**SQL SERVER**

1. **Tổng quan về SQL:**
   * SQL hay Structured Query Language là ngôn ngữ truy vấn.
   * SQL cho phép người lập trình có thể thực hiện tạo, lưu trữ và sửa đổi các nguồn dữ liệu từ nhỏ đến lớn với đủ các kiểu dữ liệu.
   * Hiện nay có rất nhiều hệ quản trị cơ sở dữ liệu
     + MySQL
     + MariaDB
     + Oracle
     + MongoDB
     + PostgreSQL.
     + Microsoft SQL Server.
     + …
   * Các thành phần trong SQL
     + SQL gồm 4 loại câu lệnh chính
       - DML (Data Multiplate Language) : đây là nhóm câu lệnh thực hiện thao tác với dữ liệu trong các bảng của DB (select , update, delete,.. ).
       - DDL (Data Define Language) : đây là nhóm câu lệnh thực hiện tạo database , tạo các bảng , thực hiện tạo cấu trúc cho DB (create, drop , truncate , …)
       - TCL (Transaction Control Language) : đây là nhóm câu lệnh thực hiện transaction trong SQL (commit, rollback).
       - DCL (Data Control Language) : đây là nhóm câu lệnh thực hiện cấp quyền cho người sử dụng (grant, Revoke).
     + Datatype trong SQL:
       - Number : gồm có kiểu
     + Số Nguyên
       - * Tinyint : 1 byte
         * SmallInt : 2 bytes
         * MediumInt : 3 bytes
         * Int : 4 bytes
         * BigInt : 8 bytes
     + Số thực :
       - Float (4 bytes)
       - Double (8 bytes)
       - Decimal
   * Chuỗi kí tự:

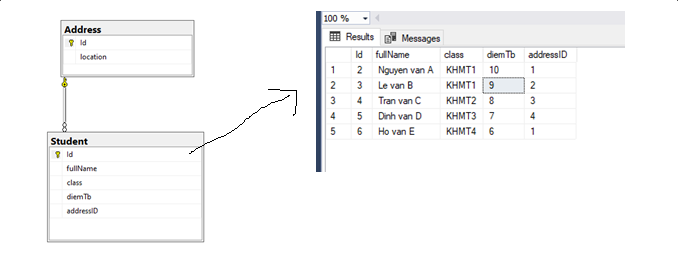
|  |  |  |
| --- | --- | --- |
|  | **CHAR** | **VARCHAR** |
| Viết tắt | Character (Ký tự) | Variable Character (nghĩa là chưa biết length của Varchar) |
| Chiều dài tối đa | Lưu tối đa 255 bytes | Lưu tối đa 65,535 bytes |
| Length | Fix cứng size lưu trữ trong ổ cứng máy tính  VD: khai báo CHAR(10) thì  'Nguyen' chiếm 10 ô trong ổ cứng máy tính  'Nguyen Van' cũng chiếm 10 ô trong ổ cứng máy tính | Length của Varchar phụ thuộc vào giá trị  VD: khai báo VARCHAR(10) thì  'Nguyen' chiếm 6 ô trong ổ cứng máy tính  'Nguyen Van' chiếm 10 ô trong ổ cứng máy tính |
| Tốc độ | Nhanh hơn | Chậm hơn |

* Date & Time : Date, DateTime : sử dụng kiểu dữ liệu thời gian
* Enum : tương tự kiểu dữ liệu Enum trong Java , định nghĩa sẵn các giá trị đầu vào cho trường đó.

