

The Relational Data Model and Relational Database Constraints

สอนโดย

ผู้ช่วยศาสตราจารย์ ดร.อารีรัตน์ ตระกูลมีทอง (Section 001)

รองศาสตราจารย์ ดร.สุรี เตชะวุฒิ (Section 002)

Outlines

- ▶ Relational Model Concepts
- ▶ Relational Model Constraints and Relational Database Schemas
- ▶ Update Operations and Dealing with Constraint Violations

2

Relational Model Concepts

▶ Relational model

- ▶ แบบจำลองเชิงสัมพันธ์ ประยุกต์ใช้ทฤษฎี ความสัมพันธ์ (Relation) ที่อิงตามแนวคิดของเรื่อง เซต (Set)
- ▶ Relational Model ถูกนำเสนอครั้งแรก โดย Dr. E.F. Codd จากบริษัท IBM Research จากการตีพิมพ์ในวารสาร ACM ในบทความชื่อ "A Relational Model for Large Shared Data Banks" ในเดือน มิถุนายน ปี 1970
 - ▶ บทความข้างต้น ทำให้เกิดการปฏิวัติครั้งใหญ่ในการ การจัดการฐานข้อมูล (Database Management) และทำให้ Dr. Codd ได้รับรางวัล ACM Turing Award

▶ Hierarchical and network models

- ▶ สำหรับแบบจำลองข้อมูลที่มีการใช้งานมาก่อนแบบจำลองเชิงสัมพันธ์ คือ แบบจำลองข้อมูลที่นำเสนอในรูปแบบ Network Model และ Hierarchical Model

3

Relational Model Concepts: Informal Definitions

▶ Informal Definition

- ▶ แบบจำลองข้อมูลเชิงสัมพันธ์ จัดเก็บข้อมูลใน Relation ที่มีลักษณะคล้ายกับ ตารางข้อมูล
- ▶ แต่ละ Relation ประกอบด้วย รายการข้อมูล หรือแถวข้อมูล (Row)

▶ Row

- ▶ แต่ละแถวข้อมูลนำเสนอข้อมูลที่เป็นข้อเท็จจริงที่เกิดขึ้นบนโลกใบนี้ และความสัมพันธ์ระหว่างข้อเท็จจริงเหล่านี้
- ▶ ในแบบจำลองข้อมูลเชิงสัมพันธ์ เรียก Row ว่า Tuple

▶ Column

- ▶ แต่ละ Row ประกอบด้วยคอลัมน์ (Column) ต่าง ๆ ในแบบจำลองข้อมูลเชิงสัมพันธ์ เรียก Column ว่า แอททริบิวต์ (Attribute)
- ▶ ชื่อคอลัมน์ (Column Name/Attribute Name) ใช้สื่อความหมายของข้อมูล เพื่อบอกว่าเป็นข้อมูลเกี่ยวกับเรื่องอะไร

4

Relational Model Concepts: Informal Definitions (Cont.)

▶ Key of a Relation

- ▶ **Key** คือ ค่าของข้อมูล 1 Attribute หรือ หลาย Attributes ที่มีค่าไม่ซ้ำกันในแต่ละ Row หรือ ค่าที่ทำให้แต่ละแถวมีความเป็นเอกลักษณ์
- ▶ ตัวอย่างเช่น รหัสประกันสังคม (SSN) คือ Key ของตารางข้อมูล STUDENT
- ▶ ในบางครั้ง มีการกำหนดค่าข้อมูลที่อยู่ในลักษณะเรียงลำดับ เช่น Running Number, Auto Increment, หรือ Row-id ให้เป็น Key สำหรับ Key ในลักษณะนี้ถูกเรียกว่า Surrogate key

5

Relational Model Concepts: Example of a Relation

Diagram illustrating the structure of a relation. The Relation Name is **STUDENT**. The Attributes are **Name**, **Ssn**, **Home_phone**, **Address**, **Office_phone**, **Age**, and **Gpa**. The Tuples are the rows of data.

Name	Ssn	Home_phone	Address	Office_phone	Age	Gpa
Benjamin Bayer	305-61-2435	373-1616	2918 Bluebonnet Lane	NULL	19	3.21
Chung-cha Kim	381-62-1245	375-4409	125 Kirby Road	NULL	18	2.89
Dick Davidson	422-11-2320	NULL	3452 Elgin Road	749-1253	25	3.53
Rohan Panchal	489-22-1100	376-9821	265 Lark Lane	749-6492	28	3.93
Barbara Benson	533-69-1238	839-8461	7384 Fontana Lane	NULL	19	3.25

Figure 5.1
The attributes and tuples of a relation STUDENT.

