

บทที่ 4

แบบจำลองฐานข้อมูลเชิงสัมพันธ์ (Relational Database Model)

แบบจำลองฐานข้อมูลแบบลำดับชั้น (Hierarchical Database Model) และแบบจำลองแบบเครือข่าย (Network Database Model) มีข้อจำกัดในการจัดการข้อมูลภายใน เนื่องจากถ้ามีการเพิ่มฟิลด์ข้อมูลลงไป ในเรคคอร์ดของแฟ้มข้อมูล จะต้องมีการจัดตำแหน่งที่อยู่ใหม่ ของเรคคอร์ดทั้งหมดในจานแม่เหล็ก จะต้องมีการเขียนโปรแกรมจัดการในด้านนี้โดยเฉพาะ โดยผู้ที่มีความเชี่ยวชาญในการเขียนโปรแกรม ทำให้การจัดการข้อมูลในฐานข้อมูลทั้งสองแบบมีความยุ่งยากมาก ด้วยเหตุผลจากข้อจำกัดดังกล่าว ในปี ค.ศ.1970 E.F.Codd จึงได้พัฒนาระบบฐานข้อมูลแบบใหม่ขึ้นมา คือ **แบบจำลองฐานข้อมูลเชิงสัมพันธ์** เป็นแบบจำลองที่มีพื้นฐานมาจากทฤษฎีทางคณิตศาสตร์ เป็นแบบจำลองฐานข้อมูลที่นิยมใช้กันมากในปัจจุบัน เพราะง่ายต่อการใช้งานและผู้ใช้ไม่จำเป็นต้องทราบว่าข้อมูลเก็บอยู่ ณ ตำแหน่งใดในสื่อบันทึกข้อมูล และไม่ต้องทราบว่าวิธีการเข้าถึงข้อมูลในสื่อบันทึกอย่างไร กล่าวได้ว่า Codd เป็นผู้วางรากฐาน แบบจำลองฐานข้อมูลเชิงสัมพันธ์ แนวคิดของเขาเป็นที่ยอมรับกันเป็นสากล

1. ความหมายและแนวคิดแบบจำลองฐานข้อมูลเชิงสัมพันธ์

1.1 ความหมายของแบบจำลองฐานข้อมูลเชิงสัมพันธ์

แบบจำลองฐานข้อมูลเชิงสัมพันธ์ เป็นฐานข้อมูลแบบหนึ่ง ซึ่งประกอบด้วยตารางข้อมูลจำนวนมาก แต่ละตารางอาจจะใช้เก็บข้อมูลเกี่ยวกับเรื่องต่าง ๆ ของคน ของสถานที่ ของเหตุการณ์ต่าง ๆ หรือ ความสัมพันธ์ระหว่างเรื่องเหล่านี้ก็ได้ และเป็นแบบจำลองฐานข้อมูลที่อยู่บนพื้นฐานจากทฤษฎีคณิตศาสตร์เรื่องความสัมพันธ์ (relations)

จากสารานุกรมไทยสำหรับเยาวชน โดยพระราชประสงค์พระบาทสมเด็จพระเจ้าอยู่หัว เล่มที่ 25 เรื่องที่ 2 **ระบบฐานข้อมูล** ได้ให้ความหมาย แบบจำลองฐานข้อมูลเชิงสัมพันธ์ ว่า “เป็นแบบจำลองข้อมูลที่ได้รับการยอมรับอย่างสูงที่สุดเสนอขึ้นเมื่อปี พ.ศ. ๒๕๑๓ โดยนายเท็ดคอดด์ มีโครงสร้างข้อมูลพื้นฐานแบบเดียวกันเท่านั้น และเป็นแบบที่ง่ายด้วย คือ ตารางความสัมพันธ์ (Relation) มีลักษณะเหมือนกับโครงสร้างตาราง นั่นเอง ฐานข้อมูลหนึ่งๆ คือ กลุ่มของตารางความสัมพันธ์ แต่ละแถวของข้อมูลในตารางหนึ่งๆ คือ ชุดข้อมูลที่มีความสัมพันธ์กัน ซึ่งอาจหมายถึงเอนทิตี หรือความสัมพันธ์ระหว่างเอนทิตีก็ได้โดยที่ข้อมูลแต่ละแถว (Row) จะถูกเรียกว่า **ทูเปิล (Tuple)** และชื่อของแต่ละสดมภ์ (Column) นั้นจะถูกเรียกว่า **คุณสมบัติ (Attribute)**

ตัวอย่างระบบจัดการฐานข้อมูล ที่ใช้แบบจำลองข้อมูลเชิงสัมพันธ์ ได้แก่ ดีบีทู (DB2) โอราเคิล (Oracle) อินฟอร์มิक्स (Informix) ซายเบส (Sybase) อินเกรส (Ingress) แอ็กเซส (Access) อินเทอร์เบส (Interbase) เป็นต้น ”

จึงอาจกล่าวโดยสรุปได้ว่าแบบจำลองฐานข้อมูลเชิงสัมพันธ์ เป็นแบบจำลองที่แสดงข้อมูลในเชิงคณิตศาสตร์ในรูปแบบของตารางสองมิติ การจัดเก็บข้อมูลจะอยู่ในลักษณะของตารางข้อมูลนั่นเอง ในฐานข้อมูลหนึ่ง ๆ สามารถที่จะมีตารางได้ตั้งแต่ 1 ตารางเป็นต้นไป และในแต่ละตารางก็สามารถมีได้หลายคอลัมน์ หลายแถว และตารางข้อมูลแต่ละตารางในฐานข้อมูลจะมีความสัมพันธ์กัน การเชื่อมโยงข้อมูลระหว่างตาราง จะใช้แอททริบิวท์ ที่มีอยู่ทั้งสองตารางเป็นตัวเชื่อมโยงข้อมูล แบบจำลองข้อมูลเชิงสัมพันธ์ได้ออกแบบมาเพื่อลดความซ้ำซ้อนของการเก็บข้อมูล และสามารถเรียกใช้ข้อมูลได้อย่างมีประสิทธิภาพ เป็นแบบจำลองที่มีความนิยมใช้กันมากในปัจจุบัน

ระบบจัดการฐานข้อมูล ช่วยให้ผู้ออกแบบฐานข้อมูลหรือผู้ใช้งานสามารถใช้งานเพียงแคในระดับตรรกะเท่านั้น ส่วนรายละเอียดต่าง ๆ ในระดับกายภาพ เช่นการจัดเก็บข้อมูล เส้นทางในการเข้าถึงข้อมูล หรือโครงสร้างข้อมูล จะมีระบบจัดการฐานข้อมูล (Relational Database Management System, RDMS) เป็นผู้ดูแลจัดการให้ ทำให้การจัดการของแบบจำลองฐานข้อมูลเชิงสัมพันธ์ สามารถตอบสนองความต้องการของผู้ใช้ในเรื่องการจัดการฐานข้อมูลของระบบได้อย่างมีประสิทธิภาพ

1.2 แนวคิดและกฎ 12 ข้อ ของ E.F Codd (Codd's 12 Rules)

แบบจำลองฐานข้อมูลเชิงสัมพันธ์เสนอ โดย Edgar Frank Codd นักวิทยาศาสตร์คอมพิวเตอร์ ชาวอังกฤษ ในปี ค.ศ. 1970 โดยใช้ทฤษฎีของเซต (set) ซึ่งเป็นทฤษฎีทางคณิตศาสตร์มาใช้ในการจัดการกับข้อมูลหรือกลุ่มข้อมูลเชิงเลขคณิต (Mathematical Concept of relational sets) ด้วยการนิยามทฤษฎีต่าง ๆ ให้กับโครงสร้างข้อมูลในรูปแบบของคณิตศาสตร์ เพื่อแก้ไขข้อบกพร่องของฐานข้อมูลแบบเดิมที่ใช้กันอยู่ในขณะนั้น คือแบบจำลองฐานข้อมูลแบบลำดับขั้น และแบบจำลองแบบเครือข่าย โดยที่เซตจะหมายถึงกลุ่มของสิ่งต่าง ๆ เช่น กำหนดให้ A เป็นเซตของสัตว์ปีก, $A = \{\text{นก ไก่ กา ห่าน หงส์}\}$ จำนวนของสมาชิก A หรือ $n(A) = 5$ และกำหนดให้ B เป็นเซตของผลไม้, $B = \{\text{มะม่วง ทุเรียน มังคุด เงาะ ส้ม ลำไย}\}$ จำนวนของสมาชิก B หรือ $n(B) = 6$

