“DATABASE SYSTEMS”

Modern Database Management

Jeffrey A. Hoffer,
Mary B. Prescott,
Fred R. McFadden

Instructor: Rahmat Ullah Khan

Courtesy: Dr.Muhammad Asif Khan

Normalization

Chapter 06 (B)

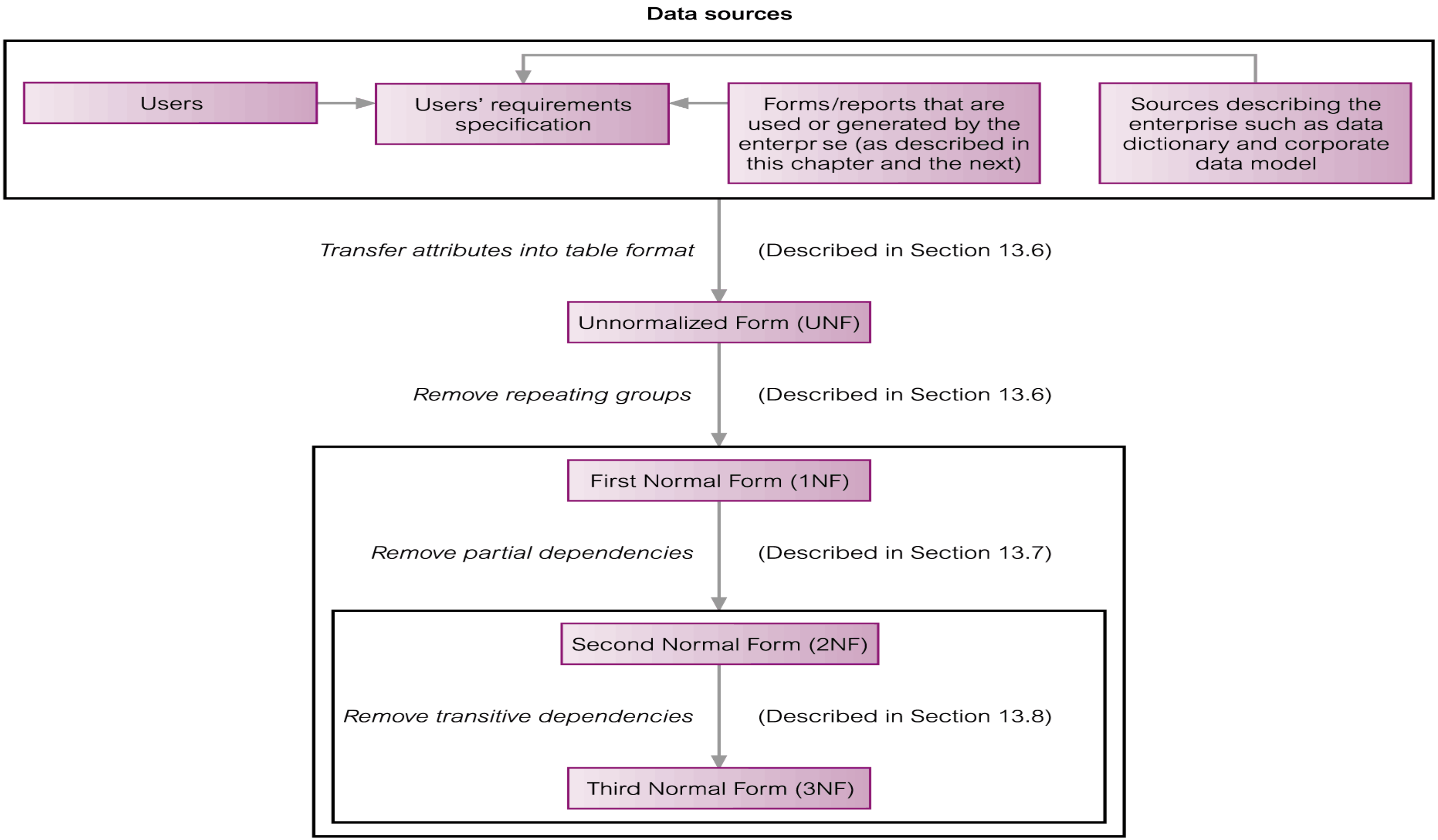
The Process of Normalization

- Normalization is a formal technique for analyzing relations based on their PK (or CKs) and functional dependencies
- The technique involves a series of rules that can be used to test individual relations so that a database can be normalized to any degree
- When a requirement is not met, the relation violating the requirement must be decomposed into relations that individually meet the requirements of normalization
- Types of normal forms are 1NF, 2NF, 3NF, BCNF, 4NF, 5NF
- All normal forms are based on FDs except 1NF
- Normal forms 4NF, 5NF deal with the situations that are very rare

The Process of Normalization

- Normalization is often executed as a series of steps, each step corresponds to a specific normal form that has known properties
- As normalization proceeds, the relations become progressively more restricted (stronger) in format and also less vulnerable to update anomalies
- For the relational data model, it is important to recognize that it is only First Normal Form (1NF) that is critical in creating relations; all subsequent normal forms are optional
- However, to avoid the update anomalies, it is generally recommended that we proceed to at least Third Normal Form (3NF)

The Process of Normalization



Data sources

Users

Users' requirements specification

Forms/reports that are used or generated by the enterpr se (as described in this chapter and the next)

Sources describing the enterprise such as data dictionary and corporate data model

Transfer attributes into table format (Described in Section 13.6)

Unnormalized Form (UNF)

Remove repeating groups (Described in Section 13.6)

First Normal Form (1NF)

Remove partial dependencies (Described in Section 13.7)

Second Normal Form (2NF)

Remove transitive dependencies (Described in Section 13.8)

