

การพัฒนาต้นแบบของฐานข้อมูลแบบกระจายของระบบห้องสมุด

จากโมเดลที่ได้ในบทที่ 4 นั้น จะนำมาพัฒนาต้นแบบเพื่อทดสอบประสิทธิภาพของฐานข้อมูลที่ได้จากโมเดลว่าสามารถทำงานได้จริงตามเป้าหมายที่วางไว้ และหากพบข้อบกพร่องจะได้นำไปตัดแปลงแก้ไขโมเดลใหม่เพื่อให้สมบูรณ์ยิ่งขึ้น

รายละเอียดในการพัฒนาต้นแบบ

1. ทำการพัฒนาบนเครื่อง IBM Compatible 2 เครื่อง ดังนี้

1.1 SERVER Machine ซึ่งมีรายละเอียดดังนี้

1.1.1 ซีพียู 80486

1.1.2 ความถี่ 20 เมกะเฮิรท์

1.1.3 หน่วยความจำ 12 เมกะไบต์

1.1.4 ความจุฮาร์ดดิสก์ 94 เมกะไบต์

1.1.5 ดิสก์ไดรฟ์ขนาด 5.25 นิ้ว 1.2 เมกะไบต์ จำนวน 1 ไดรฟ์

1.1.6 ดิสก์ไดรฟ์ขนาด 3.5 นิ้ว 1.44 เมกะไบต์ จำนวน 1 ไดรฟ์

1.2 CLIENT Machine ดังมีรายละเอียดดังนี้

1.2.1 ซีพียู 80386

1.2.2 ความถี่ 21 เมกะเฮิรท์

1.2.3 หน่วยความจำ 8 เมกะไบต์

1.2.4 ความจุฮาร์ดดิสก์ 80 เมกะไบต์

1.2.5 ดิสก์ไดรฟ์ขนาด 5.25 นิ้ว 1.2 เมกะไบต์ จำนวน 1 ไดรฟ์

1.2.6 ดิสก์ไดรฟ์ขนาด 3.5 นิ้ว 1.44 เมกะไบต์ จำนวน 1 ไดรฟ์

2. พัฒนากายใต้ระบบปฏิบัติการเครือข่ายของโนเวล เนตแวร์ เวอร์ชัน 3.11 ซึ่ง

ใช้โทโปโลยีแบบรูปดาว ทำงานภายใต้ระบบปฏิบัติการของ DOS V.3.31

3. พัฒนาภายใต้ระบบจัดการฐานข้อมูลแบบรีเลชันนัลของออราเคิล 2 ชนิด คือ

3.1 SERVER Machine จะใช้ ORACLE Server for Netware 386 ซึ่งประกอบด้วย

3.1.1 ORACLE RDBMS V6.0.34.3.0, with Transaction Processing Option and PL/SQL V1.0.34.0.0

3.1.2 SQL\*Net V1.1

3.2 CLIENT Machine จะใช้ ORACLE RDBMS V6.0.31.2.2, with Transaction Processing Option and PL/SQL V1.0.32.3.1 ประกอบด้วยโปรแกรมอรรถประโยชน์ที่นำมาใช้ดังนี้

3.2.1 เอสคิวแอลพลัส (SQL\*PLUS V3.0.10.1.4)

3.2.2 เอสคิวแอลฟอร์ม (SQL\*Forms (Design) V3.0.16.9.5)

#### การพัฒนาต้นแบบฐานข้อมูล



จากโมเดลเชิงกายภาพที่ได้ในบทที่ 4 ได้ถูกนำมาทดลองสร้างฐานข้อมูลแบบรีเลชันนัล โดยมีขั้นตอนดังนี้

1. การสร้างตารางตามโมเดลข้อมูลเชิงกายภาพ โดยให้ชื่อเอนติตี้ เป็นชื่อตารางรีเลชัน ซึ่งมีรูปแบบดังนี้

```
CREATE TABLE table(column_spec[NOT NULL],...);
```

