

บทที่ 7 คำสั่ง SQL

- ✦ 1982 The American National Standards Institute(ANSI) ได้กำหนดมาตรฐานของชุดคำสั่ง SQL เพื่อให้ผู้ผลิตรายต่างๆ สร้างชุดคำสั่งให้อยู่ในมาตรฐานเดียวกัน
- ✦ ในปัจจุบันมีระบบจัดการฐานข้อมูลต่างๆเช่น ORACLE,DB2,SYBASE,Informix,MS-SQL,MS-Access,MS-FoxPro ,File Maker,My SQL,SQL Server ฯลฯ

หมายเหตุ : OODBMS จะมองข้อมูลในฐานข้อมูลเป็น Object ที่เก็บทั้ง Data และ Method ดังนั้นการใช้งานจะมีความรวดเร็ว ตัวอย่าง OODBMS เช่น Cache และ DBMSของ RDBMS บางตัวได้พัฒนาให้สามารถทำงานกับข้อมูลในลักษณะเชิงวัตถุได้ด้วย เช่น Oracle แต่อย่างไรก็ตามพื้นฐานของโครงสร้างหรือโมเดลข้อมูลยังเป็น RDBMS

ประวัติย่อและความเป็นมาของภาษา SQL

- ✦ ในการสอบถามข้อมูลจากฐานข้อมูล คำถามของผู้ใช้ เรียกว่า Query
- ✦ ภาษาที่ใช้ในการสอบถามข้อมูลเรียกว่า Query Language ซึ่งมีหลายภาษา แต่ละภาษาได้รับการพัฒนาต่างกันไปตาม DBMS
- ✦ ภาษา SQL(Structured Query Language) พัฒนาโดย IBM จากแนวคิดทางคณิตศาสตร์ของ E.F.Codd ในปี 1974 โดยใช้ชื่อเดิมว่า SEQUEL (Structured English Query Language) เป็นภาษาที่นิยมใช้ใน RDBMS
- ✦ ระบบฐานข้อมูล Oracle ของ Oracle Corporation นำ SQL มาใช้ในการพัฒนาระบบการจัดการฐานข้อมูลในเชิงพาณิชย์ เป็นรายแรก ได้รับความนิยม จึงมีผลิตภัณฑ์อื่นๆ ตามมา เช่น Ingress ,Access,DB2 เป็นต้น

วัตถุประสงค์ของ SQL (Objective of SQL)

- ✦ สร้างฐานข้อมูลและโครงสร้างรีเลชัน
- ✦ สนับสนุนงานด้านการจัดการฐานข้อมูลพื้นฐาน คือ การเพิ่ม ปรับปรุงและลบข้อมูลจากรีเลชัน
- ✦ สนับสนุนการ Query ข้อมูลพื้นฐานตลอดจนข้อมูลชั้นสูงที่มีความซับซ้อน รวมทั้งสามารถแปลงข้อมูลดิบให้เป็นสารสนเทศ

การใช้งานภาษา SQL

1.แบบโต้ตอบ (Interactive SQL)

เป็นการปฏิบัติการที่ผู้ใช้เรียกคำสั่ง SQL ให้มีการดำเนินการกับข้อมูลแบบโต้ตอบระหว่างผู้ใช้กับคอมพิวเตอร์โดยตรง โดยการพิมพ์คำสั่ง SQL ผ่านคอมพิวเตอร์ และผลลัพธ์ของข้อมูลจากฐานข้อมูลจะปรากฏบนจอภาพในทันที (User ต้องมีความรู้ในเบื้องต้นและจะต้องได้รับสิทธิในการเข้าถึงข้อมูล)

2.แบบฝังในตัวโปรแกรม (Embedded SQL)

เป็นการเขียนชุดคำสั่งภาษา SQL ไว้ในโปรแกรม ปัจจุบันมีภาษาโปรแกรมหลายภาษาที่สนับสนุนคำสั่ง SQL รวมทั้งภาษาในยุคที่ 3 เช่น C, COBOL, Pascal ในการปฏิบัติการลักษณะนี้จำเป็นต้องพึ่งพา Programmer ในการเขียนชุดคำสั่งด้วยการฝังไว้ในตัวโปรแกรม (Embedded) ซึ่งทำให้ลดข้อจำกัดบางอย่างของชุดคำสั่งภาษา SQL ได้ เช่น ไม่มีคำสั่ง Loop, Do..While รวมถึงเงื่อนไขที่มีความสลับซับซ้อน

