Chapter 7

对冗余的控制（反范式模式）

Consider Introduction of Controlled Redundancy

Consider the Introduction of Controlled Redundancy

- Determine whether introducing redundancy in a controlled manner by relaxing the normalization rules will improve system performance.

Denormalization

- Refinement to relational schema such that the degree of normalization for a modified table is less than the degree of at least one of the original tables.

- Also use term more loosely to refer to situations where two tables are combined into one new table, which is still normalized but contains more nulls than original tables.
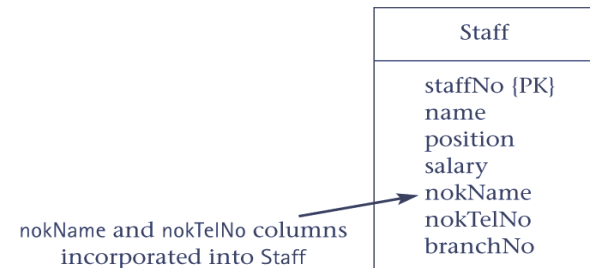
Consider the Introduction of Controlled Redundancy

- Consider denormalization in following situations, specifically to speed up frequent or critical transactions:
    - Pattern 1 Combining 1:1 relationships
    - Pattern 2 Duplicating nonkey columns in 1:* relationships to reduce joins
    - Pattern 3 Duplicating FK columns in 1:* relationships to reduce joins
    - Pattern 4 Duplicating columns in *:* relationships to reduce joins
    - Pattern 5 Introducing repeating groups
    - Pattern 6 Creating extract tables
    - Pattern 7 Partitioning tables.
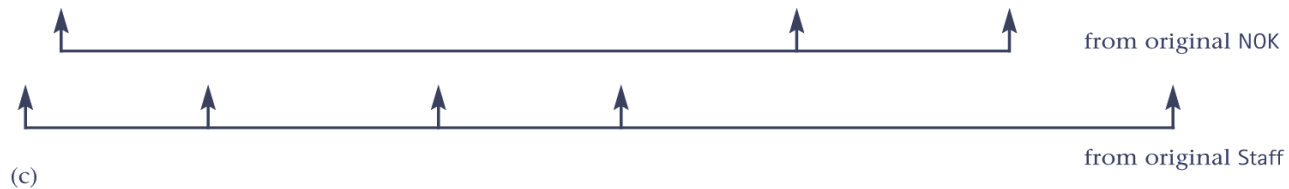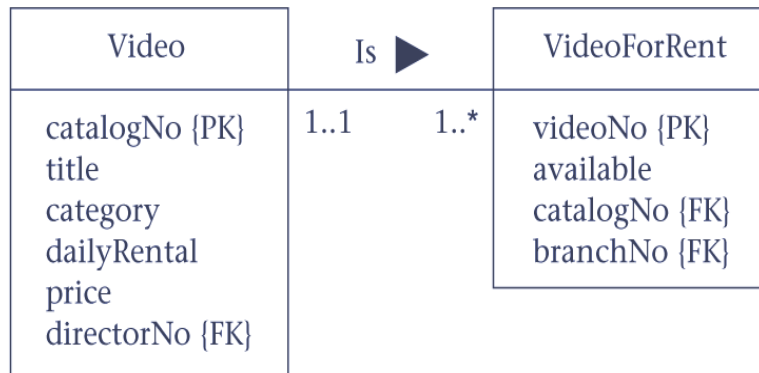
Pattern 1 Combining 1:1 relationships

Staff | RelatedTo ▶ | NOK
staffNo {PK}, name, position, salary, branchNo | 1..1   0..1 | staffNo {PK}, nokName, nokTelNo

(a)

Staff
staffNo {PK}
name
position
salary
nokName
nokTelNo
branchNo

nokName and nokTelNo columns
incorporated into Staff

(b)

