

Data Models

01418221 Fundamentals of Database Systems

ตัวแบบข้อมูล (Data Model)

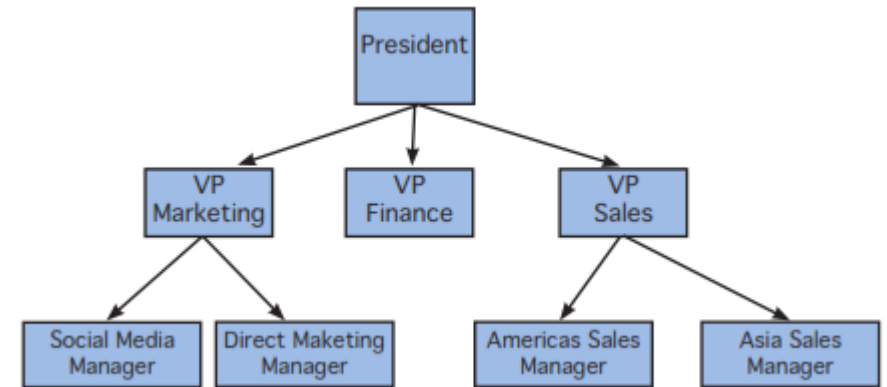
- กำหนดวิธีการสร้างแบบจำลองโครงสร้างเชิงตรรกะของฐานข้อมูล ตัวแบบข้อมูลเป็นเอนทิตีพื้นฐานที่จะนำเสนอนามธรรมใน DBMS
- ตัวแบบข้อมูลกำหนดวิธีการเชื่อมต่อข้อมูลซึ่งกันและกันและวิธีการประมวลผลและจัดเก็บภายในระบบ
- ตัวแบบข้อมูลแรกอาจเป็นตัวแบบข้อมูลแบบเรียบ (flat data-models) ซึ่งข้อมูลทั้งหมดที่ใช้จะถูกเก็บไว้ในระนาบเดียวกัน
 - แบบจำลองข้อมูลก่อนหน้านี้ไม่ได้เป็นหลักวิชาการออกแบบ ดังนั้นพวกเขาจึงมีแนวโน้มที่จะแนะนำการเก็บข้อมูลซ้ำซ้อน (lots of duplication) จำนวนมากและปรับปรุงข้อมูลที่ไม่ถูกต้อง (update anomalies)

ตัวแบบข้อมูล (Data Model)

- แบบจำลองข้อมูล (Data Model) คือ โครงสร้างข้อมูลในระดับตรรกะที่อาจเป็นการนำเสนอข้อมูลในรูปแบบกราฟฟิก หรือรูปแบบอื่น ๆ ที่นำเสนอข้อมูลและความสัมพันธ์ระหว่างข้อมูล
 - การนำเสนอข้อมูลให้อยู่ในรูปแบบที่ง่ายในการทำความเข้าใจ แบบจำลองข้อมูลถูกนำไปใช้โดยนักออกแบบฐานข้อมูล
 - ส่วนประกอบของแบบจำลองข้อมูลคือ
 - ส่วนโครงสร้าง (Structural Part)
 - ส่วนการจัดการ (Manipulative Part)
 - กฎความคงสภาพ (Integrity Rules)
 - ตัวอย่างแบบจำลองข้อมูล
 - แบบจำลองข้อมูลเชิงลำดับชั้น (Hierarchical Data Model)
 - แบบจำลองข้อมูลแบบเครือข่าย (Network Data Model)
 - แบบจำลองข้อมูลเชิงสัมพันธ์ (Relation Data Model)
 - แบบจำลองข้อมูลเชิงวัตถุ (object Data Model)

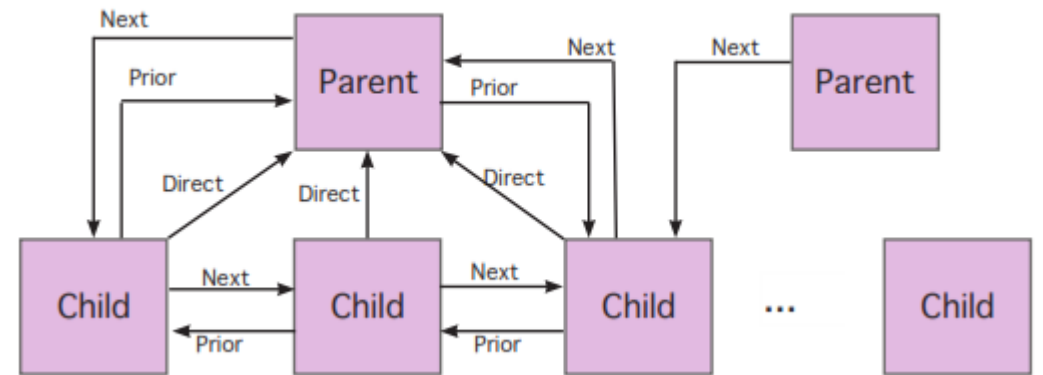
แบบจำลองข้อมูลเชิงลำดับชั้น (Hierarchical Data Model)

- แบบจำลองที่นำเสนอความสัมพันธ์ระหว่างข้อมูลในรูปแบบของโครงสร้างต้นไม้ โดยเรคอร์ดที่อยู่ระดับบนสุดเรียกว่า root
- ข้อดี
 - โครงสร้างเข้าใจง่าย
 - โครงสร้างที่เหมาะสมกับข้อมูลที่มีความสัมพันธ์แบบ One-To-Many
 - ป้องกันความพลวัติในข้อมูลที่ดี
 - เหมาะกับข้อมูลที่มีการเรียงลำดับแบบต่อเนื่อง
- ข้อเสีย
 - ไม่สามารถรองรับข้อมูลที่มีความสัมพันธ์ในลักษณะ Many-To-Many
 - เรียกใช้ข้อมูลจำเป็นต้องผ่าน Root เสมอ
 - การพัฒนาโปรแกรมค่อนข้างยาก