5

ประเภทของคำสั่งในภาษา SQL

- **ภาษานิยามข้อมูล (Data Definition Language : DDL)** เป็นกลุ่มคำสั่งในการสร้างฐานข้อมูล กำหนดโครงสร้างของ Attribute ชนิดข้อมูล รวมทั้งการเพิ่ม ลบ แก้ไข Attribute ในรีเลชัน รวมทั้งการสร้างดัชนี(Index)
- **ภาษาจัดการข้อมูล(Data Manipulation Language :DML)** เป็นกลุ่มคำสั่งใช้ในการจัดการข้อมูล คือ การเพิ่ม ลบ แก้ไข และคิวรีข้อมูล
- **ภาษาควบคุมข้อมูล(Data Control Language :DCL)** เป็นกลุ่มคำสั่งที่ช่วยให้ DBA สามารถควบคุมสิทธิ์ของผู้ใช้ในการใช้ รวมทั้งควบคุมการใช้งานฐานข้อมูลจาก MultiUser เพื่อป้องกันความปลอดภัยของฐานข้อมูลรวมทั้งการจัดการ Transaction

7

Embedded SQL แบ่งออกเป็น

- **Embedded SQL Statements** เป็นชุดคำสั่ง (SQL Statement) ที่เขียนฝังอยู่ในตัวโปรแกรม (Source Code) โดยนำ SQL Statements เขียนผสมกับภาษาโปรแกรม โดยทาง ISO กำหนดให้ภาษาที่สนับสนุนมีดังนี้ คือ C, COBOL, Fortran, MUMPS, Pascal และ PL/1 เป็นต้น
- **Application Programming Interface (API)** โดยอนุญาตให้ Programmer สามารถใช้กลุ่มของฟังก์ชันในการร้องขอการใช้งานผ่าน API

6

ตารางที่ 1.1 ประเภทของข้อมูลที่เหมาะสมใน SQL เปรียบเทียบระหว่าง RDBMS 3 ตัว และ SQL มาตรฐาน

SQL มาตรฐาน	Oracle	Access	DB2
character(n) : n คือจำนวนของตัวอักษร	char(n) : จำกัดความยาวสูงสุดไม่เกิน 255 ตัวอักษร	text : จำกัดความยาวสูงสุดไม่เกิน 255 ตัวอักษร	character(n) : ความหมายเดียวกับ char(n) ใน Oracle
character varying(n) : จัดเก็บจำนวนตัวอักษรได้ถึง n ตัว แต่ขนาดจริงจะปรับตามจำนวนข้อมูลที่เก็บไว้	varchar2(n) : ในการจัดเก็บสามารถปรับความยาวได้ตามข้อมูล โดยกำหนดไว้ได้สูงสุด 2,000 ตัวอักษร	text : สามารถปรับความยาวได้สูงสุดถึง 255 ตัวอักษร หรือ Memo : สามารถปรับความยาวได้สูงสุดถึง 64,000 ตัวอักษร	varchar(n) : ความหมายเดียวกับ varchar2(n) ใน Oracle

float(p) : p คือ จำนวนหลักทั้งหมดของข้อมูลตัวเลข	number : แทนค่าข้อมูลที่เป็นตัวเลข	single หรือ double : ขึ้นอยู่กับช่วงของค่าข้อมูล	float : ความหมายเดียวกับ number ใน Oracle
decimal(p,s) : ตัวเลขจำนวนจริงมีทั้งหมด p หลัก และเป็นทศนิยม s หลัก	number(p,s) : ตัวเลขจำนวนจริงมีทั้งหมด p หลัก และเป็นทศนิยม s หลัก โดย p มีค่าตั้งแต่ 1 ถึง 38 และ s มีค่าตั้งแต่ -84 ถึง 127	integer หรือ long integer : ขึ้นอยู่กับช่วงของค่าข้อมูล	integer : ความหมายเดียวกับ number(p,s) ใน Oracle
Date/Time	date : วันที่และเวลา มีค่าได้ตั้งแต่ 1 มกราคม 4712 (ก่อนคริสตศักราช) ถึง 31 ธันวาคม 4712 (หลังคริสตศักราช)		

ภาษานิยามข้อมูล (Data Definition Language :DDL)

การสร้างเทเบิล (Create Table)

การสร้างเทเบิลด้วย SQL สามารถทำได้ด้วยคำสั่ง CREATE TABLE ซึ่งจะกำหนดลักษณะของข้อมูลเป็นคอลัมน์ (column) และมีรูปแบบดังนี้

```
CREATE TABLE <table_name>
(<column_name_1> <data_type> [NOT NULL][UNIQUE],
 <column_name_2> <data_type> [NOT NULL][UNIQUE],
 ...
 <column_name_n> <data_type> [NOT NULL][UNIQUE]),
[PRIMARY KEY (column_name),]
[FOREIGN KEY (column_name) REFERENCES table_name];
```

