บทที่ 9

ภาษา SQL (Standard Query Language)

9.1 ภาษา Standard relational database Query Language (SQL)

ภาษาที่กลายเป็นภาษามาตรฐานสำหรับระบบฐานข้อมูลคือ ภาษา Standard relational database Query Language หรือเอสคิวแอล (SQL) หรือซีควอล (SE-QUEL) ซึ่งเป็นภาษาที่พัฒนาขึ้นมาโดยบริษัทไอบีเอ็ม ภาษา SQL (Standard Query Language) เป็นส่วนหนึ่งของระบบฐานข้อมูลแบบรีเลชั่นเนล (Relational Database) ที่ได้รับความนิยมมากเพราะง่ายต่อความเข้าใจ และอยู่ในรูปภาษาอังกฤษ ภาษา SQL แบ่งออกเป็น 2 ส่วน คือ

- ภาษาที่ใช้สำหรับนิยามข้อมูล (Data Definition Language-DDL)
- ภาษาสำหรับการจัดการข้อมูลล (Data Manipulation Language: DML)
  - ภาษาควบคุม (Control Language)
  - ภาษาในการเลือกข้อมูล (Data Query Language)

รูปแบบการใช้คำสั่ง SQL สามารถใช้ได้เป็น 2 รูปแบบ ดังนี้ คือ

- คำสั่ง SQL ที่ใช้เรียกดูข้อมูลได้ทันที (Interactive SQL)

  เป็นการเรียกใช้คำสั่ง SQL สั่งงานบนจอภาพ เพื่อเรียกดูข้อมูลในขณะที่ทำงานได้ทันที เช่น

```
SELECT CITY
FROM SUPPLIER
WHERE SNO = 'SE';
```

- คำสั่ง SQL ที่ใช้เขียนร่วมกันโปรแกรมอื่น ๆ (Embedded SQL)

เป็นคำสั่ง SQL ที่ใช้ร่วมกับคำสั่งของโปรแกรมภาษาต่าง ๆ เช่น PL/1 PASCAL ฯลฯ หรือแม้แต่กับคำสั่งในโปรแกรมที่ระบบจัดการฐานข้อมูลนั้นมีใช้เฉพาะ เช่น ORACLE มี PL/SQL (Procedural Language /SQL) ที่สามารถเขียนโปรแกรมและนำคำสั่ง SQL มาเขียนร่วมด้วย เป็นต้น

ตัวอย่างการใช้คำสั่ง SQL ในภาษา PL/1

```
EXEC SQL SELECT CITY
INTO :XCITY
FROM SUPPLIER
WHERE SNO = 'S4';
```

9.2 ภาษาที่ใช้สำหรับนิยามข้อมูล (Data Definition Language-DDL)

Data Definition Language (DDL) เป็นภาษาที่ใช้นิยามโครงสร้างข้อมูล เพื่อเปลี่ยนแปลง หรือยกเลิกโครงสร้างฐานข้อมูลตามที่ออกแบบไว้ โครงสร้างดังกล่าวคือ สคีมา (Schema) นั้นเอง ตัวอย่างเช่น การกำหนดให้ฐานข้อมูลประกอบด้วยตารางอะไรบ้าง ชื่ออะไร ประเภทใด มีอินเด็กซ์ (Index)

ภาษา DDLประกอบด้วย 3 คำสั่งคือ

- คำสั่งการสร้าง (Create) ได้แก่ การสร้างตารางและอินเด็กซ์

```
CREATE TABLE <Table name>
(  Attribute 1  Type 1,
Attribute 2 Type 2 ,
)
CREATE Unique Index on X<Table name>
```

เช่น

```
CREATE TABLE S11
(SNO CHAR(5)  Not NULL,
SNAME CHAR(10) ,
STATUS integer
)
CREATE Unique Index XS11  on S11(SNO)
```

- คำสั่งเปลี่ยนแปลงโครงสร้าง

ALTER TABLE < ชื่อตารางที่ตั้งขึ้น >
<คำสั่งการเปลี่ยนแปลง> (<ชื่อคอลัมน์ ประเภทข้อมูล>);

