



บทที่ 5

การพัฒนาดัชนีแบบฐานข้อมูลของบุคลากร

สภาพแวดล้อมในการพัฒนาดัชนีแบบฐานข้อมูล

ในการพัฒนาระบบฐานข้อมูลได้พัฒนาบนเครื่องซัน (Sun) ภายใต้ระบบปฏิบัติการโซลาริส 2.3 (Solaris version 2.3) โดยใช้ระบบจัดการฐานข้อมูลเชิงสัมพันธ์ของออราเคิล (ORACLE RDBMS version 7.2) และเพื่อให้ใช้โปรแกรมอรรถประโยชน์ของออราเคิล และส่วนฟรอนท์เอนด์ (Front-End) ได้ใช้ไมโครซอฟท์แอคเซส (MSACCESS version 2.0)

การพัฒนาดัชนีแบบฐานข้อมูล

ดัชนีแบบฐานข้อมูลถูกสร้าง ภายหลังจากการออกแบบแบบจำลองข้อมูล โดยการใช้นำโปรแกรมอรรถประโยชน์ต่าง ๆ ของออราเคิล ดังนี้

1. SQL*Plus version 3.0.6.5.1 แบ่งประโยคที่ใช้ออกเป็น 4 พวก คือ
 - 1.1 ประโยคสอบถาม (Quires) เพื่อเรียกดูข้อมูล ที่ต้องการ โดยมีรูปแบบคำสั่งดังนี้

```
SELECT [DISTINCT] {*[user.]select-column[expr|literal] [alias]
FROM [(user.)table[alias] [, user.]table[alias]] .....]
[WHERE clause][AND clause][OR clause]
[GROUP BY clause][HAVING clause]
[ORDER BY clause [ASC | DESC]]
```

- 1.2 ประโยคจัดการข้อมูล (Data Manipulation Statements) ใช้ในการเปลี่ยนแปลงข้อมูล ดังนี้

การเพิ่มของข้อมูลในตาราง มีรูปแบบคำสั่งดังนี้

```
INSERT INTO [user.]table-name [(column-name
    [,column-name...])]
```

```
VALUES (value[,value...]) | query-block
```

ในที่นี้มี ใช้คำสั่งนี้ ในการเพิ่มข้อมูลในตารางสืบค้นต่าง ๆ
การแก้ไขค่าของคอลัมน์ มีรูปแบบคำสั่งดังนี้

```
UPDATE [user.]table-name
```

```
SET {[column-name = value
```

```
    [,column-name = value,...]} |
```

```
    [column[,column]...] = (query-block[,...])}
```

```
[ WHERE clause]
```

การลบแถวของข้อมูลจากตาราง มีรูปแบบคำสั่งดังนี้

```
DELETE FROM [user.]table-name[alias]
```

```
[WHERE clause]
```

นอกจากนี้คำสั่ง COMMIT [WORK] และ ROLLBACK [WORK] ก็เป็น
ประโยคจัดการข้อมูลที่ใช้ด้วยเหมือนกัน

1.3 ประโยคจำกัดความข้อมูล (Data Definition Statements)

ประกอบด้วยคำสั่งในการสร้าง คำสั่งที่ใช้ได้แก่

คำสั่งในการสร้างตาราง มีรูปแบบคำสั่งดังนี้

```
CREATE TABLE [user.]table-name
```

```
    [(column-name data-type|table-constraint]
```

```
    [(column-name data-type|table-constraint]...))
```

ในที่นี้ในการสร้างตารางตามแบบจำลองข้อมูลเชิงกายภาพจะใช้คำสั่งนี้
และใช้ชื่อตารางตามชื่อเอนทิตีที่เป็นภาษาอังกฤษ ในภาคผนวก ข

คำสั่งในการสร้างวิว มีรูปแบบคำสั่งดังนี้

```
CREATE VIEW [user.]view-name[(alias[,alias]...)]
```

```
AS query-block
```

```
[WITH CHECK OPTION [CONSTRAINT constraint]]
```

การใส่ WITH CHECK OPTION เป็นวิธีการหนึ่งในการตรวจสอบความถูกต้อง
ของข้อมูลในการเพิ่ม หรือการแก้ไขข้อมูลผ่านวิวได้ และยังสามารถบังคับให้เป็นไปตาม
กฎความเป็นบูรณาภาพของข้อมูลได้ด้วย ซึ่งจะใช้ในการกำหนดความปลอดภัย

คำสั่งในการสร้างเลขลำดับ มีรูปแบบคำสั่งดังนี้

```
CREATE SEQUENCE [user.]sequence
```