นอกจากนี้ยังต้องมอบสิทธิ์ให้กับผู้ใช้ต่าง ๆ โดยมีรูปแบบคำสั่งดังนี้

```
GRANT {privilege,privilege,...!ALL}
```

```
ON table TO {user,usr,..!PUBLIC} [WITH GRANT OPTION];
```

2. นำเอาชื่อของแอตทริบิวมาเป็นชื่อของคอลัมภ์ของตารางรีเลชันนัล โดยที่แอตทริบิวที่เป็นคีย์หลัก จะทำการระบุไว้เลยว่าห้ามเป็นนัล ตั้งแต่ตอนสร้างตาราง

3. สร้างดรรชนีสำหรับแต่ละตารางเพื่อให้มีความเป็นหนึ่งเดียว โดยมีรูปแบบคำสั่งดังนี้

```
CREATE [UNIQUE] INDEX name
ON table (column[ASC|DESC],column[ASC|DESC],...);
```

4. ในการพัฒนาเมนูโดยใช้โปรแกรมเอสคิวแอลฟอร์มนั้น ถ้าเป็นเงื่อนไขที่ซับซ้อนหรือมีการใช้บ่อยเราสามารถสร้างวิวซึ่งเหมือนเป็นตารางสมมติขึ้นไว้ก่อนได้โดยใช้คำสั่งรูปแบบดังนี้

```
CREATE VIEW name [(alias,alias,...)] AS query
[WITH CHECK OPTION];
```

5. การสร้างลิงค์ระหว่างสถานี เพื่อให้ผู้ใช้จากสถานีท้องถิ่น สามารถเรียกใช้ข้อมูลจากสถานีอื่น ๆ ได้สะดวกขึ้น โดยที่ผู้ใช้ท้องถิ่น จะต้องมียุทธศาสตร์และสิทธิการใช้ทรัพยากรบนสถานีที่ต้องการอยู่ด้วย โดยใช้คำสั่งดังนี้

```
CREATE [PUBLIC] DATABASE LINK link
[CONNECT TO user IDENTIFIED BY password]
USING 'SQL*Net_string'
```

ตัวอย่างเช่น สมมุติมีการสร้างลิงค์ชื่อ CUMAIN ไว้ที่สถานีของหอสมุดกลาง สถาบันวิทยบริการ ดังนี้

```
CREATE PUBLIC DATABASE LINK CUMAIN
USING 'X:ORASRV'
```

หลังจากนี้ ผู้ใช้จากสถานีเจ้าของสิ่งดังกล่าว หากต้องการเรียกข้อมูลจากสถานีหอสมุดกลาง ก็  
สามารถทำได้ง่าย ด้วยคำสั่งดังนี้

```
SELECT * FROM TITLE@CUMAIN
```

โดยหากไม่ทำการสร้างลิงค์แล้ว ผู้ใช้จะต้องรู้ถึงที่อยู่ของสถานีนั้น ๆ เช่น จากคำสั่งข้างต้น จะ  
ได้คำสั่งดังนี้

```
SELECT * FROM TITLE@X:ORASRV
```

6. สร้างลำดับที่ (Sequence) ขึ้น 2 ตัว เพื่อใช้แทนคอลัมภ์ เลขอ้างอิงหนังสือ  
และเลขทะเบียนหนังสือ ดังนี้

```
CREATE SEQUENCE [user.]sequence
  [INCREMENT BY [1 / n]]
  [START WITH n]
  [ (MAXVALUE n / NOMAXVALUE) ]
  [ (MINVALUE n / NOMINVALUE) ]
  [ (CYCLE / NOCYCLE) ]
  [ (CACHE (n / 20) / NOCACHE) ]
  [ (ORDER / NOORDER) ]
```

