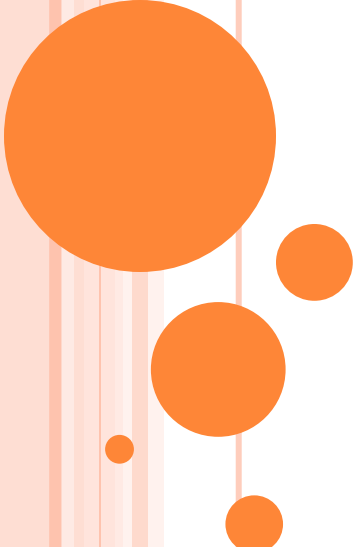


# **LAB 1. Creating and Managing Tables (SQL : Data Definition Language)**

**week#1**



**By Kanokwan Atchariyachanvanich  
Faculty of Information Technology  
KMITL  
Database System Concepts  
2/2565**

# OUTLINE ก่อนสอบกลางภาค

Date	SQL
9, 11 JAN 2023	<ul style="list-style-type: none"><li>• Lab Introduction</li><li>• Introduction to DBLearn (SQL tool)</li></ul>
16, 18 JAN 2023	LAB 1 - Creating and Managing Tables (DDL)
23, 25 JAN 2023	LAB 2 - Including Constraints (DDL)
30 JAN, 1 FEB 2023	LAB 3 - Manipulating Data (DML)
6, 8 FEB 2023	LAB 4 - SQL SELECT Statements <ul style="list-style-type: none"><li>• Writing Basic</li></ul>
13, 15 FEB 2023	LAB 5 - SQL SELECT Statements <ul style="list-style-type: none"><li>• Restricting and Sorting Data</li></ul>
20, 22 FEB 2023	- ทวนก่อนสอบ Quiz 1
27 FEB, 1 MAR 2023	<b>Quiz 1: LAB 1 – LAB 5</b>

# OUTLINE หลังสอบกลางภาค

Date	SQL
8 MAR 2023	<b>MIDTERM examination (9.30-12.30)</b>
20, 22 MAR 2023	LAB 6 - Displaying Data from Multiple Tables (join)
27, 29 MAR 2023	LAB 7 - Displaying Data from Multiple Tables (outer join)
3, 5 APR 2023	LAB 8 - Group function
10, 12 APR 2023	LAB 9 - Subqueries
17, 19 APR 2023	นำเสนอคำสั่ง SQL ที่ใช้สร้างรายงานและตัวอย่างรายงานจากฐานข้อมูลในโปรเจค
24, 26 APR 2023	ทวนก่อนสอบ Quiz 2
1, 3 MAY 2023	<b>Quiz 2 : LAB 6 – LAB 9</b>
10 MAY 2023	<b>FINAL examination (9.30-12.30)</b>

# STRUCTURED QUERY LANGUAGE (SQL) STATEMENT

Type	SQL Statement	
<b>Data Definition Language (DDL)</b> อธิบายส่วนของ SQL ที่อนุญาตให้สร้าง, เปลี่ยน, และทำลายอ็อบเจกต์ฐานข้อมูล อ็อบเจกต์ฐานข้อมูลเหล่านี้รวมถึงแบบแผน, ตาราง, มุมมอง, ลำดับ, แคตาล็อก, ตรรกชนี, และ alias	<b>CREATE</b> <b>ALTER</b> <b>DROP</b> <b>RENAME</b> <b>ANALYZE</b> <b>AUDIT</b> <b>COMMENT</b> <b>ASSOCIATE STATISTICS</b> <b>DISASSOCIATE STATISTICS</b>	<b>FLASHBACK</b> <b>GRANT</b> <b>NOAUDIT</b> <b>PURGE</b> <b>REVOKE</b> <b>TRUNCATE</b> <b>UNDROP</b>
<b>Data Manipulation Language (DML)</b> ภาษาสำหรับการจัดการข้อมูล คือส่วนของประโยค SQL ที่อนุญาตให้คุณควบคุมหรือจัดการข้อมูล	<b>SELECT</b> <b>INSERT</b> <b>UPDATE</b> <b>DELETE</b> <b>MERGE</b> <b>CALL</b> <b>EXPLAIN PLAN</b> <b>LOCK TABLE</b>	
<b>Transaction Control</b> จัดการ transaction จากการเปลี่ยนแปลงที่เกิดจาก DML	<b>COMMIT</b> <b>ROLLBACK</b> <b>SAVEPOINT</b> <b>SET TRANSACTION</b>	

