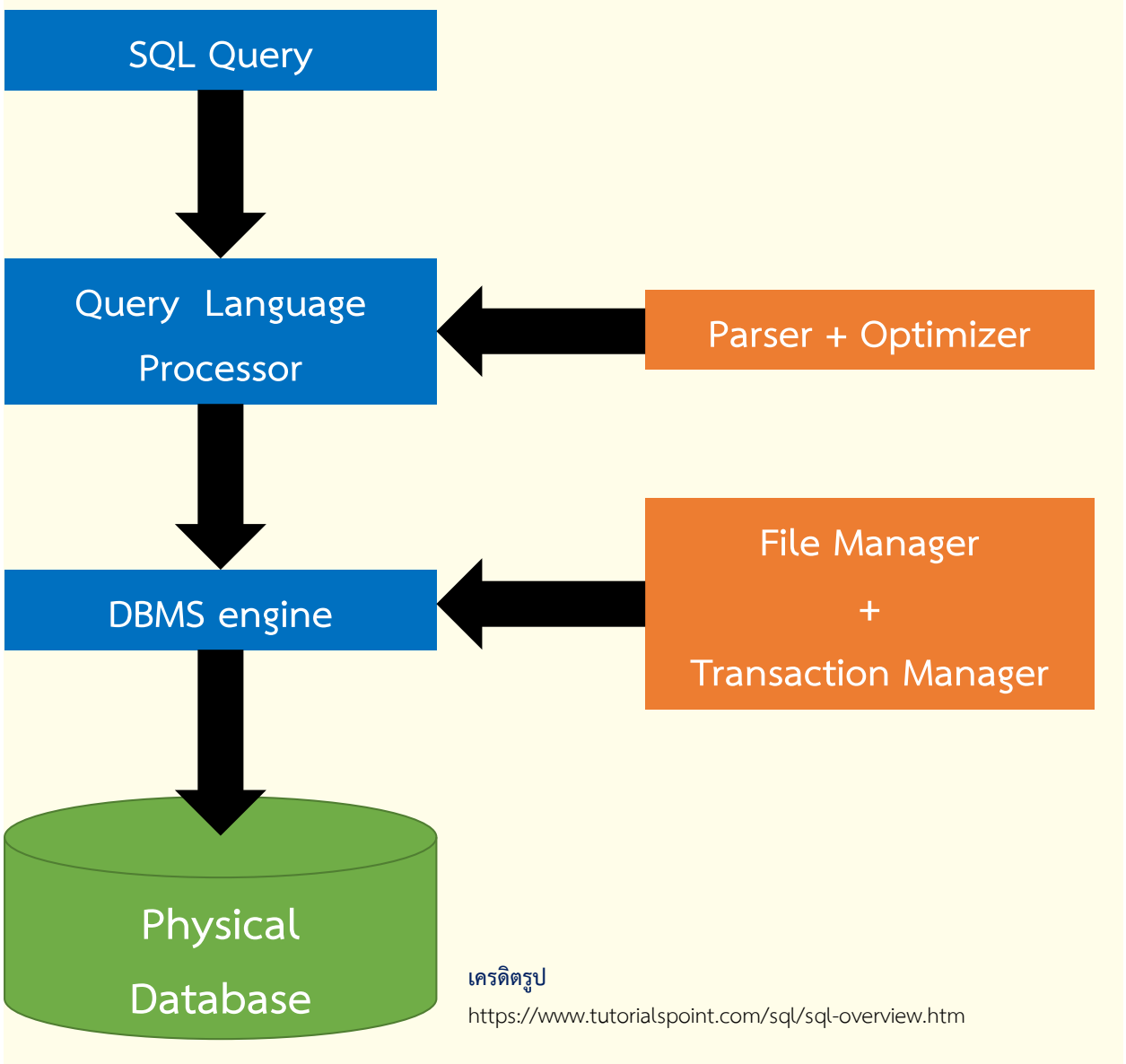
# สรุปไวยากรณ์ ภาษา SQL เบื้องต้น

(แก้ไขใหม่ 18/02/2563)




