# DATABASE

## Tạo DataBase

CREATE DATABASE <tên database>

Ví dụ:

CREATE DATABASE SQLDBQUERY

## Xoá DataBase

DROP DATABASE <tên database cần xóa>

Ví dụ:

CREATE DATABASE SQLDBQUERY

## Comment

--Nội dung Comment

## Sử dụng Database

USE  <tên database sử dụng>

# Bảng

## Tạo bảng

CREATE TABLE <Tên Table>

(

<Tên trường 1> <Kiểu dữ liệu>,

<Tên trường 2> <Kiểu dữ liệu>,

<Tên trường n> <Kiểu dữ liệu>

);

Ví dụ:

-- Tạo Table SINHVIEN có các thuộc tính MASV,TENSV,KHOA

CREATE TABLE SINHVIEN

(

            MASV NCHAR(10),

            TENSV NVARCHAR(100),

            KHOA NVARCHAR(100)

);

Lưu ý: Tạo xong mà không tìm thấy bảng thì: Click chuột phải vào Table -> Refresh

## Xoá bảng

DROP TABLE <Tên Table>

### Thao tác trong bảng

#### Thêm cột

ALTER TABLE <Tên Table> ADD <Tên Column> <Kiểu dữ liệu>

Ví dụ:

ALTER TABLE PhongBan

ADD COLUMN TruongPhong varchar(50);

#### Xoá cột

ALTER TABLE <Tên Table> DROP <Tên Column>

Ví dụ:

ALTER TABLE PhongBan

DROP COLUMN TruongPhong;

#### Thêm ràng buộc

ALTER TABLE table\_name

ADD CONSTRAINT constraint\_name

CHECK (condition);

Ví dụ:

Thêm ràng buộc kiểm tra lương tối thiểu

ALTER TABLE NHAN\_VIEN

ADD CONSTRAINT CHK\_LUONG\_TOITHIEU

CHECK (LUONG >= 100);

Thêm một ràng buộc Khoá chính- Primary key:

ALTER TABLE NHAN\_VIEN

ADD CONSTRAINT PRI\_NhanVien

PRIMARY KEY (ID\_NHANVIEN);

Thêm ràng buộc UNIQUE

ALTER TABLE NHAN\_VIEN

ADD CONSTRAINT NHANVIEN\_UNQ\_EMAIL

UNIQUE (EMAIL);

## Thêm dữ liệu

### Thêm dữ liệu theo cột tuỳ chọn

INSERT INTO TEN\_BANG (cot1, cot2, cot3,...cotN)

VALUES (gia\_tri1, gia\_tri2, gia\_tri3,...gia\_triN);

Ví dụ:

INSERT INTO NHANVIEN (ID,TEN,TUOI,DIACHI,LUONG)

VALUES (1, 'Thanh', 24, 'Haiphong', 2000.00 );

INSERT INTO NHANVIEN (ID,TEN,TUOI,DIACHI,LUONG)

VALUES (2, 'Loan', 26, 'Hanoi', 1500.00 );

### Thêm dữ liệu theo cột mặc định

INSERT INTO TEN\_BANG

VALUES (gia\_tri1, gia\_tri2, gia\_tri3,...gia\_triN);

Ví dụ:

INSERT INTO NHANVIEN

VALUES (7, 'LAM', 29, 'Hanoi', 15000.00 );

# Khoá chính

## Tạo khoá chính lúc khởi tạo bảng

Tạo khoá chính trong quá trình khai báo cột

CREATE TABLE <Tên Table>

(

           <column1> <kiểu dữ liệu> PRIMARY KEY,

            <column2> <kiểu dữ liệu>,

            …

   <columnn> <kiểu dữ liệu>

)

## Ví dụ:

CREATE TABLE HOCSINH

(

MAHS CHAR(5) PRIMARY KEY,

TEN NVARCHAR(30),

NAM BIT,

NGAYSINH DATETIME,

DIACHI VARCHAR(20),

DIEMTB FLOAT

)

## Tạo khoá chính sau khi khai báo các cột

CREATE TABLE <Tên Table>

(

            <column1> <kiểu dữ liệu>,

            <column2> <kiểu dữ liệu>,

            ...

            <columnn> <kiểu dữ liệu>

            PRIMARY KEY (columnKey1, columnKey2…, columnKeyn)

)

Ví dụ:

CREATE TABLE GIAOVIEN

(

TEN NVARCHAR(30) NOT NULL,

Nam BIT,

DIACHI VARCHAR(20),

LUONG MONEY,

PRIMARY KEY (TEN, NGAYSINH)

)

## Tạo khoá chính cho bảng có sẵn bàng ALTER

ALTER TABLE <Tên Table> ADD PRIMARY KEY(column)

ALTER TABLE <TênTable> ADD PRIMARY KEY(column**1**,column**2**,…)

Ví dụ:

ALTER TABLE dbo.DIEM ADD CONSTRAINT PK\_DIEM PRIMARY KEY(MAHS,MALOP)

## Đặt tên cho khoá chính với CONSTRAIN

CONSTRAINT PK\_<Tên Table> PRIMARY KEY (Column**Key1**, Column**Key2**,…, Column**Keyn**)