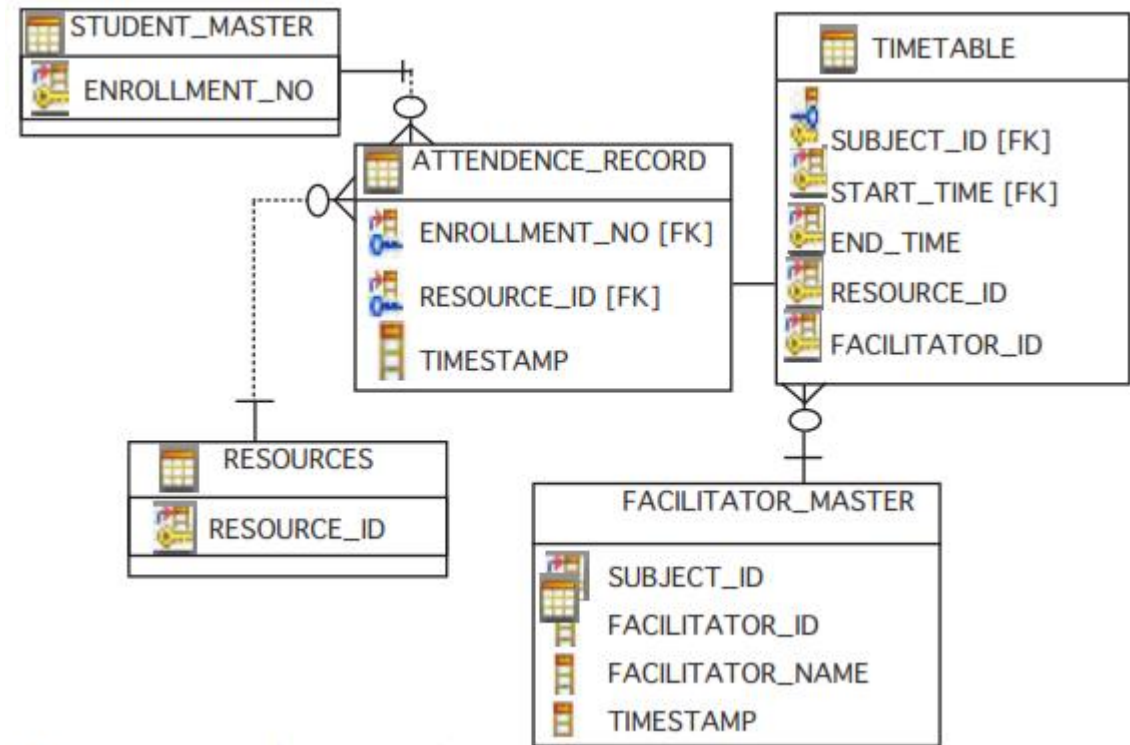
แบบจำลองข้อมูลแบบเครือข่าย (Network Data Model)

- การนำเสนอความสัมพันธ์ระหว่างข้อมูลในรูปแบบมัลติลิสต์ หรือเชื่อมโยงระหว่างจุดข้อมูล
- ข้อดี
 - สนับสนุนความสัมพันธ์แบบ Many-To-Many
 - ความซับซ้อนในข้อมูลเกิดขึ้นน้อยกว่าแบบจำลอง
 - สามารถเชื่อมโยงข้อมูลไป-กลับได้
 - มีความยืดหยุ่นในด้านการหาข้อมูล
- ข้อเสีย
 - สามารถเข้าถึงเรคอร์ดได้โดยตรง ทำให้การป้องกันข้อมูลมีน้อย
 - สิ้นเปลืองเนื้อที่หน่วยความจำ
 - การเปลี่ยนแปลงในโครงสร้างข้อมูลทำได้ยาก



แบบจำลองข้อมูลเชิงสัมพันธ์ (Relational Data Model)

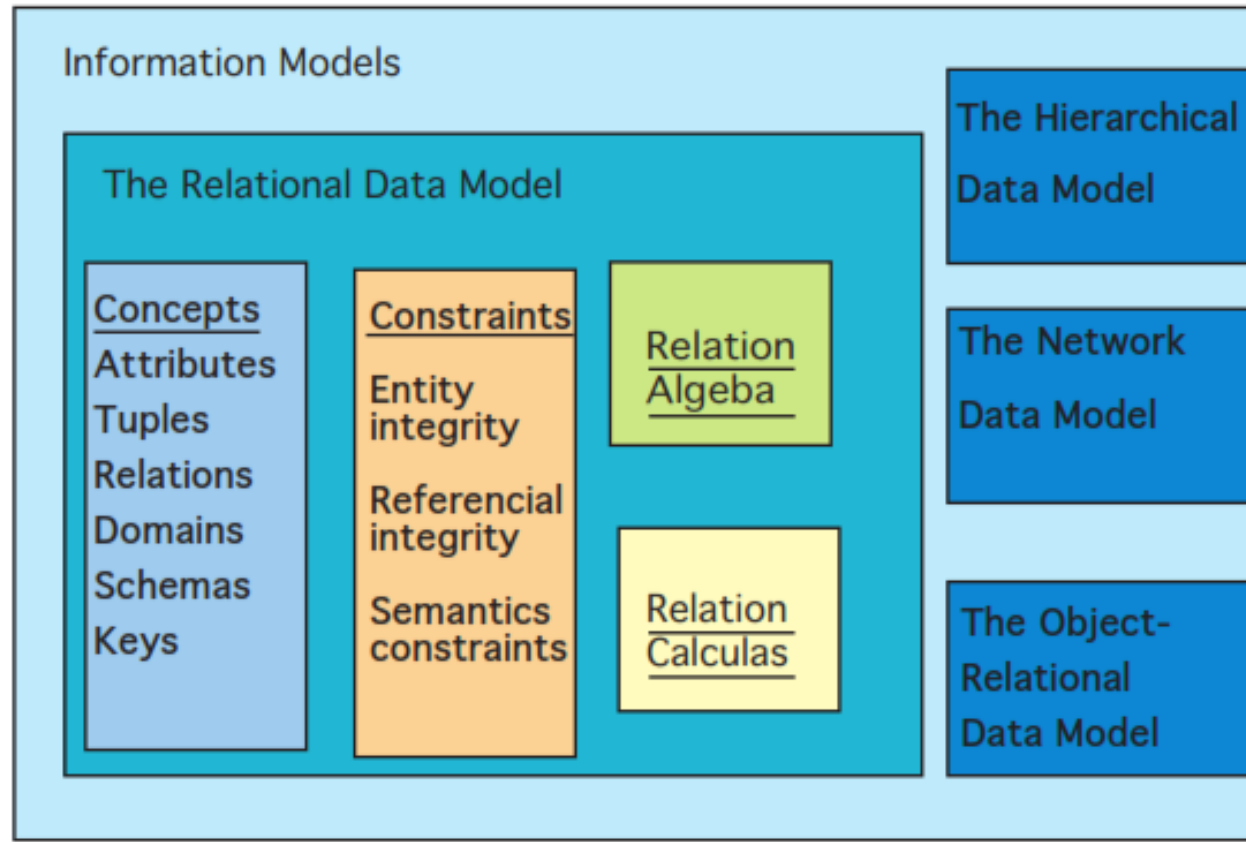
- การนำเสนอความสัมพันธ์ระหว่างข้อมูลในรูปรีเลชันที่เป็นตารางสองมิติ ประกอบด้วยแถว คอลัมน์
- ข้อดี
 - มีความเข้าใจและสื่อสารได้เข้าใจง่าย
 - สามารถเลือกข้อมูลตามเงื่อนไขได้หลายคีย์ฟิลต์
 - ความซับซ้อนในข้อมูลมีน้อย
 - โครงสร้างข้อมูลมีความอิสระจากโปรแกรม
 - มีระบบความปลอดภัยที่ดี
- ข้อเสีย
 - จำเป็นต้องเสียค่าใช้จ่ายในระบบค่อนข้างสูง