INCREMENT BY {1|INTEGER}}

[START WITH integer]

คอลัมน์ในบางตาราง เช่น รหัสผลงานวิชาการ เป็นต้น มีการกำหนดให้มีค่าเป็นตัวเลขเรียงลำดับ ดังนั้น จึงใช้คำสั่งดังนี้เพื่อกำหนดลำดับ ซึ่งเมื่อมีการเพิ่มเรคคอร์ดในตารางก็จะเพิ่มค่าเลขลำดับตามที่กำหนดไว้โดยอัตโนมัติ เป็นตัวเลขจำนวนเต็ม และตัวเลขเหล่านี้จะเป็นเอกลักษณ์

คำสั่งในการสร้างดัชนี มีรูปแบบคำสั่งดังนี้

CREATE [UNIQUE] INDEX index-name

(ON table-name (column-name [ASC | desc])

[(column-name [ASC | desc])...])

ในที่นี้ สร้างดัชนีบนคอลัมน์ที่เป็นคีย์หลักในตารางที่สำคัญ

คำสั่งในการสร้างคำเหมือน (synonym) มีรูปแบบคำสั่งดังนี้

CREATE [PUBLIC]SYNONYM [user.]synonym-name

FOR [user.]table-name | view-name

เนื่องจากใช้ตารางซึ่งได้รับอนุญาตมาจากมุมมองต่างๆของตารางจริง จึงต้องมีการสร้างคำเหมือนเพื่อสะดวกในการอ้างถึง

คำสั่งในการแก้ไขตาราง มีรูปแบบคำสั่งดังนี้

ALTER TABLE [user.]table-name

[ADD ((column-name data-type)

table-constraint)

[(column-name data-type)

table-constraint],...])

[MODIFY (column-name data-type

[,column-name data-type,...])

คำสั่งในการ drop ตาราง, วิว และดัชนี มีรูปแบบคำสั่ง ดังนี้

DROP [INDEX [user.]index-name] |

[TABLE [user.]table-name] |

[VIEW [user.]view-name]

1.4 ประโยคควบคุมข้อมูล (Data Control Statements)

ควบคุมการเข้าถึงฐานข้อมูล (Database Level Security) ได้แก่ คำสั่งในการอนุญาต มีรูปแบบคำสั่งดังนี้

GRANT {CONNECT | RESOURCE | DBA}

TO userid1[,userid2,...]

[IDENTIFIED BY password1[,password2,...]]

ซึ่งการกำหนดสิทธิ์ให้ CONNECT หมายถึงยอมให้มีการเข้าถึง และอ่านตาราง รวมทั้งสามารถใช้คำสั่งดีเอ็มแอล ในการทำงานกับตารางที่ตนมีสิทธิ์ และยอมให้สามารถสร้างวิและค่าเหมือนได้ แต่ไม่สามารถสร้างตาราง คลัสเตอร์ และดัชนี แต่ถ้าได้รับสิทธิ์ให้ใช้ RESOURCE สามารถสร้างตาราง ดัชนี และคลัสเตอร์ รวมทั้งอนุญาตผู้อื่นในการใช้ของตน และสามารถตรวจสอบผู้ปฏิบัติงานนั้นได้ (auditability) สำหรับการได้สิทธิ์เป็น DBA นั้นจะสามารถใช้สิทธิ์ในการเป็นผู้บริหารฐานข้อมูล

ส่วนคำสั่งถอนการอนุญาต มีรูปแบบคำสั่งดังนี้

REVOKE {CONNECT | RESOURCE | DBA}

FROM userid

ควบคุมการเข้าถึงข้อมูลในฐานข้อมูล (Table Level Security) ได้แก่ คำสั่งในการอนุญาต มีรูปแบบคำสั่งดังนี้

GRANT {privilege | ALL}

ON { table-name | view-name }

TO {userid | PUBLIC [,user]...}

[WITH GRANT OPTION]

privilege := SELECT, INSERT, DELETE, UPDATE [(column-name)], ALTER, INDEX

โดยที่ WITH GRANT OPTION เป็นการยอมให้ผู้อื่นเห็น, เปลี่ยนแปลง และใช้ข้อมูลในตาราง รวมทั้งสามารถส่งผ่านสิทธิ์ให้ผู้อื่นต่อไปได้อีก ส่วนคำสั่งถอนการอนุญาต มีรูปแบบคำสั่งดังนี้