# OBJECTIVES

After completing this lesson, you should be able to do the following:

- Learn **Data Definition Language (DDL)**
- **Create** tables
- **Describe** the data types that can be used when specifying column definition
- **Alter table** definitions
- **Drop, and rename** tables

# DATA DEFINITION LANGUAGE (DDL)

- SQL includes commands to create database objects as showed below

Object	Description
Table	Basic unit of storage; composed of rows and columns
View	Logically represents subsets of data from one or more tables
Sequence	Numeric value generator
Index	Improves the performance of some queries
Synonym	Gives alternative names to objects

Note: More database objects are available but are not covered in this course.

# EXAMPLE: TABLES IN REGISTRATION SYSTEM

**STUDENT**

Student_Name	Student_Number	Class	Major
Wirat	13	1	IT
Nannapas	18	2	IT

**COURSE**

Course_Name	Course_Number	Credit	Department
Database System	6100103	3	IT
Discrete Mathematics	4100103	3	MATH
Economics of Information Technology	6100110	3	IT
Management Information System	6100106	3	IT

**SECTION**

Section_Number	Course_Number	Semester	Year	Instructor
10	6100103	1	07	Srinual
11	6100103	1	07	Srinual
12	6100110	2	07	Warunee
20	4100103	2	07	Prapan

**GRADE REPORT**

Student_Number	Section_Number	Grade
13	10	A
13	12	B
18	11	A
18	20	C

**PREREQUISITE**

Course_Number	Prerequisite_Number
6100103	4100103
6100110	6100106

รูปที่ 1.1 แสดงฐานข้อมูลระบบลงทะเบียน

# EXAMPLE: TABLES IN REGISTRATION SYSTEM

## Meta-data

### RELATIONS

Relation_Name	No_of_Columns
STUDENT	4
COURSE	4
SECTION	5
GRADE_REPORT	3
PREREQUISITE	2

### COLUMNS

Column_Name	Data_Type	Belongs_to_relation
Name	Character(30)	STUDENT
Student_Number	Character(4)	STUDENT
Class	Integer(1)	STUDENT
Major	Character(30)	STUDENT
Course_Name	Character(30)	COURSE
....	....	....
....	....	....
Prerequisite_Number	Character(8)	PREREQUISITE

## User data

### STUDENT

Student_Name	Student_Number	Class	Major
Wirat	13	1	IT
Nannapas	18	2	IT



# THE CREATE TABLE STATEMENT SYNTAX

- Creates a new table in the user's database schema

```
CREATE TABLE TABLE_NAME (column_name datatype,  
                             [column_name datatype, ... ] );
```

- You must have:
  - CREATE TABLE privilege
  - A storage area
- You specify:
  - The table name can be specified as *db\_name.tbl\_name*
  - Table name
  - Column name, column data type, and column size

# CREATING TABLES

- Example: Create the table named “dept”

```
CREATE TABLE dept
(
    deptno    INT (3),
    dname     VARCHAR(14),
    loc       VARCHAR(13)
);
```

- Confirm table creation.

```
DESCRIBE dept ;
```

# NAMING RULES FOR TABLES

Table names and column names:

- ควรตั้งชื่อให้มีความยาว ตั้งแต่ 1- 64 ตัวอักษร
- ควรจะกำหนดแบบ alphanumeric ประกอบด้วย A-Z, a-z, 0-9, \_, \$, สามารถขึ้นต้นด้วยตัวเลข แต่ไม่ใช่ตัวเลขทั้งหมด
- ไม่ควรใช้ spaces ในการตั้งชื่อ
- ต้องไม่ตั้งชื่อตารางซ้ำกันภายใต้เจ้าของตารางเดียวกัน
- ชื่อตารางไม่ควรมี '.' และ '/' หรือ '\'
- If an identifier contains special characters or is a **reserved word**, you *must* quote- the **backtick** (``)
- Table Alias' name can contain 1- 256 characters
- Column name is not case sensitive on any platform
- Table name is not case sensitive. (server runs on Windows)

Note: For example, EMPLOYEES is treated as the same name as eMPloyees or eMpLOYEES.