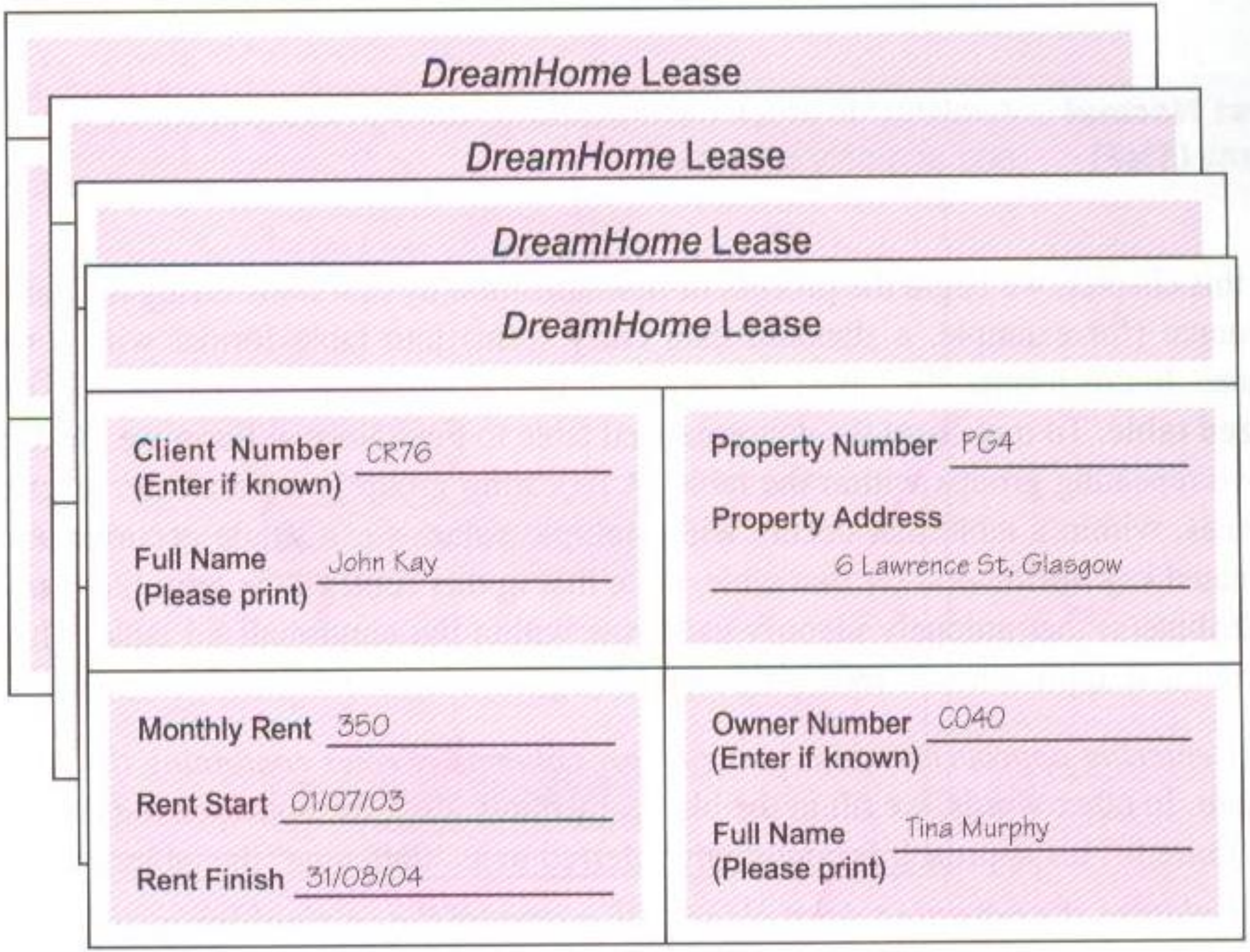
Third Normal Form (3NF)

Unnormalized Form (UNF)

- A table that contains one or more repeating groups
- To create an unnormalized table
  - Transform the data from the information source (e.g. form) into table format with columns and rows

Lease Forms: DreamHome

Figure 13.9 Collection of (simplified) *DreamHome* leases.



DreamHome Lease

DreamHome Lease

DreamHome Lease

DreamHome Lease

Client Number CR76
(Enter if known)

Full Name John Kay
(Please print)