ภาษานิยามข้อมูล

(Data Definition Language :DDL)

- CREATE สร้างตาราง,ดัชนี,วิว
- DROP ลบตาราง,ดัชนี,วิว
- ALTER เปลี่ยนแปลงโครงสร้างตาราง

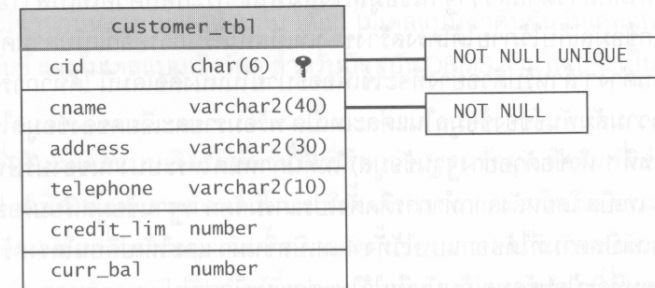
ตัวอย่าง 2-1

การสร้างเทเบิลตามโครงสร้างที่ออกแบบไว้ในบทที่ 1 (ตัวอย่างฐานข้อมูล)

>> ปัญหา ▶

สร้างเทเบิล 'customer_tbl'

>> ผังภาพ ▶



>> SQL ▶

```
CREATE TABLE customer_tbl
(cid char(6) NOT NULL UNIQUE,
 cname varchar2(40) NOT NULL,
 address varchar2(30),
 telephone varchar2(10),
 credit_lim number,
 curr_bal number,
 PRIMARY KEY (cid));
```

การเปลี่ยนโครงสร้างของเทเบิล (Alter Table)

ในการทำงานจริงนั้น แม้ว่าจะสร้างและป้อน (insert) ข้อมูลไว้ในเทเบิลเป็นจำนวนมากแล้วก็ตาม ก็อาจต้องเปลี่ยนโครงสร้างข้อมูลในเทเบิลนั้น เนื่องจากการออกแบบที่ผิดพลาดในตอนต้น หรือจากการเปลี่ยนความต้องการของผู้ใช้งาน ในกรณีที่เทเบิลมีข้อมูลอยู่แล้วควรระวังเป็นพิเศษ เพราะการแก้ไขโครงสร้างอาจมีผลกระทบต่อข้อมูลเหล่านั้น ในการแก้ไขโครงสร้างของเทเบิลทำได้ด้วยคำสั่ง ALTER TABLE โดยมีรูปแบบดังนี้

รูปแบบ

```
ALTER TABLE <table_name>
[MODIFY] [<column_name> <data_type>,]
[ADD] [<column_name> <data_type>];
[DROP] [<column name> <data type>]
```

2. เปลี่ยนแปลงโครงสร้างของเทเบิลที่ต้องการด้วยคำสั่งดังนี้

```
ALTER TABLE employee_tbl
ADD deptno char(6);
```

SQL ▶

```
ALTER TABLE employee_tbl
MODIFY (deptno char(5));
```

13

การสร้างวิว (CREATE VIEW)

การสร้างวิวเป็นการสร้างตารางสมมติ (virtual table) ขึ้นมาให้กับผู้ใช้งานบางกลุ่ม จุดประสงค์เพื่อต้องการให้ผู้ใช้งานเหล่านั้นสามารถเรียกข้อมูลที่ต้องการจากวิวที่กำหนดให้เท่านั้น โดยมีรูปแบบดังนี้

```
CREATE VIEW view_name [(column_name [, ...])]
AS subselect [WITH [CASCADED | LOCAL] CHECK OPTION];
```

```
CREATE VIEW STAFF2
AS SELECT sno, fname, lname, position, bno
FROM staff ;
```

15

การลบ Table ออกจากฐานข้อมูล (Drop Table)

รูปแบบ DROP TABLE <table_name>

เช่น DROP TABLE Order_Detail;

การสร้างดัชนีและลบดัชนี (CREATE/DROP INDEX)

คำสั่งที่ใช้ในการสร้าง/ลบดัชนีคือคำสั่ง CREATE INDEX และ DROP INDEX โดยมีรูปแบบดังนี้

```
CREATE [UNIQUE] INDEX index_name
```

```
ON table_name (column [ASC | DESC] [...])
```

```
DROP INDEX index_name
```

```
CREATE UNIQUE INDEX sno_ind ON staff(sno);
CREATE UNIQUE INDEX pno_ind ON property_for_rent(pno);
CREATE INDEX rent_ind on property_for_rent (area, rent);
DROP INDEX rent_ind;
```