ตัวอย่างเช่น

```
ALTER TABLE SUPPLIER
ADD (LAST_SNAME  Char(10));
```

- คำสั่งยกเลิก (Drop) ต่างๆ

การลบโครงสร้างตาราง

DROP TABLE < ชื่อตารางที่ตั้งขึ้น >

ภาษาดังกล่าวคือ ภาษาที่ใช้สร้างฐานข้อมูลลงในคอมพิวเตอร์ หลังจากที่เราได้ออกแบบแล้วว่าฐานข้อมูลมีกี่รีเลชั่น แต่ละรีเลชั่นมีความสัมพันธ์อย่างไร จากนั้นการใช้ภาษา DDL นี้แปลงรีเลชั่นต่างๆ ให้อยู่ในรูปภาษาสำหรับนิยามข้อมูล เพื่อป้อนเข้าสู่ระบบฐานข้อมูล เพื่อสร้างฐานข้อมูลที่แท้จริงให้เกิดขึ้นในคอมพิวเตอร์ ภาษา DDL สามารถสรุปคำสั่งต่างๆได้ดังตอไปนี้

คำสั่ง	**ความหมาย**
CREATE TABLE	นิยามโครงสร้างข้อมูลในรูปตารางบนฐานข้อมูล
DROP TABLE	ลบโครงสร้างตารางข้อมูลออกจากระบบ
ALTER TABLE	แก้ไขปรับปรุงโครงสร้างตาราง

คำสั่ง	ความหมาย
CREATE INDEX	สร้างดัชนีของตาราง
DROP INDEX	ลบ ดัชนีของตารางออกจากระบบ
CREATE VIEW	กำหนดโครงสร้างวิวของผู้ใช้
DROP VIEW	ลบโครงสร้างวิวออกจากระบบ

9.2.1 คำสั่งนิยามโครงสร้างตาราง

การสร้างตารางใน ฐานข้อมูลแบบรีเลชั่นเนล โดยเฉพาะฐานข้อมูลขนาดใหญ่บนระบบ UNIX จะทำด้วยการป้อนคำสั่งในลักษณะเท็กซ์โหมด (Text Mode) เข้าไปในระบบฐานข้อมูล ดังรูปแบบต่อไปนี้

CREATE TABLE <ชื่อตาราง>

(<ชื่อคอลัมน์ ประเภทของข้อมูล>[,<ชื่อคอลัมน์ ประเภทของข้อมูล>]....);

9.2.2 ประเภทของข้อมูล

ประเภทของข้อมูลแบ่งเป็น 5 ประเภทใหญ่ ๆ ขึ้นอยู่กับซอฟต์แวร์ระบบฐานข้อมูลที่ใช้ว่าคืออะไร ตัวอย่างเช่น CHAR, INTEGER, DATE ฯลฯ

คำสั่งการลบโครงสร้างตาราง

DROP TABLE <ชื่อตารางที่ต้องการลบ>

คำสั่งการเปลี่ยนแปลงโครงสร้างตาราง

ในกรณีที่ต้องการเปลี่ยนแปลงโครงสร้างตารางที่เคยนิยามไว้ สามารถใช้คำสั่งต่อไปนี้

ALTER TABLE <ชื่อตารางที่ต้องการเปลี่ยนแปลง>

<คำสั่งการเปลี่ยนแปลง><[,<ชื่อคอลัมน์ ประเภทของข้อมูล>]>

คำสั่งดัชนี

ดัชนี (INDEX) มีความสำคัญมากต่อฐานข้อมูลเชิงสัมพันธ์ เนื่องจาก ระบบฐานข้อมูลแบบรีเลชั่นเนล (RDBMS) จะใช้ดัชนีในการค้นหาระเบียนที่ต้องการได้อย่างรวดเร็ว โดยดัชนีที่ถูกสร้างขึ้น จะเก็บไว้แยกจากตารางในพื้นที่ต่างหาดของคอมพิวเตอร์ โดยปกติ ถ้าไม่มีการประกาศดัชนี ไว้การค้นหาข้อมูลในตาราง นั้นจะต้องทำแบบเรียงลำดับจากแถวที่หนึ่งจนถึงแถวสุดท้าย การสร้างดัชนีสำหรับตารางใด ๆ จะทำได้โดยการเลือกคอลัมน์ใดคอลัมน์หนึ่งจากตารางมาเป็นดัชนี และตารางหนึ่ง ๆ สามารถมีได้หลายดัชนี

นอกจากเพิ่มความรวดเร็วในการดึงข้อมูลแล้ว ยังสามารถนำไปใช้ในการควบคุมคอลัมน์ที่นำมาสร้างเป็นดัชนีให้มีการเก็บข้อมูลที่ไม่ซ้ำกัน(Unique) อีกด้วย

การสร้างดัชนีจะใช้คำสั่ง CREATE INDEX แล้วตามด้วยชื่อดัชนีที่เราตั้งขึ้น ดังรูปแบบต่อไปนี้