เครดิตรูป

https://www.tutorialspoint.com/sql/sql-overview.htm

# สรุปไวยากรณ์ภาษา SQL เบื้องต้น

## ประโยค SELECT

```
SELECT * FROM table_name;

SELECT column1, column2, ....
FROM table_name;
```

**ข้อสังเกต** database บางระบบต้องการเซมิโคลอน (;) ต่อท้ายเพื่อจบประโยคคำสั่ง SQL

## Comments

**คอมเมนต์เพียงบรรทัดเดียว จะเริ่มต้นด้วย --**

```
-- SELECT * FROM Students;
```

**คอมเมนต์หลายบรรทัด จะเริ่มต้นด้วย /* และสิ้นสุดด้วย */**

```
/*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 บางเวอร์ชั่นอาจใช้ !=) |
| > | 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

*** 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

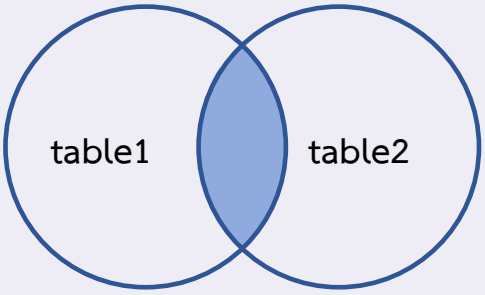
```
UPDATE table_name
SET column1 = value1, column2 = value2, ....
WHERE  condition;
```

## ประโยค DELETE

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

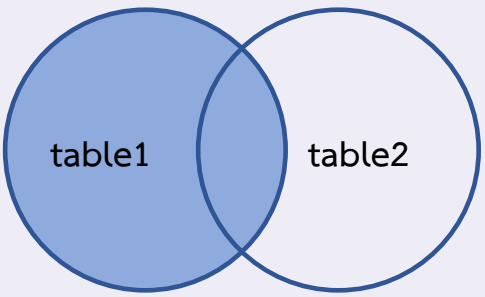
**ข้อสังเกต** ถ้าละประโยค WHERE เรคอร์ดทั้งหมดในตารางจะถูกลบออกไปหมด ซึ่งสามารถเขียนได้ด้วยประโยค DELETE * FROM table_name; หรือเขียนสั้นๆ DELETE FROM table_name;

## การทำ INNER JOIN



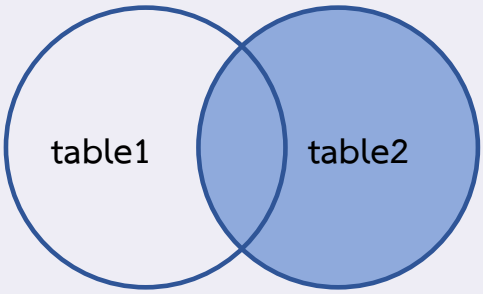

```
SELECT column1, column2, ....
FROM table1
INNER JOIN  table2
ON
table1.column_name = table2.column_name;
```

## การทำ LEFT JOIN



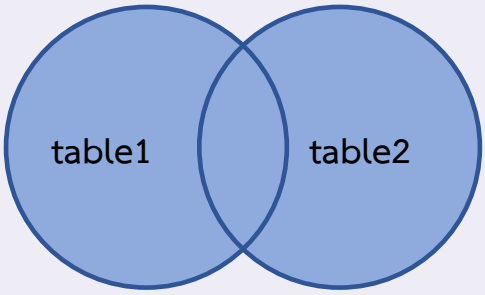

```
SELECT column1, column2, ....
FROM table1
LEFT JOIN  table2
ON
table1.column_name = table2.column_name;
```

## การทำ RIGHT JOIN



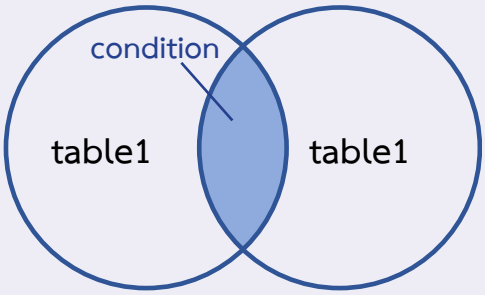

```
SELECT column1, column2, ....
FROM table1
RIGHT JOIN  table2
ON
table1.column_name = table2.column_name;
```

## 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, ....
FROM table2;
WHERE condition
```

## 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 [1]

```
ALTER TABLE table_name
DROP INDEX index_name;
```

## ประโยค DESC [1]

```
DESC table_name;
```

## ประโยค ALTER TABLE [1]

```
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 ตัวอื่นอาจใช้รูปแบบเหมือนกัน หรือต่างกันก็ได้)

**อ้างอิง**

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



Written by Jaturapat Patanasongsivilai