

บทที่ 7 คำสั่งของ MySQL

1. เข้าโปรแกรม C:\xampp\Mysql\bin\mysql -u root -p
2. ออกโปรแกรม C:\xampp\Mysql\bin\quit
3. การใช้เครื่องหมายปิดประโยค ;
4. สร้างฐานข้อมูล CREATE DATABASE <ชื่อฐานข้อมูล>; เช่น CREATE DATABASE TT;
5. แสดงฐานข้อมูล SHOW DATABASES;
6. เรียกใช้ฐานข้อมูล USE <ชื่อฐานข้อมูล>; เช่น USE TT;
7. ลบฐานข้อมูล DROP DATABASE <ชื่อฐานข้อมูล>; เช่น DROP DATABASE TT;
8. สร้างเพิ่มข้อมูลใหม่

CREATE TABLE ชื่อตาราง

(Field1 ชนิดข้อมูล(ความกว้าง) not null unique auto_increment,
 Field2 ชนิดข้อมูล(ความกว้าง) not null unique,.....,
 Primary key(field1,...),index(field1,...));

EX1 สร้างเพิ่มชื่อ SKILL โดยมีข้อมูลดังนี้

SKILLID	SKILLNAME	RATE	HOUR
1	Electric	150.00	8
2	Water	300.00	7
3	Structure	250.50	8

ด้วยคำสั่ง

- CREATE TABLE SKILL(SKILLID INTEGER NOT NULL UNIQUE AUTO_INCREMENT,
 → SKILLNAME VARCHAR(30) NOT NULL UNIQUE, RATE DECIMAL(6,2), HOUR
 TINYINT,
 → PRIMARY KEY(SKILLID),INDEX(SKILLID,SKILLNAME));

EX2 สร้างเพิ่มชื่อ EMPLOYEE โดยมีข้อมูลดังนี้

EMPID	EMPNAME	BIRTHDAY	SKILLID	SEX	CITY
1	Suwat	1975/10/15	1	M	Chiang mai
2	Supee	1970/01/20	1	F	Lumphun
3	Anupong	1960/01/01	2	M	Lumphun
4	Aniruth	1960/05/21	1	M	

- CREATE TABLE EMPLOYEE (EMPID INT NOT NULL UNIQUE AUTO_INCREMENT,
 → EMPNAME VARCHAR(50) NOT NULL UNIQUE, BIRTHDAY DATETIME, SKILLID INT
 → NOT NULL, SEX CHAR(1) NOT NULL,CITY VARCHAR(50),

➔ PRIMARY KEY(EMPID),INDEX(EMPID,EMPNAME,SEX,CITY));

9. เรียกดูโครงสร้างเพิ่มข้อมูล

- EXPLAIN <ชื่อเพิ่มข้อมูล>
- DESCRIBE <ชื่อเพิ่มข้อมูล>
- SHOW COLUMNS FROM <ชื่อเพิ่มข้อมูล>

เช่น EXPLAIN SKILL;

10. เปลี่ยนชื่อตาราง

Rename table ชื่อเดิม1 to ชื่อใหม่1, ชื่อเดิม2 to ชื่อใหม่2,..... ;

เช่น Rename table skill to Skill2

11. ลบเพิ่มข้อมูล DROP TABLE <ชื่อเพิ่มข้อมูล> เช่น DROP TABLE SKILL;

12. แก้ไขโครงสร้างเพิ่มข้อมูล

a. เพิ่ม Field ในเพิ่มข้อมูล

ALTER TABLE <ชื่อเพิ่มข้อมูล> ADD ชื่อฟิลด์ ชนิดฟิลด์ คุณสมบัติฟิลด์ ;

EX1 เพิ่มฟิลด์ใหม่ชื่อ SNOOK ชนิด Varchar(30) ในเพิ่ม SKILL โดยไม่ให้มีค่าว่าง

. ALTER TABLE SKILL ADD SNOOK VARCHAR(30) NOT NULL;

b. แก้ไข Field ในเพิ่มข้อมูล

ALTER TABLE <ชื่อเพิ่มข้อมูล> MODIFY ชื่อฟิลด์ ชนิดฟิลด์ คุณสมบัติฟิลด์ ;

EX2 เปลี่ยนฟิลด์ชื่อ SNOOK จาก Varchar(30) เป็น TINYINT ในเพิ่ม SKILL และให้

ค่าเริ่มต้น

. ALTER TABLE SKILL MODIFY SNOOK TINYINT DEFAULT '1';

c. ลบ Field ในเพิ่มข้อมูล

ALTER TABLE <ชื่อเพิ่มข้อมูล> DROP ชื่อฟิลด์;

