừng việc thực thi các câu truy vấn lại cho đến khi câu truy vấn trước đó được thực hiện xong. Một số câu lệnh cần phải sử dụng câu lệnh GO mới thực hiện được.

Sau đó, chúng ta khai báo cấu trúc bảng theo cú pháp sau:

```
CREATE TABLE <Tên bảng> (  
    <Tên thuộc tính 1> <Kiểu dữ liệu>,  
    <Tên thuộc tính 2> <Kiểu dữ liệu>,  
    ...  
    <Tên thuộc tính n> <Kiểu dữ liệu>  
);
```

Ví dụ:

```
CREATE TABLE KHACHHANG (  
    MAKH nvarchar(5),  
    TENKH nvarchar(30),  
    DIACHI nvarchar(50),  
    DT nvarchar(15),  
    EMAIL nvarchar(30)  
);
```

Với các thuộc tính là khóa chính, bổ sung thêm từ khóa PRIMARY KEY.

Với các thuộc tính không cho phép giá trị NULL, bổ sung thêm từ khóa NOT NULL.

Ví dụ:

```
CREATE TABLE KHACHHANG (  
    MAKH nvarchar(5) PRIMARY KEY,  
    TENKH nvarchar(30) NOT NULL,  
    DIACHI nvarchar(50),  
    DT nvarchar(15),  
    EMAIL nvarchar(30)  
);
```

Trong trường hợp khóa chính có từ 2 thuộc tính trở lên, bổ sung thêm cú pháp:

```
CONSTRAINT <Tên ràng buộc>1 PRIMARY KEY (<Thuộc tính 1>, <Thuộc tính 2>, ...)
```

Ràng buộc (*constraint*) sẽ được học trong các buổi sau.

Ví dụ:

```
CREATE TABLE CHITIETHOADON (  
    MAHD nvarchar(10),  
    MAVT nvarchar(5),  
    SL int,  
    KHUYENMAI int,  
    GIABAN int,  
    CONSTRAINT PK_CHITIETHOADON PRIMARY KEY (MAHD, MAVT)  
);
```

### III. Khóa ngoại trong CSDL

#### 1. Khái niệm về khóa ngoại

Khóa ngoại (*foreign key*) dùng để tạo mối quan hệ giữa các bảng. Khóa ngoại trỏ tới khóa chính của 1 bảng khác.

Mỗi bảng có thể có nhiều khóa ngoại. Khóa ngoại có thể bao gồm 1 hoặc nhiều thuộc tính.

Giả sử có bảng KHACHHANG như sau:

---

<sup>1</sup> Tên ràng buộc khóa chính thường có dạng PK\_<Tên bảng>

Bảng KHACHHANG				
MAKH	TENKH	DIACHI	DT	EMAIL
KH01	Nguyễn Thị Bé	Tân Bình	8457895	bnt@yahoo.com
KH02	Lê Hoàng Nam	Bình Chánh	9878987	bnt@yahoo.com
KH03	Trần Thị Chiêu	Tân Bình	8457895	
KH04	Mai Thị Quế Anh	Bình Chánh		
KH05	Lê Văn Sang	Quận 10		sanglv@hcm.vnn.vn
KH06	Trần Hoàng Khải	Tân Bình	8457897	

Mỗi khách hàng xem như được định danh bằng khóa chính MAKH.

Với bảng HOADON, mỗi hóa đơn cần phải lưu lại thông tin khách hàng nào là người mua hóa đơn này. Do đó, trong bảng HOADON cần phải có MAKH, đại diện cho toàn bộ thông tin của khách hàng đó:

Bảng HOADON		
MAHD	NGAY	MAKH
HD001	12/05/2015	KH01
HD002	25/05/2015	KH02
HD003	25/05/2015	KH01
HD004	25/05/2015	KH04
HD005	26/05/2015	KH04
HD006	02/06/2015	KH03
HD007	22/06/2015	KH04
HD008	25/06/2015	KH03
HD009	15/08/2015	KH04
HD010	30/09/2015	KH01
HD011	27/12/2015	KH06
HD012	27/12/2015	KH01

Như vậy, thuộc tính MAKH trong bảng HOADON là khóa ngoại, trỏ tới khóa chính MAKH của bảng KHACHHANG. Việc đưa khóa ngoại vào bảng sẽ đảm bảo việc lưu dữ liệu không bị trùng lặp và không bị xung đột.

**Lưu ý:** 2 thuộc tính ở 2 đầu của khóa ngoại phải cùng kiểu dữ liệu và cùng kích thước.

## 2. Cách tạo khóa ngoại

Với các thuộc tính là khóa ngoại, bổ sung thêm cú pháp sau:

**FOREIGN KEY REFERENCES** <Tên bảng đích>(<Khóa chính của bảng đích>)

Ví dụ:

```
CREATE TABLE HOADON (
    MAHD nvarchar(10) PRIMARY KEY,
    NGÀY date,
    MAKH nvarchar(5) FOREIGN KEY REFERENCES KHACHHANG(MAKH),
    TONGTG int
);
```

## IV. Thay đổi cấu trúc bảng

Ngôn ngữ SQL cung cấp câu lệnh ALTER TABLE để thay đổi cấu trúc bảng đã có.