Codd ได้ศึกษาและกำหนดคุณลักษณะที่เป็นบรรทัดฐานของระบบฐานข้อมูลแบบจำลองฐานข้อมูลเชิงสัมพันธ์ และได้สรุปเป็นเกณฑ์หรือกฎ (Rules) ของแบบจำลองนี้ว่า จะต้องมียุคสมบัติ 12 ข้อ ที่รู้จักกันดีในนาม กฎ 12 ข้อของ Codd (Codd's 12 rules) และจากกฎ 12 ข้อ นี้ ระบบปฏิบัติการฐานข้อมูล จึงได้ถูกพัฒนาขึ้นมาอย่างจริงจัง

กฎ 12 ข้อของ Codd ประกอบกลุ่มของกฎ 13 ข้อ (ข้อที่ 0 ถึง 12) ดังนี้

Rule 0 : The foundation rule : กฎพื้นฐาน

“ระบบฐานข้อมูลใดๆ ที่จะจัดเป็นแบบจำลองฐานข้อมูลเชิงสัมพันธ์ ระบบฐานข้อมูลนั้นจะต้องสามารถที่จะจัดการข้อมูลทั้งหมดด้วยความสามารถทางรีเลชัน” (For any system that is advertised as, or claimed to be, a relational data base management system, that system must be able to manage data bases entirely through its relational capabilities.)

Rule 1 : The information rule : กฎสารสนเทศ

“สารสนเทศหรือข้อมูลทั้งหมดในระบบฐานข้อมูล ต้องแสดงอยู่ในรูปแบบเดียวกัน เท่านั้นคือรูปแบบของตาราง และสามารถกำหนดรูปแบบความสัมพันธ์ของข้อมูลได้” (All information in a relational data base is represented explicitly at the logical level and in exactly one way — by values in tables.)

Rule 2 : The guaranteed access rule : กฎการรับประกันการเข้าถึงข้อมูล

“ข้อมูลทุกอย่างที่ถูกจัดเก็บไว้ในฐานข้อมูล ต้องสามารถอ้างอิงหรือเข้าถึงได้ ด้วยการระบุ ชื่อตาราง ค่าของคีย์หลัก (primary key) และชื่อคอลัมน์ (Column) ที่ต้องการ” (Each and every data (atomic value) in a relational data base is guaranteed to be logically accessible by resorting to a combination of table name, primary key value and column name.)

Rule 3 : Systematic treatment of null values : กฎการจัดการค่าว่างอย่างเป็นระบบ

“ระบบจัดการฐานข้อมูลจะต้องมีการรองรับการแทนค่าสำหรับข้อมูลบางอย่างที่ยังไม่มีค่าด้วยค่า null เช่น ข้อมูลที่ขาดหายไป หรือไม่มีข้อมูลที่เหมาะสม อย่างเป็นระบบ” (ค่า null จะต่างจากค่าโดยทั่วไป เช่น แตกต่างจากค่าศูนย์ หรือค่าตัวเลข) (Null values, distinct from the empty character string or a string of blank characters and distinct from zero or any other number, are supported in fully relational DBMS for representing missing information and inapplicable information in a systematic way, independent of data type.)

Rule 4 : Active *online catalog* based on the relational model : แอดทีฟออนไลน์

แคตตาล็อกที่อยู่บนพื้นฐานของแบบจำลองฐานข้อมูลเชิงสัมพันธ์

“ระบบจัดการฐานข้อมูลต้องสนับสนุนให้ผู้ใช้สามารถเข้าถึงได้โดยที่มีสิทธิ์ผ่าน ทางภาษาคิวรี” (The data base description is represented at the logical level in the same way as ordinary data, so that authorized users can apply the same relational language to its interrogation as they apply to the regular data.)

Rule 5 : The comprehensive data sublanguage rule : กฎภาษาข้อมูล

“ระบบแบบจำลองฐานข้อมูลเชิงสัมพันธ์ จะต้องรองรับภาษาที่เป็นรีเลชันอย่างน้อย 1 ภาษาที่มีคุณสมบัติดังต่อไปนี้ :

- 1) สามารถนิยามโครงสร้างระบบฐานข้อมูล
- 2) นิยามวิว
- 3) การจัดการข้อมูล (ด้วยวิธีการใช้คำสั่งแบบ interactive และการเขียนโปรแกรม)
- 4) ความถูกต้อง/มั่นคงของข้อมูล และเงื่อนไข Integrity
- 5) การจัดการสิทธิการใช้งาน
- 6) มีการทำงานในรูปแบบของ transaction

(A relational system may support several languages and various modes of terminal use (for example, the fill-in-the-blanks mode). However, there must be at least one language whose statements are expressible, per some well-defined syntax, as character strings and that is comprehensive in supporting all of the following items :

- 1) Data definition.
- 2) View definition.
- 3) Data manipulation (interactive and by program).
- 4) Integrity constraints.
- 5) Authorization.
- 6) Transaction boundaries (begin, commit and rollback).

Rule 6 : The *view updating rule* : กฎการแก้ไขข้อมูลผ่านทางวิว

“ระบบจะต้องรองรับความสามารถในการแก้ไข วิวทุกวิวที่สามารถเปลี่ยนแปลงแก้ไขข้อมูลได้” (All views that are theoretically updatable are also updatable by the system.)

Rule 7 : *High-level insert, update, and delete* : กฎความสามารถในการเพิ่ม ลบ และแก้ไขข้อมูล

“ระบบจะต้องมีความสามารถในการแทรก แก้ไข และลบข้อมูลในลักษณะของเซตได้ด้วย คำสั่งเดียว”(The capability of handling a base relation or a derived relation as a single operand applies not only to the retrieval of data but also to the insertion, update and deletion of data.)

Rule 8 : *Physical data independence* : กฎความเป็นอิสระของข้อมูลเชิงกายภาพ

“ระบบจะต้องรองรับการจัดการ การเปลี่ยนแปลงในระดับโครงสร้าง (schema) ทางกายภาพได้ โดยที่ไม่มีผลกับผู้ใช้งาน ๆ ไม่จำเป็นจะต้องรับรู้เกี่ยวกับการเปลี่ยนแปลงวิธีการจัดเก็บและการเข้าถึงข้อมูล (Application programs and terminal activities remain logically unimpaired whenever any changes are made in either storage representations or access methods.)

Rule 9 : *Logical data independence* : กฎความเป็นอิสระของข้อมูลในเชิงตรรกะ

“ระบบจะต้องรองรับการจัดการ การเปลี่ยนแปลงโครงสร้างทางกายภาพ โดยไม่มีผลกระทบต่อโครงสร้างของโปรแกรมประยุกต์”(การเปลี่ยนแปลงข้อมูลในระดับตรรกะ รวมถึงการเปลี่ยนแปลงโครงสร้างข้อมูล มีผลต่อคำสั่ง และโปรแกรมมากกว่าการเปลี่ยนแปลงทางกายภาพ) (Application programs and terminal activities remain logically unimpaired when information-preserving changes of any kind that theoretically permit un impairment are made to the base tables.)

Rule 10 : *Integrity independence* : กฎความคงสภาพที่เป็นอิสระของข้อมูล

“เงื่อนไขของอินทิกริตีต้องมีการกำหนดแยกต่างหากจากโปรแกรมประยุกต์ และจัดเก็บไว้ในแคตตาล็อก ทั้งนี้เพื่อให้เกิดความเป็นไปได้ในการที่จะเปลี่ยนแปลงเงื่อนไขของอินทิกริตีโดยไม่มีผล กระทบต่อโปรแกรมประยุกต์ที่มีอยู่”(การควบคุมข้อมูลให้มีความคงสภาพและเป็นไปตามกฎที่ตั้งไว้ ต้องสามารถ ระบุได้จากแอปพลิเคชันโปรแกรม และเก็บในรายการ ซึ่งเงื่อนไขต่าง ๆ สามารถเปลี่ยนแปลงได้ ตาม ความเหมาะสม โดยไม่กระทบต่อแอปพลิเคชันที่ใช้งานอยู่) (Integrity constraints specific to a particular relational data base must be definable in the relational data sublanguage and storable in the catalog, not in the application programs.)