6

Formal Definitions: Schema

▶ Relation Schema R :

- ▶ โครงสร้างของรีเลชัน (Relation) R ประกอบด้วย แอททริบิวต์ A_1, \dots, A_n แสดงด้วยสัญลักษณ์ $R(A_1, A_2, \dots, A_n)$ โดยที่ R คือ Relation Name และ A คือ Attribute Name

▶ Attribute A_i

- ▶ Attribute A_i แต่ละตัว เป็นองค์ประกอบของ Relation R ใช้อธิบายคุณลักษณะของ Relation R
- ▶ แต่ละ Attribute คือ 1 Domain หรือกลุ่มของข้อมูลที่มีค่าถูกต้อง (Valid)
 - ▶ ตัวอย่างเช่น Domain ของ Cust-id คือ ตัวเลข 6 หลัก เป็นต้น

▶ Example: CUSTOMER (Cust-id, Cust-name, Address, Phone#)

- ▶ CUSTOMER คือ Relation Name
- ▶ ข้อมูลในแต่ละแถว ประกอบด้วย 4 Attributes คือ Cust-id, Cust-name, Address, และ Phone#

7

Formal Definitions: Tuple, Domain

- ▶ **Tuple** คือ ชุดของค่าข้อมูลที่อยู่ภายในเครื่องหมาย <> (Angled Brackets)
- ▶ ข้อมูลแต่ละตัวที่อยู่ภายใน Tuple จะต้อง มี Domain ที่เหมาะสม และสะท้อนถึงความหมายของข้อมูลแต่ละรายการ
- ▶ ตัวอย่างเช่น แต่ละแถวข้อมูลใน CUSTOMER Relation คือ รายการข้อมูลที่ประกอบด้วยข้อมูล 4 ค่า (มาจาก 4 Attributes) เรียกว่า 4-tuple
 <632895, "John Smith", "101 Main St. Atlanta, GA 30332", "(404) 894-2000">
- ▶ **Domain** คือ กลุ่ม หรือชุดของข้อมูล ที่มีการกำหนด ชนิดข้อมูล (Data Type) และรูปแบบข้อมูล (Format) หรือเรียกว่า **ขอบเขตข้อมูล** เช่น
 - ▶ Domain ชื่อ USA_phone_numbers ใช้กำหนดขอบเขตของข้อมูลหมายเลขโทรศัพท์ในประเทศสหรัฐอเมริกา
 Data Type = ตัวเลข 10 หลัก, Format = (ddd)ddd-dddd
 - ▶ Domain ชื่อ Date ใช้กำหนดขอบเขตของข้อมูล วันที่
 Data Type = Date, Format = yyyy-mm-dd (หรือ mm-dd-yyyy)

8

Formal Definitions: Tuple, Domain

- ▶ Attribute Name ใช้สื่อถึงบทบาทหน้าที่ที่แตกต่างกันของแต่ละ Attribute ใน Relation
- ▶ เราสามารถกำหนด Domain ให้กับ Attribute แต่ละตัวได้ เช่น
 - ▶ Attribute ชื่อ Invoice-date และ Payment-date มี Domain เหมือนกันคือ

Data Type = Date, Format = yyyy-mm-dd

- ▶ ถึงแม้ Invoice-date และ Payment-date จะมี Domain เหมือนกัน แต่ทั้งคู่มีความหมาย/บทบาท ที่แตกต่างกัน
 - ▶ Invoice-date คือ วันที่ออกใบแจ้งหนี้
 - ▶ Payment-date คือ วันที่ชำระเงิน

9

Formal Definitions: Relation State

▶ Relation State

- ▶ สถานะของข้อมูล (State) ใน Relation ณ เวลาใดเวลาหนึ่ง เรียกว่า Relation State (r) ซึ่งแทนด้วยชุดของ tuple หรือ record แทนด้วยสัญลักษณ์

$$r = \{t_1, t_2, \dots, t_m\}$$

- ▶ ในแต่ละ tuple (t) ประกอบด้วยค่าข้อมูล v_1, v_2, \dots, v_n ของแอททริบิวต์ A_1, A_2, \dots, A_n แสดงด้วยสัญลักษณ์

$$t = \langle v_1, v_2, \dots, v_n \rangle$$

- ▶ ค่าข้อมูลแต่ละตัว $v_i, 1 \leq i \leq n$, คือสมาชิกของแต่ละแอททริบิวต์ A_i โดยที่ค่าข้อมูล ต้องอยู่ในขอบเขต (Domain) ที่กำหนดไว้ สำหรับค่าข้อมูลสามารถจัดเก็บอยู่ในรูป NULL ก็ได้ (หากไม่มีการกำหนดข้อบังคับว่า ห้ามเป็น NULL)

10

Formal Definitions: Relation State

▶ Relation State Example

- ▶ ตัวอย่างเช่น Attribute ชื่อ Cust-name ถูกกำหนดให้มีขอบเขตข้อมูลคือ เป็น String ที่มีความยาวสูงสุดได้ไม่เกิน 25 (Maximum length = 25) แสดงด้วยสัญลักษณ์ดังนี้

dom(Cust-name) is varchar(25)

dom = domain

▶ Current Relation State

- ▶ ค่าสถานะปัจจุบันของข้อมูลใน Relation และจะต้องเป็นข้อเท็จจริงที่มีความถูกต้องเท่านั้น จึงจะสามารถจัดเก็บในแต่ละ Relation ได้

11

Formal Definitions: Summary

▶ Formally,

- ▶ Given $R(A_1, A_2, \dots, A_n)$
- ▶ $r(R) \subset \text{dom}(A_1) \times \text{dom}(A_2) \times \dots \times \text{dom}(A_n)$
- ▶ $R(A_1, A_2, \dots, A_n)$ is the schema of the relation
- ▶ R is the name of the relation
- ▶ A_1, A_2, \dots, A_n are the attributes of the relation
- ▶ $r(R)$: a specific state (or "value" or "population") of relation R
 - ▶ this is a set of tuples (rows)
 - ▶ $r(R) = \{t_1, t_2, \dots, t_m\}$ where each t_i is an n -tuple
 - ▶ $t_i = \langle v_1, v_2, \dots, v_n \rangle$ where each v_j element-of $\text{dom}(A_j)$