Property Number PG4

Property Address
6 Lawrence St, Glasgow

Monthly Rent 350

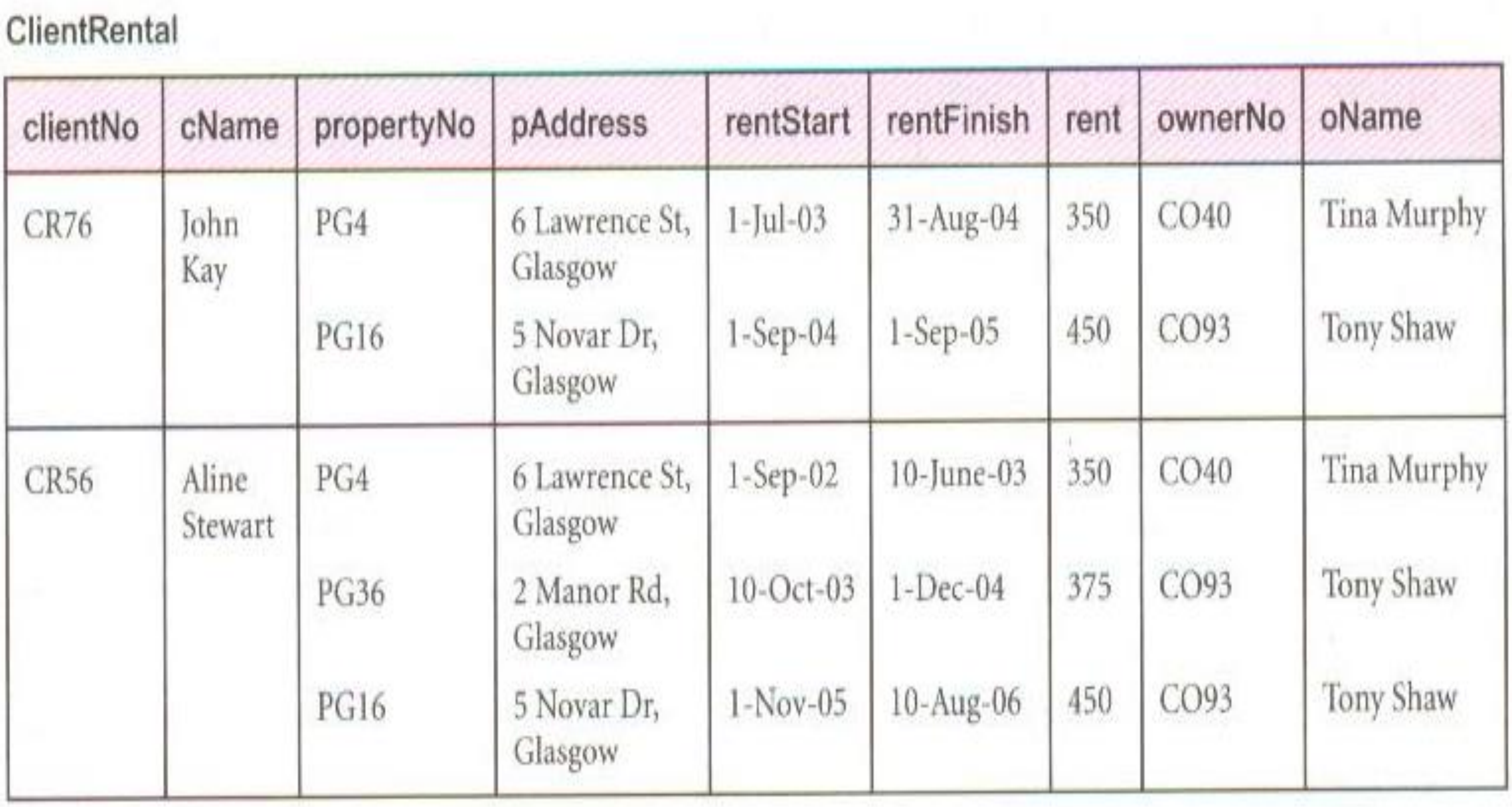
Rent Start 01/07/03

Rent Finish 31/08/04

Owner Number C040
(Enter if known)

Full Name Tina Murphy
(Please print)

Transformation into UNF Table



ClientRental

clientNo	cName	propertyNo	pAddress	rentStart	rentFinish	rent	ownerNo	oName
CR76	John Kay	PG4	6 Lawrence St, Glasgow	1-Jul-03	31-Aug-04	350	CO40	Tina Murphy
		PG16	5 Novar Dr, Glasgow	1-Sep-04	1-Sep-05	450	CO93	Tony Shaw
CR56	Aline Stewart	PG4	6 Lawrence St, Glasgow	1-Sep-02	10-June-03	350	CO40	Tina Murphy
		PG36	2 Manor Rd, Glasgow	10-Oct-03	1-Dec-04	375	CO93	Tony Shaw
		PG16	5 Novar Dr, Glasgow	1-Nov-05	10-Aug-06	450	CO93	Tony Shaw

First Normal Form (1NF)

- A relation in which the intersection of each row and column contains one and only one value

UNF to 1NF

- Nominate an attribute or group of attributes to act as the key for the unnormalized table
- Identify the repeating group(s) in the unnormalized table which repeats for the key attribute(s)

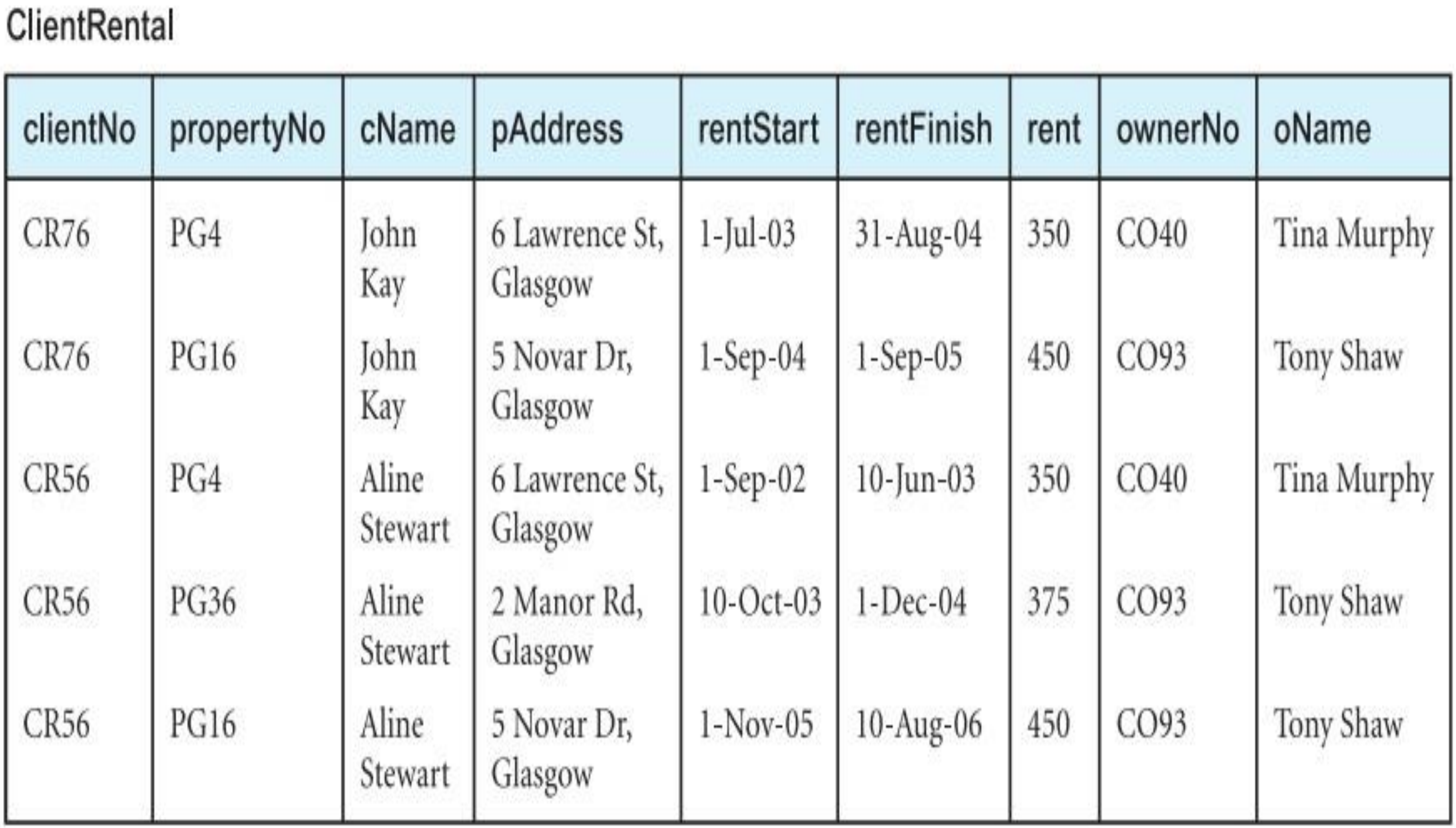
UNF to 1NF

- Remove the repeating groups by
  - Entering appropriate data into the empty columns of rows containing the repeating data (‘flattening’ the table)
  - By Placing the repeating data along with a copy of the original key attribute(s) into a separate relation