Ví dụ:

Trong lúc tạo bảng

CREATE TABLE LOPHOC

(

MALOP CHAR(5) NOT NULL UNIQUE,

TENLOP NVARCHAR(30),

SOLUONG INT,

CONSTRAINT PK\_LOPHOC PRIMARY KEY (MALOP)

)

Bảng có sẵn

ALTER TABLE dbo.DIEM ADD CONSTRAINT PK\_DIEM PRIMARY KEY(MAHS,MALOP)

## Xoá khoá chính có tên

ALTER TABLE <Tên Table> DROP CONSTRAINT <Tên Khóa chính>

Ví dụ:

ALTER TABLE dbo.DIEM DROP CONSTRAINT PK\_DIEM

# Khoá ngoại

## Tạo khoá ngoại trong khi tạo bảng

CREATE TABLE <Table Foreign>

(

…

FOREIGN KEY (<ColumnF, ColumnF1, ColumnF2,…>)

REFERENCES <Table Key> (<ColumnK, ColumnK1, ColumnK2,…>)

)

Ví dụ

CREATE TABLE GiaoVien

(

            MaGV CHAR(10) NOT NULL,

            Name NVARCHAR(100) DEFAULT N'Tên giáo viên',

            DiaChi NVARCHAR(100) DEFAULT N'Địa chỉ giáo viên',

            NgaySinh DATE,

            Sex BIT,

            MaBM CHAR(10),

            -- Tạo khóa ngoại ngay khi tạo bảng

            FOREIGN KEY(MaBM) REFERENCES dbo.BOMON(MaBM)

)

## Tạo khoá ngoại sau khi tạo bảng

ALTER TABLE <Table Foreign>

ADD FOREIGN KEY(<ColumnF, ColumnF1, ColumnF2,…>)

REFERENCES <Table Key> (<ColumnK, ColumnK1, ColumnK2,…>)

## Đặt tên cho khoá ngoại

ALTER TABLE <Table Foreign>

ADD CONSTRAINT <Tên khóa ngoại>

FOREIGN KEY(<ColumnF, ColumnF1, ColumnF2,…>)

REFERENCES <Table Key> (<ColumnK, ColumnK1, ColumnK2,…>)

Ví dụ:

ALTER TABLE HocSinh ADD CONSTRAINT FK\_HS FOREIGN KEY(MaLop) REFERENCES Lop(MaLop)

## Huỷ khoá ngoại

ALTER TABLE <Table **Foreign**> DROP CONSTRAINT <Tên khóa ngoại>

# TRUY VẤN

## Cấu trúc truy vấn

SELECT [tính chất] <danh sách column> FROM <danh sách Table/Query>

## Select \*

Để lấy tất cả dữ liệu từ 1 hay nhiều bảng

SELECT \* FROM <tên table1>, < tên table2>,…

## SELECT DISTINCT

Trong table, ngoài khóa chính, các column khác đều có thể chứa giá trị trùng lặp. Và đôi khi bạn cần lấy ra những dữ liệu hoàn toàn riêng biệt. Để đảm bảo điều này ta dùng từ khóa DISTINCT

SELECT DISTINCT <tên column> FROM <tên table>

Ví dụ:

-- Lấy tất cả mã khoa từ Table BOMON

SELECT MAKHOA

FROM BOMON

-- Lấy tất cả mã khoa không trùng nhau từ Table BOMON

SELECT DISTINCT MAKHOA

FROM BOMON

## SELECT TOP (MySQL workbench không hỗ trọ)

Sử dụng LIMIT để thay thế

SELECT column1,column2,...

FROM table

LIMIT count;

Ví dụ: SELECT \* FROM CUSTOMERS LIMIT 3;

## Truy vấn có điều kiện

SELECT [tính chất] <danh sách column>

FROM <danh sách Table/Query>

WHERE <điều kiện>

Ví dụ:

-- 1. Lấy ra Mã GV, tên GV và họ tên người thân tương ứng

SELECT GV.MAGV, HOTEN, NT.TEN FROM GIAOVIEN AS GV, NGUOITHAN AS NT

WHERE GV.MAGV=NT.MAGV

-- 2. Lấy ra Mã GV, Tên GV và Tên khoa của giáo viên đó làm việc. Gơi ý: Bộ môn nằm trong khoa

SELECT gv.MAGV, gv.HOTEN, k.TENKHOA FROM GIAOVIEN AS GV,BOMON AS BM, dbo.KHOA AS K

WHERE GV.MABM=BM.MABM AND BM.MAKHOA=K.MAKHOA

## COUNT()

Hàm COUNT() là một hàm rất hữu ích, dùng để đếm số lượng record được chỉ định.