REVOKE {privilege | ALL}

ON {table-name | view-name}

FROM {userid | PUBLIC}

แต่เมื่อพัฒนาไประยะหนึ่งจึงได้นำโปรแกรมอีอาวิน (E-R WIN) มาช่วยในการจัดการกับตารางต่าง ๆ

โดยจากแบบจำลองข้อมูลจะสามารถสร้างตารางต่าง ๆ ได้ดังนี้
(ดูรายละเอียดในการสร้างตารางทุกตารางดูได้ในภาคผนวก ข)

ตารางที่ 5.1 เอนทิตี ความหมาย และตาราง

ชื่อเอนทิตี	ความหมาย	ชื่อตาราง
ABS_ANUAL	การลาปกติ	ABS_ANUAL
ABS_EDUCATION	การลาศึกษา	ABS_EDUCATION
ABS_QUOTAS	จำนวนกำหนดวันลาสูงสุด	ABS_QUOTAS
ABS_TYPE	ชนิดการลา	ABS_TYPE
ADDRESS	ที่อยู่	ADDRESS
CHILD	บุตร	CHILD
CMD_TYPE	ชนิดคำสั่ง	CMD_TYPE
COMMAND	คำสั่ง	COMMAND
CONTACT	ผู้ติดต่อ	CONTACT
COUNTRY	ประเทศ	COUNTRY
CURRICULUM	หลักสูตรการศึกษา	CURRICULUM
DEPARTMENT	ภาควิชาหรือเทียบเท่า	DEPARTMENT
DIVISION	แผนกหรือเทียบเท่า	DIVISION
DUTY	หน้าที่ตามการจ้างงาน	DUTY
EDUCATION	การศึกษาของบุคลากร	EDUCATION
EDUCATION_LEVEL	ระดับการศึกษา	EDUCATION_LEVEL
EXT_POSITION	ตำแหน่งภายนอกองค์กร	EXT_POSITION
FACULTY	คณะหรือเทียบเท่า	FACULTY
FUND	ทุน	FUND
MARRITAL_STATUS	สถานะภาพสมรส	MARRITAL_STATUS
OLD_NAME	ชื่อเดิม	OLD_NAME
ORGANIZATION	องค์กร	ORGANIZATION
PARENT	บิดามารดา	PARENT
PC_LEVEL	ระดับ	PC_LEVEL
PERSON	บุคลากร	PERSON
PERSON_POSITION	ตำแหน่งของบุคลากร	PERSON_POSITION

ตารางที่ 5.1 เอนทิตี ความหมาย และตาราง(ต่อ)

ชื่อเอนทิตี	ความหมาย	ชื่อตาราง
PERSON_PROJECT	ผลงานทางวิชาการของบุคลากร	PERSON_PROJECT
PERSON_RESEARCH	งานวิจัยของบุคลากร	PERSON_RESEARCH
PICTURE	รูป	PICTURE
POSITION	อัตราเลขที่	POSITION
POSITION_STATUS	สถานะของอัตราเลขที่	POSITION_STATUS
POSITION_TYPE	ตำแหน่ง	POSITION_TYPE
PRE_NAME	คำนำหน้าชื่อ	PRE_NAME
PROFESSION	อาชีพ	PROFESSION
PROJECT	ผลงานทางวิชาการ	PROJECT
PROJECT_TYPE	ชนิดของผลงาน	PROJECT_TYPE
PROMOTION_TYPE	ชนิดการเลื่อนขั้น	PROMOTION_TYPE
PROVINCE	จังหวัด	PROVINCE
PUNISHMENT	การลงโทษ	PUNISHMENT
PUNISH_TYPE	ชนิดการลงโทษ	PUNISH_TYPE
RECEPT_REWARD	เครื่องราชย์ที่ได้รับ	RECEPT_REWARD
RELATION	ความสัมพันธ์	RELATION
RELIGION	ศาสนา	RELIGION
RESEARCH	งานวิจัย	RESEARCH
RESPONCE	หน้าที่ที่รับมอบหมายเพิ่ม	RESPONCE
RES_FINISH_STATUS	สถานะการปิดงานวิจัย	RES_FINISH_STATUS
REWARD	เครื่องราชย์	REWARD
SALARY_GRADE	ชั้นเงินเดือน	SALARY_GRADE
SPOUSE	คู่สมรส	SPOUSE
SPOUSE_STATUS	สถานะคู่สมรส	SPOUSE_STATUS
TERMINATION	การออกจากราชการ	TERMINATION
TERM_REASON	เหตุผลการออก	TERM_REASON

ตารางที่ 5.1 เอนทิตี ความหมาย และตาราง(ต่อ)