UNF to 1NF

- For both approaches, the resulting tables are now referred to as 1NF relations containing atomic (or single) values at the intersection of each row and column
- Although both approaches are correct, approach 1 introduces more redundancy into the original UNF table as part of the ‘flattening’ process, whereas approach 2 creates two or more relations with less redundancy than in the original UNF table

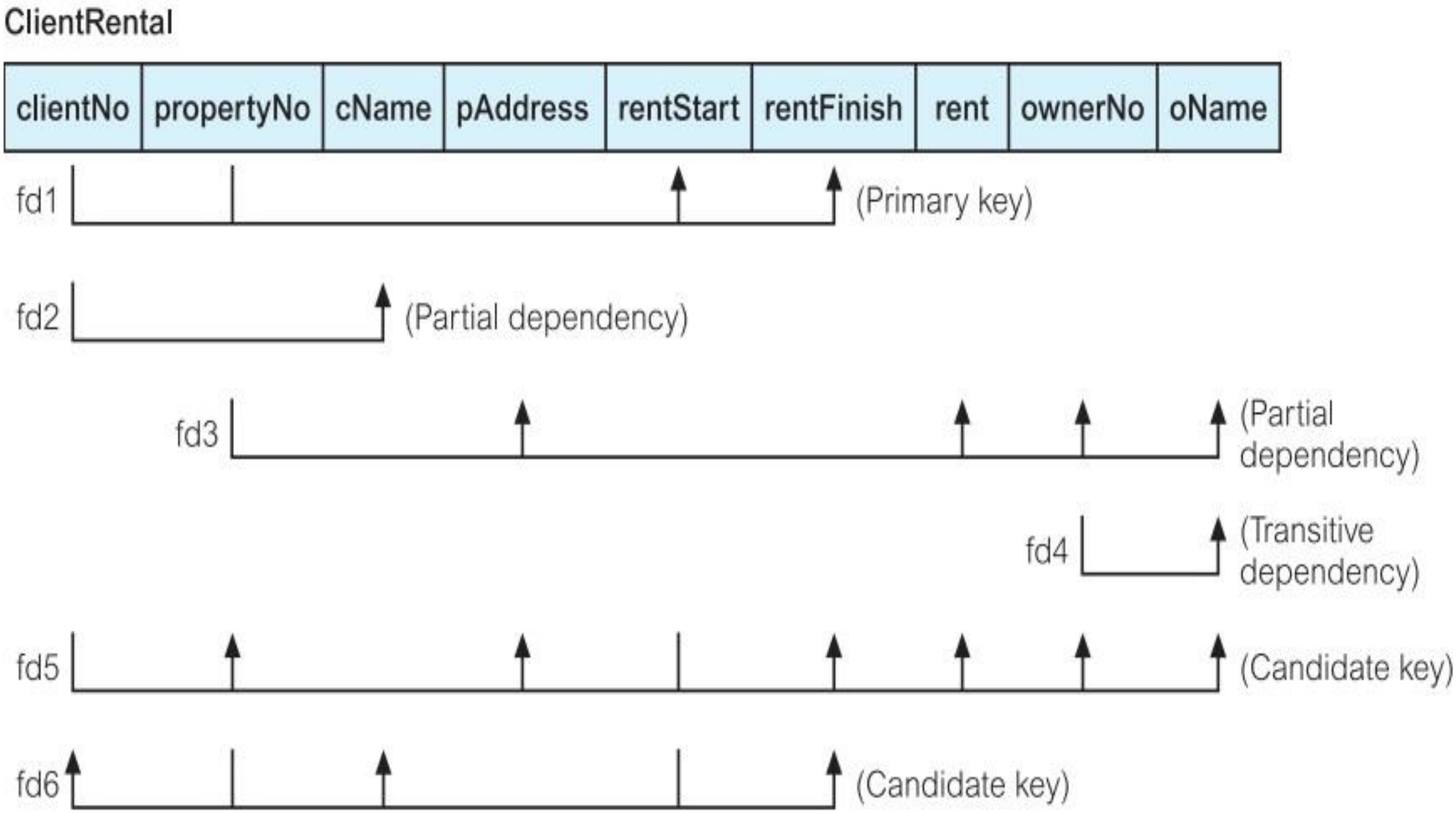
Approach 1 (1NF)