12

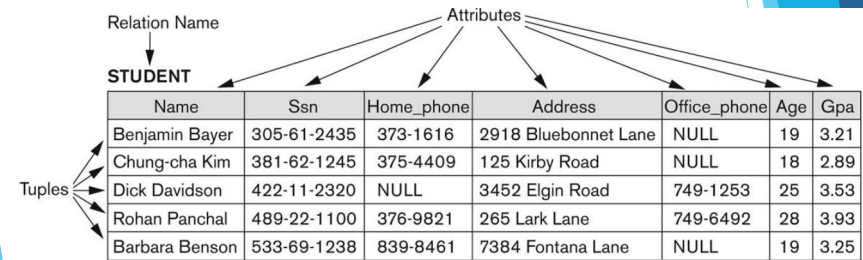
Formal Definitions: Example

- ▶ Let $R(A_1, A_2)$ be a relation schema:
 - ▶ Let $\text{dom}(A_1) = \{0, 1\}$
 - ▶ Let $\text{dom}(A_2) = \{a, b, c\}$
- ▶ Then: $\text{dom}(A_1) \times \text{dom}(A_2)$ is all possible combinations:
 $\{ \langle 0, a \rangle, \langle 0, b \rangle, \langle 0, c \rangle, \langle 1, a \rangle, \langle 1, b \rangle, \langle 1, c \rangle \}$
- ▶ The relation state $r(R) \subset \text{dom}(A_1) \times \text{dom}(A_2)$
- ▶ For example: $r(R)$ could be $\{ \langle 0, a \rangle, \langle 0, b \rangle, \langle 1, c \rangle \}$
 - ▶ this is one possible state (or "population" or "extension") r of the relation R , defined over A_1 and A_2 .
 - ▶ It has three 2-tuples: $\langle 0, a \rangle, \langle 0, b \rangle, \langle 1, c \rangle$

13

Definition Summary

Informal Terms	Formal Terms
Table	Relation
Column Header	Attribute
All possible Column Values	Domain
Row	Tuple
Table Definition	Schema of a Relation
Populated Table	State of the Relation



14

Example - A relation STUDENT

Figure 5.1
The attributes and tuples of a relation STUDENT.

Characteristics of Relations

- ▶ ลำดับของ Tuple ใน Relation
 - ▶ ข้อมูลแต่ละแถว (Tuple/Record) ที่จัดเรียงใน Relation ไม่นับเป็นลำดับ
 - ▶ สำหรับค่าข้อมูลของ Attribute ในแต่ละ Tuple $t = \langle v_1, v_2, \dots, v_n \rangle$ จะพิจารณาเรื่องของลำดับตามการประกาศโครงสร้างของ Relation $R(A_1, A_2, \dots, A_n)$
 - ▶ จากตัวอย่าง Relation ชื่อ STUDENT ด้านล่าง ค่าข้อมูล Attribute แรกในแต่ละแถว คือ ชื่อนักเรียน ส่วนค่าข้อมูล Attribute ตัวที่สองคือ รหัสประจำตัวประชาชน เป็นต้น

Figure 5.2

The relation STUDENT from Figure 5.1 with a different order of tuples.

Name	Ssn	Home_phone	Address	Office_phone	Age	Gpa
Dick Davidson	422-11-2320	NULL	3452 Elgin Road	749-1253	25	3.53
Barbara Benson	533-69-1238	839-8461	7384 Fontana Lane	NULL	19	3.25
Rohan Panchal	489-22-1100	376-9821	265 Lark Lane	749-6492	28	3.93
Chung-cha Kim	381-62-1245	375-4409	125 Kirby Road	NULL	18	2.89
Benjamin Bayer	305-61-2435	373-1616	2918 Bluebonnet Lane	NULL	19	3.21

สำหรับ Data Modeling รูปแบบอื่น เช่น NoSQL JSON ไม่มีการกำหนดลำดับค่าข้อมูลของ Attribute ตายตัว แต่โครงสร้างอยู่ในลักษณะ Self-Describing

15

Characteristics of Relations

- ▶ ค่าข้อมูลแต่ละตัวใน Tuple
 - ▶ ต้องเป็นค่าข้อมูลที่ไม่สามารถแบ่งแยกได้ (Atomic/Indivisible)
 - ▶ กำหนดให้ Tuple/Row $t = \langle v_1, v_2, \dots, v_n \rangle$
 - ▶ ค่าข้อมูลของ v_i แต่ละตัว ต้องมาจาก Domain ของแต่ละ Attribute $\text{dom}(A_i)$
- ▶ ค่า NULL ใน Tuple
 - ▶ ค่า NULL ใน Attribute หมายถึง
 - ▶ ไม่ทราบค่า (Unknown)
 - ▶ บอกไม่ได้ หรือ ณ ขณะนั้นยังไม่ทราบ
 - ▶ ไม่ได้กำหนด (Undefined)

16

Characteristics of Relations

► Relation Schema

- โครงสร้าง Relation แสดงด้วยสัญลักษณ์ $R(A_1, A_2, \dots, A_n)$
- ชื่อรีเลชัน แทนด้วยตัวอักษรตัวใหญ่ เช่น Q, R, S

