Database Processing: Fundamentals, Design, and Implementation

Chapter 3

The Relational Model and Normalization

semicolon

Learning Objectives

- **3.1** To understand basic relational terminology
- **3.2** To understand the characteristics of relations
- **3.3** To understand alternative terminology used in describing the relational model
- **3.4** To be able to identify functional dependencies, determinants, and the dependent attributes
- **3.5** To identify primary, candidate, and composite keys
- **3.6** To be able to identify possible insertion, deletion, and update anomalies in a relation
- **3.7** To be able to place a relation into BCNF normal form
- **3.8** To understand the special importance of domain key normal form
- **3.9** To be able to identify multivalued dependencies
- **3.10** To be able to place a relation in fourth normal form

Chapter Premise

- We have received one or more tables of existing data.
- The data is to be stored in a new database.
- QUESTION: Should the data be stored as received, or should it be transformed for storage?

Figure 3-1
How Many Tables?

ORDER_ITEM

	OrderNumber	SKU	Quantity	Price	ExtendedPrice
1	1000	201000	1	300.00	300.00
2	1000	202000	1	130.00	130.00
3	2000	101100	4	50.00	200.00
4	2000	101200	2	50.00	100.00
5	3000	100200	1	300.00	300.00
6	3000	101100	2	50.00	100.00
7	3000	101200	1	50.00	50.00

SKU_DATA

	SKU	SKU_Description	Department	Buyer
1	100100	Std. Scuba Tank, Yellow	Water Sports	Pete Hansen
2	100200	Std. Scuba Tank, Magenta	Water Sports	Pete Hansen
3	100300	Std. Scuba Tank, Light Blue	Water Sports	Pete Hansen
4	100400	Std. Scuba Tank, Dark Blue	Water Sports	Pete Hansen
5	100500	Std. Scuba Tank, Light Green	Water Sports	Pete Hansen
6	100600	Std. Scuba Tank, Dark Green	Water Sports	Pete Hansen
7	101100	Dive Mask, Small Clear	Water Sports	Nancy Meyers
8	101200	Dive Mask, Med Clear	Water Sports	Nancy Meyers
9	201000	Half-dome Tent	Camping	Cindy Lo
10	202000	Half-dome Tent Vestibule	Camping	Cindy Lo
11	203000	Half-dome Tent Vestibule - Wide	Camping	Cindy Lo
12	301000	Light Fly Climbing Harness	Climbing	Jerry Martin
13	302000	Locking Carabiner, Oval	Climbing	Jerry Martin

SKU_ITEM

	OrderNumber	SKU	Quantity	Price	SKU_Description	Department	Buyer
1	1000	201000	1	300.00	Half-dome Tent	Camping	Cindy Lo
2	1000	202000	1	130.00	Half-dome Tent Vestibule	Camping	Cindy Lo
3	2000	101100	4	50.00	Dive Mask, Small Clear	Water Sports	Nancy Meyers
4	2000	101200	2	50.00	Dive Mask, Med Clear	Water Sports	Nancy Meyers
5	3000	100200	1	300.00	Std. Scuba Tank, Magenta	Water Sports	Pete Hansen
6	3000	101100	2	50.00	Dive Mask, Small Clear	Water Sports	Nancy Meyers
7	3000	101200	1	50.00	Dive Mask, Med Clear	Water Sports	Nancy Meyers

Figure 3-2
A Very Strange Table!

PRODUCT_BUYER

	BuyerName	SKU_Managed	CollegeMajor
1	Pete Hansen	100100	Business Administration
2	Pete Hansen	100200	Business Administration
3	Pete Hansen	100300	Business Administration
4	Pete Hansen	100400	Business Administration
5	Pete Hansen	100500	Business Administration
6	Pete Hansen	100600	Business Administration
7	Nancy Meyers	101100	Art
8	Nancy Meyers	101100	Info Systems
9	Nancy Meyers	101200	Art
10	Nancy Meyers	101200	Info Systems
11	Cindy Lo	201000	History
12	Cindy Lo	202000	History
13	Cindy Lo	203000	History
14	Jenny Martin	301000	Business Administration
15	Jenny Martin	301000	English Literature
16	Jenny Martin	302000	Business Administration
17	Jenny Martin	302000	English Literature

- To understand why this is a very odd table, suppose that **Nancy Meyers** is assigned a new, **SKU 101300**! What addition should we make to this table? One or two rows?