### การรักษาความปลอดภัยของระบบ

เรื่องของการรักษาความปลอดภัย ในระบบต่าง ๆ จะมีวิธีการรักษาความปลอดภัยหลายวิธี  
ให้เลือกใช้ ซึ่งโดยมากจะไม่มีวิธีใดที่สามารถรักษาความปลอดภัยได้โดยสมบูรณ์ สำหรับการพัฒนา  
ต้นแบบฐานข้อมูลแบบกระจายในครั้งนี้ มีวิธีการรักษาความปลอดภัยให้เลือกใช้ได้ดังนี้

1. การรักษาความปลอดภัยในระดับระบบปฏิบัติการเครือข่าย โดยการกำหนดรหัสผ่าน  
และสิทธิการมองเห็นข้อมูลต่าง ๆ ของผู้ใช้

2. การรักษาความปลอดภัยในระดับฐานข้อมูล ในระบบจัดการฐานข้อมูลออราเคิล สามารถกำหนดสิทธิให้แก่ผู้ใช้ฐานข้อมูลได้ดังนี้

2.1 ให้สิทธิผู้ใช้ฐานข้อมูลในการเข้าถึง เพื่อการอ่านข้อมูลในตารางต่างๆ ได้ อย่างเดียวเท่านั้น ไม่อนุญาตให้สร้างตาราง แต่อนุญาตให้สร้างวิวได้ (ให้สิทธิการ CONNECT)

2.2 ให้สิทธิในการสร้างตาราง ครรชนี้ รวมถึงความสามารถในการอนุญาต ให้ผู้ใช้คนอื่น ๆ มีสิทธิต่าง ๆ กับตารางของผู้สร้างได้ (ให้สิทธิ RESOURCE)

2.3 ให้สิทธิในการเป็นผู้บริหารฐานข้อมูล (ให้สิทธิเป็น DBA)

3. ความปลอดภัยในระดับตาราง ตามปกติผู้สร้างตารางเท่านั้น จึงจะเป็นเจ้าของ ตาราง และจะเป็นผู้มีสิทธิให้ผู้อื่นสามารถเข้าถึง และแก้ไขข้อมูลในตารางของตนได้ ความปลอดภัย ในระดับตารางนี้ แบ่งออกเป็น

3.1 การสร้างวิวให้กับผู้ใช้ต่าง ๆ สมมติให้ผู้ใช้คนหนึ่งทำหน้าที่เป็นผู้บริหาร ฐานข้อมูล ซึ่งจะทำหน้าที่ในการสร้างตารางต่าง ๆ ทุกตาราง แล้วจึงทำการสร้างวิวให้กับผู้ใช้ ในระดับต่าง ๆ โดยกำหนดสิทธิตามสมควร

3.2 แบ่งผู้ใช้ออกเป็น 3 ระดับ คือ ระดับผู้บริหาร ระดับปฏิบัติการและระดับผู้ใช้ ทั่วไป โดยจะให้สิทธิในการดูข้อมูลด้านการบริหาร เช่น ข้อมูลด้านการเงิน ข้อมูลด้านการบริหารงาน แก่ผู้ใช้ระดับบริหารเท่านั้น ให้สิทธิในการปรับปรุงแก้ไขฐานข้อมูลแก่ผู้ปฏิบัติงาน และให้สิทธิในการ เรียกดู เฉพาะข้อมูลทั่วไป แก่ผู้ใช้ธรรมดา โดยผู้บริหารฐานข้อมูลจะต้องสร้างวิวขึ้นมา แล้วกำหนด สิทธิให้เหมาะสม เช่นการสร้างวิวให้กับผู้ใช้ฐานข้อมูลที่มีรหัสผู้ใช้ในระบบว่า SCOTT และ MANAGER หมายความว่า ผู้ใช้ซึ่งมีรหัสว่า SCOTT และ MANAGER เท่านั้นที่มีสิทธิในการดูข้อมูล จากวิวนี้ สามารถทำได้ดังนี้

```
CREATE VIEW VAUTHOR
```

```
AS (SELECT * FROM AUTHOR /* Local database */
```

```
UNION
```

```
(SELECT * FROM AUTHOR@CUMAIN) /* remote database */
```