CREATE [UNIQUE] INDEX < ชื่อตารางที่ตั้งขึ้น >
ON (<ชื่อตารางที่สร้างดัชนี> (< ชื่อคอลัมน์_1> [,< ชื่อคอลัมน์_2>]…);

การลบดัชนี

เมื่อต้องการลบดัชนีที่สร้างขึ้น ก็สามารถทำได้ด้วยคำสั่ง DROP INDEX แล้วตามด้วยชื่อดัชนีที่ต้องการลบ ดังรูปแบบดังนี้

DROP INDEX <ชื่อดัชนี>

9.3 ภาษาสำหรับการจัดการข้อมูล (Data Manipulation Language-DML)

หลังจากที่เราสร้างโครงสร้างฐานข้อมูลขึ้นแล้ว คำสั่งต่อไปในการป้อนข้อมูลลงในฐานข้อมูลและเปลี่ยนแปลงข้อมูล ในฐานข้อมูล โดยการใช้ภาษาสำหรับการจัดการข้อมูล (Data Manipulation Language-DML) ใช้จัดการข้อมูลภายในตารางภายในฐานข้อมูล และภาษาแก้ไขเปลี่ยนแปลงตาราง แบ่งออกเป็น 4 Statement คือ

- Select Statement : การเรียกหา (Retrieve) ข้อมูลจาก ฐานข้อมูล
- Insert Statement : การเพิ่มเติมข้อมูลลงใน ตาราง (Table) จาก ฐานข้อมูล
- Delete Statement: การลบข้อมูลลงออกจาก ตาราง (Table) จาก ฐานข้อมูล
- Update Statement: การเปลี่ยนแปลงข้อมูลลงใน ตาราง (Table) จาก ฐานข้อมูล

คำสั่ง	ความหมาย
SELECT	เรียกค้นข้อมูลในตาราง
INSERT	เพิ่มแถวข้อมูลลงในตาราง
DELETE	ลบแถวข้อมูล
UPDATE	ปรับปรุงแถวข้อมูลในตาราง

คำสั่งค้นหาข้อมูล (Query Statement)

คำสั่ง SELECT เป็นคำสั่งการเรียกดูข้อมูล หรือ ค้นข้อมูล ตามเงื่อนไขที่ระบุ เนื่องจากคำสั่ง SELECT เป็นคำสั่งที่มีรูปแบบการใช้งานที่ง่ายเพื่อช่วยในการค้นหาข้อมูลที่ซับซ้อน ดังมีรูปแบบดังนี้

SELECT <ชื่อคอลัมน์ที่ต้องการดูข้อมูล>
FROM <ชื่อตาราง>
WHERE <เงื่อนไขตามที่ระบุ>

SELECT --- เป็นคำสั่งให้ทำการเรียกดูข้อมูลในคอลัมน์ที่ระบุ ซึ่งอาจจะมากกว่า หนึ่งก็ได้ และถ้ามีมากกว่าหนึ่งคอลัมน์ต้องคั่นด้วย คอมม่า (,) และนอกจากนี้ยังสามารถใช้เครื่องหมายดอกจัน (*) เพื่อแสดงถึงการขอดูข้อมูลทั้งหมดได้อีกด้วย

FROM --- เป็นคำส่วนประกอบของคำสั่งที่บอกถึงตารางที่ต้องการดู ซึ่งอาจจะมีมากกว่าหนึ่งตารางก็ได้ ที่จะถูกเรียกใช้จากคำสั่ง SELECT

WHERE--- เป็นส่วนประกอบของคำส่ง ที่ใช้บ่งบอกเงื่อนไขที่จะใช้ในการค้นหาข้อมูล ขึ้นมาจากตารางใด ๆ ที่อยู่หลัง FROM นี้

การเรียกดูแบบซ้อนกัน (Nested SELECT Statement)

```
SELECT <ชื่อคอลัมน์>
FROM   <ชื่อตาราง>
WHERE  <ชื่อคอลัมน์>  IN
            ( SELECT   <ชื่อคอลัมน์>
              FROM     <ชื่อตาราง>
              WHERE <ชื่อคอลัมน์> )
```

คำสั่งเติมข้อมูล (Insert Statement)

```
INSERT  INTO < ชื่อตาราง >
VALUES (< ชื่อคอลัมน์_1> [,< ชื่อคอลัมน์_2>]…);
```

คำสั่งแก้ไขและลบแถว (Update Statement)

```
UPDATE < ชื่อตาราง >
SET  <ค่าที่ต้องการ>
WHERE <เงื่อนไข>
```

9.4 ภาษาควบคุม (Control Language)