ชื่อเอนทิตี	ความหมาย	ชื่อตาราง
TRAIN_VISIT	การพัฒนา	TRAIN_VISIT
TRAIN_VISIT_TYPE	ชนิดการพัฒนา	TRAIN_VISIT_TYPE
WORK_TYPE	ชนิดงานที่ได้รับมอบหมาย	WORK_TYPE

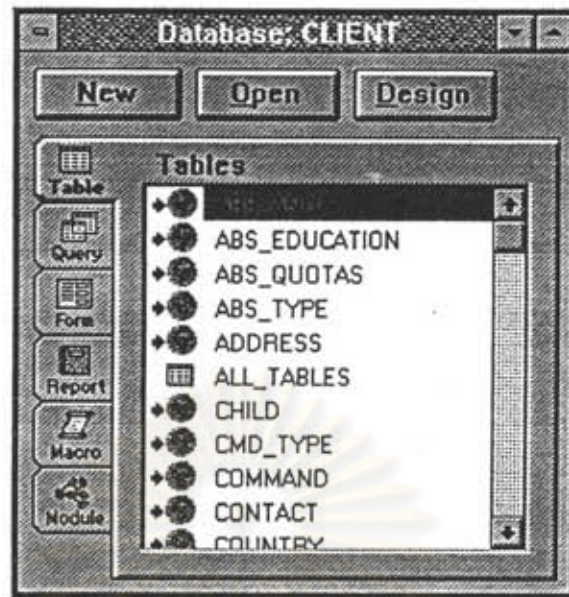
การพัฒนาส่วนฟรอนท์เอนด์

การพัฒนาส่วนฟรอนท์เอนด์นั้น ทำการพัฒนาบนระบบคอมพิวเตอร์ส่วนบุคคลที่ใช้งานระบบวินโดวส์ ซึ่งทำให้การแสดงผลเป็นไปอย่างสวยงามและง่ายต่อผู้ใช้ โดยได้ใช้โปรแกรมไมโครซอฟท์แอคเซสเพื่อพัฒนาโปรแกรมประยุกต์ ซึ่งไมโครซอฟท์แอคเซสนั้นมีการใช้งานดังต่อไปนี้

ตัวประสานผู้ใช้ไมโครซอฟท์แอคเซส

ตัวประสานผู้ใช้ไมโครซอฟท์แอคเซส (Microsoft Access User Interface) เป็นส่วนที่ใช้งานและเป็นที่ยึดของผู้ใช้ จะถูกจัดเก็บในที่บรรจุฐานข้อมูล (Database Container) ซึ่งจะประกอบด้วย 6 ส่วนใหญ่ๆดังนี้

1. ตาราง (Tables)
2. ข้อคำถาม (Queries)
3. ฟอร์ม (Forms)
4. รายงาน (Reports)
5. แมโคร (Macros)
6. มอดูล (Modules)



รูปที่ 5.1 แสดงตัวประสานผู้ใช้ของแอสเสส

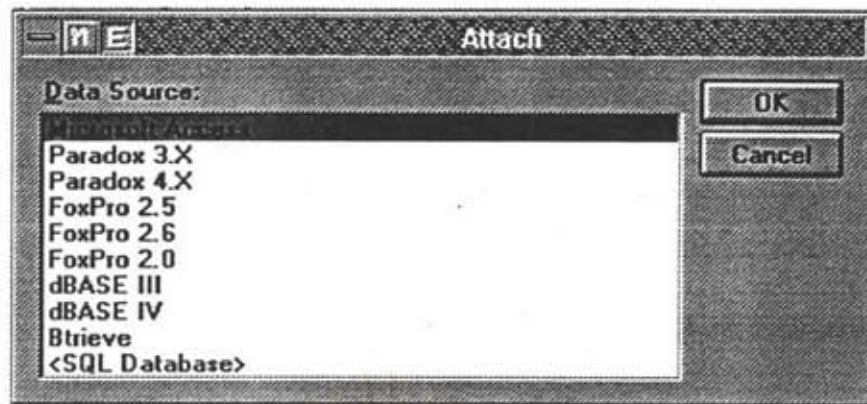
โดยทั่วไปแล้วถ้าส่วนของตารางไม่มี ตารางแบบดึงมา(Attached Table)แล้ว ข้อมูลทั้งหมดจะถูกเก็บไว้ในแฟ้มข้อมูลเพียงแฟ้มข้อมูลเดียวคือแฟ้มข้อมูลที่มีนามสกุล .MDB