► Relation State

- สถานะของข้อมูลใน Relation ณ เวลาใดเวลาหนึ่ง แทนด้วยสัญลักษณ์ที่ใช้อักขระตัวเล็ก เช่น q, r, s ตัวอย่างเช่น

$$r = \{t_1, t_2, \dots, t_m\}$$

► องค์ประกอบของข้อมูลใน Tuple/Record/Row

- $t[A_i]$ และ $t.A_i$ ใช้อ้างถึงค่าข้อมูล v_i ของ Attribute A_i ที่อยู่ใน tuple t
- $t[A_u, A_w, \dots, A_z]$ และ $t.(A_u, A_w, \dots, A_z)$ ใช้อ้างถึง sub-tuple ของข้อมูล $\langle v_u, v_w, \dots, v_z \rangle$ ซึ่งเป็นส่วนหนึ่งของค่าข้อมูลของ Attribute ที่ระบุใน tuple t
 - เนื่องจากในบางกรณีเราอาจจะต้องการอ้างถึงข้อมูลบาง Attribute โดยที่ไม่ได้อ้างถึงข้อมูลทุก Attribute

17

Relational Model Notation

► ตัวอย่างโครงสร้าง Relation ชื่อ STUDENT

- STUDENT(Name, Ssn, Home_phone, Address, ...)
- Name และ Ssn คือ Attribute ของ Relation ชื่อ STUDENT
- การอ้างถึง Attribute ของ Relation ใช้สัญลักษณ์ $R.A$ โดยที่ A คือ Attribute ของ R เช่น
 - STUDENT.Name, STUDENT.Ssn

Figure 5.2

The relation STUDENT from Figure 5.1 with a different order of tuples.

STUDENT

Name	Ssn	Home_phone	Address	Office_phone	Age	Gpa
Dick Davidson	422-11-2320	NULL	3452 Elgin Road	749-1253	25	3.53
Barbara Benson	533-69-1238	839-8461	7384 Fontana Lane	NULL	19	3.25
Rohan Panchal	489-22-1100	376-9821	265 Lark Lane	749-6492	28	3.93
Chung-cha Kim	381-62-1245	375-4409	125 Kirby Road	NULL	18	2.89
Benjamin Bayer	305-61-2435	373-1616	2918 Bluebonnet Lane	NULL	19	3.21

18

Relational Model Constraints

► Constraints

- กฎ/ข้อบังคับ ที่กำหนดสถานะของข้อมูล ที่จะจัดเก็บในฐานข้อมูล
- ข้อมูลที่จะจัดเก็บในฐานข้อมูลได้ ต้องผ่าน กฎ/ข้อบังคับ ที่กำหนดไว้เท่านั้น จึงจะสามารถจัดเก็บไว้ในฐานข้อมูลได้
- ข้อบังคับส่วนใหญ่ได้มาจากความเป็นจริงของข้อมูล เช่น อายุของสิ่งมีชีวิตมีค่า > 0 เป็นต้น

► กฎ/ข้อบังคับ แบ่งออกเป็น 3 กลุ่มหลัก

- Inherent or Implicit Constraints
- Schema-based or Explicit Constraints
- Application based or semantic constraints

19

Relational Model Constraints

► Inherent or implicit constraints

- กฎข้อบังคับที่ได้รับสืบทอดมาจากกฎข้อบังคับของแต่ละแบบจำลองข้อมูล เช่น Relational Data Model ไม่อนุญาตให้ข้อมูลใน Attribute มีค่าเป็น list หรือ มีค่าหลายค่า

► Schema-based or Explicit Constraints

- กฎข้อบังคับที่สามารถกำหนดโดยตรงในโครงสร้างฐานข้อมูล เช่น Salary $\geq 15,000$ เป็นต้น ซึ่งส่วนใหญ่จะเป็นกฎข้อบังคับที่ไม่ซับซ้อน

► Application based or semantic constraints

- กฎข้อบังคับที่ซับซ้อนที่ไม่สามารถกำหนดโดยตรงในโครงสร้างฐานข้อมูล จะถูกกำหนดไว้ในแอปพลิเคชันโปรแกรมแทน เช่น ห้ามเบิก OT เกิน 4 ชั่วโมงในวันธรรมดา เป็นต้น

20

Relational Integrity Constraints

- ▶ สถานะข้อมูลที่เกิดขึ้นอยู่ในฐานข้อมูลทุกตัว จะต้องคงสภาพความถูกต้องตามเงื่อนไขที่กำหนดไว้ในกฎ/ข้อบังคับ
- ▶ Explicit schema-based constraints ใน Relational Data Model แบ่งออกเป็น 3 กลุ่มหลัก ดังนี้
 - ▶ Key constraints
 - ▶ Entity integrity constraints
 - ▶ Referential integrity constraints
- ▶ Domain Constraint
 - ▶ เป็นอีก 1 ประเภท ของ Schema-based constraint คือ ทุกค่าข้อมูลใน Tuple จะต้องสอดคล้องกับขอบเขตข้อมูล (Domain) ที่กำหนดไว้ในแต่ละ Attribute หรือ เป็น NULL ได้ หากมีการกำหนดไว้ว่าอนุญาตให้เป็น NULL ได้