# DATA TYPES (MySQL)

Text types:

Data type	Description
CHAR(size)	Holds a fixed length string (can contain letters, numbers, and special characters). The fixed size is specified in parenthesis. Can store up to 255 characters
VARCHAR(size)	Holds a variable length string (can contain letters, numbers, and special characters). The maximum size is specified in parenthesis. Can store up to 255 characters. <b>Note:</b> If you put a greater value than 255 it will be converted to a TEXT type
TINYTEXT	Holds a string with a maximum length of 255 characters
TEXT	Holds a string with a maximum length of 65,535 characters
BLOB	For BLOBs (Binary Large Objects). Holds up to 65,535 bytes of data
MEDIUMTEXT	Holds a string with a maximum length of 16,777,215 characters
MEDIUMBLOB	For BLOBs (Binary Large Objects). Holds up to 16,777,215 bytes of data
LONGTEXT	Holds a string with a maximum length of 4,294,967,295 characters
LOBLOB	For BLOBs (Binary Large Objects). Holds up to 4,294,967,295 bytes of data
ENUM(x,y,z,etc.)	<p>Let you enter a list of possible values. You can list up to 65535 values in an ENUM list. If a value is inserted that is not in the list, a blank value will be inserted.</p> <p><b>Note:</b> The values are sorted in the order you enter them.</p> <p>You enter the possible values in this format: ENUM('X','Y','Z')</p>
SET	Similar to ENUM except that SET may contain up to 64 list items and can store more than one choice

# DATA TYPES (MySQL)

Number types:

Data type	Description
TINYINT(size)	-128 to 127 normal. 0 to 255 UNSIGNED*. The maximum number of digits may be specified in parenthesis
SMALLINT(size)	-32768 to 32767 normal. 0 to 65535 UNSIGNED*. The maximum number of digits may be specified in parenthesis
MEDIUMINT(size)	-8388608 to 8388607 normal. 0 to 16777215 UNSIGNED*. The maximum number of digits may be specified in parenthesis
INT(size)	-2147483648 to 2147483647 normal. 0 to 4294967295 UNSIGNED*. The maximum number of digits may be specified in parenthesis
BIGINT(size)	-9223372036854775808 to 9223372036854775807 normal. 0 to 18446744073709551615 UNSIGNED*. The maximum number of digits may be specified in parenthesis
FLOAT(size,d)	A small number with a floating decimal point. The maximum number of digits may be specified in the size parameter. The maximum number of digits to the right of the decimal point is specified in the d parameter
DOUBLE(size,d)	A large number with a floating decimal point. The maximum number of digits may be specified in the size parameter. The maximum number of digits to the right of the decimal point is specified in the d parameter
DECIMAL(size,d)	A DOUBLE stored as a string , allowing for a fixed decimal point. The maximum number of digits may be specified in the size parameter. The maximum number of digits to the right of the decimal point is specified in the d parameter

# DATA TYPES (MySQL)

Date types:

Data type	Description
DATE()	<p>A date. Format: YYYY-MM-DD</p> <p><b>Note:</b> The supported range is from '1000-01-01' to '9999-12-31'</p>
DATETIME()	<p>*A date and time combination. Format: YYYY-MM-DD HH:MI:SS</p> <p><b>Note:</b> The supported range is from '1000-01-01 00:00:00' to '9999-12-31 23:59:59'</p>
TIMESTAMP()	<p>*A timestamp. TIMESTAMP values are stored as the number of seconds since the Unix epoch ('1970-01-01 00:00:00' UTC). Format: YYYY-MM-DD HH:MI:SS</p> <p><b>Note:</b> The supported range is from '1970-01-01 00:00:01' UTC to '2038-01-09 03:14:07' UTC</p>
TIME()	<p>A time. Format: HH:MI:SS</p> <p><b>Note:</b> The supported range is from '-838:59:59' to '838:59:59'</p>
YEAR()	<p>A year in two-digit or four-digit format.</p> <p><b>Note:</b> Values allowed in four-digit format: 1901 to 2155. Values allowed in two-digit format: 70 to 69, representing years from 1970 to 2069</p>