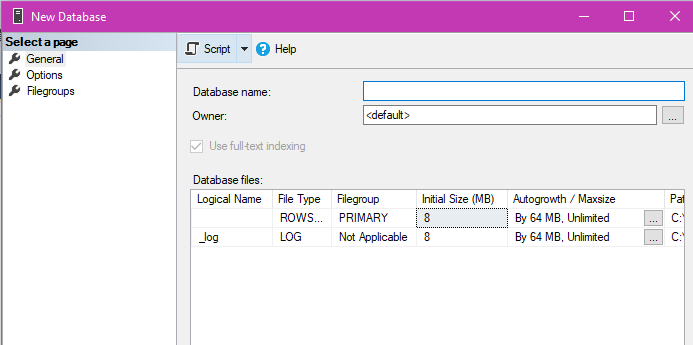
1. **MS SQL Server**

* Là hệ thống quản trị cơ sở dữ liệu quan hệ , viết tắt là RDBMS được phát triển bởi Microsoft.
* Được Microsoft bắt đầu triển khai phát triển vào 1988 và công bố phiên bản 1.0 vào 1989. Và đến nay phiên bản SQL Server 2019 là phiên bản mới nhất được công bố bởi Microsoft.
* Ưu điểm:
  + Có thể cài nhiều phiên bản MS SQL khác nhau trên cùng một máy tính.
  + Duy trì riêng biệt các môi trường sản xuất, phát triển, thử nghiệm.
  + Giảm thiểu các vấn đề tạm thời trên cơ sở dữ liệu.
  + Tách biệt các đặc quyền bảo mật.
  + Duy trì máy chủ dự phòng.
* Nhược điểm
  + Có thể mất phí cho các dịch vụ cao.
  + Hạn chế nhất định về cơ sở hạ tầng.
  + Sự phụ thuộc.

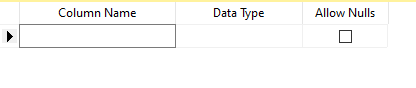
1. **Data Source**
   * là bộ dữ liệu được người lập trình thực hiện tạo nên, bao gồm các bảng , dữ liệu trong bảng.
   * Mỗi bảng sẽ có các trường đóng vai trò chứa thông tin về chủ thể mà bảng đó mô tả. Mỗi trường sẽ có các giá trị thể hiện cho 1 đại diện của bảng đó mô tả.

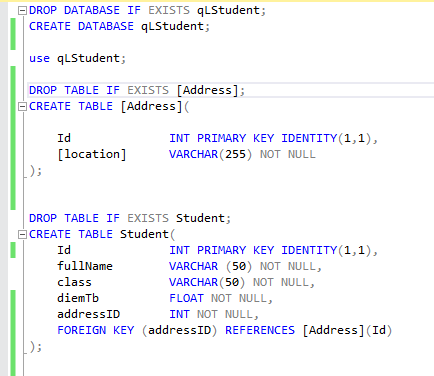
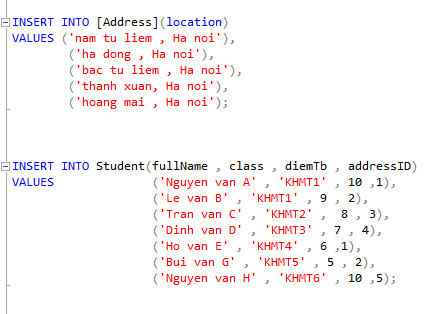
****

* + Để tạo nên một Data source : chúng ta có thể sử dụng 2 cách
    - Cách 1: Sử dụng luôn tính năng tạo của tool:

****

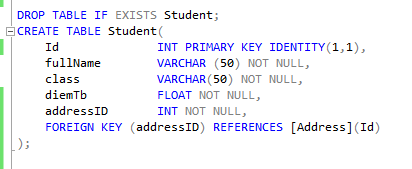
Sau đó thực hiện tạo thêm cac bảng cho database :



* Cách 2: Sử dụng các câu lệnh SQL để tạo database, bảng, và thêm dữ liệu cho các bảng.

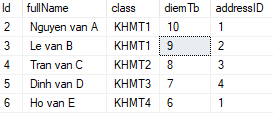
1. **Table , Column**
   * Table:
     + Là các đối tượng chứa dữ liệu trong database.
     + Mỗi bảng sẽ có các trường là các thuộc tính đại diện cho table đó, mỗi trường sẽ được thiết kế để lưu trữ về 1 tính chất của bảng đó

Vd : Bảng Student sẽ có các trường thể hiện cho thông tin của học sinh.