But First--

- We need to understand:
  - The **relational model**
  - **Relational model terminology**

The Relational Model

- Introduced in a paper published in 1970
- Created by E.F. Codd
  - IBM engineer
  - The model used mathematics known as “relational algebra”
- Now the standard model for commercial DBMS products

Figure 3-3
Important Relational Model Terms

Important Relational Terms
Relation
Functional Dependency
Determinant
Candidate Key
Composite Key
Primary Key
Surrogate Key
Foreign Key
Referential integrity constraint
Normal forms
Multivalued dependency

Entity

- An **entity** is some identifiable thing that users want to track:
  - Customers
  - Computers
  - Sales

Figure 3-4 Relation

- **Relational DBMS products** store data about entities in relations, which are a special type of table.
- A **relation** is a two-dimensional table that has the following characteristics:

Characteristics of Relations
Rows contain data about an entity.
Columns contain data about attributes of the entities.
All entries in a column are of the same kind.
Each column has a unique name.
Cells of the table hold a single value.
The order of the columns is unimportant.
The order of the rows is unimportant.
No two rows may be identical.

The Domain Integrity Constraint

- The requirement that all of the values in a column are of the same kind is known as the **domain integrity constraint**.
- The term domain means a grouping of data that meets a specific type definition.
  - **FirstName** could have a domain of names such as *Albert, Bruce, Cathy, David, Edith*, and so forth.
  - All values of **FirstName** *must* come from the names in that domain.
- Columns in different relations may have the same name.

Figure 3-5
A Sample EMPLOYEE Relation

EmployeeNumber	FirstName	LastName	Department	EmailAddress	Phone
100	Jerry	Johnson	Accounting	JJ@somewhere.com	518-834-1101
200	Mary	Abernathy	Finance	MA@somewhere.com	518-834-2101
300	Liz	Smathers	Finance	LS@somewhere.com	518-834-2102
400	Tom	Caruthers	Accounting	TC@somewhere.com	518-834-1102
500	Tom	Jackson	Production	TJ@somewhere.com	518-834-4101
600	Eleanore	Caldera	Legal	EC@somewhere.com	518-834-3101
700	Richard	Bandalone	Legal	RB@somewhere.com	518-834-3102

Figure 3-6 Tables That are Not Relations: Multiple Entries per Cell

EmployeeNumber	FirstName	LastName	Department	EmailAddress	Phone
100	Jerry	Johnson	Accounting	JJ@somewhere.com	518-834-1101
200	Mary	Abernathy	Finance	MA@somewhere.com	518-834-2101
300	Liz	Smathers	Finance	LS@somewhere.com	518-834-2102
400	Tom	Caruthers	Accounting	TC@somewhere.com Fax: Home:	518-834-2102 518-834-9911 518-723-8795
500	Tom	Jackson	Production	TJ@somewhere.com	518-834-4101
600	Eleanore	Caldera	Legal	EC@somewhere.com Fax: Home:	518-834-3101 518-834-9912 518-723-7654
700	Richard	Bandalone	Legal	RB@somewhere.com	518-834-3102

Figure 3-7 Tables That Are Not Relations
Table with Required Row Order

EmployeeNumber	FirstName	LastName	Department	EmailAddress	Phone
100	Jerry	Johnson	Accounting	JJ@somewhere.com	518-834-1101
200	Mary	Abernathy	Finance	MA@somewhere.com	518-834-2101
300	Liz	Smathers	Finance	LS@somewhere.com	518-834-2102
400	Tom	Caruthers	Accounting	TC@somewhere.com	518-834-2102
				Fax:	518-834-9911
				Home:	518-723-8795
500	Tom	Jackson	Production	TJ@somewhere.com	518-834-4101
600	Eleanore	Caldera	Legal	EC@somewhere.com	518-834-3101
				Fax:	518-834-9912
				Home:	518-723-7654
700	Richard	Bandalone	Legal	RB@somewhere.com	518-834-3102

Figure 3-8
A Relation with Values of Varying Length

EmployeeNumber	FirstName	LastName	Department	EmailAddress	Phone	Comment
100	Jerry	Johnson	Accounting	JJ@somewhere.com	518-834-1101	Joined the Accounting Department in March after completing his MBA. Will take the CPA exam this fall.
200	Mary	Abernathy	Finance	MA@somewhere.com	518-834-2101	
300	Liz	Smathers	Finance	LS@somewhere.com	518-834-2102	
400	Tom	Caruthers	Accounting	TC@somewhere.com	518-834-1102	
500	Tom	Jackson	Production	TJ@somewhere.com	518-834-4101	
600	Eleanore	Caldera	Legal	EC@somewhere.com	518-834-3101	
700	Richard	Bandalone	Legal	RB@somewhere.com	518-834-3102	Is a full-time consultant to Legal on a retainer basis.

