



Sprint 02 - Intro to Database and CRUD

☰ Tools	SQL
📅 Date	@November 1, 2023
⚙️ Status	Done

SQL Online Compiler - Next gen SQL Editor

SQL OnLine - Next gen SQL Editor: SQLite, MariaDB / MySQL, PostgreSQL, MS SQL Server. User-friendly interface for data science. No registration for start, No DownLoad, No Install. Online test SQL script. Online Open/Save SQLite file. Online view all table DB. Fiddle link SQL text and DB file. SQL

<https://sqliteonline.com/>

SQL Code for Intro to Database and CRUD

```
-- create table employee
CREATE TABLE employee (
    id INT UNIQUE,
    name TEXT,
    department TEXT,
    position TEXT,
    salary REAL
);

-- insert data into employee
INSERT INTO employee VALUES
    (1, 'David', 'Marketing', 'CEO', 100000),
    (2, 'John' , 'Marketing', 'VP', 85000 ),
    (3, 'Marry', 'Sales', 'Manager', 60000);

INSERT INTO employee VALUES
```

```

        (4, 'Harry', 'IT', 'Senior Manager', 88000),
        (5, 'Walker', 'IT', 'Manager', 68000);

-- select all columns from employee
SELECT * FROM employee;

-- create new column : new_salary + company_email
SELECT
    name,
    salary,
    salary * 1.15 AS new_salary,
    LOWER(name) || '@company.com' AS company_email
From employee ;

-- filter employee by department and salary
SELECT * from employee
    WHERE department = 'Marketing' AND salary > 90000;

SELECT * FROM employee
WHERE department IN ('Marketing', 'IT');
/* WHERE department = 'Marketing'
    OR department = 'IT'; */

SELECT * FROM employee
WHERE salary <= 90000;

-- update new salary of 1st employee
UPDATE employee
SET salary = 99000
WHERE id = 1;

-- delete just 1 employee and 2 employees at once
DELETE FROM employee
WHERE name = 'Walker';

DELETE FROM employee
WHERE id in (2, 4);

```

```
-- alter table by create new column and input name email
ALTER TABLE employee RENAME TO MyEmployee;

ALTER TABLE MyEmployee
ADD email TEXT;

UPDATE MyEmployee
SET email = 'admin@company.com';

UPDATE MyEmployee
SET email = 'ceo@company.com'
WHERE id = 1;

-- copy (backup) and drop table
CREATE TABLE MyEmployee_Backup AS
    SELECT * FROM MyEmployee; --old Data

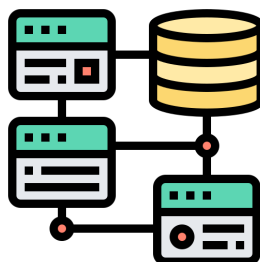
DROP TABLE MyEmployee_Backup;
DROP TABLE MyEmployee;
```

▼ EP01 - SQLite Online Editor 1

SQL

- SQL ⇒ Structured Query Language
- ทำงานกับ Database
- Since 1970

Relational Database




- Relational database = ที่เก็บ data ที่เป็น table หรือ structured data ที่เยอะมาก ๆ ๆ ๆ

Who Should Learn SQL?

- To be Data Driven company
 - All people have **basic and understand** of **SQL and Excel**
 - **SQL (database) and Excel**

SQLite Online Editor

SQL Online Compiler - Next gen SQL Editor

SQL OnLine - Next gen SQL Editor: SQLite, MariaDB / MySQL, PostgreSQL, MS SQL Server.
User-friendly interface for data science. No registration for start, No Download, No Install.
Online test SQL script. Online Open/Save SQLite file. Online view all table DB. Fiddle link SQL
 <https://sqliteonline.com/>

▼ EP02 - Create Table 2

Delete Table demo

- ใส่คำสั่ง **DROP TABLE** = เพื่อลบ database demo ออก
- กด run

```
DROP TABLE demo;
```

Create New Table

- กดปุ่ม + มุมบนขวา
- ใส่คำสั่ง **CREATE TABLE**

```
CREATE TABLE employee (
  id INT UNIQUE, //UNIQUE = id ซ้ำกันไม่ได้
  name TEXT,
  department TEXT,
```

```
position TEXT,  
salary REAL  
);
```



Data Types

1. INT (integer) = จำนวนเต็ม
2. TEXT = ข้อความ
3. REAL (Real number) = ตัวเลขทศนิยม

▼ EP03 - Insert Data 3

Insert New Data

- ใช้คำสั่ง `INSERT INTO` **ชื่อตาราง** `VALUES`
- **PRIMARY KEY**

(**id**, 'name', 'department', 'position', **salary**)

```
INSERT INTO employee VALUES  
    (1, 'David', 'Marketing', 'CEO', 100000),  
    (2, 'John' , 'Marketing', 'VP' , 85000),  
    (3, 'Marry', 'Sales'      , 'Manager', 60000);
```