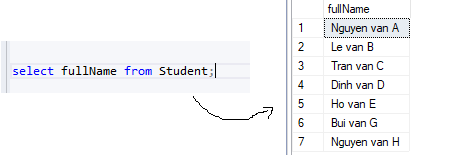


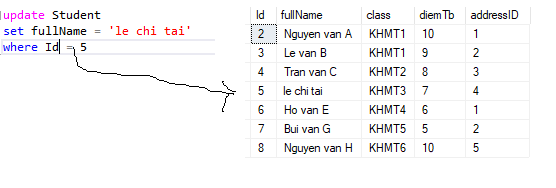
* + - Mỗi bảng sẽ có nhiều nhất 1 khóa chính (Primary key) để làm định danh cho bảng.
    - Các bảng trong database có thể liên kết với nhau thông qua các mỗi quan hệ (Foreign key) để có thể lấy thông tin của nhau.
    - Bảng trong database được thiết kế khá giống với kiểu spreedsheet (bảng tính) với các cột (trường dữ liệu) và các hàng (là các bản ghi trong bảng).
  + Column:
    - Đây là còn được biết là các trường trong bảng hay là các tính chất của chủ thế mà bảng đó lưu trữ thông tin.

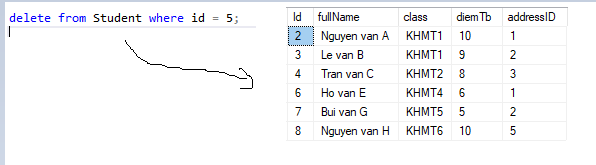
VD: Fullname , class, diemTb, addressId là các trường thể hiện tính chất thông tin của đối tượng Student



* + - Dựa vào các cột nay mà người lập trình có thể thực hiện thao tác lấy , sửa hoặc xóa dữ liệu trong bảng.







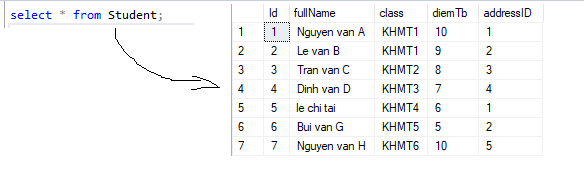
* Contraints
  + Primary Key : Khóa chính của bảng
    - Là 1 trường khác null và các giá trị trong trường không thể trùng nhau
    - Có thể là 1 trường hoặc 1 nhóm trường
    - Mỗi bảng có nhiều nhất 1 khóa chính
    - Có thể có bảng không có khóa chính nào
  + Foreign key : Khóa ngoại của bảng
    - Các giá trị có thể trùng nhau
    - Chỉ có tác dụng liên kết với bảng khác để lấy thêm thông tin
  + Unique key (khóa duy nhất)
* Nếu 1 field là Unique key thì sẽ không có giá trị nào trong field đó trùng lặp nhau (các giá trị trong cột đó khác nhau)
* Có thể chứa nhiều nhất 1 giá trị null (không thể chứa 2 giá trị null)
  + Default
    - Gán value default cho 1 trường nào đó, khi không insert value vào trường đó thì SQL sẽ tự động lấy value default cho trường đó
  + Not null
    - Gán value not null cho trường nào đó , khi không insert value vào trường thì SQL sẽ báo lỗi.

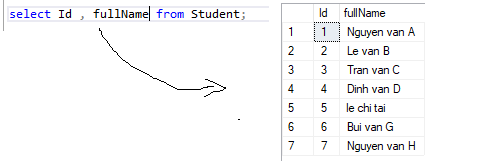
1. **Các câu lệnh cơ bản**
   * SELECT : thực hiện lấy thông tin tử bảng, có thể thực hiện lấy từ tất cả các trường (“\*”) hoặc có thể chọn 1 vài trường để lấy thông tin.