แบบจำลองข้อมูลเชิงวัตถุ (Object Data Model)

- แบบจำลองที่เกิดจากแนวคิดของการเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming: OOP) ที่มีการมองของทุกสิ่งเป็นวัตถุ
 - โดยแต่ละวัตถุจะเป็นแหล่งรวมของข้อมูล และการทำงาน (Procedure) ที่มีคลาสเป็นตัวกำหนดคุณสมบัติหรือรายละเอียดของวัตถุ รวมทั้งคุณสมบัติปกปิดความลับของวัตถุ (Encapsulation)
- ข้อดี
 - สามารถจัดการกับข้อมูลชนิดต่าง ๆ ที่มีความสลับซับซ้อนได้เป็นอย่างดี
 - มีคุณสมบัติการสืบทอด ทำให้ข้อมูลมีความคงสภาพสูง
 - มีคุณสมบัติในการกลับมาใช้ใหม่
- ข้อเสีย
 - ต้องการผู้เชี่ยวชาญโดยเฉพาะ และมีค่าใช้จ่ายระบบค่อนข้างสูง
 - ยังไม่มีมาตรฐานรองรับที่ชัดเจน

แบบจำลองข้อมูลเชิงสัมพันธ์ (Relational Data Model)



แบบจำลองข้อมูลเชิงสัมพันธ์ (Relational Data Model)

1. รีเลชัน (Relation) เป็นโครงสร้างของแบบจำลองเชิงข้อมูลเป็นตาราง 2 มิติ ซึ่งประกอบไปด้วย แถว (Row) และคอลัมน์ (Column)

CARS Relations

Attribute

คอลัมน์หรือ สดมภ์ (Column, Attribute, Field)

Header {

Body {

Tuple

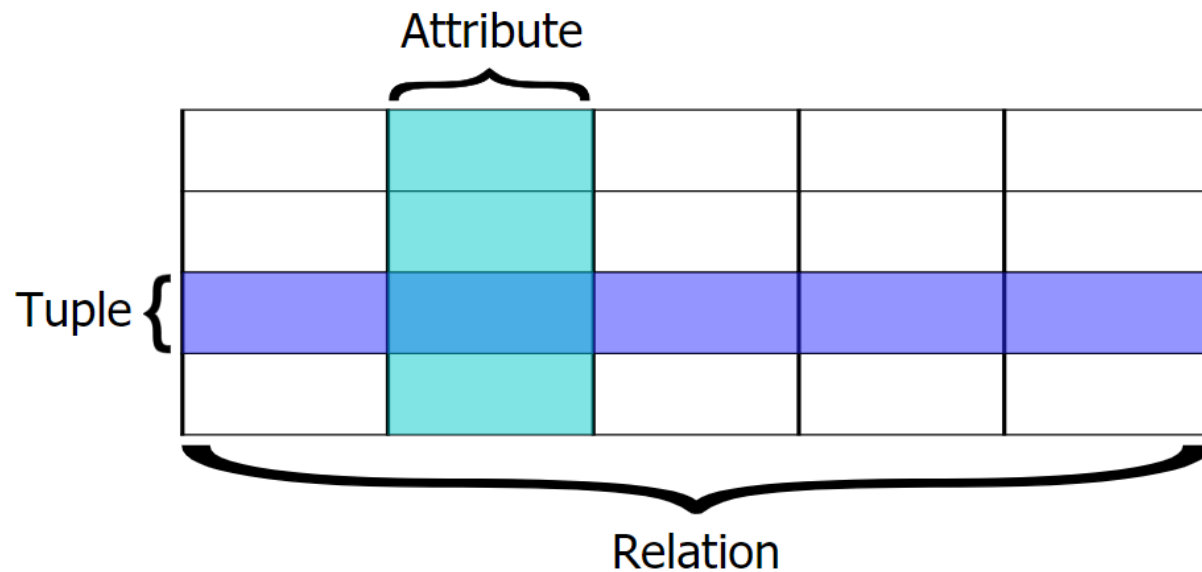
แถว (Row, Tuple, Record)

ตาราง (Table, Relation, File)

Value

TYPE	PRODUCER	MODEL	FABRICATION YEAR	COLOR	FUEL
LIMOUSINE	BMV	740	2008	BLACK	GAS
VAN	VW	TRANSPORTER	2007	RED	DIESEL
LIMOUSINE	MERCEDES	320	2008	WHITE	GAS
LIMOUSINE	AUDI	ALLROAD	2009	BLUE	DIESEL
LIMOUSINE	BMW	525	200	GREY	DIESEL

คุณสมบัติของรีเลชัน (Relation Properties)



https://en.wikipedia.org/wiki/Relational_database

- รีเลชันต้องมีชื่อเพื่อใช้ในการอ้างอิง และชื่อรีเลชันต้องไม่ซ้ำกัน
- แอตทริบิวต์ในรีเลชันต้องมีชื่อ และต้องเป็นชื่อที่แตกต่างกัน
- ลำดับก่อนหลังของแอตทริบิวต์ในรีเลชันไม่มีความสำคัญ
- ลำดับก่อนหลังของทัพเพิลในรีเลชันไม่มีความสำคัญ
- แต่ละเซลล์ในรีเลชัน (แถวและคอลัมน์ตัดกัน) ต้องเป็นข้อมูลเดี่ยว หรือ Atomic
- แต่ละทัพเพิลในรีเลชันต้องไม่ซ้ำ
- ข้อมูลในแอตทริบิวต์เดียวกันต้องใช้รูปแบบ (Format) เดียวกัน และอยู่ในโดเมน (Domain) เดียวกัน