Staff

staffNo	name	position	salary	nokName	nokTelNo	branchNo
S1500	Tom Daniels	Manager	46000	Jane Daniels	207-878-2751	B001
S0003	Sally Adams	Assistant	30000	John Adams	518-474-5355	B001
S0010	Mary Martinez	Manager	50000			B002
S3250	Robert Chin	Supervisor	32000	Michelle Chin	206-655-9867	B002
S2250	Sally Stern	Manager	48000			B004
S0415	Art Peters	Manager	41000	Amy Peters	718-507-7923	B003

from original NOK

from original Staff

(c)

Pattern 2 Duplicating nonkey columns in 1:* relationships to reduce joins

Video		VideoForRent
catalogNo {PK} title category dailyRental price directorNo {FK}	Is ▶ 1..1 1..*	videoNo {PK} available catalogNo {FK} branchNo {FK}

(a)

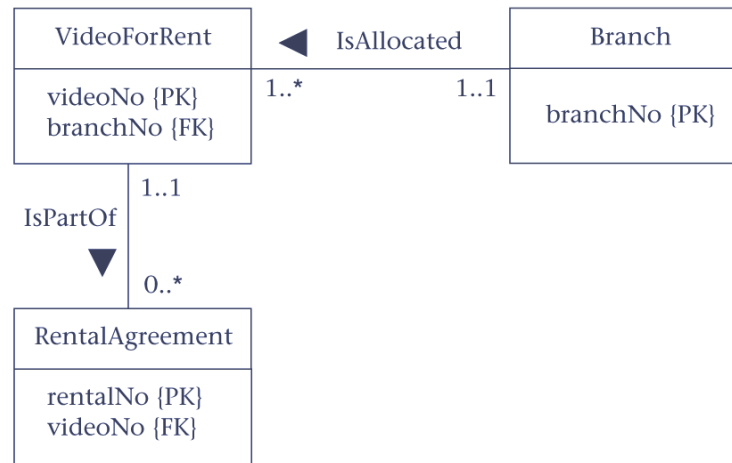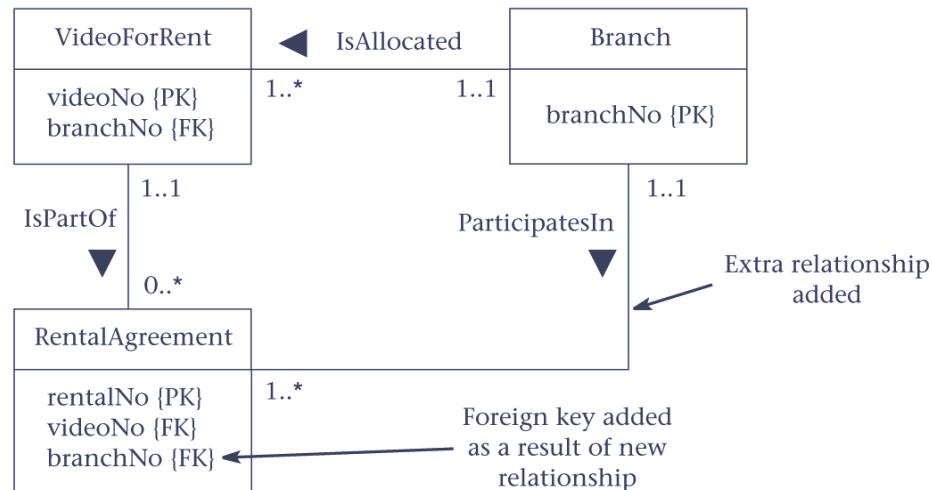| Video		VideoForRent
catalogNo {PK} title category dailyRental price directorNo {FK}	Is ▶ 1..1 1..*	videoNo {PK} available dailyRental catalogNo {FK} branchNo {FK}
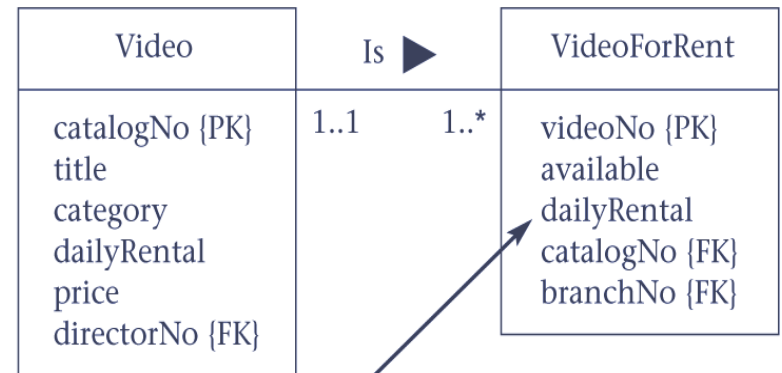
dailyRental column
duplicated in VideoForRent