Figure 3-9 Alternative Terminology

- The terms *table* and *relation* are normally used interchangeably.
- The following sets of terms are equivalent:

Table	Column	Row
Relation	Attribute	Tuple
File	Field	Record

To Key, or Not to Key That is the Question!

- In a relation as defined by Codd:
  - The rows of a relation must be *unique.*
  - These is *no requirement* for a *designated primary key.*
- The requirement for unique rows implies that a primary key can be designated.
- In the “real world,” every relation has a primary key.
- When do we designate a primary key?
- **We need some more information!**

Functional Dependencies

- A **functional dependency** occurs when the value of one (set of) attribute(s) determines the value of a second (set of) attribute(s):

  CookieCost = NumberOfBoxes x $5

  NumberOfBoxes → CookieCost
- The attribute on the left side of the functional dependency is called the **determinant**.
- Functional dependencies may be *based* on equations:

  ExtendedPrice = Quantity X UnitPrice

  (Quantity, UnitPrice) → ExtendedPrice
- Function dependencies are *not* equations!

Functional Dependencies

ObjectColor	Weight	Shape
Red	5	Ball
Blue	5	Cube
Yellow	7	Cube

ObjectColor □ Weight
ObjectColor □ Shape
ObjectColor □ (Weight, Shape)

Composite Determinants

- **Composite determinant** = a determinant of a functional dependency that consists of more than one attribute

  (StudentNumber, ClassNumber) (Grade)

Functional Dependency Rules

- If A □ (B, C), then A □ B and A□ C.
  - This is the **decomposition rule**.
- If A □ B and A□ C, then A □ (B, C).
  - This is the **union rule**.
- However, if (A,B) □ C, then *neither A nor B determines C by itself*.

Functional Dependencies in the SKU_DATA Table (1 of 2)

	SKU	SKU_Description	Department	Buyer
1	100100	Std. Scuba Tank, Yellow	Water Sports	Pete Hansen
2	100200	Std. Scuba Tank, Magenta	Water Sports	Pete Hansen
3	100300	Std. Scuba Tank, Light Blue	Water Sports	Pete Hansen
4	100400	Std. Scuba Tank, Dark Blue	Water Sports	Pete Hansen
5	100500	Std. Scuba Tank, Light Green	Water Sports	Pete Hansen
6	100600	Std. Scuba Tank, Dark Green	Water Sports	Pete Hansen
7	101100	Dive Mask, Small Clear	Water Sports	Nancy Meyers
8	101200	Dive Mask, Med Clear	Water Sports	Nancy Meyers
9	201000	Half-dome Tent	Camping	Cindy Lo
10	202000	Half-dome Tent Vestibule	Camping	Cindy Lo
11	203000	Half-dome Tent Vestibule - Wide	Camping	Cindy Lo
12	301000	Light Fly Climbing Harness	Climbing	Jerry Martin
13	302000	Locking Carabiner, Oval	Climbing	Jerry Martin

Functional Dependencies in the SKU_DATA Table (2 of 2)

SKU → (SKU_Description, Department, Buyer)

SKU_Description → (SKU, Department, Buyer)

Buyer → Department

Functional Dependencies in the ORDER_ITEM Table (1 of 2)

	OrderNumber	SKU	Quantity	Price	ExtendedPrice
1	1000	201000	1	300.00	300.00
2	1000	202000	1	130.00	130.00
3	2000	101100	4	50.00	200.00
4	2000	101200	2	50.00	100.00
5	3000	100200	1	300.00	300.00
6	3000	101100	2	50.00	100.00
7	3000	101200	1	50.00	50.00

Functional Dependencies in the ORDER_ITEM Table (1 of 2)

(OrderNumber, SKU) □

(Quantity, Price, ExtendedPrice)

(Quantity, Price) □ (ExtendedPrice)

What Makes Determinant Values Unique?

- A determinant is unique in a relation if and only if it determines every other column in the relation.
- You cannot find the determinants of all functional dependencies simply by looking for unique values in one column:
  - Must be logically a determinant

Keys

- A **key** is a combination of one or more columns that is used to identify particular rows in a relation.
- A **composite key** is a key that consists of two or more columns.

Candidate and Primary Keys

- A **candidate key** is a key with no repeated values.
- A table can have multiple **candidate key**
- It uniquely identifies each record in a table
- A **primary key** is a candidate key selected as the primary means of identifying rows in a relation.
  - There is only one primary key per relation.
  - The primary key may be a single key or a composite key.

Why we need a key

- Keys help identify any row of data in a table. In a real world application, a table could contain thousands of records. Moreover the records could be duplicated. Keys ensure that you can uniquely identify a table record despite these challenges.
- Allows you to establish a relationship between and identify the relation between tables.
- Helps to enforce identity and integrity in the relationship.