กุญแจหรือคีย์ (Key)

- แบบจำลองข้อมูลเชิงสัมพันธ์ใช้โครงสร้างตาราง 2 มิติ การอ้างอิงข้อมูลในระบบทำได้ด้วยการกำหนดให้แอตทริบิวต์ของรีเลชันทำหน้าที่เป็น “คีย์” (Key) เพื่อกำหนด (Determine) ข้อมูลในรีเลชัน
 - คีย์ทำหน้าที่เป็นผู้กำหนด (Determinant) เขียนให้อยู่ในรูป Determination ได้ดังนี้
$$A \rightarrow B$$
 - หมายถึง A กำหนด (Determine) B
 - หรืออาจกล่าวได้ว่า แอตทริบิวต์ A ทำหน้าที่เป็นผู้กำหนด ที่ทำการกำหนดแอตทริบิวต์ B

ซูเปอร์คีย์ (Super Key)

- ซูเปอร์คีย์ (Super Key) คือแอตทริบิวต์หรือกลุ่มของแอตทริบิวต์ที่ระบุความแตกต่างกันของแต่ละทUPLE ในรีเลชันได้ (Uniquely Identifies)

รหัสพนักงาน EmpID	ชื่อ FirstName	นามสกุล LastName	รหัสแผนก DepID	ตำแหน่ง Position
E991005	บุญโชค	เอื้ออำนวย	D1	ผู้จัดการแผนก
E991102	ปรีชา	เปรมปรีดิ์	D2	รองผู้จัดการแผนก
E010109	ธีระพงษ์	ทองมาก	D2	พนักงาน
E020401	ปัญญา	นิคม	D3	พนักงาน
E030202	กัญญา	ทรัพย์สมบัติ	D1	พนักงาน
E031101	จักรินส์	เปรมปรีดิ์	D3	ผู้จัดการแผนก

<https://na5cent.blogspot.com/2011/12/keys-database.html>

1. EmpID
2. EmpID + FirstName
3. EmpID+LastName
4. EmpID+DepID
5. EmpID+Position
6. EmpID+FirstName+LastName
7. EmpID+FirstName+DepID
8. EmpID+FirsttName+Position
9. EmpID+LastName+DepID
10. EmpID+LastName+Position
11. EmpID+DepID+Position
12. FirstName+LastName+DepID+Position

Composite Key

- Composite Key หมายถึง Key ที่ประกอบกัน หรือมีมากกว่า 1 Key เช่น

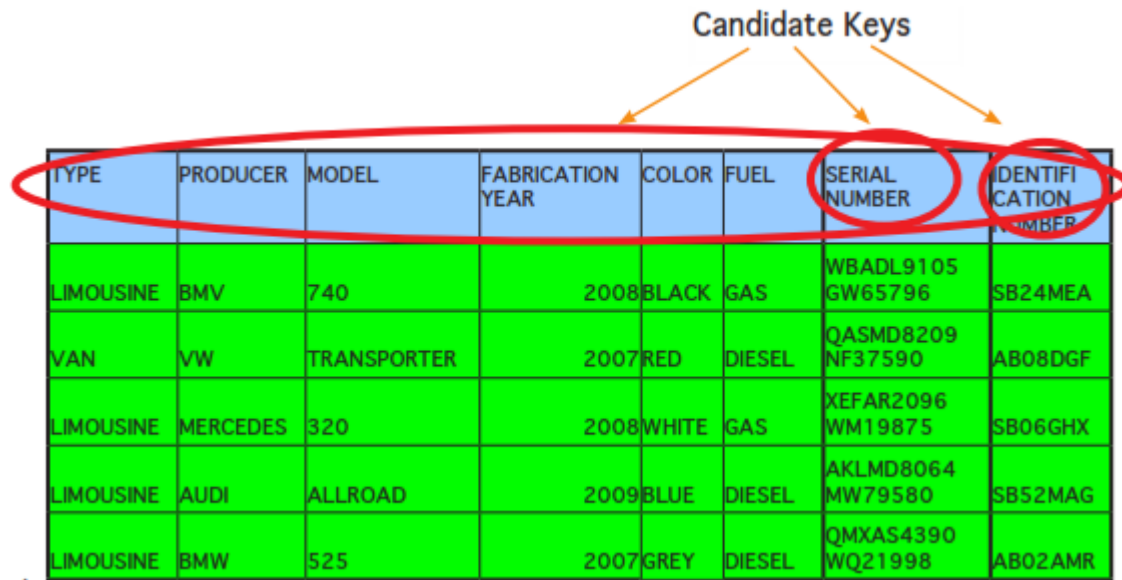
รหัสพนักงาน EmpID	ชื่อ FirstName	นามสกุล LastName	รหัสแผนก DepID	ตำแหน่ง Position
E991005	บุญโชค	เชื้ออ้วนวย	D1	ผู้จัดการแผนก
E991102	ปรีชา	เปรมปรีช	D2	รองผู้จัดการแผนก
E010109	ธีระพงษ์	ทองมาก	D2	พนักงาน
E020401	ปัญญา	มีคม	D3	พนักงาน
E030202	ภิญญา	ทรัพย์สมบัติ	D1	พนักงาน
E031101	จักรินส์	เปรมปรีช	D3	ผู้จัดการแผนก

1. EmpID + FirstName
 2. EmpID+LastName
 3. EmpID+FirstName+LastName
 4. FirstName+LastName+DepID+Position
- แต่ EmpID ไม่ใช่ Composite Key เนื่องจากเป็นเพียงแค่ Key เดียว

<https://na5cent.blogspot.com/2011/12/keys-database.html>

คีย์คู่แข่ง (Candidate Key)

