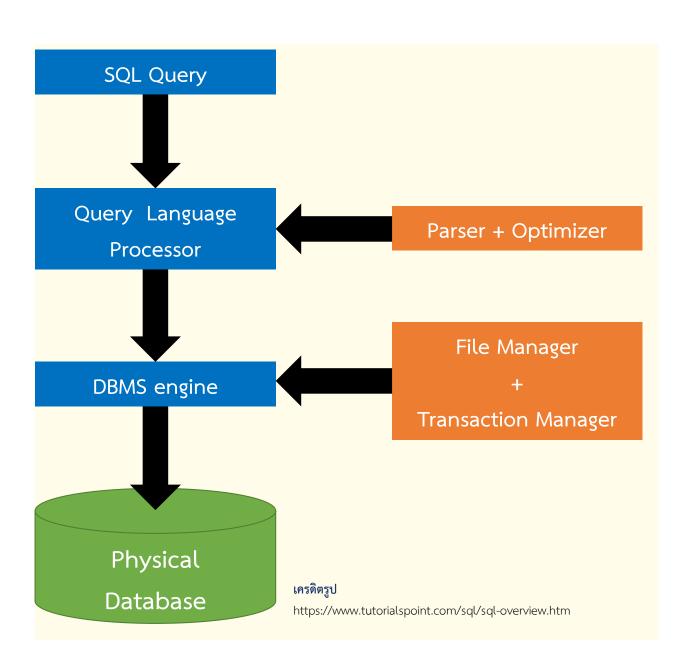
# สรุปไวยากรณ์ ภาษา SQL เบื้องต้น





# สรุปไวยากรณ์ภาษา SQL เบื้องต้น

#### ประโยค SELECT

SELECT \* FROM table name;

SELECT column1, column2, ....

FROM table name;

ข้อสังเกต database บางระบบต้องการเซมิโคลอน (;) ต่อ ท้ายเพื่อจบประโยคคำสั่ง SOL

#### Comments

คอมเมนต์เพียงบรรทัดเดียว จะเริ่มต้นด้วย --

-- SELECT \* FROM Students:

คอมเมนต์หลายบรรทัด จะเริ่มต้นด้วย /\* และสิ้น สุดด้วย \*/

/\*SELECT \* FROM Students;

SELECT \* FROM Class;

SELECT \* FROM Teacher\*/

#### **DISTINCT Clause**

SELECT DISTINCT column1, column2, ....

FROM table name;

#### **Aliases Clause**

SELECT column\_name AS alias\_name

FROM table name;

SELECT column1, column2, ....

FROM table name AS alias name;

#### WHERE Clause

SELECT column1, column2, ....

FROM table name

WHERE condition;

ข้อสังเกต ประโยค WHERE ไม่ได้ใช้แค่ในประโยค SELECT เท่านั้น ยังสามารถใช้ในประโยค UPDATE, DELETE และอื่นๆ อีกด้วย

#### โอเปอเรเตอร์ในประโยค WHERE

โอเปอเรเตอร์	คำอธิบาย
=	เท่ากับ
<>	ไม่เท่ากับ (SQL บางเวอร์ชั่นอาจ ใช้ !=)
>	มากกว่า
<	น้อยกว่า
>=	มากกว่า หรือ เท่ากับ
<=	น้อยกว่า หรือ เท่ากับ
LIKE	ค้นหาด้วยแพทเทิร์น
BETWEEN	ข้อมูลอยู่ในช่วงระหว่าง
IN	ระบุค่าที่เป็นไปได้

#### โอเปอเรอเตอร์ LIKE

SELECT column1, column2, ....

FROM table name

WHERE column name LIKE { PATTERN };

สัญลักษณ์แทนตัวอักษรใน SQL (wildcard characters)

% ใช้แทนตัวอักษรตั้งแต่ 0 ตัว 1 ตัว หรือมากกว่านั้น ใช้แทนตัวอักษรแค่ตัวเดียว

( MS Access จะใช้เครื่องหมาย ? แทน )

#### โอเปอเรอเตอร์ BETWEEN

SELECT column1, column2, ....

FROM table name

WHERE column name

**BETWEEN** value1 AND value2:

#### โอเปอเรอเตอร์ IN

SELECT column1, column2, ....

FROM table name

WHERE column name IN (value1, value2, ...);

SELECT column1, column2, ....

FROM table name

WHERE column name IN

( **SELECT** STATEMENT );

#### โอเปอเรเตอร์ AND/OR

SELECT column1, column2, ....

FROM table name

WHERE condition1 { AND | OR } condition2;

#### **NOT Clause**

SELECT column1, column2, ....

FROM table\_name

WHERE NOT condition;

#### IS NULL Clause

SELECT column1, column2, ....

FROM table name

WHERE column name IS NULL;

#### IS NOT NULL Clause

SELECT column1, column2, ....

FROM table name

WHERE column name IS NOT NULL;

#### **ORDER BY Clause**

SELECT column1, column2, ....