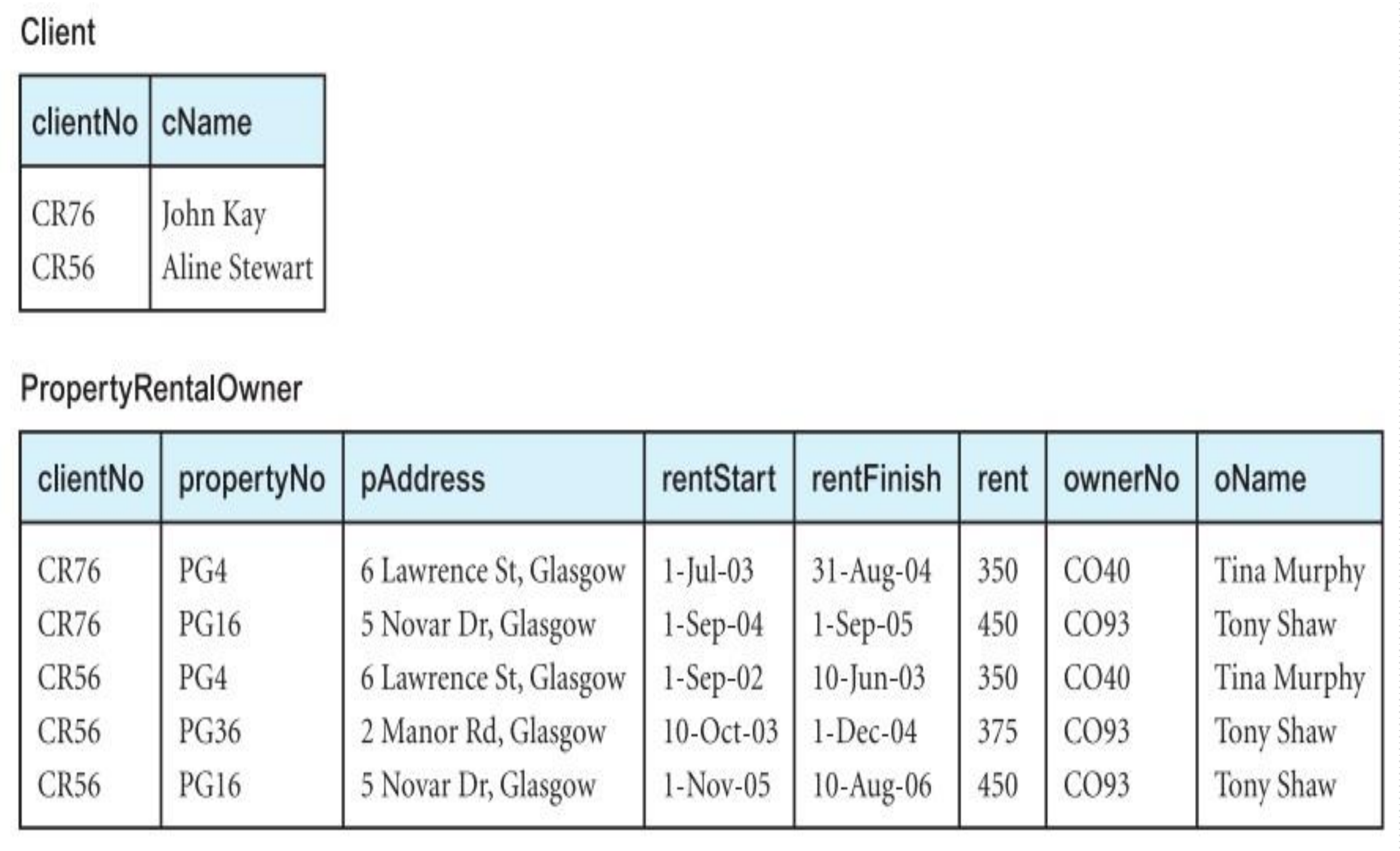
ClientRental

clientNo	propertyNo	cName	pAddress	rentStart	rentFinish	rent	ownerNo	oName
CR76	PG4	John Kay	6 Lawrence St, Glasgow	1-Jul-03	31-Aug-04	350	CO40	Tina Murphy
CR76	PG16	John Kay	5 Novar Dr, Glasgow	1-Sep-04	1-Sep-05	450	CO93	Tony Shaw
CR56	PG4	Aline Stewart	6 Lawrence St, Glasgow	1-Sep-02	10-Jun-03	350	CO40	Tina Murphy
CR56	PG36	Aline Stewart	2 Manor Rd, Glasgow	10-Oct-03	1-Dec-04	375	CO93	Tony Shaw
CR56	PG16	Aline Stewart	5 Novar Dr, Glasgow	1-Nov-05	10-Aug-06	450	CO93	Tony Shaw

FDs of ClientRental Relation



Approach 2 (1NF)



Client

clientNo	cName
CR76	John Kay
CR56	Aline Stewart

PropertyRentalOwner

clientNo	propertyNo	pAddress	rentStart	rentFinish	rent	ownerNo	oName
CR76	PG4	6 Lawrence St, Glasgow	1-Jul-03	31-Aug-04	350	CO40	Tina Murphy
CR76	PG16	5 Novar Dr, Glasgow	1-Sep-04	1-Sep-05	450	CO93	Tony Shaw
CR56	PG4	6 Lawrence St, Glasgow	1-Sep-02	10-Jun-03	350	CO40	Tina Murphy
CR56	PG36	2 Manor Rd, Glasgow	10-Oct-03	1-Dec-04	375	CO93	Tony Shaw
CR56	PG16	5 Novar Dr, Glasgow	1-Nov-05	10-Aug-06	450	CO93	Tony Shaw

Second Normal Form (2NF)

- 2NF is based on the concept of full FD
- 2NF applies to relations with composite keys, means a relation with a simple PK is automatically in at least 2NF
- A relation that is not in 2NF may suffer from the update anomalies e.g. change the rent of property number PG4 etc.

Second Normal Form (2NF)

- A relation that is in 1NF and every non-PK attribute is fully functionally dependent on the primary key
  - The normalization of 1NF relations to 2NF involves the removal of partial dependencies
  - If a partial dependency exists, we remove the partially dependent attribute(s) from the relation by placing them in a new relation along with a copy of their determinant

FDs of ClientRental Relation

fd1 clientNo, propertyNo→ rentStart, rentFinish (PK)

fd2 clientNo→ cName (Partial dependency)

fd3 propertyNo→ pAddress, rent, ownerNo, oName (Partial dependency)