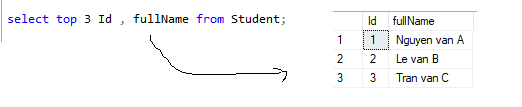
Có thể sử dụng từ khóa TOP để có thể lấy số bản ghi nhất định

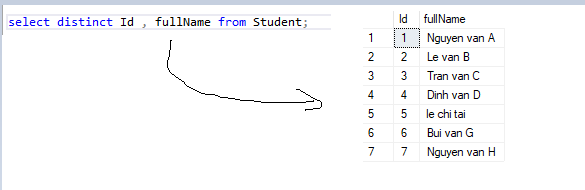
Sử dụng từ khóa DISTINCT để có thể lấy các bản ghi khác nhau.

Vd :

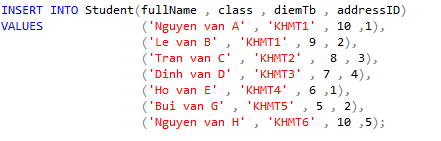


****

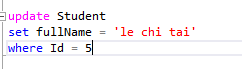
****

****

* + INSERT : đây là câu lệnh thực hiện thêm dữ liệu vào bảng (cần thêm đủ dữ liệu các trường, có thể bỏ qua các trường được thiết lập sẵn IDENTIY hoặc giá trị default)

****

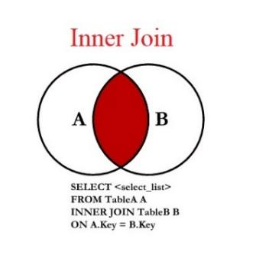
* + UPDATE : thực hiện cập nhật lại thông tin cho trường trong bảng, nên thiết lập điều kiện cập nhật để có thể cập nhật đúng bản ghi.

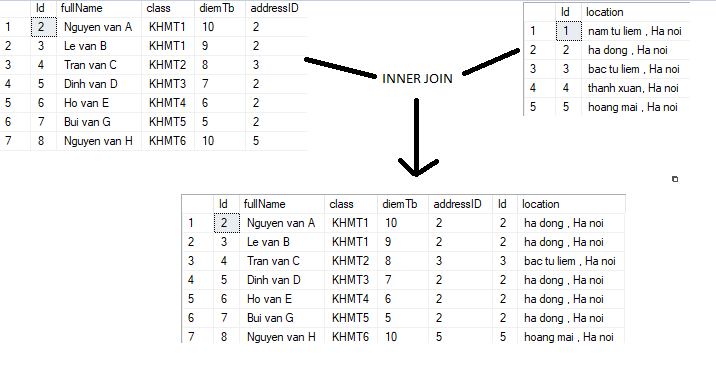
****

* + DELETE : thực hiện xóa đi bản ghi trong bảng (chú ý vấn đề khóa ngoại)

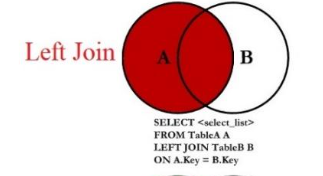
****

* + Join : đây là các câu lệnh thực hiện nối các bảng với nhau để thực hiện lấy thông tin, SQL Server bao gồm 4 loại Join
    - INNER JOIN : lấy các bản ghi phần chung giữa 2 bảng A và B

****

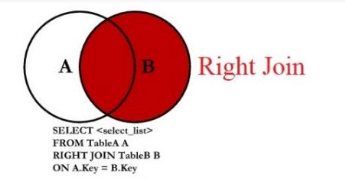
****

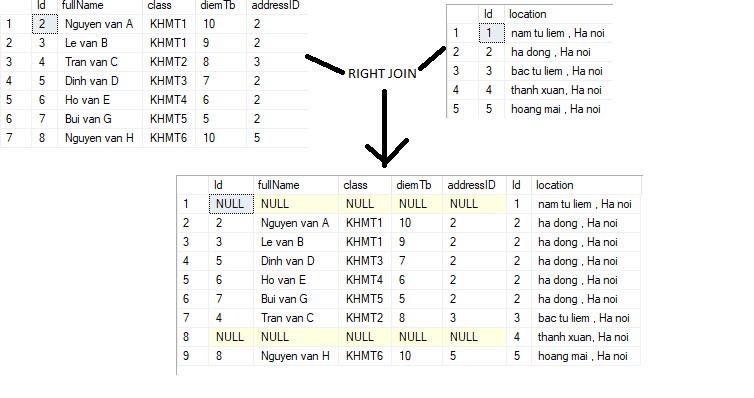
* + - LEFT [OUTER] JOIN : lấy các bản ghi chung giữa A và B cộng với các bản ghi riêng của A mà không chung với B.