## EXERCISE 1: จงสร้างตารางชื่อ EMP ซึ่งมีโครงสร้างดังต่อไปนี้

Column name	ID	LAST_NAME	FIRST_NAME	DEPT_ID
Data type	INT	VARCHAR	VARCHAR	INT
Length	7	25	25	7

```
CREATE TABLE EMP
(
    id            INT (7),
    last_name     VARCHAR(25),
    first_name    VARCHAR(25),
    dept_id       INT(7)
);
```

# THE **ALTER TABLE** STATEMENT

Use the ALTER TABLE statement to:

- Add a new column
- Modify an existing column
- Define a default value for the new column
- Drop a column



# THE **ALTER TABLE** STATEMENT SYNTAX

- Use the **ALTER TABLE** statement to add, modify, or drop columns.

```
ALTER TABLE table_name  
ADD column_name datatype [DEFAULT expr]  
[, column_name datatype]...;
```

```
ALTER TABLE table_name  
MODIFY column_name datatype [DEFAULT expr]  
[, column_name datatype]...;
```

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

# ADDING A COLUMN

## DEPT80

## New Column

EMPLOYEE_ID	LAST_NAME	ANNSAL	HIRE_DATE	JOB_ID
145	Russell	168000	01-OCT-96	
146	Partners	162000	05-JAN-97	
147	Errazuriz	144000	10-MAR-97	
148	Cambrault	132000	15-OCT-99	
149	Zlotkey	126000	29-JAN-00	
150	Tucker	120000	30-JAN-97	

**“Add a new column to the DEPT80 table.”**

## DEPT80

EMPLOYEE_ID	LAST_NAME	ANNSAL	HIRE_DATE	JOB_ID
145	Russell	168000	01-OCT-96	
146	Partners	162000	05-JAN-97	
147	Errazuriz	144000	10-MAR-97	
148	Cambrault	132000	15-OCT-99	
149	Zlotkey	126000	29-JAN-00	
150	Tucker	120000	30-JAN-97	

# ADDING A COLUMN

- You use the ADD clause to add columns.

```
ALTER TABLE dept80  
  
ADD job_id VARCHAR (9) ;
```

- The new column becomes the last column.

EMPLOYEE_ID	LAST_NAME	ANNSAL	HIRE_DATE	JOB_ID
145	Russell	168000	01-OCT-96	
146	Partners	162000	05-JAN-97	
147	Errazuriz	144000	10-MAR-97	
148	Cambrault	132000	15-OCT-99	
149	Zlotkey	126000	29-JAN-00	
150	Tucker	120000	30-JAN-97	

EXERCISE 2: เพิ่มคอลัมน์ emp\_id ในตาราง job\_it และแสดงคำสั่งเพื่อดูโครงสร้างหลังจากการลบคอลัมน์

Column	Data type & size
emp_id	int(11)

```
ALTER TABLE    job_it
```

```
ADD              emp_id    int(11) ;
```

คำสั่งเพื่อดูโครงสร้างหลังจากการแก้ไข

```
DESCRIBE job_it;
```

# MODIFYING A COLUMN

- You can change a **column's size**, and default values (old varchar(25) => new VARCHAR (30) )

```
ALTER TABLE dept80  
  
MODIFY          last_name VARCHAR (30) ;
```

Modify  
column's size

- A change to the default value affects only subsequent insertions to the table.