The Entity Integrity Constraint

- The requirement that, in order to function properly, the primary key must have unique data values inserted into every row of the table. This is known as the **entity integrity constraint**.
- The phrase *unique data values* implies that this column is NOT NULL, and does not allow a NULL value in any row.

Surrogate Keys (1 of 2)

- A **surrogate key** is an artificial column added to a relation to serve as a primary key.
  - DBMS supplied
  - Short, numeric, and never changes—an ideal primary key
  - Has artificial values that are meaningless to users
  - Normally hidden in forms and reports

Foreign Keys (1 of 2)

A **foreign key** is a column or composite of columns that is the primary key of a table other than the one in which it appears. The term arises because it is a key of a table *foreign* to the one in which it appears as the primary key.

Foreign Keys (2 of 2)

NOTE: The primary keys of the relations are underlined and any foreign keys are in *italics* in the relations below:

DEPARTMENT (DepartmentName, BudgetCode, ManagerName)
EMPLOYEE (EmployeeNumber, EmployeeLastName, EmployeeFirstName, *DepartmentName*)

The Referential Integrity Constraint

- A **referential integrity constraint** is a statement that limits the values of the foreign key to those already existing as primary key values in the corresponding relation:

SKU in ORDER_ITEM must exist in SKU in SKU_DATA

Foreign Key with a Referential Integrity Constraint

NOTE: The primary key of the relation is underlined and any foreign keys are in *italics* in the relations below:

SKU_DATA (SKU, SKU_Description, Department, Buyer)

ORDER_ITEM (OrderNumber, *SKU*, Quantity, Price, ExtendedPrice)

Where ORDER_ITEM.SKU must exist in SKU_DATA.SKU

Database Integrity

- We have defined three constraints so far in our discussion:
  - The **domain integrity constraint**
  - The **entity integrity constraint**
  - The **referential integrity constraint**
- The purpose of these three constraints, taken as a whole, is to create **database integrity**, which means that the data in our database will be useful, meaningful data.

Figures 3-10 & 3-11
Modification Anomalies

- The EQUIPMENT_REPAIR table before and after an incorrect update operation on AcquisitionCost for EquipmentType = Drill Press:

	ItemNumber	EquipmentType	AcquisitionCost	RepairNumber	RepairDate	RepairCost
1	100	Drill Press	3500.00	2000	2018-05-05	375.00
2	200	Lathe	4750.00	2100	2018-05-07	255.00
3	100	Drill Press	3500.00	2200	2018-06-19	178.00
4	300	Mill	27300.00	2300	2018-06-19	1875.00
5	100	Drill Press	3500.00	2400	2018-07-05	0.00
6	100	Drill Press	3500.00	2500	2018-08-17	275.00

	ItemNumber	EquipmentType	AcquisitionCost	RepairNumber	RepairDate	RepairCost
1	100	Drill Press	3500.00	2000	2018-05-05	375.00
2	200	Lathe	4750.00	2100	2018-05-07	255.00
3	100	Drill Press	3500.00	2200	2018-06-19	178.00
4	300	Mill	27300.00	2300	2018-06-19	1875.00
5	100	Drill Press	3500.00	2400	2018-07-05	0.00
6	100	Drill Press	5500.00	2500	2018-08-17	275.00

Types of Modification Anomalies

- Deletion anomaly
- Insertion anomaly
- Update anomaly
  - Notice that the EQUIPMENT_REPAIR table duplicates data. For example, the AcquisitionCost of the same item of equipment appears several times. Any table that duplicates data is susceptible to update anomalies. A table that has such inconsistencies is said to have **data integrity problems**.

Figure 3-12
Summary of Normalization Theory

- Relations are categorized as a **normal form** based on which modification anomalies or other problems they are subject to:

Sources of Anomaly	Normal Forms	Design Principles
Functional Dependencies	1NF, 2NF 3NF, BCNF	BCNF: Design tables so that every determinant is a candidate key
Multivalued dependencies	4NF	4NF: Move each multivalued dependency to a table of its own
Data constraints and oddities	5NF, DK/NF	DK/NF: Make every constraint a logical consequence of candidate keys and domains

First Normal Form (1NF)

- Remember that question - To Key or Not to Key?
- Codd's set of conditions for a relation does not require a primary key, but one is clearly implied by the condition that all rows must be unique. Thus, we will define 1NF as:
  - Meets the set of conditions for a relation
  - Has a defined primary key
- A review of Figure 3-4 lists the characteristics of a relation.