****

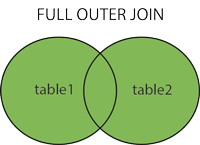
****

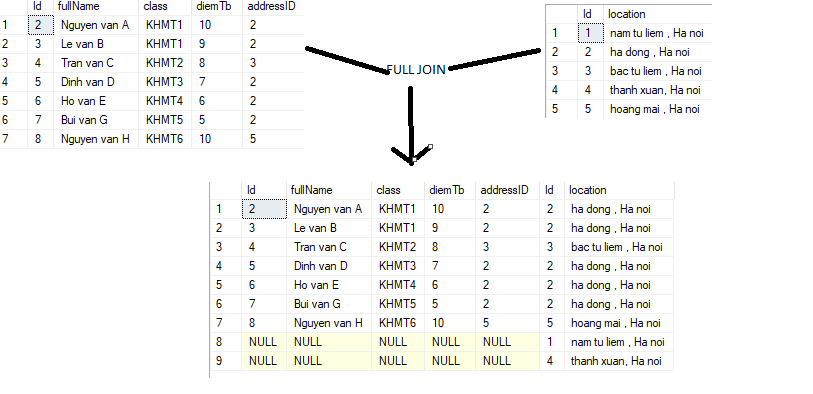
* + - RIGHT [OUTER] JOIN : lấy các bản ghi chung giữa A và B cộng với các bản ghi riêng của B mà chung với A.

****

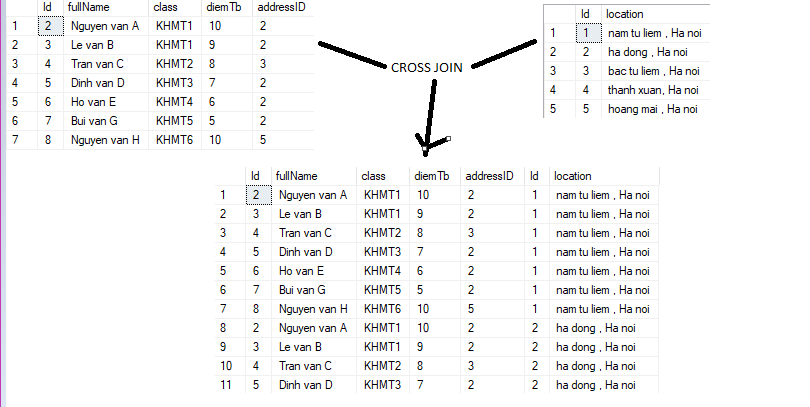
****

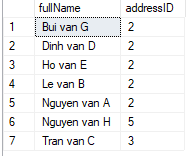
* + - FULL [OUTER] JOIN : lấy các bản ghi của 2 bảng A và B

****

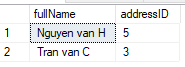
****

* + - CROSS JOIN : Thực hiện nhân tích đề các giữa các bản ghi của A với các bản ghi của B. Mỗi bản ghi của bảng A sẽ liên kết với tất cả các bản ghi của bảng B.



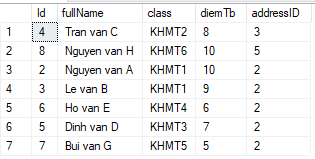
* + GROUP BY : thực hiện nhóm các bản ghi theo trường chỉ định
  + HAVING : đi cùng với Group By, để giúp giới hạn nhóm các trường, khá giống với where nhưng Having chỉ đi cùng với Group by và nó có khả năng sử dụng các Aggregate function (sum, min, max, count, …).





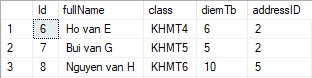
* + ORDER BY: Thực hiện sắp xếp các bản ghi trả về theo 1 thứ tự nhất định theo 1 trường chỉ định. Trong đó ASC : tăng dần, DESC giảm dần.