ตาราง

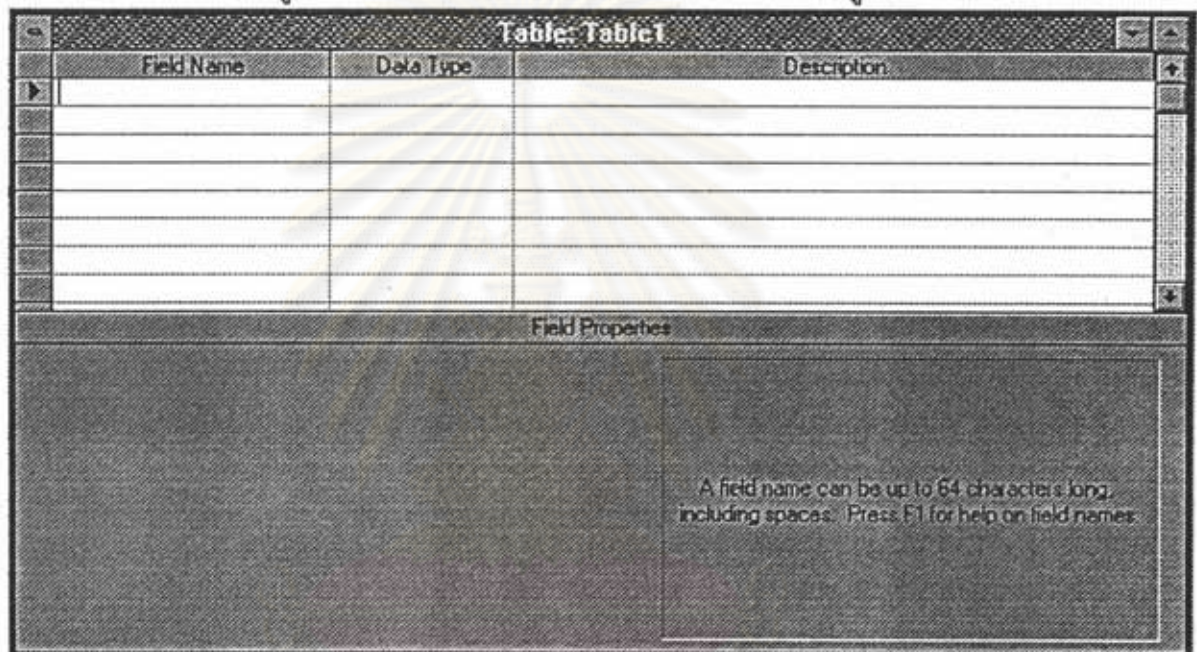
ในส่วนนี้จะเป็นการสร้างตารางของฐานข้อมูลขึ้นมา เพื่อเป็นการกำหนดว่าในแต่ละตารางมีเขต (Field) ไต่บ้าง และเป็นข้อมูลประเภทไหนเป็นต้น การสร้างตารางอาจทำได้ 2 วิธีคือ การทำตารางแบบดึงมาและการสร้างตารางในแอสเสสเอง การสร้างตารางทั้งสองวิธีนี้มีความแตกต่างกันดังนี้

1. การสร้างตารางโดยการดึงมาจะเป็นการสร้างตารางขึ้นมาเพื่อเป็นการอ้างอิงข้อมูลมาจากแหล่งข้อมูล(Data Source)อื่น เช่น ออราเคิล(Oracle) อินเกรส(Ingress) ฟอกซ์โปร(FoxPro) ดีเบส(DBase)เป็นต้น ดังนั้นตัวตารางจะเป็นลักษณะของการลอกข้อมูลรวมถึงรูปแบบของตารางด้วย ทำให้ไม่สามารถที่จะทำการเปลี่ยนแปลงประเภทของข้อมูลและรวมถึงค่าคุณสมบัติ(Property) บางตัวด้วย แต่ยังสามารถทำการแก้ไขและเพิ่มข้อมูลในตารางนั้นได้

ส่วนการสร้างตารางในแอสเสสเองนั้นจะเป็นการสร้างตารางขึ้นมาในตัวที่บรรจุฐานข้อมูล(Database Container) ในตัวแฟ้มข้อมูลของแอสเสสทำให้สามารถทำการแก้ไขและเปลี่ยนแปลงข้อมูลและประเภทของข้อมูลได้