Second Normal Form (2NF)

- A relation is in 2NF if, and only if, it is in 1NF and all non-key attributes are determined by the entire primary key. This is also known as a Partial Dependency.
- For example:
  - A relation R(A, B, N, O, P) with the composite key (A,B) means that none of the non-key attributes N, O, or P can be determined by just A or just B.

(StudentID, Activity) □ ActivityFee

Activity □x ActivityFee

Third Normal Form (3NF)

- A relation is in 3NF if, and only if, it is in 2NF and there are no non-key attributes determined by another non-key attribute. This is also known as a Transitive Dependency.
- For example:
  - A relation R(A, B, N, O, P) with the composite key (A,B) means that none of the non-key attributes N, O, or P can be determined by N, O, or P.

STUDENT_HOUSING (StudentID, Building, BuildingFee)

Building □x BuildingFee

Boyce-Codd Normal Form (BCNF)

- A relation is in BCNF if, and only if, it is in 3NF and every determinant is a candidate key.

Figure 3-19 Eliminating Anomalies from Functional Dependencies with BCNF

Process for Putting a Relation into BCNF

1. Identify every functional dependency.
2. Identify every candidate key.
3. If there is a functional dependency that has a determinate that is not a candidate key:
   - A. Move the column of that functional dependency into a new relation.
   - B. Make the determinant of that functional dependency the primary key of the new relation.
   - C. Leave a copy of the determinant as a foreign key in the original relation.
   - D. Create a referential integrity constraint between the original relation and the new relation.
4. Repeat step 3 until every determinant of every relation is a candidate key.

Note: In step 3, if there is more than one such functional dependency, start with the one with the most columns.

Figure 3-20 Putting the SKU_DATA Table into BCNF (Step-by-Step) - 1NF

SKU_DATA

	SKU	SKU_Description	Department	Buyer
1	100100	Std. Scuba Tank, Yellow	Water Sports	Pete Hansen
2	100200	Std. Scuba Tank, Magenta	Water Sports	Pete Hansen
3	100300	Std. Scuba Tank, Light Blue	Water Sports	Pete Hansen
4	100400	Std. Scuba Tank, Dark Blue	Water Sports	Pete Hansen
5	100500	Std. Scuba Tank, Light Green	Water Sports	Pete Hansen
6	100600	Std. Scuba Tank, Dark Green	Water Sports	Pete Hansen
7	101100	Dive Mask, Small Clear	Water Sports	Nancy Meyers
8	101200	Dive Mask, Med Clear	Water Sports	Nancy Meyers
9	201000	Half-dome Tent	Camping	Cindy Lo
10	202000	Half-dome Tent Vestibule	Camping	Cindy Lo
11	203000	Half-dome Tent Vestibule - Wide	Camping	Cindy Lo
12	301000	Light Fly Climbing Harness	Climbing	Jerry Martin
13	302000	Locking Carabiner, Oval	Climbing	Jerry Martin

SKU_DATA (SKU, SKU_Description, Department, Buyer)

1NF – Checking against 1NF, this relation is in 1NF.

Putting the SKU_DATA Table into BCNF (Step-by-Step) - 2NF

SKU_DATA (SKU, SKU_Description, Department, Buyer)

SKU □ (SKU_Description, Department, Buyer)
SKU_Description □ (SKU, Department, Buyer)
Buyer □ Department

- SKU and SKU_Description are candidate keys.
- A relation is in 2NF if, and only if, it is in 1NF and all non-key attributes are determined by the primary key.
- Since SKU is a single column primary key, all non-key attributes are determined by SKU, and the relation is in 2NF.

Putting the SKU_DATA Table into BCNF (Step-by-Step) - 3NF (1 of 2)

SKU_DATA (SKU, SKU_Description, Department, Buyer)

SKU □ (SKU_Description, Department, Buyer)
SKU_Description □ (SKU, Department, Buyer)
Buyer □ Department

- SKU and SKU_Description are candidate keys.
- A relation is in 3NF if, and only if, it is in 2NF and there are no non-key attributes determined by another non-key attribute.
- However, the term non-key attribute means an attribute that is neither (1) a candidate key itself, nor (2) part of a composite candidate key.
- Therefore, the only non key attribute is Buyer, and it is a determinant of Department.
- Therefore, this is not in 3NF.

Putting the SKU_DATA Table into BCNF (Step-by-Step) - 3NF (2 of 2)

- Therefore, break out the Buyer □ Department functional dependency.