และคำสั่ง



GRANT SELECT ON VAUTHOR TO SCOTT,MANAGER

เป็นการให้สิทธิในการเข้าถึงเพื่อดูวิวที่สร้างไว้ข้างต้น แก่ผู้ใช้ในระบบที่ชื่อ SCOTT และ MANAGER

และ  
GRANT SELECT, INSERT, UPDATE, DELETE  
ON AUTHOR TO SCOTT

คำสั่งนี้เป็น การให้สิทธิในการเข้าถึงเพื่อดู เพิ่ม ปรับปรุง หรือลบข้อมูลให้แก่  
ผู้ใช้ชื่อว่า SCOTT

4. ความปลอดภัยในระดับโปรแกรม ได้ทำการตรวจสอบผู้ใช้ก่อนที่จะเข้าสู่เมนูของ หน้าจอต่าง ๆ ที่ทำไว้ เพื่อให้สามารถเพิ่มเติม,ดูหรือแก้ไขได้เฉพาะข้อมูลของตนมีสิทธิ โดยใส่ไว้ได้ในทริกเกอร์ต่าง ๆ ในจุดที่ต้องการตรวจสอบ เป็นสิ่งที่โปรแกรมเอสคิวแอลฟอร์มจัดเตรียมไว้ให้ เช่น ในฟอร์มการแก้ไขข้อมูลบรรณานุกรม จะใส่ทริกเกอร์ให้ตรวจสอบรหัสของผู้ที่แก้ไขว่า เป็นส่วนหนึ่งของเจ้าหน้าที่หรือไม่ ดังนี้

```
SELECT 'x' FROM dual
WHERE :blk1.memberid IN
(SELECT member_id
FROM member
WHERE member_type = 'S')
```

#### การทดสอบและผลการทดสอบ

การทดสอบต้นแบบในการวิจัยนี้ ได้นำข้อมูลเข้าโดยการจำลองฐานข้อมูล จากระบบ บรรณานุกรมหนังสือ ของสถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย และสมมุติข้อมูลขึ้นเอง ในบางส่วนเพื่อให้สามารถทดสอบเงื่อนไขที่ต้องการได้ เช่น รายชื่อสมาชิก หลังจากใส่ข้อมูลพื้นฐาน

เรียบร้อยแล้วจึงได้ทำการทดสอบเมนูทั้งหมด ผลการทดสอบอยู่ในเกณฑ์ที่น่าพึงพอใจ สามารถตอบปัญหาที่ต้องการได้ทั้งในส่วนของบริษัทบรรณานุกรม และในส่วนที่เชื่อมกัน ในส่วนการดึงข้อมูลนั้น ไม่มีปัญหาใด ๆ เพราะสามารถที่จะจำกัดขอบเขต ให้ดึงมาเฉพาะส่วนที่ต้องการได้ แต่สำหรับการเพิ่มและแก้ไขข้อมูลบางครั้งจะพบความยุ่งยาก เนื่องจากข้อมูลถูกกระจายอยู่ตามสถานีต่าง ๆ และข้อจำกัดของระบบจัดการฐานข้อมูลของออราเคิลเอง ที่ไม่สามารถจัดการกับข้อมูลที่กระจายอยู่ตามสถานีอื่น ๆ ได้ด้วยการใช้คำสั่ง UPDATE เพียงคำสั่งเดียว หรือความยุ่งยากของระบบเอง เช่น ถ้ามีการแก้ไขตรงจุดหนึ่งต้องตามไปแก้ไขที่จุดอื่นด้วย นั่นคือระบบการจัดการฐานข้อมูลของออราเคิล ไม่สนับสนุนการทำงานแบบ LOCATION TRANSPARENCY และ FRAGMENTATION TRANSPARENCY