21

Key Constraints

- ▶ Superkey (SK) of R คือ กลุ่ม หรือ ชุด Attribute (1 Attribute หรือ > 1 ก็ได้) ใน Relation R ที่มีเงื่อนไข ดังนี้
 - ▶ มีค่าไม่ซ้ำกันใน 2 tuples ใด ๆ หรือ เขียนให้อยู่ในรูปสัญลักษณ์ ดังนี้
distinct tuples t_1 and t_2 in $r(R)$, $t_1[SK] \neq t_2[SK]$
- ▶ Key คือ Superkey ที่เล็กที่สุด (A "minimal" superkey)
 - ▶ กลุ่มข้อมูล หรือข้อมูล (1 Attribute หรือ > 1 ก็ได้) ที่จะอยู่ในสถานะ Key k ได้ จะต้องเป็น Superkey SK ของ Relation R
 - ▶ วิธีตรวจสอบว่า กลุ่มข้อมูล หรือข้อมูล ใดสามารถอยู่ในสถานะ Key k ได้หรือไม่ สามารถตรวจสอบได้ดังนี้
 - ▶ หากลบ Attribute A ใด ๆ จาก Key K ออกจะทำให้กลุ่มข้อมูล หรือข้อมูลที่เหลืออยู่ขาดคุณสมบัติของการเป็น SK ของ R
- ▶ Key k ทุกตัว จะเป็น Superkey SK โดยปริยาย
- ▶ แต่ Superkey SK ไม่จำเป็นต้องเป็น Key k

22

Key Constraints (cont'd.)

- ▶ ตัวอย่าง CAR Relation Schema:

CAR(License_number, Engine_serial_number, Make, Model, Year)

- ▶ CAR มี key 2 ตัว ดังนี้
 - Key1 = {License_number}
 - Key2 = {Engine_serial_number}
- ▶ ทั้ง Key1 และ Key2 เป็น Superkey ของ CAR
- ▶ Minimal Superkey หรือ Key ใด ๆ ไม่จำเป็นต้องมีเพียง 1 Attribute อาจจะมี > 1 Attribute ก็ได้
- ▶ {Engine_serial_number, Make} เป็น Superkey แต่ไม่เป็น Key เนื่องจาก ไม่ใช่ Minimal Superkey เนื่องจาก หากลบ ตัวใดตัวหนึ่งออก (เหลือเพียง 1 ตัว คือ Engine_serial_number หรือ Make) จะไม่คงความเป็น Superkey

23

Key Constraints (cont'd.)

- ▶ Candidate key คือ Minimal Superkey
 - ▶ ใน 1 Relation อาจจะมี Candidate Key ได้ > 1 ตัว
 - ▶ สำหรับ Candidate Key ที่ถูกเลือกให้เป็นตัวแทนในการระบุความเป็นเอกลักษณ์ของ Tuple ใน Relation (Uniquely Identify: ชุดข้อมูลที่มีค่าไม่ซ้ำกันใน 2 tuples ใด ๆ) เราเรียก Candidate Key ตัวที่ถูกเลือกว่า คีย์หลัก (Primary Key: PK) ของ Relation
- ▶ Primary key of the relation
 - ▶ Candidate Key ที่ถูกเลือกให้เป็น Primary Key จะถูกขีดเส้นใต้
 - ▶ Candidate Keys ที่เหลือที่ไม่ได้ถูกเลือก จะถูกกำหนดให้เป็น Unique Keys หรือเรียกว่า Secondary Keys
 - ▶ นอกจากนี้ PK ยังใช้ในการอ้างอิง Tuple จาก Relation อื่น ดังนั้น
 - ▶ ในการเลือก PK จึงมักจะเลือกตัวที่มีจำนวน Attribute น้อยที่สุด เพื่อให้การรวม (Join) ข้อมูลจากหลาย ๆ Relation ทำได้สะดวกขึ้น และประหยัดเวลามากขึ้น แต่ก็ไม่เสมอไป แล้วแต่สถานการณ์

24

Key Constraints (cont'd.)

CAR

License_number	Engine_serial_number	Make	Model	Year
Texas ABC-739	A69352	Ford	Mustang	02
Florida TVP-347	B43696	Oldsmobile	Cutlass	05
New York MPO-22	X83554	Oldsmobile	Delta	01
California 432-TFY	C43742	Mercedes	190-D	99
California RSK-629	Y82935	Toyota	Camry	04
Texas RSK-629	U028365	Jaguar	XJS	04

Figure 3.4

The CAR relation, with two candidate keys: License_number and Engine_serial_number.

จาก CAR Relation ด้านบน มี 2 Candidate Keys คือ

License_number และ Engine_serial_number

- ทำไม License_number จึงถูกเลือกให้เป็น PK?
- ทำไม ระบบสำนักทะเบียน จึงไม่เลือกให้ รหัสบัตรประชาชนเป็น PK?

25

Relational Database Schemas

► Relational database schema S ประกอบด้วย

► เซตของโครงสร้าง Relation ที่อยู่ในฐานข้อมูลเดียวกัน แทนด้วยสัญลักษณ์ $S = \{R_1, R_2, \dots, R_n\}$ โดยที่

► S คือ ชื่อฐานข้อมูล

► R_1, R_2, \dots, R_n คือ ชื่อของแต่ละ Relation R_i ที่อยู่ในฐานข้อมูลตัวเดียวกัน

► เซตของกฎข้อบังคับของการคงสภาพของข้อมูล เรียกว่า Integrity Constraints (IC)

26

Relational Database Schemas: COMPANY Database Schema

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
-------	-------	-------	-----	-------	---------	-----	--------	-----------	-----

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
-------	---------	---------	----------------

DEPT_LOCATIONS

Dnumber	Dlocation
---------	-----------

PROJECT

Pname	Pnumber	Plocation	Dnum
-------	---------	-----------	------

WORKS_ON

Essn	Pno	Hours
------	-----	-------

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship
------	----------------	-----	-------	--------------

Figure 5.5

Schema diagram for the COMPANY relational database schema.