SKU_DATA_2 (SKU, SKU_Description, *Buyer*)
BUYER (Buyer, Department)

Where SKU_DATA_2.Buyer must exist in BUYER.Buyer

- SKU_DATA_2 is in 3NF
- BUYER is in 3NF

Putting the SKU_DATA Table into BCNF (Step-by-Step) - BCNF

SKU_DATA_2 (SKU, SKU_Description, *Buyer*)
BUYER (Buyer, Department)

Where SKU_DATA_2.Buyer must exist in BUYER.Buyer

SKU □ (SKU_Description, Department, Buyer)
SKU_Description □ (SKU, Department, Buyer)
Buyer □ Department

- A relation is in BCNF if, and only if, it is in 3NF and every determinant is a candidate key.
- In SKU_DATA_2, both determinants are determinant keys, so SKU_DATA_2 is in BCNF
- In BUYER, the determinant is a determinate key, so BUYER is in BCNF.

Figure 3-21 The Results of the Step-by-Step to BCNF

SKU_DATA_2

	SKU	SKU_Description	Buyer
1	100100	Std. Scuba Tank, Yellow	Pete Hansen
2	100200	Std. Scuba Tank, Magenta	Pete Hansen
3	100300	Std. Scuba Tank, Light Blue	Pete Hansen
4	100400	Std. Scuba Tank, Dark Blue	Pete Hansen
5	100500	Std. Scuba Tank, Light Green	Pete Hansen
6	100600	Std. Scuba Tank, Dark Green	Pete Hansen
7	101100	Dive Mask, Small Clear	Nancy Meyers
8	101200	Dive Mask, Med Clear	Nancy Meyers
9	201000	Half-dome Tent	Cindy Lo
10	202000	Half-dome Tent Vestibule	Cindy Lo
11	203000	Half-dome Tent Vestibule - Wide	Cindy Lo
12	301000	Light Fly Climbing Harness	Jerry Martin
13	302000	Locking carabiner, Oval	Jerry Martin

BUYER_2

	Buyer	Department
1	Cindy Lo	Camping
2	Jerry Martin	Climbing
3	Nancy Meyers	Water Sports
4	Pete Hansen	Water Sports

Figure 3-20 The "Straight-to-BCNF" Method

SKU_DATA

	SKU	SKU_Description	Department	Buyer
1	100100	Std. Scuba Tank, Yellow	Water Sports	Pete Hansen
2	100200	Std. Scuba Tank, Magenta	Water Sports	Pete Hansen
3	100300	Std. Scuba Tank, Light Blue	Water Sports	Pete Hansen
4	100400	Std. Scuba Tank, Dark Blue	Water Sports	Pete Hansen
5	100500	Std. Scuba Tank, Light Green	Water Sports	Pete Hansen
6	100600	Std. Scuba Tank, Dark Green	Water Sports	Pete Hansen
7	101100	Dive Mask, Small Clear	Water Sports	Nancy Meyers
8	101200	Dive Mask, Med Clear	Water Sports	Nancy Meyers
9	201000	Half-dome Tent	Camping	Cindy Lo
10	202000	Half-dome Tent Vestibule	Camping	Cindy Lo
11	203000	Half-dome Tent Vestibule - Wide	Camping	Cindy Lo
12	301000	Light Fly Climbing Harness	Climbing	Jerry Martin
13	302000	Locking Carabiner, Oval	Climbing	Jerry Martin

The “Straight-to-BCNF” Method

SKU_DATA (SKU, SKU_Description, Department, Buyer)

SKU → (SKU_Description, Department, Buyer)
SKU_Description → (SKU, Department, Buyer)
Buyer → Department

- Therefore, break out the Buyer → Department functional dependency

SKU_DATA (SKU, SKU_Description, *Buyer*)
BUYER (Buyer, Department)

Where BUYER.Buyer must exist in SKU_DATA.Buyer

Figure 3-21 The Results of the Straight-to-BCNF

SKU_DATA_2

	SKU	SKU_Description	Buyer
1	100100	Std. Scuba Tank, Yellow	Pete Hansen
2	100200	Std. Scuba Tank, Magenta	Pete Hansen
3	100300	Std. Scuba Tank, Light Blue	Pete Hansen
4	100400	Std. Scuba Tank, Dark Blue	Pete Hansen
5	100500	Std. Scuba Tank, Light Green	Pete Hansen
6	100600	Std. Scuba Tank, Dark Green	Pete Hansen
7	101100	Dive Mask, Small Clear	Nancy Meyers
8	101200	Dive Mask, Med Clear	Nancy Meyers
9	201000	Half-dome Tent	Cindy Lo
10	202000	Half-dome Tent Vestibule	Cindy Lo
11	203000	Half-dome Tent Vestibule - Wide	Cindy Lo
12	301000	Light Fly Climbing Harness	Jerry Martin
13	302000	Locking carabiner, Oval	Jerry Martin