▼ EP04 - Select Data 4

Select Data

```
SELECT * From employee;
```

```
SELECT id, name, salary From employee LIMIT 2;
```

- ใช้คำสั่ง `SELECT` เพื่อดึง Column ที่ต้องการ จาก Table : employee
- LIMIT = จำกัดการดึง Column
- * = All columns



เราสามารถ เลือกตั้ง Column ได้ โดย พิมพ์ชื่อ Column ที่ต้องการ

▼ EP05 - Transform Columns 5

Create New Column

```
SELECT
    name,
    salary,
    salary * 1.15 AS new_salary,
    LOWER(name) || '@company.com' AS company_email
From employee ;
```



AS (Aliases) ⇒ ตั้งชื่อ Column ใหม่

LOWER() ⇒ ตัวพิมพ์เล็ก , **UPPER()** ⇒ ตัวพิมพ์ใหญ่

|| ⇒ เชื่อม Text

- การสร้าง Column เงินเดือนใหม่พนักงาน

```
salary * 1.15 AS new_salary,
```

- การสร้าง Email ของบริษัทให้พนักงาน

```
LOWER(name) || '@company.com' AS company_email
```

▼ EP06 - Filter Data 6

Filter Data

- ใช้คำสั่ง **WHERE** เพื่อ Filter



`WHERE name LIKE 'J%'`

⇒ ชื่อที่ขึ้นต้นด้วย J ตามหลังด้วยอักขระอะไรก็ได้

- พนักงาน แผนก Marketing + เงินเดือน > 90000

```
SELECT * from employee
WHERE department = 'Marketing' AND salary > 90000
```

- พนักงาน แผนก Marketing + IT

```
SELECT * FROM employee
WHERE department IN ('Marketing', 'IT');
/* WHERE department = 'Marketing'
   OR department = 'IT'; */
```

- พนักงาน เงินเดือน ≤ 90000

```
SELECT * FROM employee
WHERE salary <= 90000;
```

▼ EP07 - Update Data 7

Update Data

- ใช้คำสั่ง `UPDATE` เพื่อแก้ไข Data ใน Database

- การแก้เงินเดือนพนักงานคนที่ 1 จาก `100,000` → `99,000`
- ใช้คำสั่ง `SET` เพื่อ salary พนักงาน
- ใช้คำสั่ง `WHERE` กำหนด Column `id = 1`

```
UPDATE employee
SET salary = 99000
WHERE id = 1;

SELECT * FROM employee;
```

id	name	department	position	salary
1	David	Marketing	CEO	99 000
2	John	Marketing	VP	85 000
3	Marry	Sales	Manager	60 000
4	Harry	IT	Senior Manager	88 000
5	Walker	IT	Manager	68 000

▼ EP08 - Delete Data 8

Delete Data

- Delete ชื่อพนักงาน 1 คน

```
DELETE FROM employee
WHERE name = 'Walker';
```

- Delete ชื่อพนักงาน 2 คนพร้อมกัน

```
DELETE FROM employee
WHERE id in (2, 4);
```



ถ้าไม่ใช่ **WHERE** clause ⇒ ลบชื่อพนักงานทั้งหมด

▼ EP09 - Alter Data 9

Alter Table

- Alter Data = การเปลี่ยนแปลง Table เช่น ชื่อ table

1. ตั้งชื่อ table ใหม่

```
ALTER TABLE employee RENAME TO MyEmployee;
```

2. การเพิ่ม Column ใหม่ : email


```
ALTER TABLE MyEmployee
ADD email TEXT;
```

3. การ Update ชื่อ email + พนักงานมี email เหมือนกัน

```
UPDATE MyEmployee
SET email = 'admin@company.com';
```

4. Update email เฉพาะ David

```
UPDATE MyEmployee
SET email = 'ceo@company.com'
WHERE id = 1;

SELECT * FROM MyEmployee;
```

id	name	department	position	salary	email
1	David	Marketing	CEO	99 000	ceo@company.c...
3	Marry	Sales	Manager	60 000	admin@compan...

▼ EP10 - Copy and Drop Table 10

Copy Table = Backup Table

- ใช้คำสั่ง **CREATE TABLE** + คำสั่ง **AS** ตามด้วย Query ที่ดึงจาก Data เก่า ไปใส่ใน Table_Backup

```
CREATE TABLE MyEmployee_Backup AS
SELECT * FROM MyEmployee; --Data เก่า
```

- ใช้คำสั่ง **DROP TABLE** เพื่อลบ Table

```
DROP TABLE MyEmployee_Backup;
DROP TABLE MyEmployee;
```



ลบแล้วลบเลย + ถ้าไม่ทำ Backup ไว้
ไม่มี Undo
ต้อง Recheck table ที่จะลบดีๆ