(b)

Pattern 3 Duplicating FK columns in 1:* relationship to reduce joins

(a)

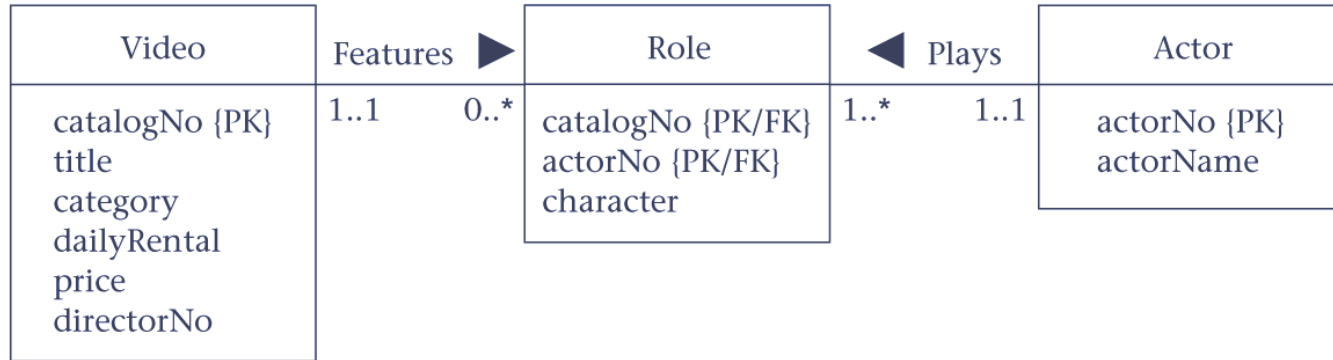| VideoForRent |
| --- |
| videoNo {PK} |
| branchNo {FK} |

◀ IsAllocated

1..*   1..1

Branch
branchNo {PK}

1..1

IsPartOf

▼

0..*

RentalAgreement
rentalNo {PK}
videoNo {FK}

(b)

VideoForRent
videoNo {PK}
branchNo {FK}

◀ IsAllocated

1..*   1..1

Branch
branchNo {PK}

1..1

IsPartOf

▼

0..*

1..1

ParticipatesIn

▼

Extra relationship
added

RentalAgreement
rentalNo {PK}
videoNo {FK}
branchNo {FK}

1..*

Foreign key added
as a result of new
relationship

Pattern 4 Duplicating columns in *:* relationships to reduce joins

(a)

Video		Features ▶		Role		◀ Plays		Actor

Video	Role	Actor
catalogNo {PK} title category dailyRental price directorNo	catalogNo {PK/FK} actorNo {PK/FK} character	actorNo {PK} actorName

Features ▶ 1..1   0..*   ◀ Plays 1..*   1..1

(a)

(b)

Video	Role	Actor
catalogNo {PK} title category dailyRental price directorNo	catalogNo {PK/FK} actorNo {PK/FK} character title	actorNo {PK} actorName

Features ▶ 1..1   0..*   ◀ Plays 1..*   1..1

title column from Video table duplicated in Role table

(b)

Pattern 5 Introducing repeating groups

Pattern 6 Creating extract tables

- Reports can access derived data and perform multi-table joins on same set of base tables. However, data report based on may be relatively static or may not have to be current.

- Can create a single, highly denormalized extract table based on tables required by reports, and allow users to access extract table directly instead of base tables.

Pattern 7 Partitioning tables

- Rather than combining tables, could decompose a table into a smaller number of <u>partitions</u>.

- <u>Horizontal partition</u>: distribute records across a number of (smaller) tables.