Mặc định của orderby là asc.



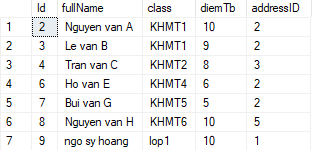
1. **Điều kiện và toán tử** 
   * Có thể được sử dụng so sánh , tạo điều kiện bởi where hoặc having.
     + AND : được sử dụng để kết hợp 2 hay nhiều điều kiện với nhau. Trả về đúng khi cả 2 điều kiện đều đúng.



****

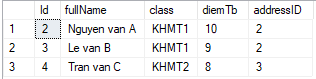
* + - OR : được sử dụng để kết hợp 2 hay nhiều điều kiện với nhau. Trả về đúng khi 1 trong các điều kiện đúng.

****

****

* + - IN và NOT IN: được sử dụng để tạo điều kiện so sánh giá trị của 1 trường có trong các giả trị chỉ định hay không. Thường được sử dụng để kiểm tra với cac giá trị được trả về từ 1 câu sub query. NOT IN là phủ định của IN.

****

****

****

****

* + - BETWEEN …AND… : được sử dụng để tạo điều kiện so sánh giá trị của trường chỉ định nằm trong 1 khoảng từ … đến … nào đó.

****

****

* + - LIKE : được sử dụng để so sánh đối với các giá trị của các trường có kiểu dữ liệu kiểu chuỗi kí tự/kí tự.
      * LIKE ‘%a’ : kết thúc bằng kí tự a.
      * LIKE ‘a%’ : bắt đầu bằng kí tự a.
      * LIKE ‘%a%’ : có chứa kí tự a.

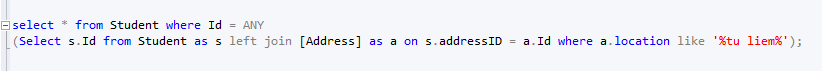
****

****

* + - Ngoài ra còn có toán tử so sánh NOT , đây là toán tử phủ định. Trả về đúng khi điều kiện sai, và ngược lại.

****

* + - Toán tử ANY, ALL,SOME, EXISTS : các toán tử này khá tương tự với IN hoặc NOT IN, nhiên chúng cần được sử dụng kết hợp với 1 toán tử so sánh (> , < , = , >= , <= , …) với ANY, ALL, SOME. Được sử dụng so sánh với kết quả trả về của 1 câu sub query.
      * ANY , SOME khá giống với IN : giá trị so sánh cần đúng với 1 trong các kết quả trả về sẽ trả về TRUE

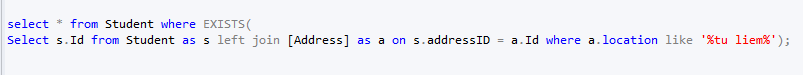
****

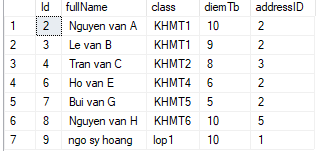
****

* + - * ALL : khá giống ANY, SOME , tuy nhiên giá trị so sánh cần đúng kết với tất cả kết quả trả về từ câu subquery.

****

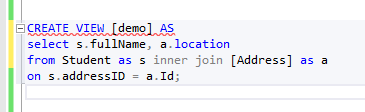
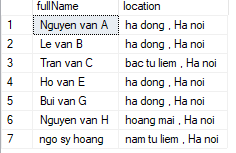
* + - * EXISTS : toán tử này được sử dụng để kiểm tra sự tồn tại kết quả trả về từ 1 câu subquery. Trả về True khi câu subquery tồn tại giá trị trả về



****

1. **VIEW**

* Khái niệm: Một view có thể coi như logic của table, và là một "bảng ảo" có nội dụng được xác định từ một câu truy vấn
* View không lưu trực tiếp data trong Database . Khi thay đổi dữ liệu của các bảng có liên quan đến View, thì khi truy xuất vào View, dữ liệu sẽ thay đổi
* View không tự thay đổi nếu cấu trúc của bảng có thay đổi
* View chỉ nên Select không nên Update, Insert, Delete Record
* Thực chất việc Update, Insert, Delete Record là gọi tới các table gốc và update trên đó