รูปที่ 5.2 แสดงหน้าต่างการเลือกชนิดแหล่งข้อมูล



รูปที่ 5.3 แสดงรายละเอียดตาราง

2. การสร้างความสัมพันธ์ของตารางนั้นจะทำเฉพาะส่วนที่เป็นการสร้างตารางใน แอคเสสหรือตารางในแอคเสสกับตัวตารางแบบดั้งเดิม โดยมากส่วนที่เป็นตัวตารางแบบดั้งเดิมด้วยกันจะทำความสัมพันธ์มาจากตัวแหล่งข้อมูลมาแล้ว

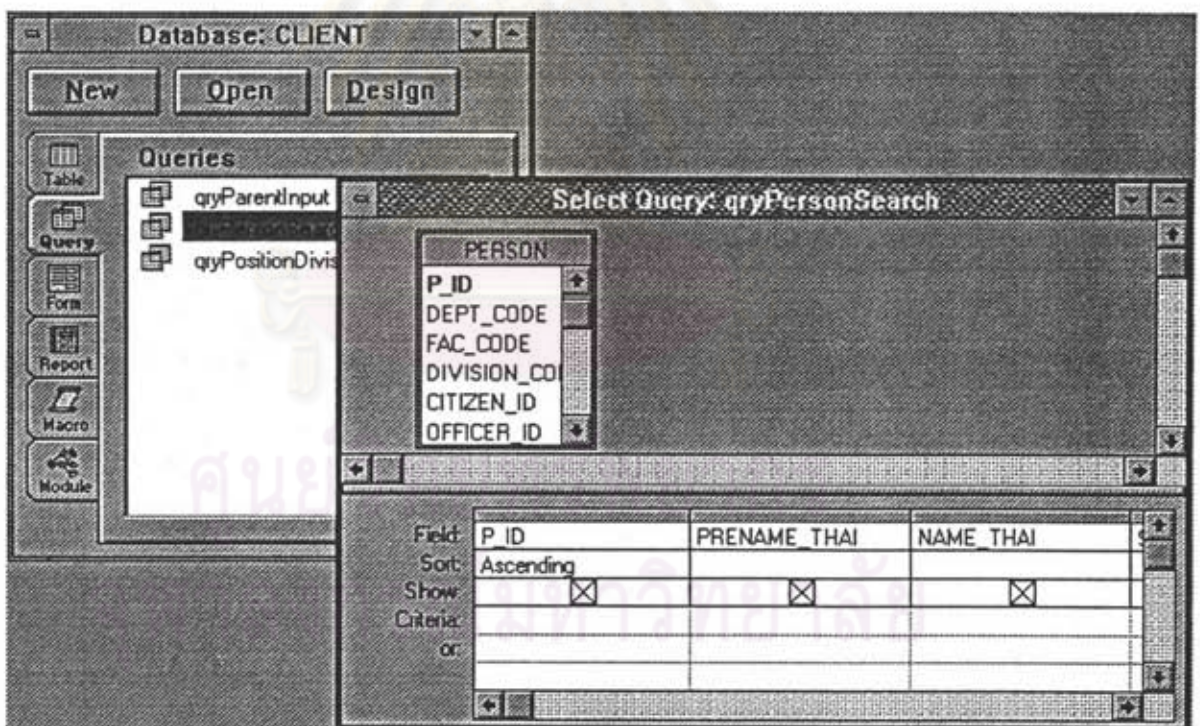
ข้อคำถาม

ในส่วนนี้จะเป็นการสร้างข้อคำถามโดยที่ตัวแอคเสสจะสร้างคำสั่ง SELECT ของ เอสคิวเอล เพื่อนำไปใช้ในการดึงข้อมูลของตารางที่สร้างไว้ ซึ่งจะทำการดึงข้อมูลเพื่อจุด

ประสงค์ต่างกัน เช่น เรียกข้อมูลที่เป็นตำแหน่งของบุคคลากร หรือการผลงานของบุคคลากร เป็นต้น

การสร้างข้อความของแอสเสสนั้น จะเป็นลักษณะแบบตั้งค่าโดยตัวอย่าง(Query By Example) นิยมย่อว่าคิวบีอี(QBE) ที่เป็นแบบกราฟิก ซึ่งจะช่วยให้การทำงานของผู้อัฒนา ง่ายและรวดเร็วขึ้น โดยการทำการเรียกตารางเท่าที่ต้องการขึ้นมา แล้วเลือกเขตที่ต้องการ แสดงออกมาพร้อมกำหนดข้อจำกัด (Criteria) ในการดึงข้อมูลของแต่ละเขต ในการเลือก ตารางหลายๆ ตารางเราสามารถกำหนดประเภทของการเชื่อม(Join) ได้ 3 รูปแบบ คือ

