

CÁC CÂU LỆNH TRONG SQL

Câu lệnh	Cú pháp
CSDL	
Tạo mới	CREATE DATABASE <Tên_Database>
Xóa	DELETE DATABASE <Tên_Database>
TẠO MỚI TABLE	
Tạo mới bảng	CREATE TABLE <Tên_Table> (<Tên_Cột_1> <Kiểu_DL> [NULL / NOT NULL], <Tên_Cột_2> <Kiểu_DL>) Có thể có thêm (Độ Dài) nếu cột đó xác định độ dài DL
Đặt khóa chính cho 1 cột khi đang tạo bảng:	CREATE TABLE <Tên_Table> (Tên_Cột_1 Kiểu_DL PRIMARY KEY , Tên_Cột_2 Kiểu_DL)
Đặt khóa chính cho nhiều cột khi đang tạo bảng:	CREATE TABLE <Tên_Table> (Tên_Cột_1 Kiểu_DL, Tên_Cột_2 Kiểu_DL PRIMARY KEY (Tên_Cột_1, Tên_Cột_2))
Đặt khóa ngoại cho 1 cột khi đang tạo bảng:	CREATE TABLE <Tên_Table> (<Tên_Cột_1> <Kiểu_DL> REFERENCES <Tên_TableTC> (Tên_KhóaChính_TC), <Tên_Cột_2> <Kiểu_DL>) CREATE TABLE <Tên_Table> (<Tên_Cột> <Kiểu_DL>, FOREIGN KEY (<Tên_Cột>) REFERENCES <Tên_TableTC> (Tên_KhóaChính_TC))
Tập ràng buộc CHECK	CREATE TABLE <Tên_Table> (Tên_Cột Kiểu_DL CHECK (<Biểu_Thức>)) Các toán tử điều kiện: So sánh: >, >=, <, <=

	Luận lý: AND, OR Khác: IN, BETWEEN, LIKE
HIỆU CHỈNH + XÓA TABLE ĐÃ TẠO	
Thêm cột	ALTER TABLE <Tên_Table> ADD <Tên_Cột> <Kiểu_Dữ_Liệu> [NULL / NOT NULL]
Xóa cột	ALTER TABLE <Tên_Table> DROP COLUMN <Tên_Cột>
Hiệu chỉnh cột	ALTER TABLE <Tên_Table> ALTER COLUMN <Tên_Cột> <Kiểu_DL_Mới> [NULL / NOT NULL]
Thêm ràng buộc khóa chính	ALTER TABLE <Tên_Table> ADD CONSTRAINT <Tên_Khóa_Chính> PRIMARY KEY (Tên_Cột)
Thêm ràng buộc khóa ngoại	ALTER TABLE <Tên_Table> ADD CONSTRAINT <Tên_Khóa_Ngoại> FOREIGN KEY (Tên_Cột) REFERENCES <Tên_Table_TC> (<Khóa_Chính_TC>)
Thêm ràng buộc CHECK	ALTER TABLE <Tên_Table> ADD CHECK <Biểu_Thức>
Xóa table	DROP TABLE <Tên_Table>
QUẢN TRỊ DỮ LIỆU	
Các toán tử so sánh:	=, >, >=, <, <=, <>, !
Toán tử đại diện:	_ : 1 ký tự % : nhiều ký tự [] : các ký tự thuộc khoảng trong ngoặc vuông [^] : các ký tự KHÔNG thuộc khoảng trong ngoặc vuông
Toán tử luận lý:	AND OR NOT
Các toán tử khác	
Thêm dữ liệu	INSERT [INTO] <Tên_Table> VALUES (Các Giá Trị Cách Nhau Bằng Dấu Phẩy)
Thêm dữ liệu vào Table1 từ bảng Table2	INSERT [INTO] <Tên_Table_1> SELECT <Tên_Cột> FROM <Tên_Table_2>
Cập nhật dữ liệu	UPDATE <Tên_Table> SET <Tên_Cột_1> = <Giá_Trị_1>, <Tên_Cột_2> = <Giá_Trị_2>
Xóa dữ liệu	DELETE FROM <Tên_Table> DELETE FROM <Tên_Table> WHERE <Điều_Kiện> TRUNCATE TABLE <Tên_Table>
TRUY VẤN	
Truy vấn thông thường	SELECT <DS_Tên_Cột> FROM <Tên_Table>