- <u>Vertical partition</u>: distribute columns across a number of (smaller) tables. PK duplicated to allow reconstruction.

- Partitions useful for applications that store and analyze large amounts of data.

Pattern 7 Partitioning tables

Horizontal

Vertical
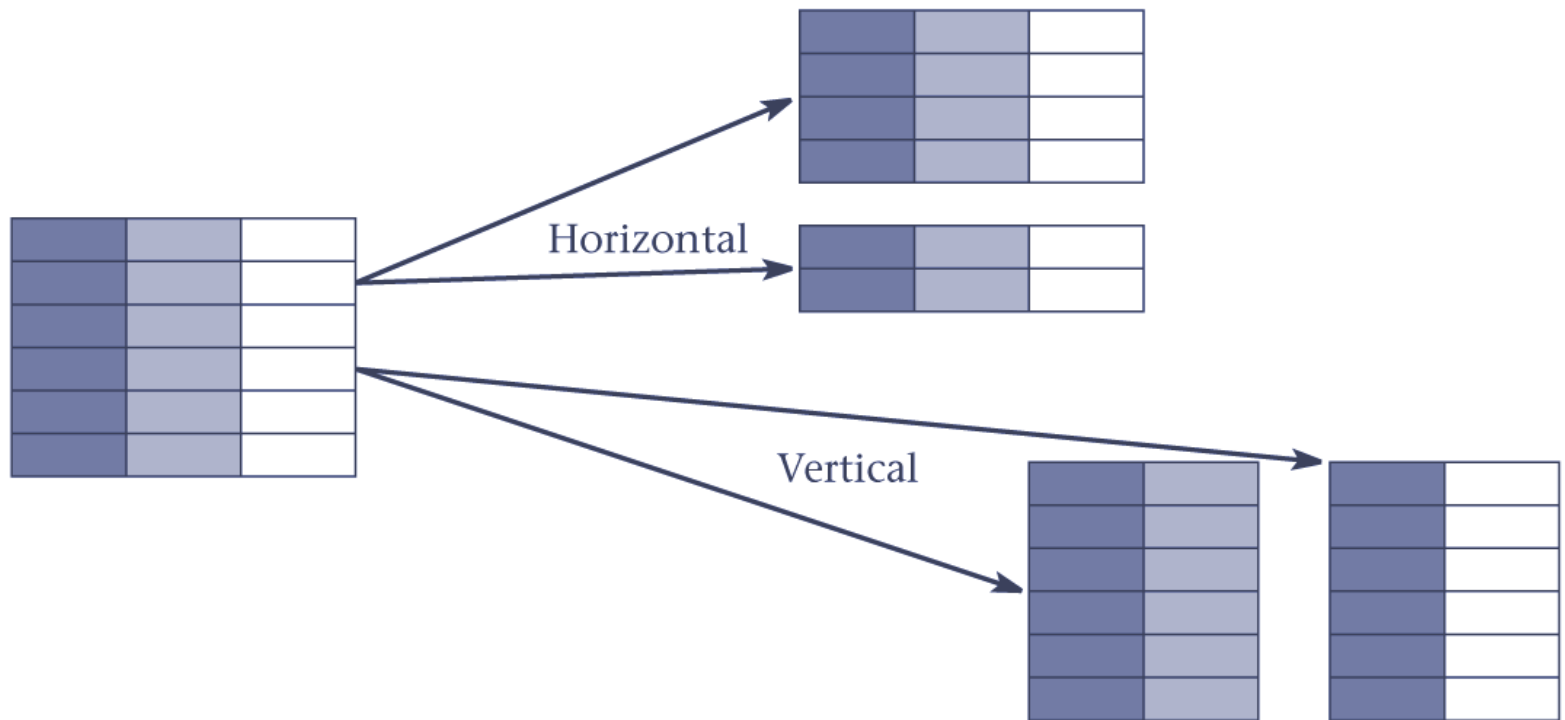
Pattern 7 Partitioning tables

- Advantages:
  - Improved load balancing
  - Improved performance
  - Increased availability
  - Improved recovery
  - Security.
- Disadvantages:
  - Complexity
  - Reduced performance
  - Duplication.