

5.

Data Definition Language : DDL

(คำสั่ง SQL ในการจัดการโครงสร้างของฐานข้อมูล)

Data Manipulation Language : DML

(คำสั่ง SQL ในการจัดการข้อมูล)

DB304-DB404 การจัดการฐานข้อมูลทางธุรกิจดิจิทัล

อาจารย์เฉลิมพล ศิริกายน และ ดร.ปาณิตา ชูสรานนท์

ประเภทของคำสั่ง SQL: Structured Query Language

- คำสั่ง SQL ในการจัดการโครงสร้างของฐานข้อมูล (Data Definition Language: **DDL**)
 - คำสั่งในการสร้างฐานข้อมูล (Create DB)
 - คำสั่งในการสร้างตาราง (CREATE TABLE)
 - ชนิดของข้อมูล (Data Type)
 - คำสั่งในการแก้ไขโครงสร้างตาราง (ALTER TABLE)
 - คำสั่งในการลบตารางและลบฐานข้อมูล (DROP)
- คำสั่ง SQL ในการจัดการข้อมูล (Data Manipulation Language: **DML**)
 - INSERT, SELECT, UPDATE, DELETE

คำสั่ง SQL ในการจัดการโครงสร้างของฐานข้อมูล Data Definition Language (DDL)

The SQL CREATE DATABASE Statement

The CREATE DATABASE statement is used to create a new SQL database.

Syntax

```
CREATE DATABASE databasename;
```

The SQL DROP DATABASE Statement

The DROP DATABASE statement is used to drop an existing SQL database.

Syntax

```
DROP DATABASE databasename;
```

The SQL CREATE TABLE Statement

The CREATE TABLE statement is used to create a new table in a database.

Syntax

```
CREATE TABLE table_name (  
    column1 datatype,  
    column2 datatype,  
    column3 datatype,  
    ....  
);
```

SQL Data Types

String data types:

Data type	Description
CHAR(size)	A FIXED length string (can contain letters, numbers, and special characters). The <i>size</i> parameter specifies the column length in characters - can be from 0 to 255. Default is 1
VARCHAR(size)	A VARIABLE length string (can contain letters, numbers, and special characters). The <i>size</i> parameter specifies the maximum column length in characters - can be from 0 to 65535

SQL Data Types

Numeric data types:

Data type	Description
BIT(<i>size</i>)	A bit-value type. The number of bits per value is specified in <i>size</i> . The <i>size</i> parameter can hold a value from 1 to 64. The default value for <i>size</i> is 1.
TINYINT(<i>size</i>)	A very small integer. Signed range is from -128 to 127. Unsigned range is from 0 to 255. The <i>size</i> parameter specifies the maximum display width (which is 255)
BOOL	Zero is considered as false, nonzero values are considered as true.
BOOLEAN	Equal to BOOL

SQL Data Types

Numeric data types:

Data type	Description
INT(<i>size</i>)	A medium integer. Signed range is from -2147483648 to 2147483647. Unsigned range is from 0 to 4294967295. The <i>size</i> parameter specifies the maximum display width (which is 255)
INTEGER(<i>size</i>)	Equal to INT(<i>size</i>)
BIGINT(<i>size</i>)	A large integer. Signed range is from -9223372036854775808 to 9223372036854775807. Unsigned range is from 0 to 18446744073709551615. The <i>size</i> parameter specifies the maximum display width (which is 255)
FLOAT(<i>size</i> , <i>d</i>)	A floating point number. The total number of digits is specified in <i>size</i> . The number of digits after the decimal point is specified in the <i>d</i> parameter. This syntax is deprecated in MySQL 8.0.17, and it will be removed in future MySQL versions

5E-18

0.000000000000000005

SQL Data Types

Numeric data types:

Data type	Description
DOUBLE(<i>size</i> , <i>d</i>)	A normal-size floating point number. The total number of digits is specified in <i>size</i> . The number of digits after the decimal point is specified in the <i>d</i> parameter
DECIMAL(<i>size</i> , <i>d</i>)	An exact fixed-point number. The total number of digits is specified in <i>size</i> . The number of digits after the decimal point is specified in the <i>d</i> parameter. The maximum number for <i>size</i> is 65. The maximum number for <i>d</i> is 30. The default value for <i>size</i> is 10. The default value for <i>d</i> is 0.

ตัวอย่างการสร้างตาราง

```
CREATE TABLE people (  
    id CHAR(13),  
    name VARCHAR(50),  
    dob DATE,  
    driver_licence BOOLEAN,  
    height INTEGER(3),  
    weight FLOAT,  
    account_balance DECIMAL(11,2)  
);
```

SQL NOT NULL on CREATE TABLE

The following SQL ensures that the "ID", "LastName", and "FirstName" columns will NOT accept NULL values when the "Persons" table is created:

Example

```
CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255) NOT NULL,  
    Age int  
);
```

The SQL DROP TABLE Statement

The DROP TABLE statement is used to drop an existing table in a database.