Candidate Keys



TYPE	PRODUCER	MODEL	FABRICATION YEAR	COLOR	FUEL	SERIAL NUMBER	IDENTIFICATION NUMBER
LIMOUSINE	BMV	740	2008	BLACK	GAS	WBADL9105 GW65796	SB24MEA
VAN	VW	TRANSPORTER	2007	RED	DIESEL	QASMD8209 NF37590	AB08DGF
LIMOUSINE	MERCEDES	320	2008	WHITE	GAS	XEFAR2096 WM19875	SB06GHX
LIMOUSINE	AUDI	ALLROAD	2009	BLUE	DIESEL	AKLMD8064 MW79580	SB52MAG
LIMOUSINE	BMW	525	2007	GREY	DIESEL	QMXAS4390 WQ21998	AB02AMR

- คีย์คู่แข่ง (Candidate Key) คือซัพเปอร์คีย์ขนาดเล็กที่สุดที่สามารถระบุความแตกต่างของแต่ละทUPLEในรีเลชันได้ เราสามารถตรวจสอบว่าแอตทริบิวต์ที่รวมกันเป็นคีย์คู่แข่งนั้นเป็นจำนวนน้อยที่สุดหรือไม่
- แอตทริบิวต์ที่เป็นคีย์คู่แข่งต้องมีคุณสมบัติ 2 ข้อต่อไปนี้
 - ยูนิคเนส (Uniqueness) แอตทริบิวต์ใด ๆ ที่จะเป็นคีย์ได้ ต้องมีค่าที่มีความแตกต่างกันหรือไม่มีค่าซ้ำกัน
 - มินิมอลลิตี (Minimality) จะมีจำนวนแอตทริบิวต์ที่น้อยที่สุด
- แคนดิเดตคีย์ในบางครั้งเรียกว่า ยูนิคคีย์ (unique key) ยูนิคคีย์ถูกใช้ในภาษากำหนดโครงสร้าง (DDL) ที่มีการใช้พารามิเตอร์ UNIQUE ตามหลังชื่อของแอตทริบิวต์
 - ถ้ารีเลชันมีแคนดิเดตคีย์มากกว่าหนึ่งตัว หนึ่งในนั้นจะถูกเลือกให้เป็นคีย์หลัก (primary key) ที่เหลือจะถูกใช้เป็นคีย์ทดแทนเรียกว่า อัลเทอเนตคีย์ (alternate keys)

คีย์หลัก (Primary Key)

- คีย์คู่แข่งที่ถูกเลือกให้เป็นคีย์หลักเพื่อใช้ในการระบุหรืออ้างอิงข้อมูลที่ต้องการในรีเลชัน คีย์หลักเป็นสิ่งที่ใช้แสดงความเป็นหนึ่งเดียวของแต่ละทูเปิลในรีเลชัน
 - คีย์หลักคือหนึ่งในคีย์คู่แข่งที่ถูกเลือกของรีเลชันในฐานะข้อมูลเพื่อระบุความเป็นหนึ่งเดียวของแต่ละทูเปิลในรีเลชัน รีเลชันในฐานะข้อมูลควรมีการระบุคีย์หลักเสมอ
- ระบบจัดการฐานข้อมูล (DBMS) สามารถกำหนดคีย์หลักในขั้นตอนการสร้างรีเลชัน (ตารางข้อมูล) ภาษาที่ใช้สำหรับกำหนดโครงสร้าง (DDL) จะใช้คำว่า PRIMARY KEY ในการกำหนดคีย์หลัก
 - ค่าของแอตทริบิวต์ดังกล่าวต้องเป็นค่ายูนิคและไม่เป็นค่าว่าง (NOT NULL) สำหรับทุกทูเปิลในรีเลชัน
- คีย์รอง (Alternate Key) คือคีย์คู่แข่งตัวใด ๆ ที่ไม่ได้ถูกเลือกเป็นคีย์หลัก

คีย์นอก (Foreign Key)

Primary key						Foriegn key	
TYPE	PRODUCER	MODEL	FABRICATION YEAR	COLOR	FUEL	SERIAL NUMBER	IDENTIFI CATION NUMBER
LIMOUSINE	BMV	740	2008	BLACK	GAS	WBADL9105 GW65796	SB24MEA
VAN	VW	TRANSPORTER	2007	RED	DIESEL	QASMD8209 NF37590	AB08DGF
LIMOUSINE	MERCEDES	320	2008	WHITE	GAS	XEFAR2096 WM19875	SB06GHX
LIMOUSINE	AUDI	ALLROAD	2009	BLUE	DIESEL	AKLMD8064 MW79580	SB52MAG
LIMOUSINE	BMW	525	2007	GREY	DIESEL	QMXAS4390 WQ21998	AB02AMR

คีย์นอก (Foreign Key) คือแอตทริบิวต์หรือกลุ่มของแอตทริบิวต์ของรีเลชันที่มีค่าตรงกันกับค่าของคีย์หลักของรีเลชันอื่น

- ความสัมพันธ์คีย์นอกไปยังคีย์หลัก (Foreign-to-primary-key) ที่ตรงกันเป็นการอ้างอิงรีเลชันหนึ่งไปยังอีกรีเลชันหนึ่ง เหมือนเป็นกาวที่ยึดโยงรีเลชันต่างๆไว้ด้วยกันในฐานข้อมูล
- ความสัมพันธ์คีย์นอกไปยังคีย์หลักเป็นความสัมพันธ์ที่เชื่อมโยงกันระหว่างทุเปิล

กฎความคงสภาพ (Integrity Rules)

- รีเลชันต่าง ๆ ในระบบฐานข้อมูลเชิงสัมพันธ์ มีการอ้างอิงความสัมพันธ์ของแต่ละรีเลชันโดยใช้คีย์ในการอ้างอิง หรือใช้คีย์เป็นตัวเชื่อมโยงระหว่างรีเลชัน จึงจำเป็นต้องมีกฎความคงสภาพเพื่อเป็นการควบคุมความถูกต้องในระบบฐานข้อมูล
- ในการสร้างแบบจำลองข้อมูลเชิงสัมพันธ์ กฎควบคุมความถูกต้อง (integrity rules) หรือเงื่อนไขการควบคุมข้อมูล (constraint) เป็นส่วนหนึ่งที่สำคัญในการกำหนดความถูกต้องของข้อมูล
- ถ้าต้องการให้ฐานข้อมูลเชิงสัมพันธ์มีความถูกต้องเชิงโครงสร้าง (schema level) โดยการควบคุมข้อมูลถูกใช้ในระดัการออกแบบโครงสร้างฐานข้อมูล ซึ่งถ้าผู้ใช้พยายามที่จะละเมิดการควบคุมข้อมูลดังกล่าวระบบจะทำการปฏิเสธการดำเนินการนั้นหรือหาทางเลือกอื่นให้กับผู้ใช้เพื่อไม่ให้กระทบต่อข้อมูลในฐานข้อมูล ในลำดับต่อไปจะอธิบายถึงการควบคุมข้อมูลต่างๆที่ใช้ในฐานข้อมูลเชิงสัมพันธ์