Cú pháp

SELECT COUNT(<Tên column>)  
FROM <Tên table>  
WHERE <Điều kiện>

Đếm tất cả bản ghi

SELECT COUNT(\*) FROM <Tên Table>

Ví dụ 1: Đếm số lượng giáo viên có trong Table Giáo Viên

​​​​​​​SELECT COUNT(\*) AS N'Số lượng giáo viên' FROM GIAOVIEN

Ví dụ 2: Đếm số lượng người thân của Giáo viên có mã GV là 007

SELECT COUNT(\*) AS N'Số lượng người thân'

FROM dbo.GIAOVIEN, dbo.NGUOITHAN

WHERE GIAOVIEN.MAGV = '007'

AND GIAOVIEN.MAGV = NGUOITHAN.MAGV

## AVG

Hàm AVG() là một hàm tính toán, dùng để lấy giá trị trung bình của một column có giá trị số được chỉ định

Cú pháp:

SELECT AVG(<Tên column>)  
FROM <Tên table>  
WHERE <Điều kiện>

Ví dụ:

SELECT AVG(LUONG) AS 'TBLUONG' FROM GIAOVIEN

## SUM()

Hàm SUM() là hàm tính toán, dùng để lấy tính tổng giá trị số của một column được chỉ định.

Cú pháp:

SELECT SUM(<Tên column>)  
FROM <Tên table>  
WHERE <Điều kiện>

****Ví dụ:****Xuất ra tổng kinh phí dành cho các đề tài có thời gian kết thúc trước năm 2009

​​​​​​​SELECT SUM(KINHPHI) AS 'TONG KINH PHI' FROM DETAI

WHERE YEAR(NGAYKT) < 2009

## Tìm kiếm gần đúng

Cấu trúc:

SELECT [tính chất] <danh sách column>

FROM <danh sách Table/Query>

WHERE <column> LIKE <dữ liệu mẫu>

### Một số dạng tìm kiếm gần đúng

|  |  |
| --- | --- |
| **Dạng tìm kiếm** | **Mệnh đề WHERE** |
| Tìm kiếm dữ liệu bắt đầu bằng ký tự K | …WHERE <column> LIKE ‘K%’ |
| Tìm kiếm dữ liệu kết thúc bằng ký tự K | …WHERE <column> LIKE ‘%K’ |
| Tìm kiếm dữ liệu có chứa ký tự Kt ở vị trí bất kỳ | …WHERE <column> LIKE ‘%Kt%’ |
| Tìm kiếm dữ liêu có ký tự K ở vị trí thứ hai | …WHERE <column> LIKE ‘\_K%’ |
| Tìm kiếm dữ liệu bắt đầu bằng ký tự K, và có ít nhất có chiều dài là 3 ký tự | …WHERE <column> LIKE ‘K\_%\_%’ |
| Tìm kiếm dữ liệu bắt đầu bằng ký tự K, kết thúc bằng ký tự m | …WHERE <column> LIKE ‘K%m’ |

**Ví dụ 1:**Xuất ra thông tin giáo viên mà tên bắt đầu bằng chữ l

SELECT \* FROM GIAOVIEN

WHERE HOTEN like 'l%'

## INNER JOIN

Cấu trúc:

SELECT <Danh sách column>

FROM <Table A> INNER JOIN <Table B>

ON <table A>.<tên Column> = <table B>.<tên Column>

(có thể bỏ INNER)

****Ví dụ 2:**** Xuất ra thông tin giáo viên (mã GV và tên) và tên người thân tương ứng của giáo viên đó

SELECT GV.MAGV,GV.HOTEN, NT.TEN AS "NGUOI THAN"

FROM GIAOVIEN AS GV

INNER JOIN NGUOITHAN AS NT ON NT.MAGV = GV.MAGV

## INNER JOIN nhiều bảng

SELECT <Danh sách column>

FROM <Table A>

INNER JOIN <Table B> ON <điều kiện join B>

INNER JOIN <Table C> ON <điều kiện join C>….

LƯU Ý: Có thể kết hợp điều kiện vào INNER JOIN

****Ví dụ:**** Xuất ra thông tin các giáo viên nam, bộ môn và khoa mà giáo viên đó giảng dạy.

SELECT GV.MAGV,GV.HOTEN,GV.PHAI,BM.TENBM, K.TENKHOA

FROM dbo.BOMON AS BM

     JOIN dbo.GIAOVIEN AS GV ON GV.MABM = BM.MABM

     JOIN dbo.KHOA AS K ON K.MAKHOA = BM.MAKHOA

WHERE GV.PHAI ='NAM'

## LEFT/ RIGHT OUTER JOIN

SELECT <Danh sách column>

FROM <Table A>  LEFT | RIGHT  JOIN <Table B>

ON <Điều kiện kết hợp AB>

****Ví dụ 1:**** Xuất ra danh sách giáo viên chủ nhiệm hoặc chưa chủ nhiệm đề tài. Ở đây ta chỉ chọn những thông tin chính cần thiết đáp ứng nhu cầu truy vấn

SELECT GV.MAGV, GV.HOTEN, DT.MADT,DT.TENDT

FROM dbo.GIAOVIEN AS GV

LEFT JOIN dbo.DETAI AS DT ON DT.GVCNDT = GV.MAGV

## ORDER BY

Cấu trúc:

SELECT [DISTINCT] Column(s) FROM TableName

[WHERE Conditions ]

ORDER BY Column(s) [asc|desc]

Ví dụ:

SELECT \* from NHAN\_VIEN ORDER BY ID\_NhanVien