# MODIFYING A COLUMN

- You can change a column's data type & size
- Old varchar(25) => New **CHAR (30)**

```
ALTER    TABLE    dept80
```

```
MODIFY   last_name  CHAR (30) ;
```

Modify data  
type

EXERCISE 3: แก้ไขคอลัมน์ job\_title ในตาราง job\_it ให้มีขนาดเป็น 50 และแสดงคำสั่งเพื่อดูโครงสร้างหลังจากการแก้ไข

```
ALTER TABLE job_it
```

```
MODIFY job_title VARCHAR(50) ;
```

คำสั่งเพื่อดูโครงสร้างหลังจากการแก้ไข

```
DESCRIBE job_it;
```

# ALTER TABLE แก้ไขชื่อคอลัมน์ ชนิดข้อมูลและขนาดข้อมูล

```
ALTER TABLE ชื่อตาราง  
CHANGE ชื่อคอลัมน์เก่า ชื่อคอลัมน์ใหม่ ชนิดข้อมูล(ขนาดข้อมูล);
```

```
ALTER TABLE sales_reps  
CHANGE reps_id id varchar(30);
```



# ALTER TABLE แก้ไขชื่อคอลัมน์

หมายเหตุ : ใช้กับ MySQL v.8 ขึ้นไป ดังนั้นใช้กับ DBLearn ไม่ได้ เนื่องจาก  
ระบบ DBLearn ใช้ MySQL v.5.7.4

```
ALTER TABLE      ชื่อตาราง  
RENAME COLUMN ชื่อคอลัมน์เก่า TO ชื่อคอลัมน์ใหม่ ;
```

```
ALTER TABLE sales_reps  
RENAME COLUMN name TO firstname ;
```

# DROPPING A COLUMN

- Use the DROP COLUMN clause to drop columns you no longer need from the table.

```
ALTER TABLE dept80  
DROP job_id ;
```

- **Guidelines**

- คอลัมน์นั้นอาจจะมีหรือไม่มีข้อมูลก็ได้
- ตารางต้องมีอย่างน้อย 1 คอลัมน์เหลืออยู่หลังจากใช้ ALTER TABLE statement
- เมื่อคอลัมน์ถูกลบแล้ว จะไม่สามารถกู้คืนคอลัมน์นั้นได้อีก

EXERCISE 4: ลบคอลัมน์ MAX\_SALARY จากตาราง job\_it และแสดงคำสั่งเพื่อดูโครงสร้างหลังจากการลบคอลัมน์

```
ALTER TABLE    job_it  
  
DROP            max_salary ;
```

คำสั่งเพื่อดูโครงสร้างหลังจากการแก้ไข

```
DESCRIBE job_it;
```

# ADD, MODIFY, DROP มากกว่า 1 คอลัมน์

## ○ ตัวอย่าง

```
ALTER TABLE    sales_reps
ADD              address varchar(30),
ADD              tel varchar(15);
```

```
ALTER TABLE    sales_reps
MODIFY          name varchar(30),
MODIFY          salary int(6);
```

```
ALTER TABLE    sales_reps
DROP            salary,
DROP            commission_pct;
```

# CHANGING THE NAME OF AN OBJECT

- To change the name of a table, view, sequence, or synonym, you execute the RENAME statement.
- You must be the owner of the object.

```
RENAME TABLE old_table_name TO new_table_name ;  
  
ALTER TABLE old_table_name RENAME new_table_name;
```

```
RENAME TABLE dept80 TO detail_dept ;  
  
ALTER TABLE dept80 RENAME detail_dept;
```

หมายเหตุ RENAME TABLE สามารถแก้ไขชื่อตารางตั้งแต่ 2 ตารางขึ้นไปพร้อมกันได้

# DROPPING A TABLE

- All data and structure in the table is deleted.
- Any pending transactions are committed.
- All indexes are dropped.
- You cannot roll back the DROP TABLE statement.

```
DROP TABLE    table_name ;
```

```
DROP TABLE    detail_dept ;
```

# SUMMARY

- In this lesson, you should have learned how to use DDL statements to create, alter, drop, and rename tables.

Statement	Description
CREATE TABLE	Creates a table
ALTER TABLE	Modifies table structures
DROP TABLE	Removes the rows and table structure
RENAME TABLE	Changes the name of a table