

# Cơ sở dữ liệu ứng dụng



MIS

# SQL

# Giới thiệu

SQL (Structured Query Language) là một ngôn ngữ máy tính được thiết kế cho việc quản lý dữ liệu trong Hệ thống quản trị cơ sở dữ liệu quan hệ (RDBMS)

# Lịch sử

- Được phát triển ở IBM từ những năm 1970 với tên gọi ban đầu là SEQUEL.
- Năm 1986: được ANSI/ISO chuẩn hóa.
- Các phiên bản mới được cập nhật liên tục qua các năm, mới nhất là SQL:2016 (ISO/IEC 9075:2016) (bạn có thể mua tại [đây](#), ~ \$2000).

Year	Name	Alias	Comments
1986	SQL-86	SQL-87	First formalized by ANSI.
1989	SQL-89	FIPS 127-1	Minor revision that added integrity constraints, adopted as FIPS 127-1.
1992	SQL-92	SQL2, FIPS 127-2	Major revision (ISO 9075), <i>Entry Level</i> SQL-92 adopted as FIPS 127-2.
1999	SQL:1999	SQL3	Added regular expression matching, <a href="#">recursive queries</a> (e.g. <a href="#">transitive closure</a> ), <a href="#">triggers</a> , support for procedural and control-of-flow statements, non-scalar types (arrays), and some object-oriented features (e.g. <a href="#">structured types</a> ). Support for embedding SQL in Java ( <a href="#">SQL/OLB</a> ) and vice versa ( <a href="#">SQL/JRT</a> ).
2003	SQL:2003		Introduced <a href="#">XML</a> -related features ( <a href="#">SQL/XML</a> ), <a href="#">window functions</a> , standardized sequences, and columns with auto-generated values (including identity-columns).
2006	SQL:2006		ISO/IEC 9075-14:2006 defines ways that SQL can be used with XML. It defines ways of importing and storing XML data in an SQL database, manipulating it within the database, and publishing both XML and conventional SQL-data in XML form. In addition, it lets applications integrate queries into their SQL code with <a href="#">XQuery</a> , the XML Query Language published by the World Wide Web Consortium ( <a href="#">W3C</a> ), to concurrently access ordinary SQL-data and XML documents. <sup>[35]</sup>
2008	SQL:2008		Legalizes ORDER BY outside cursor definitions. Adds INSTEAD OF triggers, TRUNCATE statement, <sup>[36]</sup> FETCH clause.
2011	SQL:2011		Adds temporal data (PERIOD FOR) <sup>[37]</sup> (more information at: <a href="#">Temporal database#History</a> ). Enhancements for <a href="#">window functions</a> and FETCH clause. <sup>[38]</sup>
2016	SQL:2016		Adds row pattern matching, polymorphic table functions, <a href="#">JSON</a> .
2019	SQL:2019		Adds Part 15, multidimensional arrays (MDarray type and operators).

# Phân loại mệnh đề SQL

SQL bao gồm nhiều mệnh đề khác nhau để tương tác trong RDBMS, người ta thường phân các mệnh đề này thành 4 nhóm:

- DQL (Data Query Language)
- DDL (Data Definition Language)
- DCL (Data Control Language)
- DML (Data Manipulation Language)

# Thực hiện trong sản phẩm

Tuy SQL được ANSI/ISO chuẩn hóa, nhưng vì nhiều nguyên nhân nên có sự khác nhau trong ngôn ngữ SQL (được thực hiện trong phần mềm) giữa các RDBMS.

Điều này dẫn đến là đôi khi phải chỉnh sửa câu lệnh khi chuyển đổi giữa các RDBMS.