1. การเชื่อมภายใน (Inner Join) เป็นการเรียกข้อมูลจาก ตารางทั้งสองที่มีเขตที่ เข้าร่วมมีค่าเท่ากัน
2. การเชื่อมทางซ้าย (Left Outer Join) เป็นการเรียกข้อมูลจากตารางทางซ้ายทั้งหมดและตารางทางขวาที่มีเขตที่เข้าร่วมมีค่าเท่ากัน
3. การเชื่อมทางขวา (Right Outer Join) เป็นการเรียกข้อมูลจากตารางทางขวาทั้งหมดและตารางทางซ้ายที่มีเขตที่เข้าร่วมมีค่าเท่ากัน



รูปที่ 5.4 แสดงข้อสอบถาม

ฟอร์ม

การสร้างฟอร์มไว้เพื่อใช้เป็นตัวประสานงานระหว่างตัวตารางและผู้ใช้งานฐานข้อมูล การสร้างฟอร์มของแอคเสสนั้นเป็นแบบรหัสอยู่เบื้องหลังฟอร์ม(Code-Behind-Forms) กล่าวคือ เราสามารถบรรจุคำสั่งลงไปในแต่ละส่วน(element)ของฟอร์ม หรือของตัวฟอร์มเอง ด้วยคุณสมบัตินี้ทำให้ตัวแอคเสสสามารถสร้างฟอร์มได้ง่ายและรวดเร็ว

ในตัวแอคเสสนี้มีชุดคำสั่งอยู่ 3 ประเภทคือ เอสคิวเอล(SQL) แอคเสสเบสิค(Access Basic) และ แมโคร(Macro) ซึ่งในแต่ละประเภทสามารถนำไปใช้ในลักษณะต่าง ๆ กันได้ และสามารถนำมาใช้ร่วมกันในแต่ละฟังก์ชันหรือในโปรแกรมย่อย

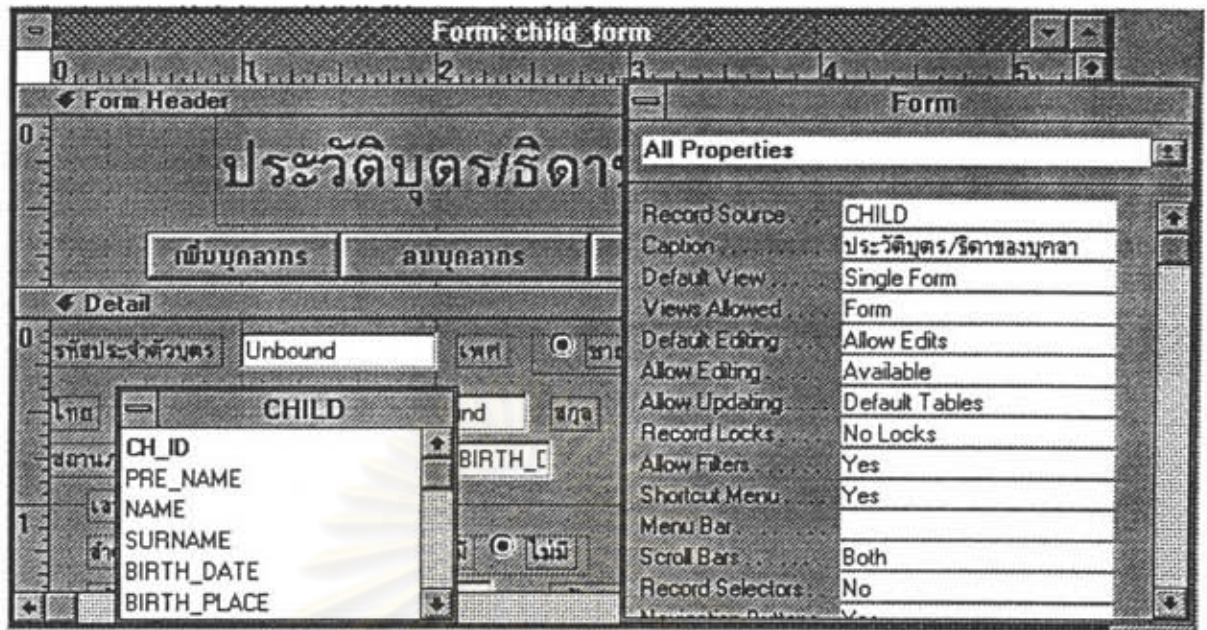
ตัวฟอร์มจะประกอบด้วย 2 ส่วนหลัก ๆ ดังนี้