27

Relational Database State

► Relational database state หรือ Relational database snapshot

► เซตของ สถานะของข้อมูล แทนด้วยสัญลักษณ์

$$DB = \{r_1, r_2, \dots, r_m\}$$

► แต่ละ r_i คือ สถานะของข้อมูลในแต่ละ Relation R_i

► แต่ละ r_i จะต้องไม่ละเมิดกฎข้อบังคับของการคงสภาพของข้อมูล IC ของแต่ละ Relation ที่กำหนดไว้ในโครงสร้างฐานข้อมูล หรือเรียกว่า อยู่ในสถานะ Valid State

► สถานะข้อมูลที่จะละเมิดกฎ IC กฎใดกฎหนึ่ง จะเรียกว่า อยู่ในสถานะ Invalid State

28

Populated Database State

- ▶ สถานะ ปัจจุบัน ของข้อมูล ในแต่ละ Relation มาจาก สถานะข้อมูลของหลาย ๆ Tuples ใน Relation
 - ▶ Relational database state คือ การรวมกันของสถานะ ข้อมูลจากทุก Relation
- ▶ เมื่อไหร่ที่ฐานข้อมูลมีการเปลี่ยนแปลง จะมีข้อมูล สถานะใหม่/ค่าใหม่ เกิดขึ้น
- ▶ การดำเนินการพื้นฐานที่ทำให้ข้อมูลเกิดการ เปลี่ยนแปลง คือ
 - ▶ INSERT คือ การเพิ่มข้อมูลเข้าไปใน Relation
 - ▶ DELETE คือ การลบข้อมูลออกจาก Relation
 - ▶ MODIFY คือ การเปลี่ยนแปลงข้อมูลของ Attribute(s) ใน Tuple

29

Populated Database State for COMPANY

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

Figure 5.6

30 One possible database state for the COMPANY relational database schema.

Populated Database State for COMPANY (Cont.)

WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

PROJECT

Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship
333445555	Alice	F	1986-04-05	Daughter
333445555	Theodore	M	1983-10-25	Son
333445555	Joy	F	1958-05-03	Spouse
987654321	Abner	M	1942-02-28	Spouse
123456789	Michael	M	1988-01-04	Son
123456789	Alice	F	1988-12-30	Daughter
123456789	Elizabeth	F	1967-05-05	Spouse

Figure 5.6

31 One possible database state for the COMPANY relational database schema.

Entity Integrity

- ▶ Entity integrity constraint
 - ▶ เป็น กฎ/ข้อบังคับ ที่เกี่ยวข้องกับ 1 Relation
 - ▶ เนื่องจาก ค่าคีย์หลัก (PK) ใช้สำหรับระบุแถวข้อมูล tuples
 - ▶ ดังนั้น ค่าข้อมูลของ PK ในทุก Relation $r(R)$ ใน ฐานข้อมูล S จะต้องไม่เป็น NULL
 - ▶ ถ้า PK ประกอบด้วยหลาย Attributes, ค่าข้อมูลใน Attribute ใด ๆ ห้ามมีค่าเป็น NULL
- ▶ $t[PK] \neq \text{null for any tuple } t \text{ in } r(R)$
- ▶ เราสามารถกำหนดให้ Attribute ใด ๆ ห้ามมีค่าเป็น NULL ได้ ถึงแม้ Attribute ดังกล่าว จะไม่ใช่ส่วนหนึ่งของ PK

32

Referential Integrity Constraint

- ▶ Referential integrity constraint
 - ▶ เป็น กฎ/ข้อบังคับ ที่ถูกกำหนดระหว่าง 2 Relations
 - ▶ ใช้สำหรับ กำหนดความสัมพันธ์ระหว่างข้อมูลใน Tuples ระหว่าง 2 Relations
 - ▶ ใช้สำหรับรักษาไว้ซึ่งความถูกต้องตรงกันของ ข้อมูลใน tuples ที่เชื่อมโยงกันระหว่าง 2 Relations
 - ▶ ใช้อ้างอิงข้อมูลระหว่าง 2 Relations