Rule 11 : *Distribution independence* : ความเป็นอิสระของการกระจาย

“การกระจายข้อมูลเป็นคุณสมบัติประการหนึ่งของฐานข้อมูลที่ต้องการจะเพิ่มพื้นที่ในการเก็บข้อมูล ผู้ใช้ไม่ต้องให้ความสนใจว่าข้อมูลจะถูกจัดเก็บอย่างไร หรือกระบวนการทำงานเป็นอย่างไร ระบบฐานข้อมูลจะทำการกระจายผ่านโปรแกรมประยุกต์” (The end-user must not be able to see that the data is distributed over various locations. Users should always get the impression that the data is located at one site only. This rule has been regarded as the foundation of distributed database systems.)

Rule 12: *The non subversion rule* : กฎการไม่มีเวอร์ชันย่อย

“ระบบจะต้องมีการควบคุมกฎเกณฑ์ต่างๆ ที่ตั้งไว้ได้ โดยจะต้องไม่สามารถถูกปรับเปลี่ยน หรือแก้ไขได้โดยเครื่องมือหรือภาษา อื่น ๆ”.(กฎ integrity จะต้องสามารถควบคุมความถูกต้องของข้อมูลในระบบได้ตลอดเวลา ไม่ว่าผู้ใช้จะเข้าถึงข้อมูลด้วยเครื่องมือใด ๆ ก็ตาม) (If a relational system has a low-level (single-record-at-a-time) language, that low level cannot be used to subvert or bypass the integrity rules and constraints expressed in the higher level relational language (multiple-records-at-a-time)).

แบบจำลองฐานข้อมูลเชิงสัมพันธ์ เป็นแบบจำลองฐานข้อมูลที่ได้รับความนิยมสูงสุด ในปัจจุบันเพราะสามารถใช้งานได้กับเครื่องคอมพิวเตอร์ทุกระดับตั้งแต่ไมโครคอมพิวเตอร์ จนกระทั่งถึงเมนเฟรม คอมพิวเตอร์ มีโครงสร้างข้อมูลพื้นฐานแบบเดียว และเป็นแบบง่าย ๆ โครงสร้างข้อมูลจะถูกเก็บอยู่ในรูปแบบของตาราง (Table) หรือเรียกอีกชื่อว่า รีเลชัน (relation) ภายในตารางจะแบ่งออกเป็นแถว (row) เรียกว่า *ทูเพิล* (Tuple) และ คอลัมน์ (column) เรียกว่า *แอททริบิวต์* (Attribute) แต่ละแถวของข้อมูลในแต่ละตารางคือข้อมูล 1 เรคอร์ด (Record) หรือ ระเบียบ หรือ ชุดข้อมูลที่มีความสัมพันธ์กัน ซึ่งแสดงค่าของข้อมูลสำหรับวัตถุ หรือสิ่งของที่เรานสนใจที่เก็บอยู่ในฐานข้อมูลหนึ่งๆ ในระบบงานนั้นๆ หรือที่เรียกกันว่า เอนทิตี (Entity) ส่วนแต่ละคอลัมน์หรือแอททริบิวต์คือข้อมูล 1 ฟิลด์ (Field) ซึ่งจะแสดงคุณสมบัติต่าง ๆ ของรีเลชันนั้น โดยจะต้อง มีการกำหนดชื่อให้กับแต่ละคอลัมน์

กลุ่มของข้อมูลของแต่ละแอททริบิวต์คือโดเมน (Domain) ของแอททริบิวต์นั้น ๆ โดเมนจะเป็นกรอบหรือขอบเขตของค่าต่าง ๆ ที่เป็นไปได้ เช่น โดเมนของค่าลำดับของเดือนคือ 1 ถึง 12 โดเมนของค่าของอุณหภูมิอาจอยู่ระหว่าง -10 ถึง 100 องศาเซลเซียส โดเมนของชื่อคือข้อความที่มีความยาวไม่เกิน 50 ตัวอักษร โดเมนของหมายเลขโทรศัพท์เคลื่อนที่จะประกอบด้วยข้อมูลที่เป็นตัวเลขจำนวน 10 ตัว เป็นต้น

จำนวนทูเพิลในหนึ่งรีเลชันคือจำนวนแถวในหนึ่งตารางเรียกว่า *คาร์ดินัลลิตี* (Cardinality) ของรีเลชันนั้น จำนวนแอททริบิวต์ในหนึ่งรีเลชันคือจำนวนคอลัมน์ในหนึ่งตาราง เรียกว่า *ดีกรี* (Degree) ของรีเลชัน

แบบจำลองฐานข้อมูลเชิงสัมพันธ์เป็นแบบจำลองที่ใช้งานกันอย่างกว้างขวาง เนื่องจาก ควบคุมได้ง่าย มีความยืดหยุ่นมาก นอกจากนี้การจัดการข้อมูลในรูปแบบของตารางทำให้ผู้ใช้สามารถเข้าใจโครงสร้างของข้อมูลได้ง่าย เนื่องจากมองเห็นข้อมูลทั้งหมด การเชื่อมโยงข้อมูลระหว่างตารางทำได้โดยการใช้ฟิลด์ที่มีข้อมูลร่วมกันระหว่างตาราง การเปลี่ยนแปลงโครงสร้างของฐานข้อมูล ทำได้โดยการเพิ่มหรือลบคอลัมน์ออกจากตาราง โดยไม่จำเป็นต้องสร้างโครงสร้างฐานข้อมูลใหม่ทั้งหมด ตัวอย่างระบบจัดการฐานข้อมูลเชิงสัมพันธ์ที่เป็นที่นิยมแพร่หลายในปัจจุบันได้แก่ MS ACCESS, SQL Server, DB2, ORACLE, INGRES, MySQL และ SYBASE เป็นต้น

คำศัพท์แบบจำลองเชิงสัมพันธ์ (Relational Database Terminology)

เนื่องจากแบบจำลองข้อมูลเชิงสัมพันธ์ เกิดจากทฤษฎีทางคณิตศาสตร์เรื่องเซต และเป็นการเก็บข้อมูลในรูปของตารางหลาย ๆ ตารางที่มีความสัมพันธ์กัน ในทางทฤษฎีจะมีคำศัพท์เฉพาะที่แตกต่างออกไป ดังนี้

ตาราง แถว คอลัมน์ คีย์หลัก Primary key คีย์นอก Foreign key โดเมน datatype

- ตาราง (Tables) หรือ รีเลชัน (Relations) หรือไฟล์ (files) หรือเอนทิตี เซต (entity sets) เป็นการแสดงถึงรูปแบบของตาราง 2 มิติ ที่ประกอบด้วยคอลัมน์และแถวของข้อมูล
- แอททริบิวต์ หรือ คอลัมน์ เป็นการแสดงถึงคุณลักษณะของรีเลชัน อาจจะเรียกว่า *เขตข้อมูล* ก็ได้ แต่ละคอลัมน์ต้องมีการกำหนดชื่อที่แตกต่างกัน โดยทั่วไปชื่อของคอลัมน์จะแสดงความหมายของข้อมูลที่จะถูกจัดเก็บไว้ในคอลัมน์นั้น และลำดับของคอลัมน์ไม่มีความสำคัญ เช่น รีเลชัน “นิสิต” ประกอบด้วยคอลัมน์ที่แสดงถึง แอททริบิวต์ต่างๆ ได้แก่ รหัสนิสิต ชื่อนิสิต คณะวิชา สาขาวิชา วันเดือนปีเกิด ที่อยู่ เป็นต้น
- ทูเพิล หรือ แถว หรือ เรคอร์ด (record) แถวจะเป็นที่เก็บสมาชิกของรีเลชัน ดังนั้นแถวแต่ละแถวในรีเลชันจะต้องมีคุณสมบัติที่เหมือนกันหมด
- คาร์ดินอลลิตี (Cardinality) คือ จำนวนของทูเพิลในหนึ่งรีเลชัน หรือ จำนวนแถวในหนึ่งตาราง
- ดีกรี (Degree) คือ จำนวนแอททริบิวต์ในหนึ่งรีเลชัน หรือ จำนวนคอลัมน์ในหนึ่งตาราง
- โดเมน (Domain) คือ เซตหรือกลุ่มของข้อมูลที่เป็นไปได้ของแต่ละแอททริบิวต์
- ค่าว่าง (Null Value) หมายถึงแอททริบิวต์ที่ยัง *ไม่มีค่า* ข้อมูล เนื่องจากอาจจะยังไม่ทราบค่าข้อมูลที่จะต้องใส่ลงไปแอททริบิวต์นั้น ๆ ภายหลังเมื่อทราบค่าข้อมูลในแอททริบิวต์นั้น อาจจะมีการกลับมาใส่ข้อมูลลงไปใหม่ก็ได้ (ยกเว้นแอททริบิวต์ที่เป็นคีย์หลักที่ไม่สามารถทำให้เป็นค่าว่างได้)
- คีย์หลัก (Primary Key) คือ แอททริบิวต์ที่สามารถใช้เจาะจงแถวใดแถวหนึ่งในรีเลชัน โดยข้อมูลแต่ละแถวจะไม่ซ้ำซ้อนกัน บางครั้งอาจเรียกสั้น ๆ ว่า คีย์