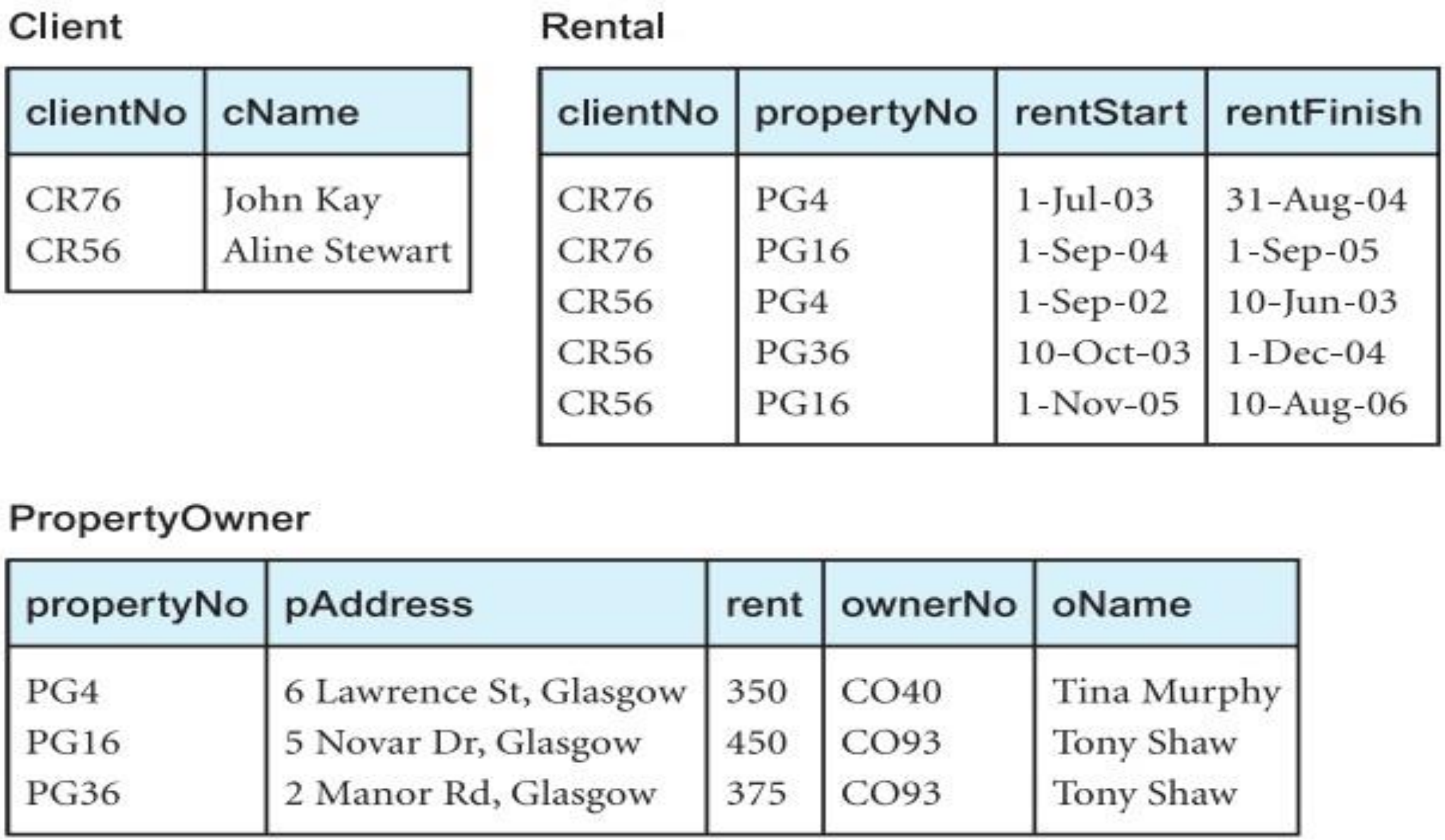
fd4 ownerNo→ oName (Transitive dependency)

fd5 clientNo, rentStart→ propertyNo, pAddress, rentFinish, rent, ownerNo, oName (Candidate key)

fd6 propertyNo, rentStart→ clientNo, cName, rentFinish **(Candidate key)**

1NF to 2NF

- **By removing partial functional dependencies**



Client

clientNo	cName
CR76	John Kay
CR56	Aline Stewart

Rental

clientNo	propertyNo	rentStart	rentFinish
CR76	PG4	1-Jul-03	31-Aug-04
CR76	PG16	1-Sep-04	1-Sep-05
CR56	PG4	1-Sep-02	10-Jun-03
CR56	PG36	10-Oct-03	1-Dec-04
CR56	PG16	1-Nov-05	10-Aug-06

PropertyOwner

propertyNo	pAddress	rent	ownerNo	oName
PG4	6 Lawrence St, Glasgow	350	CO40	Tina Murphy
PG16	5 Novar Dr, Glasgow	450	CO93	Tony Shaw
PG36	2 Manor Rd, Glasgow	375	CO93	Tony Shaw

Third Normal Form (3NF)

- Although 2NF relations have less redundancy than those in 1NF, they may still suffer from update anomalies e.g. if we want to update the name of an owner, such as Tony Shaw (ownerNo CO93) etc.

Third Normal Form (3NF)

- A relation that is in 1NF and 2NF and in which no non-primary-key attribute is transitively dependent on the primary key
    - The normalization of 2NF relations to 3NF involves the removal of transitive dependencies
    - If a transitive dependency exists, we remove the transitively dependent attribute(s) from the relation by placing the attribute(s) in a new relation along with a copy of the determinant

FDs of 2NF Relations

<u>Client</u>

fd2 clientNo → cName (Primary key)