. ALTER TABLE SKILL DROP SNOOK ;

13. เพิ่ม Record ใหม่ในเพิ่มข้อมูล

INSERT INTO <ชื่อตาราง> (ชื่อฟิลด์1,ชื่อฟิลด์2,...,ชื่อฟิลด์n) Values(ค่าที่จะใส่ในฟิลด์1,ฟิลด์2,...,ฟิลด์n)

Ex เพิ่มเรคคอร์ดที่ 1 ในเพิ่ม Skill ในหน้าที่ 1

. INSERT INTO SKILL (SKILLNAME,RATE,HOUR) Values('Electric',150.00,8);

Ex2 เพิ่มเรคคอร์ดที่ 1 ในเพิ่ม Employee ในหน้าที่ 1

. INSERT INTO EMPLOYEE (EMPNAME,BIRTHDAY,SKILLID,SEX,CITY)

➔ Values('Suwat','1975/10/15',1,'M','Chiang mai');

14. แก้ไขข้อมูล Record ในเพิ่มข้อมูล

```
UPDATE <ชื่อตาราง>
SET ชื่อฟิลด์1 = ข้อมูลที่จะแก้,ชื่อฟิลด์2 = ข้อมูลที่จะแก้
WHERE เงื่อนไขที่จะแก้ในเรคคอร์ดนั้นๆ
```

หมายเหตุ ถ้าไม่มี Where จะทำการ update ให้ทุกเรคคอร์ด

Ex แก้ไขฟิลด์ชื่อ SEX เป็น E ทุกเรคคอร์ดในแฟ้ม Employee

```
. UPDATE EMPLOYEE
```

```
→ SET SEX = 'E';
```

Ex2 แก้ไขฟิลด์ชื่อ SEX เป็น M เฉพาะเรคคอร์ดที่ EMPID = 1 ในแฟ้ม Employee

```
. UPDATE EMPLOYEE
```

```
→ SET SEX = 'M'
```

```
→ WHERE EMPID = 1;
```

หมายเหตุ เราสามารถใช้เครื่องหมาย >, <, >=, <=, <> และใช้ and , or ในการเชื่อมประโยค
เงื่อนไขหลัง WHERE

Ex3 แก้ไขฟิลด์ชื่อ SEX เป็น F เฉพาะเรคคอร์ดที่ EMPID เป็น 2 ขึ้นไป ในแฟ้ม Employee

```
. UPDATE EMPLOYEE
```

```
→ SET SEX = 'M'
```

```
→ WHERE EMPID >= 2;
```

Ex4 แก้ไขฟิลด์ชื่อ BIRTHDAY เป็น 1990/09/09 เฉพาะเรคคอร์ดที่ EMPID เป็น 2 และ CITY เป็น LUMPHUN ในแฟ้ม Employee

```
. UPDATE EMPLOYEE
```

```
→ SET BIRTHDAY = '1990/09/09'
```

```
→ WHERE EMPID= 2 AND CITY = 'Lumphun';
```

Ex5 แก้ไขเรคคอร์ดที่พนักงานรหัสที่ 1 โดยแก้ไขฟิลด์ ชื่อพนักงานเป็น SUWATS และ เมืองเป็น

เชียงใหม่

```
. UPDATE EMPLOYEE
```

```
→ SET EMPNAME = 'SuwatS', CITY = 'Chiang rai'
```

```
→ WHERE EMPID= 1
```

14.1 แก้ไขข้อมูลในตาราง Skill เฉพาะที่ Skillid เป็น 1 ให้ค่า Hour เป็น 25

```
Update skill set Hour = 25 where Skillid = 1 ;
```

14.2 แก้ไขข้อมูลในตาราง Skill เฉพาะที่ Rate มากกว่า 250 โดยให้ Hour เป็น 60

```
Update skill set Hour = 60 where Rate > 250 ;
```

14.3 แก้ไขทุกเรคคอร์ดใน Skill โดยให้ Hour เป็น 100

```
Update Skill set Hour = 100;
```

14.4 แก้ไข Skilltype เป็น water เฉพาะ Skillid เป็น 2

Update skill set Skilltype = "water" where skillid = 2;

15. ลบ Record ในแฟ้มข้อมูล

DELETE FROM <ชื่อตาราง>

WHERE เงื่อนไขที่จะลบในเรคคอร์ดนั้นๆ

Ex ลบทุกเรคคอร์ดในแฟ้ม SKILL

. DELETE FROM SKILL;