2. องค์ประกอบแบบจำลองฐานข้อมูลเชิงสัมพันธ์

แบบจำลองฐานข้อมูลเชิงสัมพันธ์ประกอบด้วยองค์ประกอบ 3 ส่วน ดังนี้

- *ส่วนโครงสร้างของข้อมูล* (Data Structure) เป็นส่วนการจัดเก็บข้อมูลในรูปแบบของตารางที่ประกอบด้วยคอลัมน์และแถว
- *ส่วนจัดการข้อมูล* (Data Manipulation) เป็นส่วนของคำสั่งที่ใช้จัดการข้อมูลเก็บอยู่ในฐานข้อมูล
- *ส่วนควบคุมความคงสภาพของข้อมูล* (Data Integrity) เป็นข้อกำหนดหรือกฎเกณฑ์ที่ถูกสร้างขึ้นมาเพื่อใช้ควบคุมความคงสภาพของข้อมูล

2.1 คุณสมบัติของรีเลชัน (Properties of Relations)

รีเลชันหรือตารางจะอยู่ในลักษณะตาราง 2 มิติ ประกอบด้วยแถวและคอลัมน์ โดยทั่วไป รีเลชันหนึ่ง ๆ จะมีคุณสมบัติต่าง ๆ ดังนี้

- 1) รีเลชันต้องมีชื่อกำกับ แต่ละชื่อต้องแตกต่างกันจะซ้ำกันไม่ได้
- 2) ค่าของข้อมูลในแต่ละแอททริบิวต์ จะเป็นค่าของข้อมูลประเภทเดียวกัน และเป็นค่าๆ เดียว (atomic) ไม่เป็นเซตของค่าหลายค่า ทำให้ข้อมูลในตารางมีความเป็นปรกติหรืออยู่ในบรรทัดฐาน (Normalized)
- 3) ค่าของข้อมูลที่อยู่ในแต่ละแอททริบิวต์ จะต้องอยู่ในขอบเขตของโดเมนที่กำหนดไว้สำหรับแอททริบิวต์นั้น
- 4) ชื่อของแต่ละแอททริบิวต์ จะต้องมีการเรียกที่แตกต่างกัน
- 5) ลำดับที่ของแอททริบิวต์ (การเรียงลำดับก่อนหลัง หรือจากซ้ายไปขวา) ไม่มีความสำคัญ
- 6) ข้อมูลในแต่ละแถวจะไม่ซ้ำกัน (ไม่มีทูเพิลคู่ใด ๆ ที่ซ้ำกัน)
- 7) การเรียงลำดับของทูเพิล (การเรียงลำดับที่ของแถวแต่ละแถว) ไม่มีความสำคัญ
- 8) แต่ละฐานข้อมูล จะมีรีเลชันก็ได้ ขึ้นอยู่กับความซับซ้อนของฐานข้อมูล

2.2 รูปแบบของรีเลชัน

รูปแบบของรีเลชัน ประกอบด้วย :

- ชื่อของรีเลชัน
- รายชื่อของแอททริบิวต์ ๆ ทั้งหมดของรีเลชันนั้นอยู่ในวงเล็บ โดยใช้เครื่องหมาย

จุลภาค (,) คั่นระหว่างแอททริบิวต์แต่ละตัว ดังรูปแบบข้างล่าง

รูปแบบ : ชื่อรีเลชัน (ชื่อแอททริบิวต์ 1 , ชื่อแอททริบิวต์ 2 , , ชื่อแอททริบิวต์สุดท้าย :

รีเลชัน “**นิสิต**” (รหัสนิสิต, ชื่อ , สกุล, คณะวิชา, ที่อยู่)

รีเลชันชื่อ “**นิสิต**” ประกอบด้วยแอททริบิวต์ “รหัสนิสิต” “ชื่อ-สกุล” “คณะวิชา” และ “ที่อยู่”

จากภาพข้างล่างจะเห็นว่าข้อมูลปรากฏในรูปของตารางสองมิติที่มีลักษณะสัมพันธ์กัน ประกอบด้วยแถวและคอลัมน์ ตารางข้อมูลทั้งหมดเรียกว่ารีเลชัน หรือ ตาราง (ส่วนใหญ่นิยมเรียกว่าตาราง เพราะ โครงสร้างการจัดเก็บเป็นแบบตาราง ข้อมูลในแต่ละแถวของรีเลชัน เรียกว่าทูเพิล ข้อมูลในแต่ละคอลัมน์ของรีเลชันเรียกว่าแอททริบิวต์ จำนวนของแอททริบิวต์ใน

ตารางเรียกดีกรี (degree) จำนวนแถวในตารางเรียกว่า *คาร์ดินัลลิตี* (cardinality) จากภาพ มี 4 คาร์ดินัลลิตี และ 5 ดีกรี เขียนในรูปแบบของรีเลชัน : นิสิต (รหัสนิสิต, ชื่อ, สกุล, คณะวิชา, ที่อยู่)

ชื่อรีเลชัน “นิสิต”

แอททริบิวต์ 1 แอททริบิวต์ 2 แอททริบิวต์ 3 แอททริบิวต์ 4 แอททริบิวต์ 5

	รหัสนิสิต	ชื่อ	สกุล	คณะวิชา	สาขาวิชา	
รีเลชัน	580103	น.ส.นารี	สดใส	พยาบาล	การพยาบาล	ทูเพิล 1
	580314	นายสะอาด	ใจดี	วิทยาศาสตร์	ชีววิทยา	ทูเพิล 2
	580619	นายมานะ	อดทน	วิศวกรรมศาสตร์	วิศวกรรมไฟฟ้า	ทูเพิล 3
	580907	น.ส.วิไล	โลภา	สหเวชศาสตร์	กายวิภาค	ทูเพิล 4

← ดีกรี →

รูปที่ 3.1 ตัวอย่างรูปแบบจำลองฐานข้อมูลเชิงสัมพันธ์

2.3 ชนิดของรีเลชัน

ในระบบจัดการฐานข้อมูลทั่วไป รีเลชันอาจจำแนกออกได้เป็น 2 ประเภท ดังนี้

1) *รีเลชันหลัก (Base Relation)* เป็นรีเลชันที่ถูกกำหนดขึ้นเพื่อเก็บข้อมูลและเพื่อนำข้อมูลไปใช้เมื่อมีการสร้างรีเลชัน โดยใช้ภาษาในการนิยามข้อมูล (Data Definition Language) เช่นในภาษา SQL คำสั่ง Create Table เป็นการสร้าง *รีเลชันหลัก* คำสั่ง Alter Table เป็นคำสั่งในการแก้ไข รีเลชันหลัก เป็นต้น หลังจากนั้นก็จะทำการเก็บข้อมูลเพื่อการเรียกใช้ข้อมูลในภายหลัง รีเลชันหลักจะเป็นตารางที่ใช้จัดเก็บข้อมูลจริงไว้

2) *รีเลชันแบบเสมือน (Virtual Relation) หรือ วิว (View)* เป็นตาราง หรือ รีเลชันที่สร้างขึ้นตามความต้องการใช้ข้อมูลของผู้ใช้แต่ละคน เนื่องจากผู้ใช้แต่ละคนอาจจะต้องการใช้ข้อมูลในลักษณะที่แตกต่างกัน เพื่อความสะดวกในการใช้ข้อมูล และได้ข้อมูลที่ตรงกับความต้องการของผู้ใช้ จึงมีการกำหนด *วิว* ของผู้ใช้แต่ละคนขึ้นมาจากรีเลชันหลัก นอกจากนั้นยังเป็นการช่วยรักษาความปลอดภัยของฐานข้อมูลไว้ด้วย อย่างไรก็ตาม *วิว* จะเป็นเพียงตารางสมมุติเท่านั้น จะไม่มีการจัดเก็บข้อมูลจริงลงในระบบฐานข้อมูลแต่อย่างใด