<u>Rental</u>

fd1 clientNo, propertyNo → rentStart, rentFinish **(PK)**

fd5′ clientNo, rentStart → propertyNo, rentFinish **(CK)**

fd6′ propertyNo, rentStart → clientNo, rentFinish **(CK)**

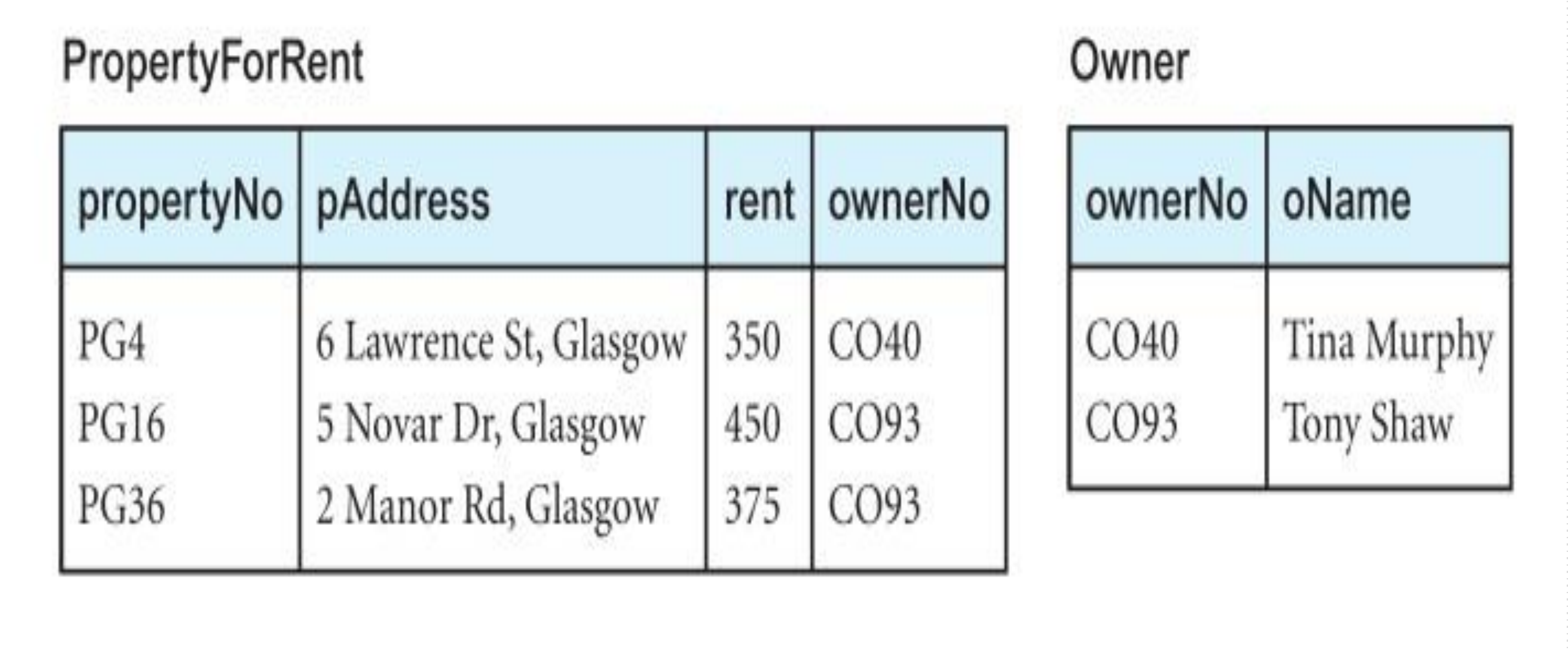
<u>PropertyOwner</u>

fd3 propertyNo → pAddress, rent, ownerNo, oName **(PK)**

fd4 ownerNo → oName **(Transitive dependency)**

2NF to 3NF

- **By removing transitive dependency**



PropertyForRent

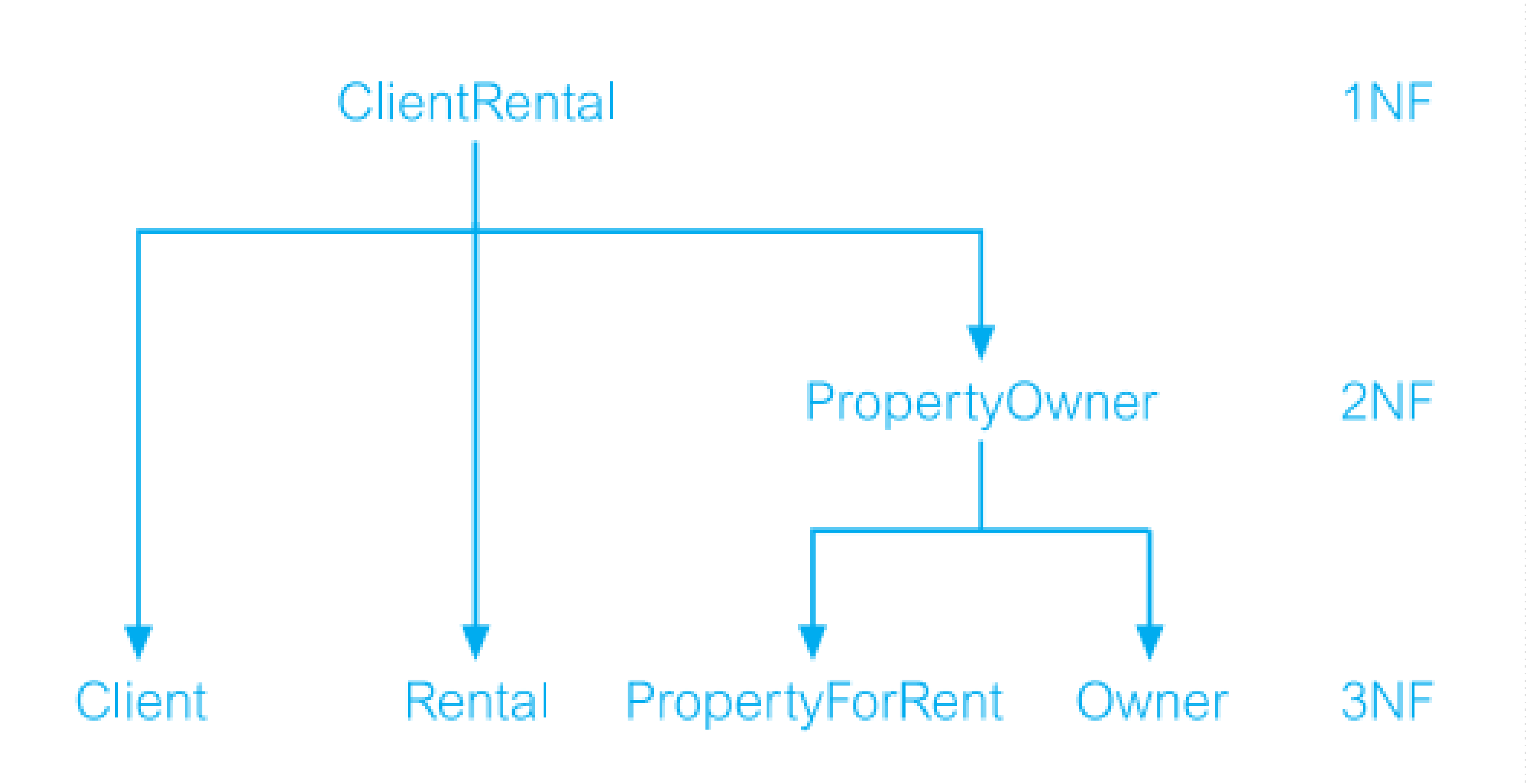
propertyNo	pAddress	rent	ownerNo
PG4	6 Lawrence St, Glasgow	350	CO40
PG16	5 Novar Dr, Glasgow	450	CO93
PG36	2 Manor Rd, Glasgow	375	CO93