Vd :

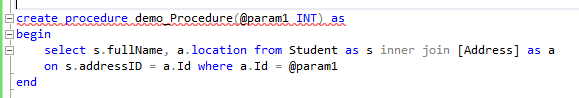
* Ưu điểm
* Bảo mật dữ liệu
* Đơn giản hóa các thao tác truy vấn dữ liệu
* Tập trung và đơn giản hóa dữ liệu
* Độc lập với structure của table
* Thường dùng làm report
* Nhược điểm
  + Query lâu hơn bình thường (do phải chuyển đổi từ select 🡪 View)

1. **STORE PROCEDURE**

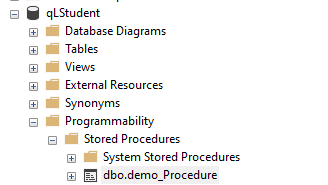
* Được sử dụng như một thủ tục thực hiện một nhiệm vụ, tác vụ nào đó khi được gọi đến.
* Có khả năng thực hiện các câu lệnh insert , select, update, delete,…
* Có khả năng truyền tham số vào trả về các giá trị (có thể trả về bảng các bản ghi.)
* Sử dụng từ khóa EXEC để gọi đến procedure.
* Set no count on/off:
  + SET NO COUNT OFF : đây là dạng mặc định , khi thực thi các tác vụ thay đổi db thì sẽ hiện thị ra số hàng trong bảng bị thay đổi trên màn hình out put.
  + SET NO COUNT ON : khi sử dụng câu lệnh này thì khi thực thi thay đổi DB sẽ không hiển thị số hàng bị thay đổi trên màn hình out put.

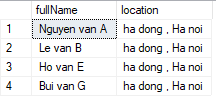
Vd:

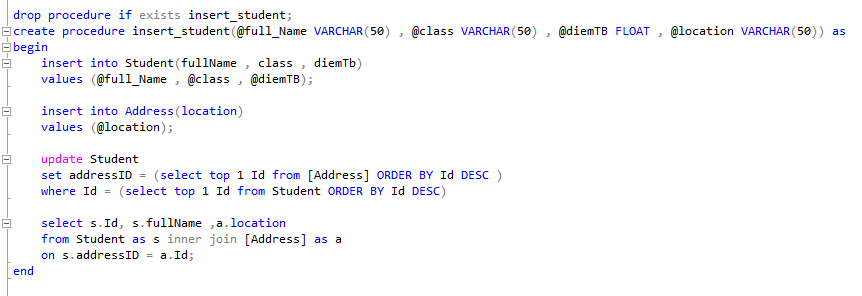
Thực hiện tạo procedure:



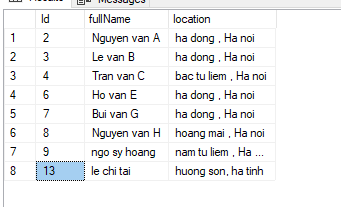
Sau khi tạo xong , có thể thực hiện kiểm tra đã hoàn thành tạo:

****

**** Sử dụng EXEC để gọi procedute và truyền tham số

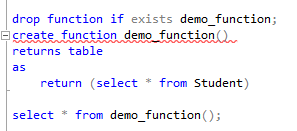
****

****

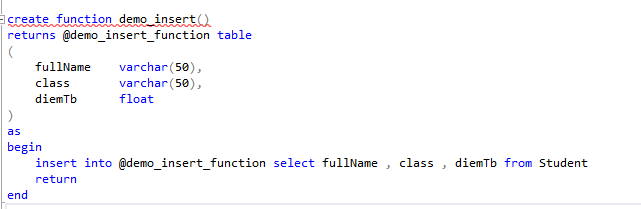
****

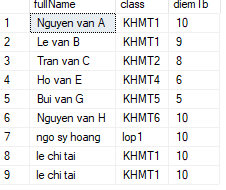
1. **Function.**

* Được sử dụng để thực hiện tính toán, thao tác 1 tác vụ nào đó được sử dụng nhiều lần trong SQL , từ đó gói thành function để có thể dễ dàng gọi đến khi cần.
* Có thể trả về giá trị của cả bảng sql.
* Sử dụng câu lệnh select để gọi đén giá trị trả về của function
* Vd :