BUYER_2

	Buyer	Department
1	Cindy Lo	Camping
2	Jerry Martin	Climbing
3	Nancy Meyers	Water Sports
4	Pete Hansen	Water Sports

Figure 3-24 The STUDENT_ACTIVITY Normalization Example (1 of 4)

STUDENT_ACTIVITY

	StudentID	StudentName	Activity	ActivityFee	AmountPaid
1	100	Jones	Golf	65.00	65.00
2	100	Jones	Skiing	200.00	0.00
3	200	Davis	Skiing	200.00	0.00
4	200	Davis	Swimming	50.00	50.00
5	300	Garrett	Skiing	200.00	100.00
6	300	Garrett	Swimming	50.00	50.00
7	400	Jones	Golf	65.00	65.00
8	400	Jones	Swimming	50.00	50.00

The STUDENT_ACTIVITY Normalization Example (2 of 4)

- **STUDENT_ACTIVITY (StudentID, StudentName, Activity, ActivityFee, AmountPaid)**

1. Identify the functional Dependencies

   StudentID □ StudentName

   StudentID □ Activity (depends if student can or cannot belong to just one activity!)

   Activity □ ActivityFee

   (StudentID, Activity) □ ActivityFee (assume a different fee for each different activity)

   (StudentID, Activity) □ AmountPaid

2. Thus we have three determinants (StudentID, Activity, and (StudentID, Activity)

The STUDENT_ACTIVITY Normalization Example (3 of 4)

To normalize this table, we need to construct tables so that every determinant is a candidate key. We can do this by creating a separate table for each functional dependency as before.

STUDENT (StudentID, StudentName)

ACTIVITY (Activity, ActivityFee)

PAYMENT (*StudentID*, *Activity*, AmountPaid)

These tables are in BCNF and will have no anomalies from functional dependencies.

Figure 3-25 The STUDENT_ACTIVITY Normalization Example (4 of 4)

STUDENT

	StudentID	StudentName
1	100	Jones
2	200	Davis
3	300	Garrett
4	400	Jones

ACTIVITY

	Activity	ActivityFee
1	Golf	65.00
2	Skiing	200.00
3	Swimming	50.00

PAYMENT

	StudentID	Activity	AmountPaid
1	100	Golf	65.00
2	100	Skiing	0.00
3	200	Skiing	0.00
4	200	Swimming	50.00
5	300	Skiing	100.00
6	300	Swimming	50.00
7	400	Golf	65.00
8	400	Swimming	50.00

Multivalued Dependencies

- A **multivalued dependency** occurs when a determinant is matched with a particular *set* of values:

  Employee □□ Degree

  Employee □□ Sibling

  PartKit □□ Part

- The determinant of a multivalued dependency can *never* be a primary key.

Figure 3-28 Examples of Multivalued Dependencies

EMPLOYEE_DEGREE

	EmployeeName	EmployeeDegree
1	Chau	BS
2	Green	BS
3	Green	MS
4	Green	PhD
5	Jones	AA
6	Jones	BA

EMPLOYEE_SIBLING

	EmployeeName	EmployeeSibling
1	Chau	Eileen
2	Chau	Jonathan
3	Green	Nikki
4	Jones	Frank
5	Jones	Fred
6	Jones	Sally

PARTKIT_PART

	PartKitName	Part
1	Bike Repair	Screwdriver
2	Bike Repair	Tube Fix
3	Bike Repair	Wrench
4	First Aid	Aspirin
5	First Aid	Bandaids
6	First Aid	Elastic Band
7	First Aid	Ibuprofin
8	Toolbox	Drill
9	Toolbox	Drill bits
10	Toolbox	Hammer
11	Toolbox	Saw
12	Toolbox	Screwdriver

Figure 3-29 Two Multivalued Dependencies

EMPLOYEE_DEGREE_SIBLING

	EmployeeName	EmployeeDegree	EmployeeSibling
1	Chau	BS	Eileen
2	Chau	BS	Jonathan
3	Green	BS	Nikki
4	Green	MS	Nikki
5	Green	PhD	Nikki
6	Jones	AA	Frank
7	Jones	AA	Fred
8	Jones	AA	Sally
9	Jones	BA	Frank
10	Jones	BA	Fred
11	Jones	BA	Sally

Fourth Normal Form (4NF)
Eliminating Multivalued Anomalies

- Multivalued dependencies are not a problem if they are in a separate relation, therefore:
  - Always put multivalued dependencies into their own relation
  - This is known as **Fourth Normal Form (4NF)**

Figure 3-2 Remember That Very Strange Table?

