

คำสั่ง SQL

ความเป็นมาของ SQL

- SQL เป็นภาษามาตรฐานบนระบบฐานข้อมูลเชิงสัมพันธ์ อีกทั้งยังเป็นภาษาที่ถูกนำไปใช้งานบนคอมพิวเตอร์หลายระดับด้วยกัน ไม่ว่าจะเป็น เมนเฟรมคอมพิวเตอร์ จนถึงไมโครคอมพิวเตอร์ ดังนั้น จึงไม่น่าแปลกใจเลยว่า ผลิตภัณฑ์ซอฟต์แวร์จัดการฐานข้อมูลที่มีตามท้องตลาดในปัจจุบัน ล้วนสนับสนุนการทำงานภายใต้ชุดคำสั่ง SQL แทบทั้งสิ้น ในปัจจุบัน ผลิตภัณฑ์ซอฟต์แวร์จัดการฐานข้อมูลของแต่ละค่าย ต่างก็ได้เพิ่มคุณสมบัติพิเศษเพิ่มเติม บางอย่างเข้าไปในชุดผลิตภัณฑ์ ของตน เพื่อให้มีประสิทธิภาพสูงขึ้น และนำไปใช้เป็นจุดขายในเชิงการตลาด

วัตถุประสงค์ของ SQL

1. เพื่อสร้างฐานข้อมูลและโครงสร้างของลิเลชั่น
2. สนับสนุนงานจัดการฐานข้อมูลพื้นฐาน อันได้แก่ การเพิ่ม การปรับปรุง และการลบข้อมูลจากลิเลชั่น
3. การสนับสนุนงานคิวรีข้อมูล ทั้งในรูปแบบอย่างง่ายและซับซ้อน

ประเภทของชุดคำสั่ง SQL

- 1. ภาษานิยามข้อมูล** ประกอบด้วยกลุ่มคำสั่งที่ใช้สำหรับสร้างตาราง แก้ไขตาราง และลบตาราง กล่าวคือ เป็นกลุ่มคำสั่งที่ใช้ในการสร้างตารางข้อมูล ด้วยการ กำหนดโครงสร้างข้อมูลว่ามีคอลัมน์หรือแอตทริบิวต์บ้าง มีชนิดข้อมูล เป็นชนิดใด รวมทั้งการจัดการด้านการเพิ่ม แก้ไข ลบแอตทริบิวต์ต่าง ๆ ใดก็ได้เช่นกัน และการสร้างลำดับดัชนี
- 2. ภาษাজัดการข้อมูล** เป็นคำสั่งที่ถือเป็นแกนสำคัญของ SQL กลุ่มคำสั่งเหล่านี้จะถูกนำมาใช้เพื่อการอัปเดต เพิ่ม ปรับปรุง และเรียกดูข้อมูลในฐานข้อมูล
- 3. ภาษาควบคุมข้อมูล** เป็นกลุ่มชุดคำสั่งที่ช่วยอำนวยความสะดวกแก่ผู้บริหารฐานข้อมูลในการนำไปใช้เพื่อควบคุมฐานข้อมูล ประกอบด้วยคำสั่งเพื่อการอนุญาต เลิกสิทธิในการเข้าถึงฐานข้อมูล ซึ่งเป็นกระบวนการป้องกันความปลอดภัยต่อฐานข้อมูลนั่นเอง

ประเภทของคีย์ในระบบฐานข้อมูลแบบสัมพันธ์

รูปแบบของคำสั่งแก้ไขของ SQL มีดังนี้

คำสั่ง Update ชื่อตาราง

คำสั่ง set คือคำสั่งที่ใช้การกำหนดค่าจะต้องมี

ฟิลด์1 = ข้อมูล

ฟิลด์1 = ข้อมูล

Where เป็นเงื่อนไข

- ในกรณีที่ตารางที่เป็น Key มักจะไม่แก้ไข Primary Key
- ข้อมูล text จะต้องใส่ “
- ข้อมูลตัวเลขให้ใส่ ค่าไปเลย เช่น 9.0 เป็นต้น

1. คำสั่งลบข้อมูลใน SQL

รูปแบบคำสั่ง **delete from** ตามด้วยชื่อตาราง,
แล้วตามด้วยเงื่อนไข **where**

**** ถ้าไม่ใส่เงื่อนไข ข้อมูลจะลบทั้งตาราง**

delete FROM student (ไม่ใส่เงื่อนไข)

delete FROM student where sid='6401103001'; (ใส่เงื่อนไข)