### 1. Thêm thuộc tính

Để thêm thuộc tính vào 1 bảng đã có, chúng ta dùng cú pháp sau:

```
ALTER TABLE <Tên bảng>  
ADD <Tên thuộc tính> <Kiểu dữ liệu>;
```

Ví dụ:

```
ALTER TABLE KHACHHANG  
ADD CMND varchar(20);
```

### 2. Thay đổi kiểu dữ liệu

Để thay đổi kiểu dữ liệu của một thuộc tính đã có, chúng ta dùng cú pháp sau:

```
ALTER TABLE <Tên bảng>  
ALTER COLUMN <Tên thuộc tính> <Kiểu dữ liệu mới>;
```

Ví dụ:

```
ALTER TABLE HOADON  
ALTER COLUMN NGAY datetime;
```

### 3. Thêm khóa chính

Để thêm khóa chính vào 1 bảng đã có, chúng ta dùng cú pháp sau:

```
ALTER TABLE <Tên bảng>  
ADD PRIMARY KEY (<Tên thuộc tính>;
```

Ví dụ:

```
ALTER TABLE LOAISANPHAM  
ADD PRIMARY KEY (MALOAISANPHAM);
```

**Lưu ý:** Thuộc tính được chọn làm khóa chính phải có tính chất NOT NULL ngay từ đầu.

### 4. Thêm khóa ngoại

Để thêm khóa ngoại vào 1 bảng đã có, chúng ta dùng cú pháp sau:

```
ALTER TABLE <Tên bảng>  
ADD FOREIGN KEY (<Khóa ngoại>) REFERENCES <Tên bảng đích>(<Khóa chính của  
bảng đích>;
```

Ví dụ:

```
ALTER TABLE SANPHAM  
ADD FOREIGN KEY (MALOAISANPHAM) REFERENCES LOAISANPHAM(MALOAISANPHAM);
```

## V. Xóa cấu trúc CSDL

Ngôn ngữ SQL cung cấp các câu lệnh DROP để xóa bảng và xóa CSDL:

```
DROP TABLE <Tên bảng>;  
DROP DATABASE <Tên CSDL>;
```

Ngoài ra, để xóa toàn bộ dữ liệu trong bảng, chúng ta sử dụng câu lệnh sau:

```
TRUNCATE TABLE <Tên bảng>;
```

## VI. Thêm, sửa, xóa dữ liệu trong bảng

### 1. Thêm dữ liệu

Ngôn ngữ SQL cung cấp câu lệnh INSERT để thêm dữ liệu vào các bảng.

#### a) Câu lệnh INSERT đầy đủ

Cú pháp:

**INSERT INTO** <Tên bảng> (<Danh sách thuộc tính>) **VALUES** (<Danh sách giá trị tương ứng>);

Trong đó:

- <Danh sách thuộc tính> không nhất thiết phải đủ số lượng và đúng thứ tự so với thuộc tính của bảng.
- <Danh sách giá trị> phải đúng về số lượng và tương đồng về thứ tự với <Danh sách thuộc tính>.

Ví dụ:

```
INSERT INTO KHACHHANG (MAKH, TENKH, DIACHI, DT, EMAIL) VALUES ('KH02', N'Lê Hoàng Nam', N'Bình Chánh', '9878987', 'bnt@yahoo.com');
```

```
INSERT INTO HOADON (MAHD, NGAY, MAKH) VALUES ('HD001', '2015-05-12', 'KH01');
```

## b) Câu lệnh INSERT thiếu

Cú pháp:

```
INSERT INTO <Tên bảng> VALUES (<Danh sách giá trị>);
```

Trong đó: <Danh sách giá trị> phải đủ số lượng và đúng thứ tự so với thuộc tính của bảng.

**Lưu ý:** Với các chuỗi có dấu tiếng Việt (kiểu dữ liệu nvarchar), cần thêm kí tự N phía trước chuỗi đó, ví dụ: N'Lê Hoàng Nam'.

## 2. Sửa dữ liệu trong bảng

Ngôn ngữ SQL cung cấp câu lệnh UPDATE để sửa dữ liệu trong các bảng.

Cú pháp:

```
UPDATE <Tên bảng>
```

```
SET <Thuộc tính 1> = <Giá trị 1>, <Thuộc tính 2> = <Giá trị 2>,...
```

```
WHERE <Điều kiện>;
```

Cách viết biểu thức <Điều kiện> sẽ được học trong các buổi sau.

Ví dụ:

```
UPDATE KHACHHANG
```

```
SET DT = '0905939947', EMAIL = 'huuphuoc29791@gmail.com'
```

```
WHERE MAKH = 'KH01';
```

**Lưu ý:** Nếu không có mệnh đề WHERE, câu lệnh UPDATE sẽ sửa tất cả các dòng có trong bảng.

## 3. Xóa dữ liệu trong bảng

Ngôn ngữ SQL cung cấp câu lệnh DELETE để xóa các dòng dữ liệu trong các bảng.

Cú pháp:

```
DELETE FROM <Tên bảng>
```

```
WHERE <Điều kiện>;
```

Cách viết biểu thức <Điều kiện> sẽ được học trong các buổi sau.

Ví dụ:

```
DELETE FROM KHACHHANG
```

```
WHERE MAKH = 'KH05';
```

**Lưu ý:** Nếu không có mệnh đề WHERE, câu lệnh DELETE sẽ xóa tất cả các dòng có trong bảng.

## VII. Bài tập

1. Tạo cấu trúc và thêm dữ liệu cho CSDL QLVatTu – Quản lý cửa hàng vật tư. Lược đồ và dữ liệu của CSDL có trong file *[QLVatTu] Mô tả CSDL.pdf*.
2. Tạo cấu trúc và thêm dữ liệu cho CSDL QLDeAn – Quản lý đề án công ty. Lược đồ và dữ liệu của CSDL có trong file *[QLDeAn] Mô tả CSDL.pdf*.