Ex2 ลบเฉพาะเรคคอร์ดที่ฟิลด์ RATE มีค่าไม่เท่ากับ 300.00 ในแฟ้ม SKILL

. DELETE FROM SKILL

→ WHERE RATE <> 300.00;

Ex3 ลบเฉพาะเรคคอร์ดที่ฟิลด์ EMPNAME ชื่อ SUWAT หรือ SUPEE ในแฟ้ม EMPLOYEE

. DELETE FROM EMPLOYEE

→ WHERE EMPNAME = 'Suwat' or EMPNAME = 'Supee';

15.1 ลบทุกเรคคอร์ดในตาราง Skill

Delete From Skill;

15.2 ลบเรคคอร์ดที่ Skillid มีค่ามากกว่า 3

Delete From Skill where skillid > 3,

15.3 ลบเรคคอร์ดที่ Rate เท่ากับ 250 หรือ Hour มากกว่า 30

Delete From Skill where Rate = 250 OR Hour > 30;

15.4 ลบเรคคอร์ดที่ Rate อยู่ระหว่าง 250 ถึง 300

Delete From Skill where Rate >=250 and Rate<= 300 ;

15.5 ลบเรคคอร์ดที่ Skilltype ขึ้นต้นด้วยตัว C

Delete From Skill where Skilltype like "C*";

15.6 ลบเรคคอร์ดที่ Skilltype ที่มีตัว c อยู่ในฟิลด์

Delete From Skill where Skilltype like "*C*"

16. การสืบค้นข้อมูลเพื่อนำมาแสดงผล

SELECT [distinct/* /field1,field2,...,field n] FROM <ชื่อตาราง>

WHERE เงื่อนไขในการแสดงเรคคอร์ด

GROUP BY ชื่อฟิลด์ HAVING เงื่อนไขในการจัดกลุ่ม

ORDER BY ชื่อฟิลด์1,ชื่อฟิลด์2,...,ฟิลด์ n [ASC/DESC]; การเรียงลำดับตามฟิลด์

Ex แสดงทุกเรคคอร์ดในแฟ้ม EMPLOYEE

. SELECT * FROM EMPLOYEE;

Ex3 แสดง EMPID,EMPNAME,BIRTHDAY จากแฟ้ม EMPLOYEE เฉพาะฟิลด์ SEX เป็น M

. SELECT EMPID,EMPNAME,BIRTHDAY FROM EMPLOYEE WHERE SEX = 'M';

Ex4 แสดงทุกฟิลด์จากแฟ้ม EMPLOYEE โดยเรียงลำดับเรคคอร์ดตาม EMPID จากมากไปน้อย

. SELECT * FROM EMPLOYEE ORDER BY EMPID DESC;

Ex5 แสดงทุกฟิลด์จากแฟ้ม EMPLOYEE โดยเรียงลำดับเรคคอร์ดตาม CITY จากน้อยไปมาก ถ้า CITY เหมือนกันให้เรียงตาม EMPID จากมากไปน้อย

```
. SELECT * FROM EMPLOYEE ORDER BY CITY ASC,EMPID DESC;
```

Ex6 แสดงเฉพาะเรคคอร์ดที่ CITY ไม่ซ้ำกันจากแฟ้ม EMPLOYEE

```
. SELECT DISTINCT CITY FROM EMPLOYEE;
```

Ex7 คำนวณเพื่อสร้างฟิลด์ใหม่ชื่อ Salary จาก RATE*HOUR และแสดง Skillid,Skillname จากแฟ้ม SKILL โดยเรียงลำดับตาม Salary จากน้อยไปมาก

```
. SELECT RATE*HOUR AS Salary,Skillid,Skillname FROM SKILL ORDER BY SALARY;
```

Ex8 แสดงทุกฟิลด์จากแฟ้ม EMPLOYEE เฉพาะเรคคอร์ดที่ CITY ไม่เป็นค่าว่าง

```
. SELECT * FROM EMPLOYEE WHERE CITY IS NOT NULL;
```

Ex9 แสดงทุกฟิลด์จากแฟ้ม EMPLOYEE เฉพาะเรคคอร์ดที่ CITY เป็นค่าว่าง

```
. SELECT * FROM EMPLOYEE WHERE CITY IS NULL;
```

Ex10 แสดงทุกฟิลด์จากแฟ้ม SKILL ที่ฟิลด์ RATE อยู่ระหว่าง 200 ถึง 280

```
. SELECT * FROM SKILL WHERE RATE >= 200 AND RATE <= 280;
```