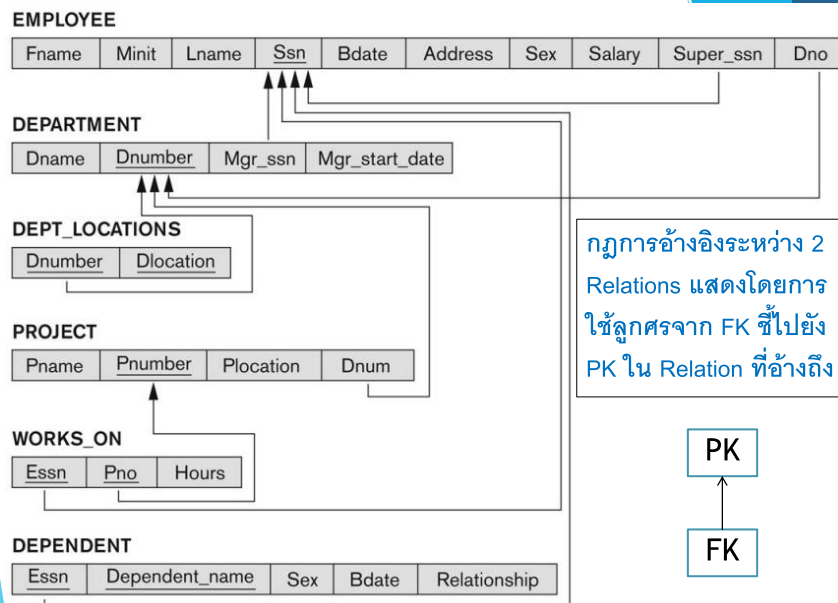
33

Referential Integrity and Foreign Keys

- ▶ Foreign Key (FK): คีย์นอก
 - ▶ FK ใช้สำหรับการอ้างอิงข้อมูลจาก ค่า Attribute ของ Tuple ใน Relation R_1 (เรียกว่า FK) ไปยัง Attribute ที่เป็น PK ใน Relation R_2
 - ▶ Attribute ที่ทำหน้าที่เป็น FK ใน Relation R_1 จะต้องเป็นกลุ่ม ข้อมูลเดียวกับ Attribute PK ใน Relation R_2
 - ▶ เราจะพูดได้ว่า tuple t_1 ใน R_1 อ้างถึง tuple t_2 ใน R_2 ถ้า $t_1[FK] = t_2[PK]$
 - ▶ ค่า FK จะต้องมิลักษณะใด ลักษณะหนึ่ง ดังต่อไปนี้
 - ▶ ค่าข้อมูลของ FK ใน tuple t_1 ของ Relation R_1 จะต้องปรากฏใน tuple t_2 ที่เป็น PK ของ Relation R_2 หรือ
 - ▶ FK มีค่า เป็น NULL ได้ (ในกรณีที่ไม่มี PK ใน R_2 ให้อ้างถึง)

34

Referential Integrity Constraints for COMPANY database



35

Other Types of Constraints

- ▶ Semantic integrity constraints
 - ▶ กฎข้อบังคับเชิงความหมาย จะขึ้นอยู่กับตรรกะในแต่ละ Application Program ที่มีลักษณะซับซ้อน ไม่สามารถกำหนดผ่าน กฎข้อบังคับที่ กล่าวมาก่อนหน้านี้ เช่น
 - ▶ ชั่วโมงทำงานสูงสุดของพนักงานแต่ละคนรวมกันทุกโครงการ ต้องไม่เกิน 56 ชั่วโมงต่อเดือน
- ▶ Triggers
 - ▶ เมื่อมีเหตุการณ์ที่ทำให้ข้อมูลมีการเปลี่ยนแปลง (Insert/Update/Delete) เกิดขึ้นในฐานข้อมูล จะกระตุ้นให้ DBMS ทำอะไรบางอย่าง เช่น
 - ▶ หาก GPA เฉลี่ยของนักศึกษาต่ำกว่า 1.50 ให้ Update ค่า StudentStatus = "พ้นสภาพ"
- ▶ Assertions
 - ▶ คำสั่งที่ใช้ในการตรวจสอบความถูกต้องของข้อมูลก่อนที่จะจัดเก็บไว้ใน ฐานข้อมูล ส่วนใหญ่จะตรวจสอบใน Application Program เช่น
 - ▶ ตรวจสอบเพื่อให้มั่นใจว่าลูกค้าไม่เบิกเงินเกินบัญชี
 - ▶ ตรวจสอบเพื่อให้แน่ใจว่าลูกค้าไม่กู้เงินเกินวงเงินที่ฝากไว้

36

Other Types of Constraints (cont'd.)

▶ Functional dependency constraint

- ▶ ความสัมพันธ์ระหว่างชุดข้อมูลใน Attribute X และ Y ในลักษณะค่าข้อมูลของ X เป็นตัวกำหนดค่าข้อมูลของ Y ซึ่งจะได้เรียนในเนื้อหาเกี่ยวกับ Data Normalization

▶ State constraints

- ▶ กฎข้อบังคับของสถานะที่ถูกต้องของฐานข้อมูล

▶ Transition constraints

- ▶ กฎข้อบังคับที่เกี่ยวข้องกับการเปลี่ยนสถานะของข้อมูลจากสถานะหนึ่ง ไปยังอีกสถานะหนึ่ง ในฐานข้อมูล

37

Update Operations on Relations

- ▶ การดำเนินการกับข้อมูลของแบบจำลองเชิงสัมพันธ์แบ่งออกเป็น 2 ลักษณะ คือ
 - ▶ การดึงข้อมูล
 - ▶ การดำเนินการที่ทำให้ข้อมูลเกิดการเปลี่ยนแปลงสถานะข้อมูลมีดังต่อไปนี้คือ
 - ▶ Insert
 - ▶ Delete
 - ▶ Update (or Modify)
- ▶ การดำเนินการใด ๆ ที่ทำให้ข้อมูลเกิดการเปลี่ยนแปลง ต้องไม่ละเมิด กฎ Integrity Constraints
- ▶ การปรับปรุงข้อมูลอาจส่งผลทำให้ข้อมูลอื่น ๆ ที่เกี่ยวข้องเกิดการเปลี่ยนแปลงไปด้วยอัตโนมัติ ซึ่งเราจะต้องจัดการข้อมูลอื่นที่เกี่ยวข้อง ไม่ให้ละเมิดกฎต่าง ๆ ที่กำหนดไว้ด้วยเช่นเดียวกัน

38

Update Operations on Relations

- ▶ ในกรณีที่เกิดมีการละเมิดกฎข้อบังคับ เราสามารถดำเนินการได้หลายอย่าง ดังนี้
 - ▶ **RESTRICT หรือ REJECT:** ยกเลิกการดำเนินการที่ทำให้เกิดการละเมิด
 - ▶ ดำเนินการ แต่แจ้งให้ผู้ใช้ทราบถึงการละเมิด
 - ▶ **TRIGGER:** กระตุ้นให้เกิดการดำเนินการเพิ่มเติม เพื่อแก้ไขการละเมิด โดยปกติ มี 2 ลักษณะ คือ CASCADE หรือ SET NULL
 - ▶ **User-specified error-correction routine:** ดำเนินการแก้ไขข้อผิดพลาดตามขั้นตอนการแก้ไขข้อผิดพลาดที่ผู้ใช้ระบุ

39

Possible violations for each operation

- ▶ **Insert Operation** การเพิ่มข้อมูลใหม่ของ Attribute ใน tuple t ของ Relation R มีโอกาสทำให้เกิดการละเมิดกฎข้อบังคับ 4 ประเภท ดังต่อไปนี้
 - ▶ **Domain constraint:** ขอบเขตข้อมูล ของ Attribute ใด Attribute หนึ่ง ไม่เป็นไปตามที่กำหนดไว้ใน Tuple ที่เพิ่มใหม่
 - ▶ **Key constraints:** ค่า Key ใน Tuple ที่เพิ่มใหม่ ซ้ำกับ ใน Tuple อื่น
 - ▶ **Entity integrity constraint:** PK ของ Tuple ที่เพิ่มใหม่ มีค่าเป็น NULL
 - ▶ **Referential integrity constraint:** FK ใน Tuple ที่เพิ่มใหม่ ไม่ปรากฏใน PK ของ Relation ที่อ้างถึง
- ▶ หากการเพิ่มข้อมูลใหม่มีการละเมิดกฎข้อบังคับแม้เพียง 1 กฎ การเพิ่มข้อมูลจะถูกปฏิเสธไม่ให้ดำเนินการ

40

Possible violations for each operation

- ▶ **Delete Operation** การลบข้อมูลสามารถทำให้ละเมิดเฉพาะกฎ **Referential Integrity Constraint**
 - ▶ หากลบ tuple ที่มีค่า PK ที่ถูกอ้างอิงโดย FK จาก tuples อื่น การละเมิดกฎดังกล่าว สามารถเลือกดำเนินการ (Delete Option) ได้ดังนี้
 - ▶ **Restrict:** ห้ามลบ คือ ปฏิเสธการลบ
 - ▶ **Cascade:** ตามไปลบทุก tuples ที่อ้างอิงด้วย FK ที่มีค่าเหมือนกับ PK ของ tuple ที่กำลังถูกลบ
 - ▶ **Set null or set default:** ตามไปเปลี่ยน Attribute ที่เป็น FK ที่อ้างอิงกับ tuple ที่กำลังถูกลบให้เป็นค่า NULL หรือค่า Default
 - ▶ ตัวอย่างเช่น STUDENT-study_in-FACULTY

STUDENT.FacultyID → FACULTY.FacultyID

41

Possible violations for each operation

- ▶ **Update Operation** ในการปรับปรุงข้อมูล จะต้องมีการระบุถึง Attribute และ tuples ที่ต้องการปรับปรุงข้อมูล
 - ▶ หาก Attribute ที่ปรับปรุง **ไม่ใช่ส่วนหนึ่งของ PK หรือ FK** จะไม่มีโอกาสละเมิดกฎข้อบังคับเกี่ยวกับ Key Constraints, Entity Integrity, หรือ Referential Integrity Constraints **แต่จะมีโอกาสละเมิดกฎ**
 - ▶ **Domain Constraint และ NOT NULL Constraint**
 - ▶ หาก Attribute ที่ปรับปรุง **เป็นส่วนหนึ่งของ PK หรือ FK** จะมีโอกาสละเมิดกฎข้อบังคับเกี่ยวกับ
 - ▶ **Updating the primary key (PK):** มีโอกาสละเมิดกฎเหมือนกับ การลบข้อมูล (Delete) ตามด้วยการเพิ่มข้อมูล (Insert) เมื่อเกิดการละเมิดกฎ สามารถดำเนินการได้เหมือนกับ Delete Option
 - ▶ **Updating a foreign key (FK):** มีโอกาสละเมิดกฎ Referential Integrity
 - ▶ **Update attribute ตัวอื่น ๆ ที่ไม่ใช่ PK และไม่ใช่ FK:** มีโอกาสละเมิดกฎ Domain constraints

42

Summary

- ▶ Presented Relational Model Concepts
 - ▶ Definitions
 - ▶ Characteristics of relations
- ▶ Discussed Relational Model Constraints and Relational Database Schemas
 - ▶ Domain constraints
 - ▶ Key constraints
 - ▶ Entity integrity
 - ▶ Referential integrity
- ▶ Described the Relational Update Operations and Dealing with Constraint Violations

43

References

- ▶ Ramez Elmasri and Shamkant B. Navathe, "Fundamentals of Database Systems Seventh Edition", Pearson, 2016.

44