ใช้เป็นภาษาที่ใช้ควบคุมระบบรักษาความปลอดภัย ของฐานข้อมูล ประกอบด้วยคำสั่ง 2 คำสั่งคือ

- คำสั่ง GRANT เป็นคำสั่งที่ใช้กำหนดสิทธิให้กับผู้ใช้แต่ละคนให้มีสิทธกระทำการใดกับข้อมูลเช่น การเพิ่มข้อมูล การแก้ไข หรือ การลบข้อมูลในตารางใดบ้าง
- คำสั่ง REVOKE เป็นคำสั่งให้มีการยกเลิกสิทธนั้นหลังจากที่ได้ GRANT แล้ว

ค่าบูลลีน (Boolean-Type Data)

- AND
- OR
- NOT

- =
- > หรือ >=
- < หรือ =<
- <>
- Bulit-In Function
- COUNT
- SUM
- AVG
- MAX
- MIN
- DISTINCT

ชุดคำสั่ง GROUP BY, ORDER BY และ HAVING

เนื่องจากข้อมูลที่สนใจมักจะเป็น ผลสรุป หรือ ข้อมูลที่ใช้ในการตัดสินใจ ภาษา SQL จึงได้ออกแบบให้มี ชุดคำสั่งพิเศษ ที่ทำหน้าที่แบ่งออกเป็นกลุ่มๆ ได้แก่

1.1 ชุดคำสั่ง GROUP BY

GROUP BY เป็นคำสั่งให้มีการจัดกลุ่มแถวข้อมูลตามคอลัมน์

โดยข้อมูลที่เหมือนกันจะถูกจัดให้อยู่ในกลุ่มเดียวกัน ดังนั้นจากตัวอย่างตาราง S

```
SELECT  S# , MAX( QTY)
FROM    SPJ
GROUP BY  S#
```

ผลลัพธ์

S#	QTY
S1	700
S2	800
S3	500
S4	300
S5	800

1.2 ชุดคำสั่ง ORDER BY

ORDER BY เป็นคำสั่งให้มีการเรียงลำดับข้อมูลในแถวข้อมูลตามคอลัมน์

โดยข้อมูลที่เหมือนกันจะถูกจัดเรียงจากน้อยไปมาก ถ้าไม่การระบุบ แต่ถ้าระบุบว่าเป็น DESC จะเรียงจากมากไปน้อย ดังนั้นจากตัวอย่างต่อไปนี้

```
SELECT  S# , MAX( QTY)
FROM      SPJ
GROUP BY  S#
ORDER BY  2  DESC
```

S#	QTY
S5	800
S2	800
S1	700
S3	500
S4	300

หมายเหตุ ตัวเลข 2 ที่อยู่หลัง ORDER BY จะเป็น การบอกระบบให้ทำการเรียงลำดับข้อมูลตามลำดับคอลัมน์ที่ 2 ของคำสั่ง SQL ซึ่งก็คือ MAX(QTY) นั่นเอง

1.3 ชุดคำสั่ง HAVING

HAVING เหมือนกับคำสั่ง WHERE ที่ต้องตามด้วยเงื่อนไข แต่ HAVING จะใช้ในกรณีที่มีการจัดกลุ่มหรือการใช้ GROUP BY

สรุปชุดคำสั่งมาตรฐานของ SQL

1. CREATE TABLE [[database.]owner.]table_name
   (column_name datatype [not null | null] IDENTITY [(seed, increment)][constraint]
   [, column_name datatype [not null | null IDENTITY [(seed, increment)]]].
   [constraint] …)
   [ON segment name]
2. DELETE [FROM] table_name
   WHERE column_name = “value”
3. ALTER TABLE [[<database.>]<owner.>]<table_name>
   ADD <column_name><datatype>NULL [Canstsaint]
   [WITH nocmeck]
   {drop [ ]
4. UPDATE table_name
   SET column_name= value
   WHERE column_name =operator_value

5. CREATE VIEW View_name AS

```
SELECT column
FROM   table_1 ….
WHERE  table_key_1=table_key_2
```

หนังสืออ้างอิง

1. ชนวัฒน์ ศรีสอ้าน, **การออกแบบและพัฒนาฐานข้อมูล**, มหาวิทยาลัยเทคโนโลยีสุรนารี : นครราชสีมา, 2542.
2. ศิริลักษณ์ โรจนกิจอำนวย, ระบบ**ฐานข้อมูล**, พิมพ์ครั้งที่ 3, ดวงกลมสมัย : กรุงเทพฯ, 2542

แบบฝึกหัดท้ายบทที่ 9

จงใช้ภาษา SQL เขียนสมการคำตอบ ของ Schema ต่อไปนี้

STUDENT(StudentNumber, Name, Class, Major)