กฎความคงสภาพ (Integrity Rules)

- การควบคุมข้อมูลเอนทิตีอินทิกริตี (Entity Integrity) เป็นกฎการควบคุมให้รีเลชันสามารถอ้างอิงค่าคีย์ได้ตลอดเวลา การควบคุมข้อมูลเอนทิตีอินทิกริตี เป็นการกำหนดว่าแอตทริบิวต์ที่ถูกใช้เป็นคีย์หลักในแต่ละรีเลชันต้องโดยมีเงื่อนไขดังนี้
 - ค่าคีย์หลักในแถวข้อมูลจะต้องไม่ซ้ำกัน กล่าวคือ ค่าใดค่าหนึ่งจะแสดงเพียงแถวเดียว ไม่ตรงกับคีย์หลักของแถวอื่น ๆ ในตาราง
 - ค่าคีย์หลักไม่อนุญาตให้เป็นค่าว่าง (null values) คำว่า Null หมายถึง ไม่มีค่า (property inapplicable) หรือค่าที่ไม่รู้จักมาก่อน (information unknown) Null เป็นเหมือนเครื่องหมายแสดงการขาดหายไปของค่า หรือเป็นค่าที่ไม่สามารถระบุได้

กฎความคงสภาพ (Integrity Rules)

artist_id	artist_name
1	Bono
2	Cher
3	Nuno Bettencourt

Link Broken

artist_id	album_id	album_name
3	1	Schizophonic
4	2	Eat the rich
3	3	Crave (single)

- การควบคุมข้อมูลเรเฟอเรนเชียลอินTEGRITY (Referential integrity) เป็นการกำหนดเงื่อนไขสำหรับความสัมพันธ์ของรีเลชัน กล่าวว่
- ถ้ารีเลชัน R2 มีคีย์นอกคือ FK อ้างอิงหรือตรงกับคีย์หลัก PK ของรีเลชันอื่นคือ รีเลชัน R1
 - ทุก FK ใน R2 ต้องเท่ากับค่า PK ใน R1 เสมอ หรือเป็นค่าว่าง (ค่าแอตทริบิวต์ที่เป็น FK เป็นค่าว่าง)
- เรเฟอเรนเชียลอินTEGRITYมีที่มาได้จากหลายสาเหตุดังต่อไปนี้
 - ถ้าทูปเปิล t2 จากรีเลชัน R2 อ้างอิงถึงบางทูปเปิล t1 จากรีเลชัน R1 ดังนั้นต้องมีค่า t1 อยู่ก่อน ไม่อย่างนั้นการอ้างอิงก็ไม่สามารถทำได้
 - ดังนั้นค่าของคีย์นอกต้องตรงกับค่าของคีย์หลัก ในการอ้างอิงผ่านความสัมพันธ์ คีย์นอกจึงควรเป็นค่าที่ไม่ใช่ค่าว่าง
 - บางครั้งในทางปฏิบัติจะอนุญาตให้คีย์นอกเป็นค่าว่างได้

กฎความคงสภาพ (Integrity Rules)

- สำหรับการระบุค่าคีย์นอกในฐานข้อมูลนั้น นักออกแบบฐานข้อมูลต้องพิจารณาประเด็นสำคัญ 3 ข้อ ดังต่อไปนี้
 1. สามารถยอมรับค่าว่าง (Null Value) ได้หรือไม่ สำหรับคำตอบในข้อนี้ นักออกแบบฐานข้อมูลไม่ต้องการให้เป็นค่าว่าง แต่ในความเป็นจริงอาจเกิดขึ้นโดยการมีการจัดเก็บค่าว่างในฐานข้อมูล
 2. ประเด็นที่ต้องพิจารณาคือจะเกิดอะไรขึ้นหากคีย์หลักที่ คีย์นอกอ้างอิงถึงถูกลบไป โดยปกติการกระทำดังกล่าว มี 3 แนวทางที่เป็นไปได้
 - CASCADE การดำเนินการลบแบบ cascades จะทำการลบทุเป็ดที่เกี่ยวข้องหรืออ้างอิงไปถึงด้วย (ทุเป็ดที่มีความสัมพันธ์ผ่านคีย์นอก) ในกรณีนี้ถ้ารถยนต์ถูกลบทุเป็ดของเจ้าของรถก็จะถูกลบทิ้งด้วย
 - RESTRICT การดำเนินการลบแบบ restricted ทำการลบข้อมูลเมื่อไม่มีการอ้างอิงถึงทุเป็ด (จะทำการยกเลิกการลบทันทีในกรณีที่มีการอ้างอิงข้อมูลผ่านคีย์นอก)
 - NULLIFIES จะเปลี่ยนค่าคีย์นอกให้เป็นค่าว่างในกรณีที่มีการอ้างอิงและทุเป็ดที่เป็นคีย์หลักจะถูก

กฎความคงสภาพ (Integrity Rules)

3. เกิดอะไรขึ้นหากมีการเปลี่ยนแปลงค่า (update) คีย์หลักที่มีค่าคีย์นอกอ้างอิงถึง
- CASCADE การดำเนินการอัปเดตแบบ cascades ทำการอัปเดตค่าของคีย์นอกที่เกี่ยวข้องหรืออ้างอิงมายังทูเปิล(รวมไปถึงทูเปิลจากความสัมพันธ์ของคีย์นอก)
 - RESTRICT การดำเนินการอัปเดตแบบ restricted ทำการอัปเดตข้อมูลเมื่อไม่มีการอ้างอิงถึงทูเปิล (ทำการยกเลิกการอัปเดตหากมีการอ้างอิง)
 - NULLIFIES จะเปลี่ยนค่าคีย์นอกให้เป็นค่าว่างในกรณีที่มีการอ้างอิงและทูเปิลที่มีคีย์หลักจะถูกอัปเดต (ในกรณีนี้ไม่สามารถทำได้หากไม่มีการกำหนดให้คีย์นอกยอมรับค่าว่าง)