3. รีเลชัน คีย์ (Relational key)

จากคุณสมบัติของรีเลชันที่ว่า *ข้อมูลในแต่ละแถวจะต้องไม่ซ้ำกัน นั่นคือไม่มีทูเพิลคู่ใด ๆ ที่ซ้ำกัน* และปัญหาจากความซ้ำซ้อนของข้อมูลที่จัดเก็บอยู่ในรูปแบบของแฟ้มข้อมูลดังที่ได้กล่าวมาแล้วในบทต้นๆ จำเป็นอย่างยิ่งต้องมีเครื่องมือเข้ามาช่วยควบคุมไม่ให้ข้อมูลเกิดความ

ซ้ำซ้อน เครื่องมือดังกล่าว คือ คีย์ (Keys) หรือ รีเลชัน คีย์ (Relational key) ซึ่งจะช่วยในการแยกความแตกต่างของข้อมูลในแต่ละแถว และยังใช้ในการเชื่อมโยงข้อมูลระหว่างตาราง ดังนั้นเราจึงสามารถที่จะระบุค่าใน แอททริบิวต์ใดแอททริบิวต์หนึ่งซึ่งไม่ซ้ำกันในแต่ละทูปเพิลได้ นอกจากนี้คีย์ยังเป็นเครื่องมือที่ช่วยในการกำหนดความสัมพันธ์ของเอนติตี คีย์ที่เกี่ยวข้องกับรีเลชัน ที่รู้จักกันดี เช่น คีย์หลัก (Primary Key) คีย์รวม (composite หรือ compound key) และ คีย์นอก (Foreign Key) เป็นต้น

คีย์ เป็นเซตข้อมูลหรือฟิลด์ที่ใช้เป็นตัวเข้าถึงข้อมูลในแต่ละระเบียนหรือบ่งชี้ว่าระเบียนใดเป็นระเบียนใด ถ้ากำหนดไว้ในตารางจะทำให้สามารถนำไปเชื่อมโยงกับตารางอื่นได้ หรืออาจจะสรุปได้ว่า คีย์ หมายถึงแอททริบิวต์หรือกลุ่มของแอททริบิวต์ ที่สามารถใช้ในการบ่งบอกความแตกต่างของแต่ละทูปเพิลนั่นเอง ดังนั้นในการอ้างอิง การประมวลผล การค้นหา การแก้ไขข้อมูล หรือการกำหนดความสัมพันธ์ระหว่างข้อมูล จะดำเนินการได้อย่างสะดวกและมีประสิทธิภาพ จะต้องกำหนดคีย์ ให้กับตารางก่อน นอกจากนี้ยังจะช่วยประหยัดพื้นที่ในการจัดเก็บข้อมูลอีกด้วย

3.1 ประเภทของคีย์ (Types of Keys)

เนื่องจากต้องใช้คีย์ในการเชื่อมโยงตารางต่าง ๆ เข้าด้วยกัน คีย์แต่ละชนิดจึงทำหน้าที่แตกต่างกัน ในแต่ละตารางจะมีคีย์หลายชนิดหรือชนิดเดียวก็ได้ คีย์หลักๆ ที่ใช้กับฐานข้อมูลมีดังนี้

1) **ซูเปอร์คีย์ (Super key)** หมายถึง “แอททริบิวต์ หรือกลุ่มของแอททริบิวต์ที่เป็นไปได้ทั้งหมดที่สามารถใช้จำแนกความแตกต่างของแต่ละทูปเพิลในรีเลชันได้” ดังนั้นในหนึ่งรีเลชันจึงสามารถ มีซูเปอร์คีย์ได้หลายซูเปอร์คีย์ ถ้าแอททริบิวต์หรือกลุ่มของแอททริบิวต์ใดที่มีค่าไม่ซ้ำกันเลยในแต่ละแถว(unique) ก็จัดได้ว่าแอททริบิวต์หรือกลุ่มของแอททริบิวต์นั้นเป็นซูเปอร์คีย์ได้” เช่น

จากคำจำกัดความของซูเปอร์คีย์ดังกล่าวข้างต้น เราอาจพิจารณาได้ว่ามี ซูเปอร์คีย์ในรีเลชัน “นิสิต” ทั้งหมด ได้ 13 แบบ ตัวอย่างของซูเปอร์คีย์ :

- รหัสนิสิต
- รหัสนิสิต + ชื่อ
- รหัสนิสิต + สกุล
- รหัสนิสิต + คณะวิชา
- รหัสนิสิต + สาขาวิชา
- รหัสนิสิต + ชื่อ + สกุล

- รหัสนิสิต + ชื่อ + คณะวิชา
- รหัสนิสิต + ชื่อ + สาขาวิชา
- รหัสนิสิต + สกุล + คณะวิชา
- รหัสนิสิต + สกุล + สาขาวิชา
- รหัสนิสิต + ชื่อ + สกุล + คณะวิชา
- รหัสนิสิต + ชื่อ + สกุล + สาขาวิชา
- รหัสนิสิต + ชื่อ + สกุล + คณะวิชา + สาขาวิชา

2) **คีย์หลัก (Primary Key)** หมายถึงแอททริบิวต์ หรือกลุ่มของแอททริบิวต์ของรีเลชัน หนึ่งๆ ที่มีข้อมูลที่เป็นเอกลักษณ์ (unique) คือมีข้อมูลที่ไม่ซ้ำกัน และจะต้องไม่เป็นค่าว่าง หรือ NULL (ค่าที่ปล่อยให้ว่างเปล่าไว้ในบางแอททริบิวต์ของทูเพิล อาจจะเป็นเพราะไม่มีค่าหรือยังไม่ทราบค่าในขณะนั้น ส่วนการกำหนดข้อมูลเป็นเลขศูนย์, 0 หรือ เครื่องหมายขีด, - นั้นไม่ใช่ค่าว่าง) คุณสมบัตินั้นจะสามารถระบุว่าข้อมูลนั้นเป็นของทูเพิลใด ในทุกรีเลชัน จะต้องกำหนดแอททริบิวต์ให้เป็นคีย์หลักของแต่ละรีเลชันเสมอ แต่ละรีเลชันควรมีคีย์หลักเพียงคีย์เดียว คีย์หลักจะใช้ประโยชน์ในการระบุข้อมูลที่ต้องการเข้าถึงเพื่อใช้ในการค้นหา การจัดเรียงรายการข้อมูลที่มีอยู่เป็นจำนวนมากได้อย่างรวดเร็วคือดัชนีหรือ index ตัวอย่างของคีย์หลัก เช่น รหัสนิสิต จะต้องไม่มีนิสิตคนใดใช้รหัสซ้ำกัน


รหัสนิสิต	ชื่อ	สกุล	คณะวิชา	สาขา
580103			พยาบาล	การพยาบาล
580314	นายสะอาด	ใจดี	วิทยาศาสตร์	ชีววิทยา
580619	นายมานะ	อดทน	วิศวกรรมศาสตร์	วิศวกรรมไฟฟ้า

ภาพที่ แสดงค่าว่าง (NULL)

คีย์หลัก อาจเป็นคีย์ที่ประกอบขึ้นจากหลายๆ แอททริบิวต์มารวมกันทำหน้าที่เป็นคีย์หลักของรีเลชัน โดยทั่ว ๆ ไปเรามักจะกำหนดคีย์หลักมาจากคีย์คู่แข่ง แต่ถ้าในรีเลชันมีคีย์คู่แข่งมากกว่า 1 คีย์ จะพิจารณาเลือกคีย์คู่แข่งที่มีขนาดเล็กที่สุดมาเป็นคีย์หลักของรีเลชัน

รีเลชัน “นิสิต”

คีย์หลัก (Primary Key)



รหัสนิสิต	ชื่อ	สกุล	คณะวิชา	สาขา
580103	น.ส.นารี	สดใส	พยาบาล	การพยาบาล
580314	นายสะอาด	ใจดี	วิทยาศาสตร์	ชีววิทยา
580619	นายมานะ	อดทน	วิศวกรรมศาสตร์	วิศวกรรมไฟฟ้า
580907	น.ส.วิไล	โสภา	สหเวชศาสตร์	กายวิภาค