COURSE (CourseNumber, CourseName, CreditHour, Department)

PREREQUISITE(PrerequisiteNumber, CourseNumber)

SECTION(SectionIdentifier, CourseNumber, Semester, Year, Instructor)

GRADE_REPORT(**StudentNumber, SectionIdentifier**, Grade)

หมายเหตุ

Class ประกอบด้วย ปีที่1(Freshy) , ปีที่2 (Junior), ปีที่ 3(Sephormor) , ปีที่4(Senoir)

1.1) จงหา ชื่อของนักศึกษาชั้นปีที่ 4 (Senior) ที่มีสาขาเน้นหนัก 'IS'

1.2) ในทุก ๆ วิชาที่สอนด้วย อาจารย์ ดร. เก่ง จงหา รหัสวิชานั้น ๆ , เทอมการศึกษา (semester) , ปีการศึกษา(Year) และ จำนวนนักเรียนที่สอนในแต่ละsection

1.3) จงหาชื่อ และ Transcript ของแต่ละ Senoir Student ที่มี Class = 5 และ มี major = 'COSC' โดย Transcript ประกอบด้วย course name, course number, credit hours, semester, year และ grade สำหรับแต่ละ course ที่นักศึกษาลงทะเบียน

1.4) จงหา ชื่อของรายวิชา ที่สอนด้วย อาจารย์ ดร. เก่ง ในปี 2540 ถึง 2541

1.5) จงหา ชื่อของนักศึกษา และ major ที่มีผลการเรียน 4.00 ตลอดการเรียน

1.6) จงหา ชื่อ และ major ของ นักศึกษา ที่ไม่เคยได้เกรด A มาเลย ตลอดการเรียน

1.7) จงหา ชื่อของนักศึกษา ที่ลงทะเบียนเรียนวิชา 204-204

1.8) จงหา ชื่อของวิชา ที่เป็น Prerequisite ของวิชา Electrical Circuit I (205-100)

จากตารางต่อไปนี้ จงหาคำตอบ

SUPPLIER (S#, SNAME, STATUS, CITY)

S#	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

PART (P#,PNAME, COLOR, WEIGHT, CITY)

P#	PNAME	COLOR	WEIGHT	CITY
P1	Nut	Red	12	London
P2	Bolt	Green	17	Paris
P3	Screw	Blue	17	Rome
P4	Screw	Red	14	London
P5	Cam	Bule	12	Paris
P6	Cog	Red	19	London

ตาราง J

J#	JNAME	CITY
J1	Sorter	Paris
J2	Runch	Rome
J3	Reader	Athens
J4	Console	Athens
J5	Collator	London
J6	Terminal	Oslo
J7	Tape	**London**

ตาราง SPJ

S#	P#	J#	QTY
S1	P1	J1	200
S1	P1	J4	700
S2	P3	J1	400
S2	P3	J2	200
S2	P3	J3	200
S2	P3	J4	500
S2	P3	J5	600
S2	P3	J6	400
S2	P3	J7	800
S2	P5	J2	100
S3	P3	J1	200
S3	P4	J2	500
S4	P6	J3	300

S#	P#	J#	QTY
S4	P6	J7	300
S5	P2	J2	200
S5	P2	J4	100
S5	P5	J5	500
S5	P5	J7	100
S5	P6	J2	200
S5	P1	J4	100
S5	P3	J4	800
S5	P4	J4	800
S5	P5	J4	400
S5	P6	J4	500

- จงหาว่า มีคนส่งของชื่อใดบ้างที่ส่ง Nut
- จงหาว่า มีคนส่งของชื่อใดบ้างที่มีสถานะ (Status) อยู่ระหว่าง 5 ถึง 30
- จงหาว่า มีคนส่งของชื่อใดบ้างที่ส่ง Nut , Screw และ Bolt
- จงหาชื่อชิ้นส่วนทั้งหมดที่ส่งโดยคนส่งของ S4
- จงหาชื่อคนส่งของ(supplier) ที่ส่ง ชิ้นส่วนที่ผลิตในเมืองที่เขาอยู่
- จงหา ชื่อของชิ้นส่วน (PNAME) และ PNO ของชิ้นส่วนที่มีสีเขียว และ สีแดง
- จงหาชื่อของคนส่งของที่ส่งชิ้นส่วน P2 และ P4
- จงหาชื่อของคนส่งของที่ส่งชิ้นส่วน P2 หรือ P4
- จงหา ของคนส่งของที่ส่งชิ้นส่วน ที่มีสีเขียว หรือ สีแดง