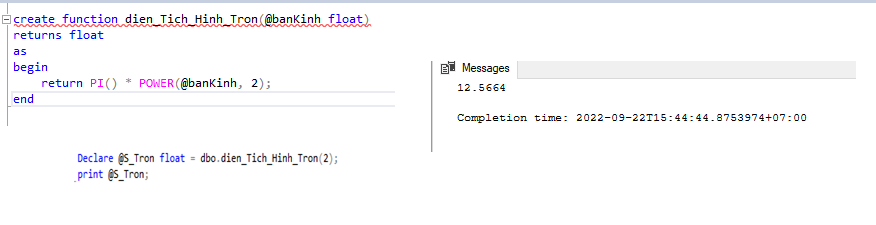
**** ****

* Vd:

****

****

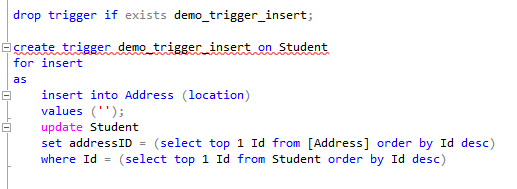
* + Sử dụng từ khóa Declare để định nghĩa biến giữ kết quả trả về của 1 function

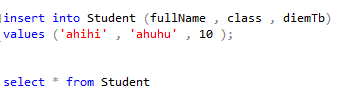
****

1. **Trigger**

* Là một dạng đặc biệt của procedure
* Tự động thực thi khi có một hoạt động nào đó được thực hiện thay đổi trong DB (upate, insert , delete).
* Sử dụng để
  + Kiểm tra tính toàn vẹn, bắt lỗi logic của cơ sở dữ liệu.
  + Các ràng buộc Contraint, giá trị default phức tạp.
  + Kiểm soát những thay đổi của dữ liệu trong table.

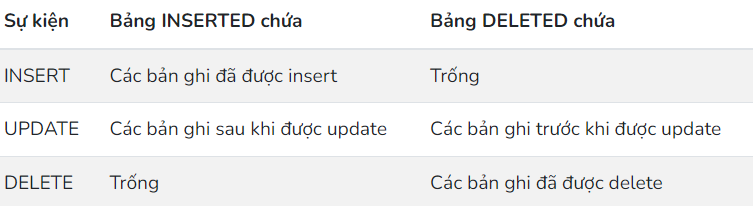
Vd :







* SQL Server cung câp 2 bảng INSERTED và DELETED chứa các dữ liệu trước và sau khi các sự kiện thay đổi xảy ra.



* Phân loại Trigger trong sql server:
  + DDL Trigger : Trigger sẽ tự động được gọi khi các sự kiện về Create, alter, … xảy ra trong DB.
  + DML Trigger : Trigger sẽ tự động được gọi khi các sự kiện về insert, update, delete xảy ra.
  + Login Trigger : Trigger sẽ tự động được gọi khi có một sự kiện đăng nhập của user vào DB.

1. **Statements**

* Các câu lệnh thực thi trong DB
  + - * DML (Data Multiplate Language) : đây là nhóm câu lệnh thực hiện thao tác với dữ liệu trong các bảng của DB (select , update, delete,.. ).
      * DDL (Data Define Language) : đây là nhóm câu lệnh thực hiện tạo database , tạo các bảng , thực hiện tạo cấu trúc cho DB (create, drop , truncate , …)
      * TCL (Transaction Control Language) : đây là nhóm câu lệnh thực hiện transaction trong SQL (commit, rollback).
      * DCL (Data Control Language) : đây là nhóm câu lệnh thực hiện cấp quyền cho người sử dụng (grant, Revoke).
      * Service Broker statements.
      * Session settings : thực thi các nhiệm vụ xử lý thơi gian chạy của section