FROM table name

WHERE condition

ORDER BY column1, column2, .... {ASC|DESC};

#### **GROUP BY Clause**

**SELECT** column name(s)

FROM table name

WHERE condition

GROUP BY column name(s)

ORDER BY column name(s);

### ฟังก์ชั่น MIN() and MAX()

SELECT MIN( column name )

FROM table name

WHERE condition:

SELECT MAX( column name )

FROM table name

WHERE condition;

#### ฟังก์ชั่น COUNT(), AVG() และ SUM()

SELECT COUNT( column\_name )

FROM table name

WHERE condition;

**SELECT AVG**( column\_name )

FROM table name

WHERE condition;

**SELECT SUM**( column name )

FROM table name

WHERE condition:

#### โอเปอเรเตอร์ EXISTS, ANY and ALL

**SELECT** column name(s)

FROM table name

WHERE EXISTS

(SELECT column name

FROM table name WHERE condition);

**SELECT** column name(s)

FROM table name

WHERE column name operator ANY

(SELECT column name

FROM table name WHERE condition);

**SELECT** column name(s)

FROM table name

WHERE column name operator ALL

(SELECT column name

FROM table name WHERE condition);

\*\*\* column name(s) หมายถึงชื่อคอลัมน์ 1 อัน หรือมากกว่านั้น

<sup>\*\*\*</sup> operators ได้แก่ =, <>, !=, >, >=, < หรือ <=



#### **HAVING Clause**

**SELECT** column name(s)

FROM table name

WHERE condition

**GROUP BY** column name(s)

HAVING (arithematic function condition);

ORDER BY column name(s)

\*\*\* arithematic functions เป็นฟังก์ชั่นได้แก่ COUNT(), AVG() และ SUM()

#### ประโยค INSERT INTO

#### **INSERT INTO**

table name(column1, column2, ....)

VALUES (value1, value2, ....);

**INSERT INTO** table name

VALUES (value1, value2, ....);

#### ประโยค INSERT INTO SELECT

#### **INSERT INTO**

table name2(column1, column2, ....)

SELECT column1, column2, ....

FROM table name1

WHERE condition;

**INSERT INTO** table name2

SELECT \* FROM table name1

WHERE condition:

#### ประโยค SELECT INTO

SELECT column1, column2, ....

INTO new table [IN externaldb]

FROM old table

WHERE condition:

**SELECT \*** 

INTO new table [ IN externaldb ]

FROM old table

WHERE condition;

#### ประโยค UPDATE

**UPDATE** table name

SET column1 = value1, column2 = value2, ....

WHERE condition;

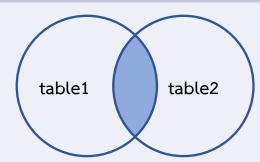
#### ประโยค DELETE

**DELETE FROM** table name

WHERE condition;

ข้อสังเกต ถ้าละประโยค WHERE เรคอร์ดทั้งหมดในตารางจะถูก ลบออกไปหมด ซึ่งสามารถเขียนได้ด้วยประโยค DELETE \* FROM table\_name; หรือเขียนสั้นๆ DELETE FROM table\_name;

#### การทำ INNER JOIN



SELECT column1, column2, ....

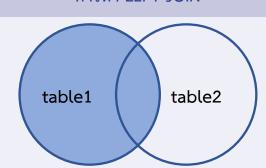
FROM table1

**INNER JOIN** table2

ON

table1.column name = table2.column name;

#### การทำ LEFT JOIN



SELECT column1, column2, ....

FROM table1

**LEFT JOIN** table2

ON

table1.column name = table2.column name;

# การทำ RIGHT JOIN table1 table2

SELECT column1, column2, ....

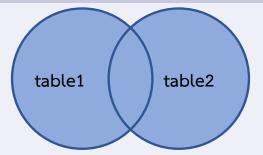
FROM table1

**RIGHT JOIN** table2

ON

table1.column name = table2.column name;

# การทำ FULL OUTER JOIN



SELECT column1, column2, ....

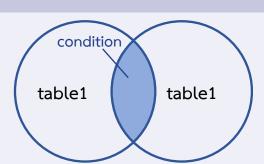
FROM table1

**FULL OUTER JOIN** table2

ON

table1.column\_name = table2.column\_name;

#### การทำ Self JOIN



SELECT column1, column2, ....

FROM table1 T1, table1 T2

WHERE condition;

#### การทำ UNION

SELECT column1, column2, ....

FROM table1

WHERE condition

UNION

SELECT column1, column2, ....

FROM table2:

WHERE condition

#### การทำ UNION ALL

SELECT column1, column2, ....

FROM table1

WHERE condition

**UNION ALL** 

SELECT column1, column2, ....

WHERE condition

FROM table2;

#### ประโยค CREATE DATABASE

**CREATE DATABASE** database name;

#### ประโยค DROP DATABASE

**DROP DATABASE** database name;

#### ประโยค CREATE TABLE

**CREATE TABLE** table\_name(

column1 datatype,

column2 datatype,

....

columnN datatype,

PRIMARY KEY( one or more columns )

);

CREATE TABLE new table name AS

SELECT column1, column2, ....

FROM existing table name

WHERE ....;



#### ประโยค DROP TABLE

DROP TABLE table\_name;

#### ประโยค TRUNCATE TABLE

TRUNCATE TABLE table name;

#### ประโยค CREATE INDEX

CREATE INDEX index\_name
ON table name ( column1, column2, ...);

#### ประโยค CREATE UNIQUE INDEX

CREATE UNIQUE INDEX index\_name
ON table\_name ( column1, column2, ...);

# ประโยค DROP INDEX

ALTER TABLE table\_name DROP INDEX index\_name;

# ประโยค DESC

**DESC** table name;

# ประโยค ALTER TABLE

ALTER TABLE table\_name
ADD column\_name { data\_type };

ALTER TABLE table\_name { DROP | MODIFY } COLUMN column\_name { data\_type };

# ประโยค ALTER TABLE (Rename) [1]

ALTER TABLE table\_name

RENAME TO new\_table\_name;

#### หมายเหตุ

[1] เป็นคำสั่ง SQL ที่ใช้กับ MySQL (ถ้าเป็น database ตัว อื่นอาจใช้รูปแบบเหมือนกัน หรือต่างกันก็ได้)

#### อ้างอิง

https://www.w3schools.com/sql/sql\_select.asp https://www.tutorialspoint.com/sql/sql-syntax.htm

Written by Jaturapat Patanasongsivilai