กฎความคงสภาพ (Integrity Rules)

- **การควบคุมข้อมูลซีแมนติกอินทิกริตี (Semantic integrity constraints)** ซีแมนติกอินทิกริตีเป็นการควบคุมข้อมูลที่แสดงถึงความถูกต้องของความหมายข้อมูล ขอบเขตและชนิดของข้อมูล
 - ตัวอย่างเช่น แอดทริบิวต์เลขที่ถนน จากรีเลชัน OWNERS ต้องเป็นเลขจำนวนเต็มบวกเพราะในความจริงนั้นเลขที่ถนนต้องเป็นเลขจำนวนเต็มบวกเท่านั้น
- ซีแมนติกอินทิกริตีถือเป็นสิ่งที่ควรปฏิบัติเพื่อความถูกต้องในสถานะของรีเลชันตามความต้องการการจัดเก็บข้อมูลของฐานข้อมูล ถ้าผู้ใช้พยายามที่จะละเมิดการควบคุมข้อมูลดังกล่าว ระบบจะปฏิเสธการดำเนินการดังกล่าวทันที หรือดำเนินการอย่างอื่น เพื่อให้คงความถูกต้องของข้อมูลในฐานข้อมูล
 1. **การควบคุมข้อมูลโดเมน (Domain constraint)** การควบคุมข้อมูลโดเมนเปรียบเสมือนเป็นการกำหนดคุณลักษณะของแอดทริบิวต์ ที่ปรากฏในรีเลชัน การควบคุมข้อมูลโดเมนจะเป็นการระบุค่าเพื่อเป็นการตรวจสอบว่าค่านั้นอยู่ในเซตที่กำหนดไว้ในโดเมนหรือไม่
 1. การควบคุมข้อมูลแบบกำหนดชื่อ (namely format)
 2. การควบคุมข้อมูลแบบขอบเขตของข้อมูล (range)
 3. การควบคุมข้อมูลแบบกำหนดชื่อ
 2. **การควบคุมข้อมูลค่าว่าง (Null value)** การควบคุมข้อมูลค่าว่างเป็นการระบุให้แอดทริบิวต์ไม่สามารถเป็นค่าว่าง
 3. **การควบคุมข้อมูลยูนิก (Unique constraint)** การควบคุมข้อมูลยูนิกเป็นการระบุให้ค่าแอดทริบิวต์ที่จัดเก็บไม่ซ้ำกัน ซึ่งจำเป็นต้องพิจารณาถึงลักษณะของข้อมูลที่จัดเก็บว่าในกรณีใดบ้างที่ค่าที่จัดเก็บนั้นจะไม่มีค่าที่ซ้ำกัน

พีชคณิตเชิงสัมพันธ์ (Relational Algebra)

- พีชคณิตเชิงสัมพันธ์เป็นแนวทางการดำเนินการสำหรับการจัดการรีเลชัน โดยที่แต่ละตัวดำเนินการของพีชคณิตเชิงสัมพันธ์จะนำเข้ารีเลชันหนึ่งหรือสองรีเลชันแล้วได้ผลลัพธ์เป็นรีเลชันที่สร้างขึ้นใหม่ ได้กำหนดตัวดำเนินการ 8 ตัว ประกอบด้วย 2 กลุ่มโดยแต่ละกลุ่มมี 4 ตัวดำเนินการ ดังนี้
 - ตัวดำเนินการที่เกี่ยวกับเซต คือ Union, Intersection, Difference, และ Cartesian product
 - ตัวดำเนินการที่เกี่ยวกับรีเลชัน คือ Select , Project, Join, และ Divide

Union

R1

Name	Age	Sex
A	20	M
C	21	M
B	21	F

R2

Name	Age	Sex
D	20	F
A	20	M
E	21	F

$$R3 = R1 \cup R2$$

Name	Age	Sex
A	20	M
C	21	M
B	21	F
D	20	F
E	21	F

- Union เป็นการดำเนินการที่เชื่อมรีเลชัน 2 ตัว เข้าด้วยกัน โดย $R1 \cup R2$ คือเซตของทุกทูเปิล t ของ R1 หรือ R2 หรือ ของทั้งคู่ ซึ่งทั้ง สองรีเลชันสามารถทำการ union กันได้ (union-compatible) รีเลชันทั้งสองต้องมีดีกรีที่เท่ากันและ มีโดเมนของแอตทริบิวต์ที่เหมือนกัน
 - สัญลักษณ์ที่ใช้แทน Union คือ \cup
 - Union เป็นการดำเนินการที่มีคุณสมบัติสามารถสลับที่ได้ และเปลี่ยนกลุ่มได้

Intersection

R1

Name	Age	Sex
A	20	M
C	21	M
B	21	F

R2

Name	Age	Sex
D	20	F
A	20	M
E	21	F

$$R3 = R1 \cap R2$$

Name	Age	Sex
A	20	M

- Intersection เป็นการดำเนินการกับรีเลชันโดยที่ $R1 \text{ INTERSECT } R2$ จะได้ผลลัพธ์เป็นเซตของทุกทูเปิล t ที่เหมือนกันของรีเลชันทั้งคู่คือ $R1$ และ $R2$
 - สัญลักษณ์ที่ใช้แทนสำหรับการดำเนินการ intersect คือ \cap
- intersect เป็นการดำเนินการที่มีคุณสมบัติสามารถสลับที่ได้และเปลี่ยนกลุ่มได้

Difference

R1

Name	Age	Sex
A	20	M
C	21	M
B	21	F

R2

Name	Age	Sex
D	20	F
A	20	M
E	21	F

$$R3 = R1 - R2$$

Name	Age	Sex
C	21	M
B	21	F

$$R3 = R2 - R1$$

Name	Age	Sex
D	20	F
E	21	F

- Difference เป็นการดำเนินการของรีเลชันโดยที่ R1 MINUS R2 คือเซตของทุกทูเปิลของรีเลชัน R1 ที่ไม่อยู่ใน R2 สัญลักษณ์ที่ใช้แทนสำหรับการดำเนินการ difference คือ '-'
- Difference เป็นการดำเนินการที่ไม่สามารถสลับที่และเปลี่ยนกลุ่มได้