ภาษาจัดการข้อมูล

(Data Manipulation Language :DML)

- SELECT ใช้สำหรับ Query ข้อมูลในฐานข้อมูล
- INSERT ใช้สำหรับเพิ่มข้อมูลในตาราง
- UPDATE ใช้สำหรับปรับปรุงข้อมูลในตาราง
- DELETE ใช้สำหรับลบข้อมูลในตาราง

16

การป้อนข้อมูลแต่ละเรคคอร์ดในตาราง

รูปแบบ **INSERT TO <table_name>**

VALUE (<value1>,<value2>,...);

```
>> SQL ▶ INSERT INTO customer_tbl  
VALUES('100','Sopha','Bangkok','2117400',300000,100000);
```

ในบางครั้งข้อมูลบางแถวอาจมีไม่ครบทุกคอลัมน์ การป้อนข้อมูลลักษณะนี้สามารถทำได้โดยมีรูปแบบ ดังนี้

```
รูปแบบ  
INSERT INTO <table-name><(list of columnname)>  
VALUES <(list of matching values)>;
```

```
>> SQL ▶ INSERT INTO customer_tbl(cid,cname,address)  
VALUES('501','Pagorn','Lampang');
```

19

การลบข้อมูลแต่ละเรคคอร์ดในตาราง

รูปแบบ **DELETE FROM <table_name>**

WHERE [Condition] ;

เช่น **DELETE FROM employee_tbl**

WHERE ename = 'Wilai';

การปรับปรุงข้อมูลแต่ละเรคคอร์ดในตาราง

รูปแบบ **UPDATE <table_name>**

SET (<Columnname1> = new_value1

,<Columnname1> = new_value2,...)

WHERE [Condition] ;

เช่น **UPDATE Employee_tbl**

SET Salary = Salary + 500.00

WHERE ename = 'Piya';

UPDATE Customer_tbl

SET Credit_lim =

Credit_lim+50000

WHERE Address = 'Bangkok'

18

การสอบถามข้อมูล (SELECT)

SELECT Statement

เป็นคำสั่งที่ใช้สำหรับเรียกดูข้อมูลจากตารางในฐานข้อมูล ซึ่งมีรูปแบบการใช้งานอยู่หลายรูปแบบด้วยกัน สามารถเรียกดูข้อมูลพร้อมเงื่อนไขประกอบ รวมทั้งการเรียกดูข้อมูลจากหลายๆ ตาราง ซึ่งประโยค SELECT จะมีรูปแบบการใช้งานดังนี้

```
SELECT [DISTINCT | ALL] { * | column_expression [ AS new_name ] } [ , ... ]  
FROM table_name [ alias ] [ , ... ]  
[ WHERE condition ]  
[ GROUP BY column_list ] [ HAVING condition ]  
[ ORDER BY column_list ] ;
```

Column_expression เป็นการแทนค่าชื่อคอลัมน์หรือ expression, table_name คือชื่อของตารางที่ต้องการเข้าถึงเพื่อวิิวข้อมูล และ alias คือชื่อย่อของตารางที่ใช้แทน table_name โดยลำดับของการประมวลผลในชุดคำสั่ง SELECT มีดังนี้

20

- ▶ FROM กำหนดตารางที่ต้องการใช้งาน
- ▶ WHERE สร้างเงื่อนไขเพื่อกรอง (filters) แถวที่ต้องการ
- ▶ GROUP BY จัดกลุ่มแถวที่มีค่าคอลัมน์เดียวกัน
- ▶ HAVING กรองกลุ่มเนื้อหาในบางเงื่อนไขจาก GROUP BY
- ▶ ORDER BY กำหนดให้เรียงลำดับผลลัพธ์ โดยที่
 - ASC คือการเรียงลำดับจากน้อยไปมาก
 - DESC คือการเรียงลำดับจากมากไปน้อย
 - (ค่าปกติเมื่อไม่ได้กำหนดจะเป็นการเรียงลำดับจากน้อยไปมาก)

21

ตัวอย่างการใช้คำสั่ง **SELECT**

- เรียกดูข้อมูลทุกฟิลด์ในตาราง ใช้ `SELECT * FROM Employee_tbl;`
- เรียกดูข้อมูลบางฟิลด์ ใช้ `SELECT ename,eaddress FROM Employee_tbl;`

22