2. การสร้างตาราง

ขั้นตอนการสร้างฐานข้อมูลจะแตกต่างกันตามผลิตภัณฑ์ ของแต่ละค่าย ตัวอย่างเช่น ในระบบมัลติยูสเซอร์ ผู้ที่มีสิทธิในการสร้างฐานข้อมูลปกติจะถูกสงวนไว้ให้กับผู้ดำเนินการได้เท่านั้นในขณะที่ระบบผู้ใช้คนเดียว ฐานข้อมูลอาจถูกสร้างขึ้นเมื่อระบบได้รับการติดตั้ง และมีการตั้งค่าคอนเฟิร์มต่าง ๆ ต่อไปนี้คือรูปแบบของการสร้างตารางข้อมูล

คำสั่ง

CREATE TABLE NAMES (Id integer PRIMARY KEY, Name text);

คำสั่งสร้างตาราง

การสร้างตาราง

- CREATE TABLE NAMES (Id integer PRIMARY KEY, Name text);

ยกตัวอย่างเช่นสร้างตาราง student

- CREATE TABLE student (student_Id text PRIMARY KEY, student_Name text);

3. เพิ่มข้อมูล

คำสั่ง

```
INSERT INTO NAMES VALUES(1,'Tom');
```

ยกตัวอย่างเช่นเพิ่มข้อมูลนักศึกษา

```
INSERT INTO student (student_id, student_name)  
VALUES ('6501103077001', 'Wattana');
```

```
INSERT INTO student (student_id, student_name)  
VALUES ('6501103077002', 'วัฒนา');
```

เข้าเว็บ SQL

[SQL OnLine IDE \(sqliteonline.com\)](http://sqliteonline.com)

ให้นักศึกษาสร้างตารางต่อไปนี้ พร้อมทั้งแสดงผลข้อมูล

```
CREATE TABLE student(student_Id text PRIMARY KEY, student_Name text  
student_address text);
```

```
INSERT INTO student (student_id, student_name)  
VALUES ('6501103077001', 'Wattana');
```

```
INSERT INTO student(student_id, student_name)  
VALUES ('6501103077002', 'วัฒนา');
```

```
/*SELECT * FROM student;*/
```

ตัวอย่างตารางที่ได้ดังนี้

SQLite 0.0.4 beta

Table

- demo
- student

Column

- student_Id TEXT
- student_Name text...

SQLite

```
1 CREATE TABLE student(student_Id text PRIMARY KEY, student_Name text
2 student_address text);
3
4 INSERT INTO student (student_id, student_name)
5 VALUES ('6501103077001', 'Wattana');
6
7 INSERT INTO student(student_id, student_name)
8 VALUES ('6501103077002', 'วัฒนา');
```

student_Id	student_Name
6501103077001	Wattana
6501103077002	วัฒนา

โจทย์ที่ 1 ให้นักศึกษาสร้างตารางดังต่อไปนี้ ในคำสั่ง SQL

แถว	A	B	C	D	E	F
1	1	1	2	a	b	a
2	2	1	1	b	c	b
3	2	3	5	c	b	b

จากตารางจงหาค่าว่าตารางใด สามารถเป็นคีย์หลักได้

BEGIN TRANSACTION;

/* Create a table called NAMES */

CREATE TABLE test(a text , b text ,c text , d text , e text, f text);

/* Create few records in this table */

INSERT INTO test(a,b,c,d,e,f) VALUES('1','1','2','a','b','a');

INSERT INTO test(a,b,c,d,e,f) VALUES('2','1','1','b','c','b');

INSERT INTO test(a,b,c,d,e,f) VALUES('2','3','5','c','b','b');

COMMIT;

SELECT * FROM test where a='2'

โจทย์ ให้นักศึกษาสร้างตาราง **STUDENT** โดยกำหนดค่าตารางดังนี้
SID TEXT PRIMARY KEY, NAME TEXT, FACULTY TEXT, SYEAR INT

```
CREATE TABLE student (sid text PRIMARY KEY, Name text, faculty text, syear int);
```

```
INSERT INTO student (sid,name,faculty,syear)
```

```
VALUES('6401103001','wattana','วิทยาศาสตร์และเทคโนโลยี','1');
```

```
INSERT INTO student (sid,name,faculty,syear)
```

```
VALUES('6401103002','วัฒนา','บัญชี','1');
```

```
INSERT INTO student (sid,name,faculty,syear)
```

```
VALUES('6301103011','อิสริยา','บริหารธุรกิจ', '3');
```

```
SELECT * FROM student;
```