หรือ

```
. SELECT * FROM SKILL WHERE RATE BETWEEN 200 AND 280;
```

Ex11 แสดง EMPID,EMPNAME จากแฟ้ม EMPLOYEE เฉพาะคนที่ชื่อขึ้นต้นด้วย S

```
. SELECT EMPID,EMPNAME FROM EMPLOYEE WHERE EMPNAME LIKE 'S%';
```

Ex12 หาค่ามากที่สุด ค่าน้อยสุดและค่าเฉลี่ยของฟิลด์ RATE จากแฟ้ม SKILL

```
. SELECT MAX(RATE),MIN(RATE),AVG(RATE) FROM SKILL;
```

Ex13 แสดงทุกฟิลด์จากแฟ้ม SKILL เฉพาะที่ฟิลด์ RATE ไม่อยู่ระหว่าง 100 ถึง 150

```
. SELECT * FROM SKILL WHERE RATE NOT BETWEEN 100 AND 150;
```

Ex14 แสดงฟิลด์ EMPID,EMPNAME,CITY จากแฟ้ม EMPLOYEE ฟิลด์ชื่อ SKILLNAME,RATE จากแฟ้ม SKILL เฉพาะที่เรคคอร์ดที่ฟิลด์ SKILLID ของทั้ง 2 แฟ้มเหมือนกัน

```
. SELECT EMPLOYEE.EMPID,EMPLOYEE.EMPNAME,EMPLOYEE.CITY
```

```
→,SKILL.SKILLNAME, SKILL.RATE FROM EMPLOYEE,SKILL
```

```
→ WHERE SKILL.SKILLID = EMPLOYEE.SKILLID;
```

Ex15 แสดงจำนวนพนักงานของแต่ละกลุ่มของ SKILLID จากแฟ้ม EMPLOYEE โดยเรียงตามจำนวนพนักงานที่ได้จากมากไปน้อย

```
. SELECT SKILLID,COUNT(SKILLID) FROM EMPLOYEE
```

```
→ GROUP BY SKILLID ORDER BY COUNT(SKILLID) DESC;
```

Ex16 แสดงจำนวนพนักงานของแต่ละกลุ่มของ CITY จากเพิ่ม EMPLOYEE โดยเฉพาะที่ CITY ที่พนักงานมีมากกว่า 1 คน

```
. SELECT CITY,COUNT(CITY) FROM EMPLOYEE
```

```
→ GROUP BY CITY HAVING COUNT(CITY) > 1;
```

หมายเหตุ 1.คำสั่งหลัง WHERE สามารถนำไปประยุกต์ใช้ใน DELETE ,UPDATE ได้

2. การใช้ ASC หมายถึง การเรียงลำดับจากน้อยไปมาก จะระบุต่อท้ายฟิลด์หรือไม่ก็ได้ ถ้าไม่ระบุโปรแกรมจะถือว่าเป็น ASC โดยอัตโนมัติ

3. การใช้ DESC หมายถึง การเรียงลำดับจากมากไปน้อย

4. หากใช้ GROUP BY จะไม่สามารถใช้คำสั่ง WHERE พร้อมๆ กันได้ เพราะเป็นข้อห้ามของการใช้คำสั่ง SELECT

5. การใช้ SELECT จะต้องใช้ร่วมกับ FROM เสมอ ส่วนคำสั่ง WHERE, GROUP BY, ORDER BY เป็นคำสั่งเสริมที่จะมีหรือไม่มีก็ได้

6. คำสั่งหลัง WHERE ถ้าฟิลด์ใดไม่ได้เป็นตัวเลข จะต้องอยู่ภายใต้เครื่องหมาย “เสมอ

17. คำสั่งให้สิทธิในการทำงาน

17.1 กำหนดสิทธิในการใช้สำหรับ user test โดยสามารถทำได้ทุกสิ่งสำหรับ database test

```
Grant all on test.* to test;
```

17.2 กำหนดสิทธิในการใช้ให้กับ user demo โดยใช้ Select อย่างเดียวใน database test

```
Grant select on test.* to demo identified by 'demo';
```

17.3 กำหนดสิทธิในการใช้ให้กับ user sale1 ให้สามารถดู เพิ่ม แก้ไขข้อมูลใน database test ได้

```
Grant select, insert, update on test.* to sales1;
```

18. คำสั่งในการถอนสิทธิในการทำงาน จะคล้ายๆ กับการกำหนดสิทธิ์ ต่างกันตรงที่ใช้ From user แทน To user เช่น

18.1 ยกเลิกการให้สิทธิการแก้ไข stock_date จาก sales3

```
revoke update(stock_date) on support.inv from sales3;
```