Cartesian product

- Cartesian product ระหว่างรีเลชัน R1 และ R2 โดยใช้ R1 TIMES R2 คือ เซตของทุกทูเปิล t โดยที่เกิดจากการเชื่อมต่อกันของทูเปิล r ของรีเลชัน R1 กับทูเปิล s ของรีเลชัน R2 การเชื่อมต่อกันของทูเปิล

R1

Name	Age	Sex
A	20	M
C	21	M

R2

Name	Age	Sex
D	20	F
E	21	F

$$R3 = R1 \times R2$$

Name	Age	Sex	Name	Age	Sex
A	20	M	D	20	F
C	21	M	D	20	F
A	20	M	E	21	F
C	21	M	E	21	F

Selection

R

Name	Age	Sex
A	20	M
M	21	F
B	20	F
F	19	M
A	20	F
R	21	F
C	21	M

$R1 = \sigma (Age=20)(R)$

Name	Age	Sex
A	20	M
B	20	F
A	20	F

$R2 = \sigma (Sex=M \text{ AND } Age>19)(R)$

Name	Age	Sex
A	20	M
C	21	M

- ตัวดำเนินการ select เป็นการเลือกทูเปิลที่เป็นซับเซต (subset) ในรีเลชัน เป็นตัวดำเนินการทางคณิตศาสตร์ ที่ใช้กับรีเลชันเดียว เพื่อให้ได้ซับเซตที่เป็นจริงตามเงื่อนไขการเลือกหรือเป็นจริงตามเงื่อนไขที่กำหนดรูปแบบของการใช้การดำเนินการ select คือ

σ <select condition> (<relation>)

- ตัวดำเนินการ select เป็นการเลือกทูเปิลที่เป็นซับเซต (subset) ในรีเลชัน

- เป็นตัวดำเนินการทางคณิตศาสตร์ ที่ใช้กับรีเลชันเดียว เพื่อให้ได้ซับเซตที่เป็นจริงตามเงื่อนไขการเลือกหรือเป็นจริงตามเงื่อนไขที่กำหนดรูปแบบของการใช้การดำเนินการ select คือ

σ <select condition> (<relation>)

Projection

R

Name	Age	Sex
A	20	M
M	21	F
B	20	F
F	19	M
A	20	F
R	21	F
C	21	M

$R1 = \pi (\text{Name, Sex})(R)$

Name	Sex
A	M
M	F
B	F
F	M
A	F

$R2 = \pi (\text{Age, Sex})(R)$

Age	Sex
20	M
21	F
20	F
19	M

- การดำเนินการ Project สร้างรีเลชันใหม่โดยการ เลือก subset ของแอตทริบิวต์ ในรีเลชัน ที่มีค่าของทูเปิลที่ไม่ซ้ำกัน เป็นตัวดำเนินการทางคณิตศาสตร์
- รูปแบบของตัวดำเนินการ project คือ

π <attribute list> (<relation>)

- โดยที่ <attribute list> เป็น subset ของแอตทริบิวต์ที่อยู่ในรีเลชัน
- รีเลชันดีกรีของผลลัพธ์เท่ากับจำนวนของแอตทริบิวต์ที่กำหนดจาก <attribute list> เพราะจะปรากฏเฉพาะแอตทริบิวต์ที่เลือกในรีเลชันผลลัพธ์
- หากแอตทริบิวต์ที่เลือกเป็นแคนดิเดตคีย์ คาร์ดินัลลิตีของรีเลชันผลลัพธ์จะน้อยกว่าหรือเท่ากับคาร์ดินัลลิตีของรีเลชันเริ่มต้น เนื่องจากมีทูเปิลที่ซ้ำกันซึ่งถูกการดำเนินการดังกล่าวตัดออก

Join

R1

First Name	Last Name
A	Mary
B	John
C	Ann

R2

Last Name	Sex
Ann	F
John	M
Mary	F
Bill	M

R3=R1 (Last Name=Last Name) R2

First Name	Last Name
A	Mary
B	John
C	Ann

Natural Join

First Name	Last Name	Sex
A	Mary	F
B	John	M
C	Ann	F

Right Outer Join

First Name	Last Name	Last Name	Sex
A	Mary	Mary	F
B	John	John	M
C	Ann	Ann	F
NULL	NULL	Bill	M

- Join เป็นตัวดำเนินการในการเชื่อมรีเลชันสองรีเลชัน บนเงื่อนไขการเชื่อม (join condition or predicate)
- รีเลชันที่จะทำการเชื่อมกันต้องมีอย่างน้อย 1 แอตทริบิวต์ที่เป็นแอตทริบิวต์กลางที่ใช้ร่วมกัน อยู่ภายใต้โดเมนเดียวกัน การ join ต้องมีการระบุแอตทริบิวต์ที่จะใช้ในการ join
- รูปแบบของการดำเนินการ join คือ

$$R1 \bowtie_{\langle \text{join condition} \rangle} R2$$
- โดยที่ $\langle \text{join condition} \rangle$ คือ $\langle \text{attribute from R1} \rangle \langle \text{comparison operator} \rangle \langle \text{attribute from R2} \rangle$
 - เครื่องหมายเปรียบเทียบที่สามารถใช้ได้มีเงื่อนไข $<, >, <=, >=, =, <>$ ใช้กับโดเมนของแอตทริบิวต์

$$R1 \bowtie_{\text{Last Name} = \text{Last Name}} R2$$

Division

R1

Name	Sex
A	M
B	F
A	F
C	F
D	M
C	M

$$R1 = R1 \div R2$$

Name
A
C

R2

Sex
M
F

- ตัวดำเนินการ Division เป็นการแบ่งรีเลชันหรือการหารรีเลชัน รีเลชัน R1 มีดีกรีเท่ากับ $(n + m)$ หารด้วยรีเลชัน R2 มีดีกรีเท่ากับ m
- ผลลัพธ์ที่ได้คือรีเลชันที่มีดีกรีเท่ากับ n ลำดับแอตทริบิวต์ $(n+i)$ ของรีเลชัน R1 และ ลำดับแอตทริบิวต์ i จากรีเลชัน R2 ควรกำหนดในโดเมนเดียวกัน
- ผลลัพธ์การดำเนินการ division ระหว่าง R1 และ R2 จะได้รีเลชันผลลัพธ์ใหม่ที่ประกอบด้วยทูเปิลทั้งหมดที่เชื่อมต่อกับทูเปิลทั้งหมดของ R2 ที่เป็นส่วนหนึ่งของรีเลชัน R1