ผล RUN ที่ได้

File Owner DB Run Export Import

SQLite

Table

demo

student

SQLite

```
1 CREATE TABLE student (sid TEXT PRIMARY KEY, Name TEXT, faculty TEXT, syear INT);
2
3 /* Create few records in this table */
4 INSERT INTO student (sid,name,faculty,syear)
5 VALUES ('6401103001','wattana', 'วิทยาศาสตร์และเทคโนโลยี', '1');
6 INSERT INTO student (sid,name,faculty,syear)
```

sid	Name	faculty	syear
6401103001	wattana	วิทยาศาสตร์และเทคโนโลยี	1
6401103002	วัฒนา	บัญชี	1
6301103011	อิสริยา	บริหารธุรกิจ	3

4. คำสั่งแสดงข้อมูล

```
SELECT * FROM Name;
```

ยกตัวอย่างเช่นการเรียกดูข้อมูล student

```
SELECT * FROM student;
```

5. คำสั่งการหยุดแสดงข้อมูล

`/* SELECT * FROM NAMES; */`

ให้ใส่เครื่องหมาย `/*` ด้านหน้าข้อความ และใส่เครื่องหมาย `*/` ไว้ด้านหลัง ระบบ
จะทำการหยุดการเรียกดูข้อมูลในบรรทัดนั้น

6. ตัวอย่างการเปลี่ยนข้อมูลหรือแก้ไขข้อมูล

คำสั่ง UPDATE

เงื่อนไข Where

ยกตัวอย่างเช่น

```
UPDATE student set name = 'kanokporn'
```

```
where name = 'วัฒนา' ;
```

```
SELECT * FROM student;
```

คำสั่งลบตารางทั้งหมด

- คำสั่ง `DROP TABLE student;`

การบ้าน

- ให้นักศึกษาไปสร้างตารางต่อไปนี้ ทุกตารางต้องมีข้อมูลอย่างน้อย 15 รายการ
 - ตารางข้อมูลหนังสือ
 - ตารางข้อมูลสินค้า
 - ตารางข้อมูลพนักงาน

ให้นักศึกษาสร้างตาราง **STUDENT** โดยจะต้องกำหนดข้อมูลดังนี้

- sid text PRIMARY KEY, name text , faculty text, syear int

ให้นักศึกษาสร้างตาราง **STUDENT** โดยจะต้องมีการกำหนดค่าดังต่อไปนี้
SID TEXT PRIMARY KEY, NAME TEXT , FACULTY TEXT, SYEAR INT

```
CREATE TABLE student(sid text PRIMARY KEY, name text , faculty text, syear int);
```

```
/* Create few records in this table */
```

```
INSERT INTO student(sid, name , faculty, syear) VALUES('64001','araya','IT', 2);
```

```
INSERT INTO student(sid, name , faculty, syear) VALUES('64002','suwit','AC', 2);
```

```
INSERT INTO student(sid, name , faculty, syear) VALUES('64003','kanok','AC', 3);
```

```
INSERT INTO student(sid, name , faculty, syear) VALUES('64004','Isariya','AC', 3);
```

```
INSERT INTO student(sid, name , faculty, syear) VALUES('64005','PIM','AC', 4);
```

คำสั่ง UPDATE แก้ไข

เปลี่ยนหลายตาราง

```
UPDATE student set name = 'kanokporn', faculty = 'IB', syear = 4  
where name = 'kanok' ;
```

เปลี่ยนตารางเดียว

```
UPDATE student set name = 'kanokporn'  
where name = 'kanok' ;
```

คำสั่งเรียกดูข้อมูล

```
SELECT * FROM student ;
```


คำสั่งลบข้อมูล

- กรณีลบบางตารางต้องใส่

```
DELETE FROM student
```

```
WHERE sid = '64001' ;
```

- กรณีลบทั้งตารางไม่ต้องใส่เงื่อนไข

```
DROP TABLE student ;
```