ภาพที่ 3.2 แสดงตัวอย่างการกำหนดคีย์หลัก

สรุป : คีย์หลัก มีคุณลักษณะดังนี้

1. คีย์หลัก ต้องมีค่าไม่ซ้ำกันในแต่ละแถว (มีค่าที่เป็น unique)
2. คีย์หลัก ต้องไม่มีค่าที่เป็นค่าว่าง หรือ NULL (ทุกคน ทุกสิ่งต้องมีรหัส เพราะจะต้องกรอกข้อมูลทั้งหมด)
3. คีย์หลัก จะถูกเปลี่ยนแปลงค่าไม่ได้ (อาจจะมีการเปลี่ยนแปลงได้ เช่น มีการเปลี่ยนชื่อ-สกุล เปลี่ยนคณะวิชา หรือสาขาวิชา)
4. แต่ละตารางจะมีหนึ่งคีย์หลัก
5. คีย์หลักสามารถสร้างจากฟิลด์เดียว หรือรวมกันจากหลายฟิลด์ก็ได้ ถ้าแบบที่รวมกันหลายฟิลด์เรียกว่า **คีย์รวม** (composite หรือ compound key)

3) คีย์รวม (Composite หรือ Compound Key) เป็นการนำเอาหลาย ๆ

ฟิลด์มา

รวมกันเป็นคีย์ เพื่อให้มีคุณสมบัติเป็นเอกลักษณ์ คือข้อมูลไม่ซ้ำกันและไม่ใช่ข้อมูลว่าง เนื่องจากในบางครั้งการสร้างคีย์หลักจากคอลัมน์เดียวอาจมีโอกาสที่จะเกิดข้อมูลซ้ำกันได้ เช่นจาก รีเลชัน “นิสิต” ประกอบด้วย ชื่อ + สกุล + คณะวิชา + สาขาวิชา สามารถจัดเป็นคีย์รวมได้หลายแบบ เช่น

- รหัสนิสิต + ชื่อ
- รหัสนิสิต + สกุล

- รหัสนิสิต + คณะวิชา
- รหัสนิสิต + สาขาวิชา
- รหัสนิสิต + ชื่อ + สกุล

คีย์หลัก

รีเลชัน “ลงทะเบียน”

รหัสนิสิต	ชื่อ	สกุล	รหัสวิชา	ชื่อรายวิชา	หน่วยกิต
580103	น.ส.นารี	สดใส	253001	คณิตศาสตร์	4
580314	นายสะอาด	ใจดี	260003	ภาษาอังกฤษ	3
580619	นายมานะ	อดทน	000101	ทักษะชีวิต	3
580907	น.ส.วิไล	โสภา	260003	ภาษาอังกฤษ	3

คีย์รวม

ภาพที่ 3.3 แสดงตัวอย่างการกำหนดคีย์รวม

4) คีย์คู่แข่ง (Candidate Key) เป็นคีย์ที่เล็กที่สุดที่มีคุณสมบัติใกล้เคียงกับคีย์หลัก สามารถแยกความแตกต่างของข้อมูลแต่ละทูเพิลได้และสามารถใช้แทนคีย์หลักได้ คีย์คู่แข่งบางคีย์อาจประกอบด้วยแอททริบิวต์เดียวหรือมากกว่าหนึ่งแอททริบิวต์ก็ได้ ที่ระบุแต่ละทูเพิลได้อย่างไม่ซ้ำกันในแต่ละรีเลชัน หรือที่มีคุณสมบัติครบถ้วนจนนำมาใช้เป็นคีย์หลัก การกำหนดคีย์คู่แข่งควรให้มีจำนวนแอททริบิวต์ที่น้อยที่สุดที่ประกอบกันเป็นคีย์หนึ่งคีย์ ในแต่ละรีเลชันสามารถมีคีย์คู่แข่งได้มากกว่า 1 คีย์

คุณลักษณะของคีย์คู่แข่ง

1. เป็นซูเปอร์คีย์ ที่มีคอลลัมน์เดียว และไม่มีค่าว่าง
2. เป็นซูเปอร์คีย์ ที่ประกอบไปด้วยหลายแอททริบิวต์ และมีค่าไม่ซ้ำกันในแต่ละแถว

จากรีเลชัน “นิสิต” เราสามารถจัดเป็น คีย์คู่แข่ง ได้ดังนี้

1. รหัสนิสิต ตรงตามเงื่อนไขข้อที่หนึ่ง คือเป็น ซูเปอร์คีย์ ที่มีคอลลัมน์เดียว
2. ชื่อ + สกุล + คณะวิชา + สาขาวิชา ตรงตามเงื่อนไขข้อ 2 คือไม่มีบางส่วนของคีย์ที่มีค่าซ้ำกันในแต่ละแถว เช่น จากรีเลชัน “นิสิต” คีย์หลักคือ รหัสนิสิตส่วน ชื่อ สกุล คณะวิชา และ สาขาวิชา ก็ไม่ซ้ำเช่นกันแต่ไม่ได้จัดเป็นคีย์หลัก จึงจัดเป็นคีย์คู่แข่ง

คีย์คู่แข่ง

คีย์หลัก

รหัสนิสิต	ชื่อ	สกุล	คณะวิชา	สาขา
580103	น.ส.นารี	สดใส	พยาบาล	การพยาบาล
580314	นายสะอาด	ใจดี	วิทยาศาสตร์	ชีววิทยา
580619	นายมานะ	อดทน	วิศวกรรมศาสตร์	วิศวกรรมไฟฟ้า
580907	น.ส.วิไล	โสกรา	สหเวชศาสตร์	กายวิภาค

ภาพที่ 3.4 แสดงตัวอย่างการกำหนดคีย์คู่แข่ง

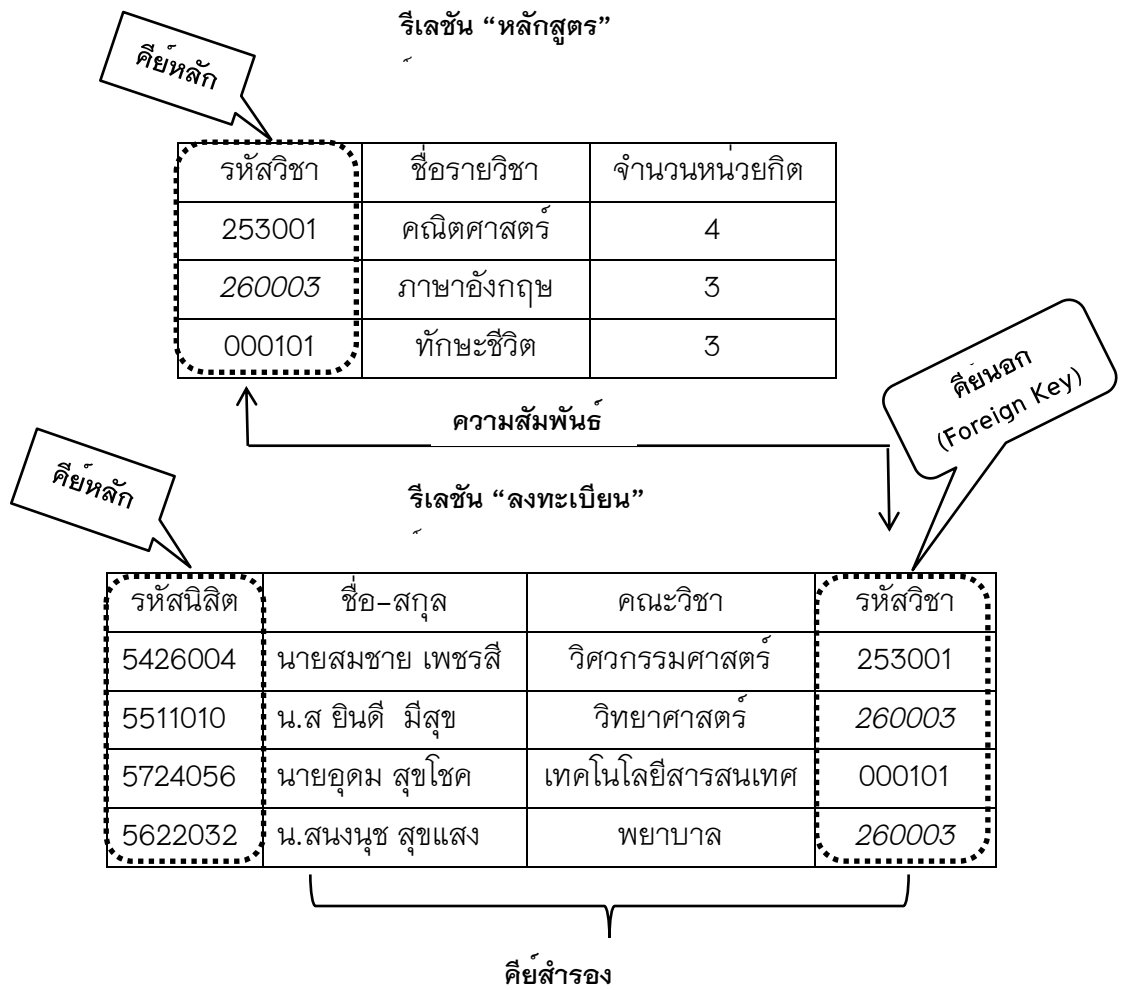
5) **คีย์นอก (Foreign Key)** เป็นคีย์ที่ใช้เชื่อมโยงหรือแสดงความสัมพันธ์ของรีเลชันที่เกี่ยวข้องกันอย่างน้อย 2 รีเลชัน และใช้ในการอ้างอิงถึงแอททริบิวต์เดียวกันนี้ในอีกรีเลชันหนึ่ง คีย์นอกสามารถมีค่าซ้ำกันหรือเป็นค่าว่างก็ได้ และไม่จำเป็นต้องมีชื่อเหมือนกับคีย์หลักของอีกรีเลชันหนึ่งที่มีความสัมพันธ์กัน