Truy vấn theo điều kiện	SELECT <DS_Tên_Cột> FROM <Tên_Table> WHERE <Điều_Kiện>
Nhóm kết quả truy vấn	SELECT GROUP BY <Tên_Cột>
Điều kiện	HAVING (<Biểu_Thức>)
Sắp xếp kết quả truy vấn	SELECT ORDER BY <Tên_Cột>
Loại bỏ các kết quả trùng nhau	SELECT DISTINCT <Tên_Cột>
Xem n mẫu tin đầu tiên	SELECT TOP n <Tên_Cột> FROM <Tên_Table> SELECT TOP 50 PERCENT <Tên_Cột> FROM <Tên_Table>
Đổi tên cột trong kết quả truy vấn	SELECT <TênCột> AS <Tên_Cột_Mới>
CÁC HÀM TRUY VẤN	
Tính tổng giá trị	SELECT SUM (<Tên_Cột>) FROM <Tên_Table>....
Đếm tổng số lượng mẫu tin	SELECT COUNT (<Tên_Cột/*>) FROM <Tên_Table>....
Tính trung bình	SELECT AVG ([ALL/DISTINCT]<Tên_Cột>) FROM <Tên_Table>....
Tính giá trị lớn nhất	SELECT MAX (<Tên_Cột>) FROM <Tên_Table>....
Tính giá trị nhỏ nhất	SELECT MIN (<Tên_Cột>) FROM <Tên_Table>....
TRUY VẤN TRÊN NHIỀU BẢNG	
Truy vấn trên 2 bảng	SELECT <Tên_Table>.<Tên_Cột> FROM <Tên_Table_1> JOIN <Tên_Table_2> ON <Điều_KiệnLK>
Truy vấn trên 3 bảng	SELECT <Tên_Table>.<Tên_Cột> FROM <Tên_Table_1> JOIN <Tên_Table_2> ON <Điều_KiệnLK1_2> JOIN <Tên_Table_3> ON <Điều_KiệnLK1/2/3>
CÁC ĐỐI TƯỢNG TRONG DB	
View	CREATE VIEW tên_view AS SELECT statment **Options: WITH CHECK OPTION
Index	CREATE INDEX tên_index

	<pre>ON tên_bảng (tên_cột) SELECT ... FROM tên_bảng WITH (INDEX = 'tên_index')</pre>
Stored procedure	<pre>CREATE PROCEDURE tên_sp AS câu_lệnh_sql EXEC tên_sp</pre>
	<pre>CREATE PROCEDURE tên_sp(đối_số_1, ...) AS câu_lệnh_sql; EXEC tên_sp giá_trị_1, ... *** Thêm OUTPUT sau return parameter</pre>
Trigger	<pre>CREATE TRIGGER tên_trigger ON tên_bảng {FOR / AFTER / INSTEAD OF } {DELETE, INSERT, UPDATE} AS câu_lệnh_sql *** Dữ liệu cập nhật lưu tạm thời trên 2 bảng : Inserted và Deleted</pre>
Scalar Functions	<pre>CREATE FUNCTION tên_hàm (@đối_số_1, ...) RETURNS kiểu_dữ_liệu AS</pre>

	<pre> BEGIN thân hàm RETURN value END Gọi: SELECT tên_hàm(giá_trị_1, ...) </pre>
Table-Valued Functions	<pre> CREATE FUNCTION tên_hàm (@đổi_số_1, ...) RETURNS TABLE AS RETURN câu_lệnh_select; Gọi: SELECT ... FROM tên_hàm(giá_trị_1, ...) </pre>