Source	Abbreviation	Full name
ANSI/ISO Standard	SQL/PSM	SQL/Persistent Stored Modules
Interbase / Firebird	PSQL	Procedural SQL
IBM DB2	SQL PL	SQL Procedural Language (implements SQL/PSM)
IBM Informix	SPL	Stored Procedural Language
IBM Netezza	NZPLSQL <sup>[20]</sup>	(based on Postgres PL/pgSQL)
Invantive	PSQL <sup>[21]</sup>	Invantive Procedural SQL (implements SQL/PSM and PL/SQL)
MariaDB	SQL/PSM, PL/SQL	SQL/Persistent Stored Module (implements SQL/PSM), Procedural Language/SQL (based on Ada) <sup>[22]</sup>
Microsoft / Sybase	T-SQL	Transact-SQL
Mimer SQL	SQL/PSM	SQL/Persistent Stored Module (implements SQL/PSM)
MySQL	SQL/PSM	SQL/Persistent Stored Module (implements SQL/PSM)
MonetDB	SQL/PSM	SQL/Persistent Stored Module (implements SQL/PSM)
NuoDB	SSP	Starkey Stored Procedures
Oracle	PL/SQL	Procedural Language/SQL (based on Ada)
PostgreSQL	PL/pgSQL	Procedural Language/PostgreSQL Structured Query Language (based on reduced PL/SQL)
SAP R/3	ABAP	Advanced Business Application Programming
SAP HANA	SQLScript	SQLScript
Sybase	Watcom-SQL	SQL Anywhere Watcom-SQL Dialect
Teradata	SPL	Stored Procedural Language



**T-SQL**

**(Transact-SQL)**

# Giới thiệu

T-SQL (Transact-SQL) là một phiên bản mở rộng của SQL được phát triển và sử dụng bởi Microsoft và Sybase.

T-SQL được sử dụng trong Microsoft SQL Server. Mọi tương tác với MSSQL đều được thực hiện thông qua T-SQL.

# Giới thiệu

Mở rộng của T-SQL bao gồm:

- Lập trình thủ tục (procedural programming).
- Biến cục bộ.
- Các hàm toán học, xử lý thời gian, chuỗi, ...
- Mở rộng mệnh đề UPDATE và DELETE.

# Đặc trưng

- Không phân biệt hoa thường.
- Khoảng trắng không quan trọng.

# Funfact

T-SQL là một ngôn ngữ lập trình.

**DDL**

# Giới thiệu

DDL (Data Definition Language) là tập hợp các mệnh đề dùng để định nghĩa các cấu trúc dữ liệu, đặc biệt là schema của cơ sở dữ liệu.

Các mệnh đề DDL dùng để tạo và thay đổi các đối tượng trong cơ sở dữ liệu như bảng, index và người dùng.

Mệnh đề DDL thường dùng: CREATE, ALTER và DROP.

# Database

Database là nơi lưu trữ dữ liệu.

Database bao gồm nhiều schema, mỗi schema lại bao gồm một hay nhiều bảng.

Để tạo database mới, ta dùng mệnh đề `CREATE DATABASE`



# Mệnh đề **CREATE DATABASE**

Dùng để tạo một cơ sở dữ liệu mới.

Cú pháp (tối giản):

Cú pháp đầy đủ

```
CREATE DATABASE <Name>
```

# Bảng

Bảng là đối tượng trong MSSQL dùng để lưu trữ dữ liệu.

Bảng thường bao gồm các thành phần sau:

- Tên bảng
- Cột và dòng.
- Index (dùng để sắp xếp dữ liệu được lưu trữ trên phần cứng)
- Các ràng buộc (khóa chính, khóa ngoại, điều kiện, ...)

Để tạo bảng mới, ta dùng mệnh đề CREATE TABLE.

# Mệnh đề CREATE TABLE

Dùng để tạo một bảng mới.