รีเลชันหรือตารางหนึ่ง ๆ อาจจะมีหรืออาจจะไม่มีคีย์นอกก็ได้ เช่น รีเลชัน “หลักสูตร” ซึ่งกำหนดให้รหัสวิชา เป็นคีย์หลักและรีเลชัน “ลงทะเบียน” ซึ่งกำหนดให้รหัสนิสิต เป็นคีย์หลัก ถ้าต้องการทราบชื่อรายวิชา และจำนวนหน่วยกิต รายวิชาที่นิสิตแต่ละคนลงทะเบียน สามารถทำได้กำหนดคีย์นอก (Foreign Key) ขึ้นเพื่อเชื่อมความสัมพันธ์ของทั้งสองรีเลชันเข้าด้วยกัน โดยการกำหนดให้ฟิลด์รหัสวิชา ในรีเลชัน “ลงทะเบียน” เป็นคีย์นอกที่ไปสัมพันธ์กับฟิลด์รหัสวิชา ในรีเลชัน “ลงทะเบียน” ที่เป็นคีย์หลัก ในลักษณะความสัมพันธ์แบบ *หนึ่งต่อกลุ่ม* นั่นก็หมายความว่า รหัสวิชา 1 รหัส สามารถให้นิสิตลงทะเบียนได้มากกว่า 1 คน ดังนั้นจึงมีรหัสวิชาซ้ำกันได้ ในรีเลชัน “ลงทะเบียน”

6) คีย์สำรอง หรือ คีย์รอง หรือ ดัชนี (Secondary/Alternate Key or Index)

ในกรณี ที่มีคีย์คู่แข่งหลายคีย์ คีย์สำรองจะหมายถึงคีย์คู่แข่งอื่น ๆ ที่เหลือ ที่ไม่ได้ถูกเลือกให้เป็นคีย์หลักของ รีเลชัน ดังนั้นคีย์สำรองจึงเป็นคีย์ที่สามารถใช้แยกแยะข้อมูลได้เช่นเดียวกันกับคีย์หลัก แต่จะไม่ใช้คีย์สำรองในการเชื่อมโยงข้อมูลระหว่างตาราง เช่น รีเลชัน ข้อมูลนิสิต มีฟิลด์รหัสนิสิต รหัสบัตรประชาชน ชื่อ นามสกุล สาขาวิชา จะเห็นได้ว่าฟิลด์รหัสนิสิต และ ฟิลด์รหัสบัตรประชาชน เป็นฟิลด์ที่มีคุณสมบัติที่มีข้อมูลที่เป็นเอกลักษณ์ จะเป็นค่าว่างไม่ได้ ซึ่งฟิลด์เหล่านี้สามารถถูกเลือกเป็นคีย์หลักได้ หากเลือกให้ฟิลด์รหัสนิสิตเป็นคีย์หลัก

ฟิลต์รหัสบัตรประชาชนจะเป็นกลายจะคีย์สำรอง หรือในทางเดียวกันหากเลือกให้ฟิลต์รหัสบัตรประชาชนเป็นคีย์หลัก ฟิลต์รหัสนิติบุคคลก็จะกลายเป็นคีย์สำรองเป็นต้น



ภาพที่ 3.5 แสดงตัวอย่างการกำหนดคีย์นอก

4. กฎความคงสภาพของข้อมูล (Data Integrity Rules)

เนื่องจากโครงสร้างแบบจำลองฐานข้อมูลเชิงสัมพันธ์ จะประกอบด้วยหลาย ๆ รีเลชัน ดังนั้นจำเป็นต้องมีการควบคุมข้อมูลให้มีความถูกต้อง เป็นจริง และสามารถนำมาใช้เชื่อมโยงกันได้ กฎของความคงสภาพของข้อมูล จะเป็นข้อกำหนดที่ใช้ในการควบคุมความถูกต้องของฐานข้อมูล ทำให้ข้อมูลในระบบฐานข้อมูลมีความถูกต้องอยู่เสมอ การเปลี่ยนแปลงข้อมูลจากผู้ใช้งานข้อมูล หรือโดยระบบจัดการฐานข้อมูลจะต้องมีผลให้ข้อมูลและความสัมพันธ์ระหว่างข้อมูลถูกต้องตามโครงสร้างของฐานข้อมูล

ในการรักษาความถูกต้องของข้อมูลในฐานข้อมูลจะต้องเป็นไปตามกฎเกณฑ์ที่ถูกต้องได้แก่กฎเกณฑ์ในเรื่องความถูกต้องของตัวข้อมูลเอง กฎเกณฑ์ในเรื่องความสัมพันธ์ระหว่าง

ข้อมูลและกฎเกณฑ์ ในการประมวลผลรายการเปลี่ยนแปลงในฐานข้อมูล ซึ่งจะป้องกันไม่ให้ข้อมูลผิดไปจากความเป็นจริง ความถูกต้องของข้อมูลจะถูกบังคับโดยกฎเกณฑ์ซึ่งมี DBMS คอยดูแลตรวจสอบ

การสูญเสียความคงสภาพของข้อมูล เกิดมาจากหลายสาเหตุด้วยกัน เช่น การเพิ่ม การลบ หรือการแก้ไขข้อมูลในรีเลชันหนึ่งที่มีความสัมพันธ์กับอีกรีเลชันหนึ่ง ทำให้เกิดความไม่สอดคล้องกันของข้อมูลระหว่างรีเลชันที่เกี่ยวข้องกัน

แบบจำลองฐานข้อมูลเชิงสัมพันธ์มีกฎความคงสภาพที่สำคัญ 2 ข้อ ได้แก่

1) กฎความคงสภาพของเอนติตี (Entity Integrity Rule) มีใจความสำคัญคือ “ค่าของคีย์หลักจะเป็นค่าว่างไม่ได้” และ “คีย์หลักของรีเลชัน จะต้องมียุคข้อมูลที่ไม่ซ้ำกัน”

เนื่องจากคุณสมบัติของคีย์หลักจะเป็นแอททริบิวต์ที่ใช้ในการเจาะจงแถวข้อมูลใด ๆ ในรีเลชัน ดังนั้นค่าข้อมูลของคีย์หลักจะเป็นค่าว่างไม่ได้ และคีย์หลักของรีเลชันจะต้องมีค่าข้อมูลที่ไม่ซ้ำกัน คือจะต้องมีคุณสมบัติที่เป็นเอกลักษณ์ สามารถระบุข้อมูลแอททริบิวต์อื่น ๆ ที่อยู่ในทูเพิลเดียวกันได้ เช่น รีเลชัน “นิสิต” มีแอททริบิวต์ “รหัสนิสิต” ทำหน้าที่เป็นคีย์หลัก จากกฎความคงสภาพของข้อมูลข้อนี้ จึงทำให้แอททริบิวต์ “รหัสนิสิต” ไม่สามารถมีค่าว่างได้ และทุกรีเลชันต้องมีหนึ่งคีย์หลัก