Syntax

```
DROP TABLE table_name;
```

SQL ALTER TABLE Statement

The ALTER TABLE statement is used to add, delete, or modify columns in an existing table

The ALTER TABLE statement is also used to add and drop various constraints on an existing table

ALTER TABLE - ADD Column

To add a column in a table, use the following syntax:

```
ALTER TABLE table_name  
ADD column_name datatype;
```

ALTER TABLE - DROP COLUMN

To delete a column in a table, use the following syntax (notice that some database systems don't allow deleting a column):

```
ALTER TABLE table_name  
DROP COLUMN column_name;
```

การแก้ไขโครงสร้างตาราง

ALTER TABLE - ALTER/MODIFY COLUMN

To change the data type of a column in a table, use the following syntax:

SQL Server / MS Access:

```
ALTER TABLE table_name  
ALTER COLUMN column_name datatype;
```


SQL NOT NULL on ALTER TABLE

To create a NOT NULL constraint on the "Age" column when the "Persons" table is already created, use the following SQL:

```
ALTER TABLE Persons  
MODIFY Age int NOT NULL;
```

คำสั่ง SQL ในการจัดการข้อมูล

Data Manipulation Language (DML)

CRUD operations

- Create
 - INSERT
- Read
 - SELECT
- Update
 - UPDATE
- Delete
 - DELETE

<https://www.w3schools.com/sql/>

<https://mariadb.com/kb/en/sql-statements-structure/>

<https://mariadb.com/kb/en/built-in-functions/>

เพิ่มข้อมูลลงในตาราง

The SQL INSERT INTO Statement

The INSERT INTO statement is used to insert new records in a table.

INSERT INTO Syntax

It is possible to write the INSERT INTO statement in two ways.

The first way specifies both the column names and the values to be inserted:

```
INSERT INTO table_name (column1, column2, column3, ...)  
VALUES (value1, value2, value3, ...);
```

If you are adding values for all the columns of the table, you do not need to specify the column names in the SQL query. However, make sure the order of the values is in the same order as the columns in the table. The INSERT INTO syntax would be as follows:

```
INSERT INTO table_name  
VALUES (value1, value2, value3, ...);
```

เพิ่มข้อมูลลงในตาราง (แบบไม่ระบุชื่อ column)

```
INSERT INTO people  
VALUES ('3749383726178', 'Panita',  
'1974-08-13' , 1, 160, 53.5, 800.25  
);
```

ตารางที่สร้างไว้

```
CREATE TABLE people (  
    id CHAR(13),  
    name VARCHAR(50),  
    dob DATE,  
    driver_licence BOOLEAN,  
    height INTEGER(3),  
    weight FLOAT,  
    account_balance DECIMAL(11,2)  
);
```

เพิ่มข้อมูลลงในตาราง (แบบระบุชื่อ column)

```
INSERT INTO people (name, id, weight)
VALUES ('Mike', '3300123836200', 78.4);
```

ตารางที่สร้างไว้

```
CREATE TABLE people (
    id CHAR(13),
    name VARCHAR(50),
    dob DATE,
    driver_licence BOOLEAN,
    height INTEGER(3),
    weight FLOAT,
    account_balance DECIMAL(11,2)
);
```

การแก้ไขข้อมูลในตาราง

The SQL UPDATE Statement

The UPDATE statement is used to modify the existing records in a table.

UPDATE Syntax

```
UPDATE table_name  
SET column1 = value1, column2 = value2, ...  
WHERE condition;
```

การแก้ไขข้อมูลในตาราง

```
UPDATE people
SET dob='2007-11-30', height=170,
driver_licence=0
WHERE name='Mike'
```

ตารางที่สร้างไว้

```
CREATE TABLE people (
    id CHAR(13),
    name VARCHAR(50),
    dob DATE,
    driver_licence BOOLEAN,
    height INTEGER(3),
    weight FLOAT,
    account_balance DECIMAL(11,2)
);
```


การแก้ไขข้อมูลในตาราง

```
UPDATE people  
SET account_balance=50000, weight=88.86557  
WHERE name='Mike';
```

ตารางที่สร้างไว้

```
CREATE TABLE people (  
    id CHAR(13),  
    name VARCHAR(50),  
    dob DATE,  
    driver_licence BOOLEAN,  
    height INTEGER(3),  
    weight FLOAT,  
    account_balance DECIMAL(11,2)  
);
```

การแก้ไขข้อมูลในตาราง

```
UPDATE people  
SET account_balance = account_balance*1.1;
```

```
UPDATE people  
SET account_balance = account_balance*1.2  
WHERE account_balance < 5000;
```

การแก้ไขข้อมูลในตาราง

```
UPDATE people  
SET account_balance = account_balance+500  
WHERE driver_licence;
```

```
UPDATE people  
SET driver_licence=1  
WHERE dob < '2010-01-01';
```

The SQL DELETE Statement

The DELETE statement is used to delete existing records in a table.

DELETE Syntax

```
DELETE FROM table_name WHERE condition;
```

```
DELETE FROM people  
WHERE name='Mike';
```