1. ส่วนของหัวฟอร์มและท้ายฟอร์ม(Header/Footer)ในส่วนนี้ยังแบ่งออกเป็นหัวฟอร์มและท้ายฟอร์มของฟอร์ม และหัวฟอร์มและท้ายฟอร์มของหน้า การทำหัวฟอร์มและท้ายฟอร์มในส่วนนี้ส่วนใหญ่ทำเพื่อเป็นการบอกผู้ใช้ว่าฟอร์มที่เข้านี้เป็นฟอร์มของอะไร เป็นหน้าที่เท่าไร มีคำสั่งหรือหัวข้ออะไรบ้าง เป็นต้น

2. ส่วนของรายละเอียด(Detail) ในส่วนนี้จะป็นรายละเอียดของฟอร์มหรือรายการของแต่ละแถวข้อมูล(Record) หรือเป็นฟอร์มรอง(Sub-Form) ก็ได้

ทั้งสองส่วนนี้จะสามารถกำหนดขนาดในการแสดงผลเท่าไรก็ได้ หรือแม้กระทั่งไม่แสดงผลเลยก็ได้ทำให้การกำหนดรูปแบบของฟอร์มจะมีลักษณะหลากหลาย

ศูนย์วิทยพัทยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 5.5 แสดงแบบฟอร์ม

รายงาน(Report)

ในส่วนนี้จะมีความคล้ายคลึงกับส่วนของการสร้างฟอร์มแต่สิ่งที่แตกต่างกันออกไป คือ การสร้างรายงานจะไม่มีคำสั่งถึงการเปลี่ยนแปลงข้อมูล หรือการเลื่อนจากเขต (Field) หนึ่งไปยังอีกเขตหนึ่ง หรือการทำตรวจสอบข้อมูล(validate) เป็นต้น

นอกจากนี้ในตัวรายงานยังสามารถจัดแบ่งประเภทของข้อมูลออกเป็นกลุ่มระดับได้ (Group Level) เช่น ต้องการจัดกลุ่มของบุคลากรแยกออกตามคณะ ภาควิชา สาขาวิชา และ แผนก เป็นชั้นๆ เป็นต้น ตัวรายงานจะจัดลำดับการแสดงผลออกมาตามการเรียงข้อมูลที่ได้จากการดึงข้อมูลจากฐานข้อมูลให้ ดังนั้นผลของการทำรายงานโดยแอสเซสจึงดูออกมาสวยงาม

แมโคร

นอกจากการเขียนโปรแกรมประยุกต์(Application) บนแอสเซสด้วยแอสเซสเบสิค (Access Basic Code) แล้ว เรายังสามารถเขียนด้วยแมโครที่กำหนดลักษณะของการกระทำต่างๆในแอสเซส ข้อดีของแมโครคือสามารถพัฒนาโปรแกรมประยุกต์ได้ง่ายขึ้นในบางกรณี

แต่ข้อจำกัดของมันคือไม่ผู้พัฒนาไม่สามารถที่จะเขียนรหัส(Code)ที่จะดักการทำงานเมื่อเกิดการผิดพลาด หรือเกิดเหตุการณ์ที่ต้องการได้เนื่องจากตัวแมโครจะจัดการให้ทั้งหมด ส่วนข้อเสียคือไม่สามารถที่จะเรียกใช้ฟังก์ชันที่อยู่ในดีเอลเอลได้ ไม่สามารถที่จะควบคุมการประมวลผลของข้อมูลที่เปลี่ยนแปลง เป็นต้น

ในความเป็นจริงแล้วการพัฒนาโปรแกรมประยุกต์นั้นมักใช้แอสเซมบลีคอบดูไปกับแมโคร เนื่องจากการดักแป้นพิมพ์จากรายการเลือก(Menu Key Trapping) และการทำให้เกิดผลแบบอัตโนมัติ(Auto-Execution With Menu) มักเป็นส่วนหนึ่งของโปรแกรมประยุกต์เสมอ ซึ่งการทำแมโครกับการกระทำในบางกรณีจะทำให้การพัฒนาง่ายขึ้น

มอดูล

มอดูลมีไว้เพื่อเป็นการแยกประเภทของการทำงานและจัดหมวดหมู่ของฟังก์ชันต่าง ๆ ให้มีระบบระเบียบ ในส่วนของมอดูลนี้จะช่วยได้เป็นอย่างมาก โดยภาษาที่ใช้จะเป็นแอสเซมบลี การเรียกใช้ฟังก์ชันต่างในมอดูลนั้นทำได้โดยเรียกชื่อฟังก์ชันโดยตรง

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย