DATABASE MANAGEMENT SYSTEM

Unit 4: Relational Database Design Process

CE: 4.6
Normalization

And the solution is…

Normalization

Normalization is a process of decomposition of a relation (table) into more relations to avoid database anomalies (Insertion, updation, deletion)

Normalization is based on Functional Dependency

4.6. Normalization (Conti...)

One attribute in a relation uniquely determines another attribute.

A -> B means "B is functionally dependent upon A."

Teaching

Class	Teacher	RoomNo	Size
BE(IT Sem-1)	Prof. Sharma	C301	300
BE(IT Sem-1)	Dr. Patel	C301	300
BE(IT Sem-1)	Prof. Desai	C301	300
BE(IT Sem-2)	Prof. Anjali	C302	250
BE(IT Sem-2)	Dr. Neha	C302	250
BE(IT Sem-2)	Prof. Desai	C302	250
BE(IT Sem-3)	Prof. Desai	C303	300

RoomNo → Size

4.6.1. Atomic Domains and First Normal Form

1NF: ATOMIC VALUE

A relation R is in first normal form (1NF) if and only if all attributes contain atomic values.

Student

(Name, Course, Address)

Ajay | MCA | Amit Park, Surat, Gujarat

1st NF: Student

(Name, Course, Address, City, State)

Ajay | MCA | Amit Park | Surat | Gujarat

4.6.2. Second Normal Form

2NF: PRIMARY KEY

A relation R is in second normal form (2NF) if and only if it is in 1NF and every non-key attribute is fully dependent on the primary.

PK → All other attributes

4.6.2. Second Normal Form (Conti...)

Example: 1NF but not 2NF

SUPPLIER (supplier_no, status, city, part_no, quantity)

Functional Dependencies:

(supplier_no, part_no) → quantity

(supplier_no) → state

(supplier_no) → city

city → state (Supplier's state is determined by location)

Comments:

Non-key attributes are not mutually independent (city → state).

Non-key attributes are not fully functionally dependent on the primary key (i.e., state and city are dependent on just part of the key, namely supplier_no).

4.6.2. Second Normal Form (Conti...)

SUPPLIER (supplier_no, status, city, part_no, quantity)

Decomposition (into 2NF):

SUPPLIER (supplier_no, status, city)

SUPPLIER_PART (part_no, quantity, supplier_no,)

4.6.3. Third Normal Form

3NF: NOTHING BUT PRIMARY KEY

A relation R is in third normal form (3NF) if and only if it is in 2NF and every non-key attribute dependent on the primary key ONLY.

Example (2NF but not 3NF):

supplier (supplier_no, status, city)

Functional Dependencies:

supplier_no → status

supplier_no → city

city → status

status → city

4.6.3. Third Normal Form (Conti...)

Example (2NF but not 3NF):

SUPPLIER (supplier_no, status, city)

Comments:

Lacks mutual independence among non-key attributes.

city → status.

SUPPLIER (supplier_no, status, city)

Decomposition (into 3NF):

SUPPLIER_CITY (supplier_no, city)

CITY_STATUS (city, status)

4.6.3. Third Normal Form (Conti...)

1NF : ATOMIC VALUE

SUPPLIER (supplier_no, status, city, part_no, quantity)

2NF : PRIMARY KEY

SUPPLIER (supplier_no, status, city)

SUPPLIER_PART (supplier_no, part_no, quantity)

3NF : ONLY PRIMARY KEY

SUPPLIER_CITY (supplier_no, city)

CITY_STATUS (city, status)

Exercise

Identify the Normal Form of following Relation

Part	Warehouse	Quantity	WarehouseAddress
42	Boston	2000	24 Main St
333	Boston	1000	24 Main St
390	New York	3000	99 Broad St

Second normal form

Part	Warehouse	Quantity	WarehouseAddress
42	Boston	2000	24 Main St
333	Boston	1000	24 Main St
390	New York	3000	99 Broad St

- All non-key fields must be a function of the full key
  - Example that violates second normal form:
    - Key is Part + Warehouse
    - Someone found it convenient to add Address, to make a report easier
    - WarehouseAddress is a fact about Warehouse, not about Part
  - Problems:
    - Warehouse address is repeated in every row that refers to a part stored in a warehouse
    - If warehouse address changes, every row referring to a part stored in that warehouse must be updated
    - Data might become inconsistent, with different records showing different addresses for the same warehouse
    - If at some time there were no parts stored in the warehouse, there may be no record in which to keep the warehouse's address.

Second normal form

- **Solution**
  - Two entity types: Inventory, and Warehouse
  - Advantage: solves problems from last slide
  - Disadvantage: If application needs address of each warehouse stocking a part, it must access two tables instead of one. This used to be a problem but rarely is now.

Part	Warehouse	Quantity
42	Boston	2000
333	Boston	1000
390	New York	3000

Warehouse	WarehouseAddress
Boston	24 Main St
New York	99 Broad St

Exercise

Identify the Normal Form of following Relation

Customer_Details

Cust_Id	Name	Spouse	Children	City	Contact No
1001	Ajay	Vibha	Neelam Vinay	Surat	9912876
1002	Jyoti	Nimesh	Rahul	Baroda	9898222 8366254

Exercise

Identify the Normal Form of following Relation

Employee	Department	DepartmentLocation
234	Finance	Boston
223	Finance	Boston
399	Operations	Washington

Third normal form

Employee	Department	DepartmentLocation
234	Finance	Boston
223	Finance	Boston
399	Operations	Washington

- **Non-key fields cannot be a function of other non-key fields**
  - Example that violates third normal form
    - Key is employee
    - Someone found it convenient to add department location for a report
    - Department location is a function of department, which is not a key
  - Problems:
    - Department location is repeated in every employee record
    - If department location changes, every record with it must be changed
    - Data might become inconsistent
    - If a department has no employees, there may be nowhere to store its location

Third normal form

- **Solution**
  - Two entity types: Employee and department

Employee	Department
234	Finance
223	Finance
399	Operations

Department	DepartmentLocation
Finance	Boston
Operations	Washington

TV: "The truth, the whole truth, and nothing but the truth"
DB: "The key, the whole key, and nothing but the key"

Exercise

Consider the following relation and convert it up to 3rd Normal Form

R = {Student ID, Std Last Name, Std First Name, Course ID, Course Section, Course Name, Grade, Professor Last Name, Professor First Name, Bldg, Office #}

Exercise - Answer

R = {Student ID, Std Last Name, Std First Name, Course ID, Course Section, Course Name, Grade, Professor Last Name, Professor First Name, Bldg, Office #}

Student {Student ID, Last Name, First name}

Course {Student ID, Course ID, Course Section, Course Name, Grade}

Faculty {Course ID, Professor Last Name, Professor First Name, Bldg, Office#}

Exercise - Answer

Student

Student_ID	L_ Name	F_Name
001	Smith	John
002	Smith	Susan
003	Beal	Fred
004	Thomson	Marie

Courses

Student_ID	Semester	C_ID	C_Section	CName	Grade
001	Fall03	Eng01	1N	English	A
001	Spr04	Ger01	2N	German	B
002	Fall03	Eng01	1N	English	A
002	Spr04	Ger01	2N	German	B
003	Spr04	Ger01	1N	German	A
004	Spr04	Ita01	1N	Italian	B

Faculty

Semester	C_ID	C_Section	Prof_LName	Prof_FName	Bldg	Office#
Fall03	Eng01	1N	Ruger	John	LG	102
Spr04	Eng01	1N	Ruger	John	LG	102
Spr04	Ger01	1N	Findling	Holger	LG	101
Spr04	Ger01	1N	Findling	Holger	LG	101
Spr04	Ital01	2N	Fresco	Luise	LG	103

Exercise - Answer

StudentPersonalDetails = {Student ID, Last Name, First name}

StudentCourse = {Student ID, Course ID, Course Section, Course Name, Grade}

ProfessorDetails = {Course ID, Professor Last Name, Professor First Name, Bldg, Office#}

Set of Functional Dependencis

- Student ID -> Student ID
- Student ID -> Last Name, First Name
- Student ID -> Course ID
- Student ID -> C_Section, C_ Name
- Student ID -> Grade
- Course ID -> Prof_LName
- Course ID -> Prof_FName
- Course ID -> Bldg, Office #

Exercise

Consider the following relation and convert it up to 3rd Normal Form

CUSTOMER_ID	LAST_NAME	FIRST_NAME	PHONES
JS1	Smith	John	310-456-4022 (W) 310-444-8712 (H)
PH1	Pocahontas		310-432-2813 (H)

Exercise

Consider the following relation and convert it up to 3rd Normal Form

COURSE	SECTION	INSTRUCTOR	COURSE NAME
Cosc 250	1	Nguyen	Computer Science for Business
Cosc 250	2	Warford	Computer Science for Business
Cosc 250	3	Nguyen	Computer Science for Business
Cosc 250	4	Zimmerman	Computer Science for Business
Cosc 480	1	Nguyen	Programming Languages

Exercise - Answer

COURSE	SECTION	INSTRUCTOR	COURSE NAME

COURSE	COURSE NAME
Cosc 250	Computer Science for Business
Cosc 480	Programming Languages

COURSE	SECTION	INSTRUCTOR
Cosc 250	1	Nguyen
Cosc 250	2	Warford
Cosc 250	3	Nguyen
Cosc 250	4	Zimmerman
Cosc 480	1	Nguyen