PRODUCT_BUYER

	BuyerName	SKU_Managed	CollegeMajor
1	Pete Hansen	100100	Business Administration
2	Pete Hansen	100200	Business Administration
3	Pete Hansen	100300	Business Administration
4	Pete Hansen	100400	Business Administration
5	Pete Hansen	100500	Business Administration
6	Pete Hansen	100600	Business Administration
7	Nancy Meyers	101100	Art
8	Nancy Meyers	101100	Info Systems
9	Nancy Meyers	101200	Art
10	Nancy Meyers	101200	Info Systems
11	Cindy Lo	201000	History
12	Cindy Lo	202000	History
13	Cindy Lo	203000	History
14	Jenny Martin	301000	Business Administration
15	Jenny Martin	301000	English Literature
16	Jenny Martin	302000	Business Administration
17	Jenny Martin	302000	English Literature

Now we understand why this is a very strange table. It has **multivalued dependencies!**

Figure 3-31 4NF

- Use **4NF** to resolve the **multivalued dependencies**!

PRODUCT_BUYER_SKU

	BuyerName	SKU_Managed
1	Cindy Lo	201000
2	Cindy Lo	202000
3	Jenny Martin	301000
4	Jenny Martin	302000
5	Nancy Meyers	101100
6	Nancy Meyers	101200
7	Pete Hansen	100100
8	Pete Hansen	100200

PRODUCT_BUYER_MAJOR

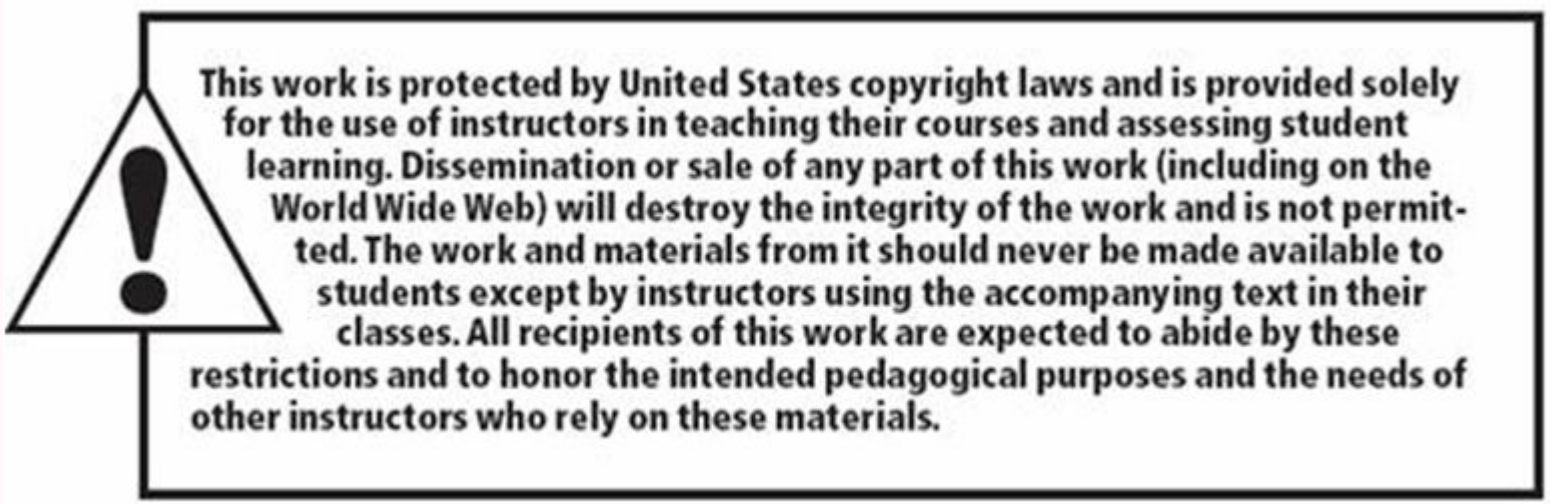
	BuyerName	CollegeMajor
1	Cindy Lo	History
2	Jenny Martin	Business Administration
3	Jenny Martin	English Literature
4	Nancy Meyers	Art
5	Nancy Meyers	Info Systems
6	Pete Hansen	Business Administration

Database Processing
Fundamentals, Design, and Implementation (14th Edition)

End of Presentation:

Chapter Three

Copyright



This work is protected by United States copyright laws and is provided solely for the use of instructors in teaching their courses and assessing student learning. Dissemination or sale of any part of this work (including on the World Wide Web) will destroy the integrity of the work and is not permitted. The work and materials from it should never be made available to students except by instructors using the accompanying text in their classes. All recipients of this work are expected to abide by these restrictions and to honor the intended pedagogical purposes and the needs of other instructors who rely on these materials.