2) กฎความคงสภาพของการอ้างอิง (Referential Integrity Rule) กฎข้อนี้จะเกี่ยวข้องกับสร้างความสัมพันธ์ระหว่างรีเลชัน 2 รีเลชันในฐานข้อมูลเชิงสัมพันธ์ ซึ่งสามารถกำหนดด้วยคีย์นอก โดยมีนิยามดังนี้ “ค่าของคีย์นอกในรีเลชัน จะต้องมียุคข้อมูลอยู่ในอีก รีเลชันหนึ่ง ที่คีย์นอกของรีเลชันนั้นอ้างอิงถึง” หมายความว่าถ้ารีเลชันใดมีแอททริบิวต์ที่เป็นคีย์นอกอยู่ ข้อมูลที่เป็นคีย์นอกนั้นจะต้องมีค่าตรงกับข้อมูลข้อมูลของคีย์หลักของอีกรีเลชันที่มีความสัมพันธ์กัน หรืออาจให้ข้อมูลที่เป็นคีย์นอกนั้นจะต้องมีค่าเป็นค่าว่าง หากมีการแก้ไขหรือลบข้อมูลของรีเลชัน “หลักสูตร” ในแอตทริบิวต์ รหัสวิชา ซึ่งมีความสัมพันธ์อยู่กับรีเลชัน “ลงทะเบียน” จะทำให้ความสัมพันธ์ของข้อมูลเสียหาย ดังนั้นจึงต้องเลือก การกระทำเพื่อไม่ให้ความสัมพันธ์ของข้อมูลสูญเสียไป

ในการแก้ไขหรือการลบฟิลด์ข้อมูลที่เป็นคีย์หลักในรีเลชันที่สัมพันธ์กันจะไม่มีผลกระทบถึงคีย์นอกในอีกรีเลชันหนึ่ง ระบบจัดการฐานข้อมูลจะยอมให้สามารถกระทำได้ใน 3 เงื่อนไข ดังนี้

เงื่อนไขที่ 1 Restrictionหรือการกระทำแบบมีข้อจำกัด เป็นโอเปอเรเตอร์ทางคณิตศาสตร์ที่ใช้ในการเลือกข้อมูลจากรีเลชันหนึ่ง ๆ ที่มีเงื่อนไขตรงตามที่ระบุไว้ กล่าวอีกนัยหนึ่งคือ ใช้ในการแสดงข้อมูลของทูเพิลที่เป็นไปตามเงื่อนไขที่ระบุไว้ เช่น $=$ (เท่ากับ) $<$ (น้อยกว่า) $>$ (มากกว่า) $<=$ (ไม่เท่ากับ) เป็นต้น ในกรณีที่มีเงื่อนไขมากกว่าหนึ่งเงื่อนไขอาจใช้คำว่า OR (หรือ) AND (และ) ประกอบกันเป็นเงื่อนไขที่ต้องการได้ โดยไม่ยอมให้ทำการแก้ไข

(update) หรือลบ (delete) ข้อมูลในฟิลด์ที่เป็นคีย์หลัก นอกจากจะมีเงื่อนไขกำหนดไว้จึงจะทำได้ ดังนี้

การแก้ไขข้อมูล

- ห้ามทำการแก้ไขข้อมูลในรีเลชันที่ถูกอ้างอิง เนื่องจากจะทำให้ข้อมูลในรีเลชันที่อ้างอิงมา ไม่สามารถอ้างอิงข้อมูลได้
- อนุญาตให้ทำการแก้ไขข้อมูลในรีเลชันที่ถูกอ้างอิงถึงได้ แต่จะต้องตามไปแก้ไขข้อมูล ในรีเลชันที่อ้างอิงมาให้ตรงกับข้อมูลที่แก้ไขใหม่ทั้งหมด
- อนุญาตให้ทำการแก้ไขข้อมูลในรีเลชันที่ถูกอ้างอิงถึงได้ โดยการแก้ไขข้อมูลในรีเลชัน ที่อ้างอิงมาให้มีค่าเป็น ค่าว่าง

การลบข้อมูล

- ห้ามทำการลบข้อมูลในรีเลชันที่ถูกอ้างอิงถึงนั้น เนื่องจากจะทำให้ข้อมูลใน รีเลชันที่อ้างอิงมา ไม่สามารถอ้างอิงข้อมูลได้
- อนุญาตให้ทำการลบข้อมูลในรีเลชันที่ถูกอ้างอิงถึงได้ แต่จะต้อง ตามไปลบข้อมูลในรีเลชัน ที่อ้างอิงมาทั้งหมด
- อนุญาตให้ทำการลบข้อมูลในรีเลชันที่ถูกอ้างอิงถึงได้ โดยการแก้ไขข้อมูลในรีเลชัน ที่อ้างอิงมาให้มีค่าเป็น ค่าว่าง (Null value)

เงื่อนไขที่ 2 Cascade หรือ การกระทำแบบต่อเนื่อง ยอมให้กระทำการเปลี่ยนแปลง(update) หรือลบ(delete) ข้อมูลในฟิลด์ที่เป็นคีย์หลักได้ แต่ต้องกระทำอย่างต่อเนื่องในอีกรีเลชันหนึ่งที่มีความสัมพันธ์กัน

เงื่อนไขที่ 3 Nullify หรือการกระทำโดยใส่ค่าว่าง ยอมให้กระทำการเปลี่ยนแปลง(update) หรือลบ(delete) ข้อมูลในฟิลด์ที่เป็นคีย์หลักได้ได้โดยการกำหนดค่าของคีย์นอกให้เป็นค่าว่าง

5. ข้อดีของแบบจำลองฐานข้อมูลเชิงสัมพันธ์

1. เป็นรูปแบบจำลองของฐานข้อมูลที่เข้าใจง่าย เพราะเป็นกลุ่มของรีเลชันหรือตารางที่ข้อมูลได้จัดเก็บเป็นแถวและคอลัมน์ทำให้ผู้ใช้เห็นภาพของข้อมูลได้ง่าย สะดวกต่อการใช้งาน และการเขียนโปรแกรม
2. มีเครื่องมือที่ช่วยให้ผู้ใช้สามารถจัดการกับข้อมูลได้ด้วยคำสั่งง่ายๆ โดยผู้ใช้ไม่ต้องรู้ว่ามียาละเอียดของการจัดเก็บข้อมูลอย่างไร
3. สามารถใช้ภาษาที่ง่ายในการเรียกดูข้อมูล เช่น ภาษา SQL เป็นภาษาที่มีลักษณะคล้ายภาษาอังกฤษ และไม่จำเป็นต้องเขียนเป็นลำดับขั้นตอนของคำสั่งอย่างในภาษาอื่น ๆ

4. การเรียกใช้หรือเชื่อมโยงข้อมูลทำได้ง่าย โดยใช้ Operator ทางคณิตศาสตร์ ไม่จำเป็นต้องใช้พอยน์เตอร์ (Pointer)

5. มีระบบความปลอดภัยที่ดี เนื่องจากโครงสร้างนี้ผู้ใช้งานจะไม่ทราบถึงกระบวนการจัดเก็บข้อมูลภายในฐานข้อมูลแท้จริงว่าเป็นอย่างไร

6. DBMS ที่พัฒนาในปัจจุบันล้วนรองรับเทคโนโลยีฐานข้อมูลเชิงสัมพันธ์

6. ข้อเสียของแบบจำลองฐานข้อมูลเชิงสัมพันธ์

1. ต้องมีการลงทุนสูงเนื่องจากต้องใช้ Hardware และ Software ที่มีความสามารถสูง
2. แนวคิดฐานข้อมูลเชิงสัมพันธ์ในภาพรวมนั้นง่ายต่อการนำไปใช้งาน ดังนั้นบุคลากรที่ไม่ได้รับการฝึกอบรมหรือผู้ที่มีความรู้ไม่ดีพอ ได้นำเครื่องมือไปใช้งานที่ผิด ทำให้ระบบที่ดีต้องแย่ง และหากไม่ได้รับการตรวจสอบ อาจทำให้เกิดข้อมูลซ้ำซ้อนได้เช่นเดียวกับระบบแฟ้ม