Cú pháp:

```
CREATE TABLE <Name> (  
    col_1_name <datatype> [column constraint],  
    ...  
    [index],  
    [constraint],  
)
```

# Các kiểu dữ liệu trong MSSQL

Kiểu dữ liệu số:

- Số chính xác:
  - bit: 0, 1 và NULL
  - int:  $-2^{31}$  (-2,147,483,648) to  $2^{31}-1$  (2,147,483,647)
  - bigint:  $-2^{63}$  (-9,223,372,036,854,775,808) to  $2^{63}-1$  (9,223,372,036,854,775,807)
  - decimal: độ chính xác tối đa 38
- Số dấu phẩy động
  - float

# Các kiểu dữ liệu trong MSSQL

Kiểu dữ liệu chuỗi:

- Chuỗi độ dài cố định:
  - char(n)
  - nchar(n)
- Chuỗi có độ dài thay đổi:
  - varchar(n)
  - nvarchar(n)
- Chuỗi rất dài:
  - varchar(max)
  - nvarchar(max)

# Các kiểu dữ liệu trong MSSQL

## Kiểu dữ liệu thời gian

- date: lưu trữ ngày
- time: lưu trữ thời gian (00:00:00 đến 23:59:59)
- datetime: lưu trữ ngày giờ
- datetime2(n):  $1 \leq n \leq 7$ , giống datetime nhưng độ chính xác cao hơn.

# Các kiểu dữ liệu trong MSSQL

Các kiểu dữ liệu khác

- binary, varbinary
- filestream
- ...

# Bài tập

Xét CSDL của một trường đại học, viết câu lệnh CREATE TABLE cho các bảng dữ liệu sau:

- Sinh viên
- Giảng viên
- Khoa
- Môn học



# Ràng buộc cột

Là các ràng buộc mà dữ liệu trong cột phải tuân theo, dựa theo mô hình dữ liệu.

Một cột có thể có một hoặc nhiều ràng buộc, hoặc có thể không có ràng buộc nào.

# Các ràng buộc cột thường dùng

- NOT NULL: dữ liệu trong cột phải khác NULL.
- UNIQUE: các giá trị trong cột phải khác nhau.
- DEFAULT: gán giá trị mặc định cho cột.
- IDENTITY: giá trị tự động tăng
- PRIMARY KEY: dùng để chỉ ra khóa chính
- FOREIGN KEY: dùng để chỉ ra khóa phụ
- CHECK : kiểm tra một điều kiện nào đó

# NOT NULL

Từ khóa NOT NULL có ý nghĩa là không thể chứa giá trị NULL trong cột dữ liệu.

Cú pháp:

```
<column_name> <datatype> NOT NULL
```

# UNIQUE

Từ khóa UNIQUE có ý nghĩa là giá trị trong cột phải khác nhau.

Lưu ý, trong cột có thể chứa giá trị NULL, nhưng chỉ xuất hiện đúng 1 lần.

Cú pháp:

```
<column_name> <datatype> UNIQUE
```

# PRIMARY KEY

Từ khóa dùng để xác định cột khóa chính cho bảng.

Phải dùng chung với NOT NULL.

Cú pháp:

```
<column_name> <datatype> NOT NULL PRIMARY KEY
```

# Khóa chính

Dùng để phân biệt các dòng trong bảng, tức là hai dòng khác nhau phải có khóa chính khác nhau. Ngoài ra, khóa chính phải khác NULL.

Khóa chính có thể là một hay nhiều cột.

Có hai dạng khóa chính: tự nhiên và nhân tạo.

# Khóa chính

Khóa tự nhiên: trong dữ liệu đã có sẵn trường (cột) giúp phân biệt các dữ liệu khác nhau. Ví dụ: số CMND.

Khóa nhân tạo: có thể hiểu đơn giản là số thứ tự khi ghi xuống csdl.

Có một số khóa nhân tạo thường dùng: số tăng dần, uuid.

# IDENTITY

Dùng để chỉ rằng đây là cột có dữ liệu tăng dần.

Thường dùng để tạo dãy tăng dần cho khóa chính nhân tạo.

Cú pháp:

```
<column_name> <datatype> IDENTITY(<start>, <step>)
```

Ý nghĩa: Dòng đầu tiên sẽ có giá trị là <start>, mỗi dòng kế tiếp số sẽ tăng thêm <step>.



# FOREIGN KEY

Dùng để xác định cột khóa phụ trong mô hình dữ liệu quan hệ.

Cú pháp

```
FOREIGN KEY REFERENCES <table_name>(<column_name>)
```

Luôn phải chỉ ra cột tham chiếu bằng REFERENCES

<table\_name>(<column\_name>).

Cột tham chiếu <column\_name> phải là khóa chính của <table\_name>.

# FOREIGN KEY

Ngoài ra, bạn có thể xác định thêm một số hành động với dòng chứa khóa ngoại khi có sự thay đổi trên khóa chính bằng

`ON (UPDATE|DELETE) (NO ACTION|SET NULL|SET DEFAULT|CASCADE)`

Mặc định khi tạo bảng là `ON (UPDATE|DELETE) NO ACTION`

# DEFAULT

Từ khóa DEFAULT xác định giá trị mặc định cho cột. Tức là khi thêm dữ liệu vào bảng, nếu không chỉ ra giá trị cho cột thì nó sẽ lấy giá trị mặc định trong DEFAULT.

Cú pháp:

```
DEFAULT <default_value>
```

# CHECK

Từ khóa CHECK dùng để kiểm tra điều kiện mà dữ liệu trong cột bắt buộc phải thỏa.

Cú pháp:

CHECK <condition>

# Index

Là một đối tượng của csdl, dùng để sắp xếp dữ liệu trên phần cứng.

Có 2 loại là clustered và non-clustered.

Một bảng chỉ có 1 (một) clustered index nhưng có thể có nhiều non-clustered index.

Nếu không chỉ cụ thể, một clustered index sẽ được tạo ra trên cột khóa chính.

# Ràng buộc bảng

Là những ràng buộc phức tạp trên bảng.

Thường có tác dụng trên nhiều cột.

Cú pháp

```
CONSTRAINT <constraint_name> <constraint>
```

## Lưu ý

<constraint\_name> phải duy nhất trong toàn database.

Với mỗi <constraint> sẽ có cấu trúc khác nhau.

# PRIMARY KEY

Cú pháp

```
CONSTRAINT <constraint_name> PRIMARY KEY (<list_column>)
```



# FOREIGN KEY

Cú pháp

```
CONSTRAINT <constraint_name> FOREIGN KEY (<list_column>) REFERENCES  
<table_name>(<list_column>)
```

# DEFAULT

Cú pháp


```
CONSTRAINT <constraint_name> DEFAULT <default_value> FOR <column_name>
```

# CHECK


Cú pháp:

```
CONSTRAINT <constraint_name> CHECK <condition>
```


**employees**

Column Name	Data Type	Allow Nulls	Description
 id	int	<input type="checkbox"/>	
birth_date	date	<input checked="" type="checkbox"/>	
first_name	nvarchar(20)	<input checked="" type="checkbox"/>	
last_name	nvarchar(20)	<input checked="" type="checkbox"/>	
gender	bit	<input type="checkbox"/>	
hire_date	date	<input checked="" type="checkbox"/>	
		<input type="checkbox"/>	

**salaries \***

Column Name	Data Type	Allow Nulls	Description
 id	int	<input type="checkbox"/>	
emp_id	int	<input type="checkbox"/>	
salary	int	<input checked="" type="checkbox"/>	>= 1,500,000
from_date	date	<input checked="" type="checkbox"/>	
to_date	date	<input checked="" type="checkbox"/>	> from_date
		<input type="checkbox"/>	

**titles \***

Column Name	Data Type	Allow Nulls	Description
 id	int	<input type="checkbox"/>	
emp_id	int	<input type="checkbox"/>	
from_date	date	<input type="checkbox"/>	
to_date	date	<input checked="" type="checkbox"/>	> from_date
title	nvarcha...	<input checked="" type="checkbox"/>	
		<input type="checkbox"/>	

**Thay đổi định  
nghĩa bảng**

# Giới thiệu

Trong suốt vòng đời của mình, định nghĩa của một bảng luôn có sự thay đổi để phù hợp với yêu cầu lưu trữ dữ liệu.

T-SQL cũng có các mệnh đề để thực hiện các hành động thay đổi định nghĩa bảng.

# Giới thiệu

Các hành động thay đổi định nghĩa bảng có thể chia làm 2 loại:

- Thêm một thành phần nào đó (cột, ràng buộc)
- Xóa một thành phần nào đó (cột, ràng buộc)

Ngoài ra, còn một mệnh đề giúp thay đổi kiểu dữ liệu của cột.

# Giới thiệu

Các mệnh đề trên có cú pháp chung là:

```
ALTER TABLE <table_name> <action>
```



# Thêm cột mới

Cú pháp:

```
ALTER TABLE <table_name>
```

```
ADD <column_name> <datatype> [column_constraint]
```

# Thêm ràng buộc mới

Có 2 cú pháp:

```
ALTER TABLE <table_name>
```

```
ADD <constraint>
```

hoặc

```
ALTER TABLE <table_name>
```

```
ADD CONSTRAINT <constraint_name> <constraint>
```

# Thêm ràng buộc mới

Điểm khác nhau giữa hai cú pháp:

- Khi thực hiện cú pháp 1, MSSQL sẽ tự động sinh ra một tên cho <constraint> mới thêm vào.
- Khi thực hiện cú pháp 2, MSSQL sẽ lấy <constraint\_name> làm tên của <constraint> mới.

Lưu ý, tên của constraint là duy nhất trong một database.

# Thay đổi ràng buộc

Cú pháp cụ thể cho một số trường hợp (dùng cú pháp 1):

- Thêm PRIMARY KEY

```
ALTER TABLE <table_name>
```

```
ADD PRIMARY KEY (<column_list>)
```

# Thay đổi ràng buộc

Thêm FOREIGN KEY

```
ALTER TABLE <table_name>
```

```
ADD FOREIGN KEY (<column_list>
```

```
REFERENCES <table_2_name>(<column_list_2>)
```

# Thay đổi ràng buộc

Thêm ràng buộc UNIQUE

```
ALTER TABLE <table_name>
```

```
ADD UNIQUE (<column_list>)
```

# Thay đổi ràng buộc

Thêm ràng buộc DEFAULT (ràng buộc này chỉ có thể dùng dạng 2)

```
ALTER TABLE <table_name>
```

```
ADD CONSTRAINT <constraint_name>
```

```
DEFAULT <default_value> FOR <column_name>
```

# Thay đổi ràng buộc

Thêm ràng buộc CHECK

```
ALTER TABLE <table_name>
```

```
ADD CHECK <condition>
```



# Xóa cột

Cú pháp:

```
ALTER TABLE <table_name>
```

```
DROP COLUMN <column_name>
```

# Xóa ràng buộc

Cú pháp:

```
ALTER TABLE <table_name>
```

```
DROP CONSTRAINT <constraint_name>
```

# Một số thao tác chỉnh sửa khác

Thay đổi kiểu dữ liệu và ràng buộc cột

```
ALTER TABLE <table_name>
```

```
ALTER COLUMN <column_name> <new_datatype> [new_constraints]
```

# Một số thao tác chỉnh sửa khác

Thay đổi tên cột

- Dùng giao diện của SQL Server Management Studio, hoặc
- Dùng cú pháp

EXEC sp\_rename <old\_name>, <new\_name>, 'COLUMN'

# Một số thao tác chỉnh sửa khác

Xóa giá trị mặc định của cột

```
ALTER TABLE <table_name>
```

```
ALTER COLUMN <column_name> DROP DEFAULT
```

# Xóa bảng

Cú pháp

```
DROP TABLE <table_name>
```