Owner

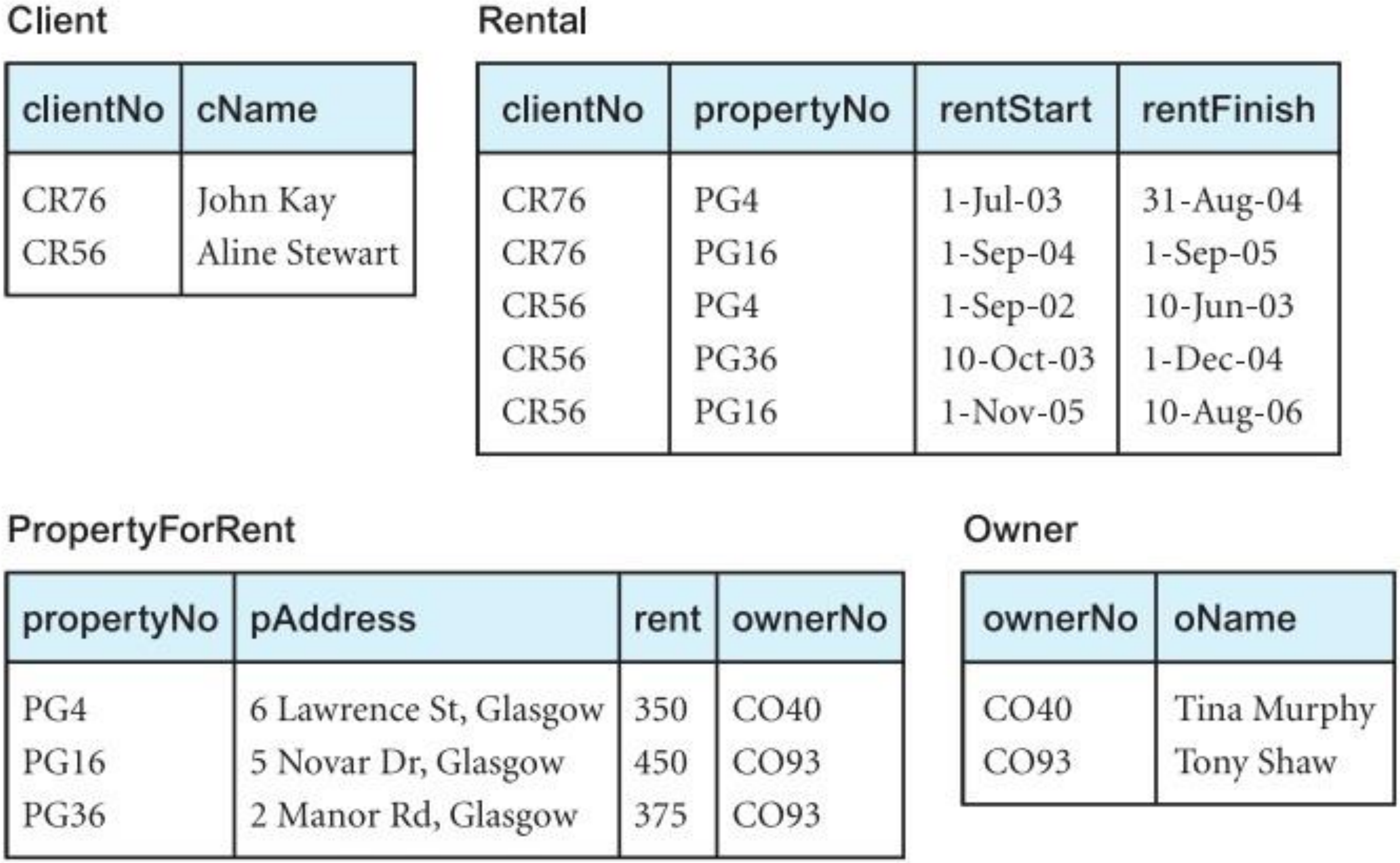
ownerNo	oName
CO40	Tina Murphy
CO93	Tony Shaw

Decomposition

- **The decomposition of ClientRental 1NF relation into 3NF relations**



3NF Relations



Client

clientNo	cName
CR76	John Kay
CR56	Aline Stewart

Rental

clientNo	propertyNo	rentStart	rentFinish
CR76	PG4	1-Jul-03	31-Aug-04
CR76	PG16	1-Sep-04	1-Sep-05
CR56	PG4	1-Sep-02	10-Jun-03
CR56	PG36	10-Oct-03	1-Dec-04
CR56	PG16	1-Nov-05	10-Aug-06

PropertyForRent

propertyNo	pAddress	rent	ownerNo
PG4	6 Lawrence St, Glasgow	350	CO40
PG16	5 Novar Dr, Glasgow	450	CO93
PG36	2 Manor Rd, Glasgow	375	CO93

Owner

ownerNo	oName
CO40	Tina Murphy
CO93	Tony Shaw

General Definitions of 2NF and 3NF

- Second normal form (2NF)
  - A relation that is in first normal form and every non-primary-key attribute is fully functionally dependent on any candidate key
- Third normal form (3NF)
  - A relation that is in first and second normal form and in which no non-primary-key attribute is transitively dependent on any candidate key

Using General Def. of 2NF and 3NF

- When using the general definitions of 2NF and 3NF
  - Look for partial and transitive dependencies on all candidate keys and not just the primary key
  - Makes the process of normalization more complex
  - However, the general definitions place additional constraints on the relations and may identify hidden redundancy in relations that could be missed

Using General Def. of 2NF and 3NF

- The tradeoff between
  - Keeping the process of normalization simpler
  - Increase the opportunity to identify missed redundancy
  - In fact, it is often the case that whether we use the definitions based on primary keys or the general definitions of 2NF and 3NF, the decomposition of relations is the same
  - For verification apply general definitions of 2NF and 3NF to ClientRental relation

Inference Rules for FDs

- The complete set of functional dependencies for a given relation can be very large
- Important to find an approach that can reduce the set to a manageable size
  - Need to identify a set of functional dependencies (represented as X) for a relation that is smaller than the complete set of functional dependencies (represented as Y) for that relation and has the property that every functional dependency in Y is implied by the functional dependencies in X

Inference Rules for FDs

- The set of all functional dependencies that are implied by a given set of functional dependencies X is called the closure of X, written X+
- A set of inference rules, called Armstrong's axioms, specifies how new functional dependencies can be inferred from given ones

Inference Rules for FDs

- Let A, B, and C be subsets of the attributes of the relation R. Armstrong's axioms are as follows:

1. Reflexivity

   If B is a subset of A, then A → B

2. Augmentation

   If A → B, then A,C → B,C

3. Transitivity

   If A → B and B → C, then A → C

Inference Rules for FDs

- Further rules can be derived from the first three rules that simplify the practical task of computing X+. Let D be another subset of the attributes of relation R, then:

4. Self-determination

   A → A

5. Decomposition

   If A → B,C, then A → B and A → C

6. Union

   If A → B and A → C, then A → B,C

7. Composition

   If A → B and C → D then A,C → B,D

Boyce–Codd Normal Form (BCNF)

- BCNF is based on functional dependencies that take into account all candidate keys in a relation, however BCNF also has additional constraints compared with the general definition of 3NF
- Boyce–Codd normal form (BCNF)
  - A relation is in BCNF if and only if every determinant is a candidate key

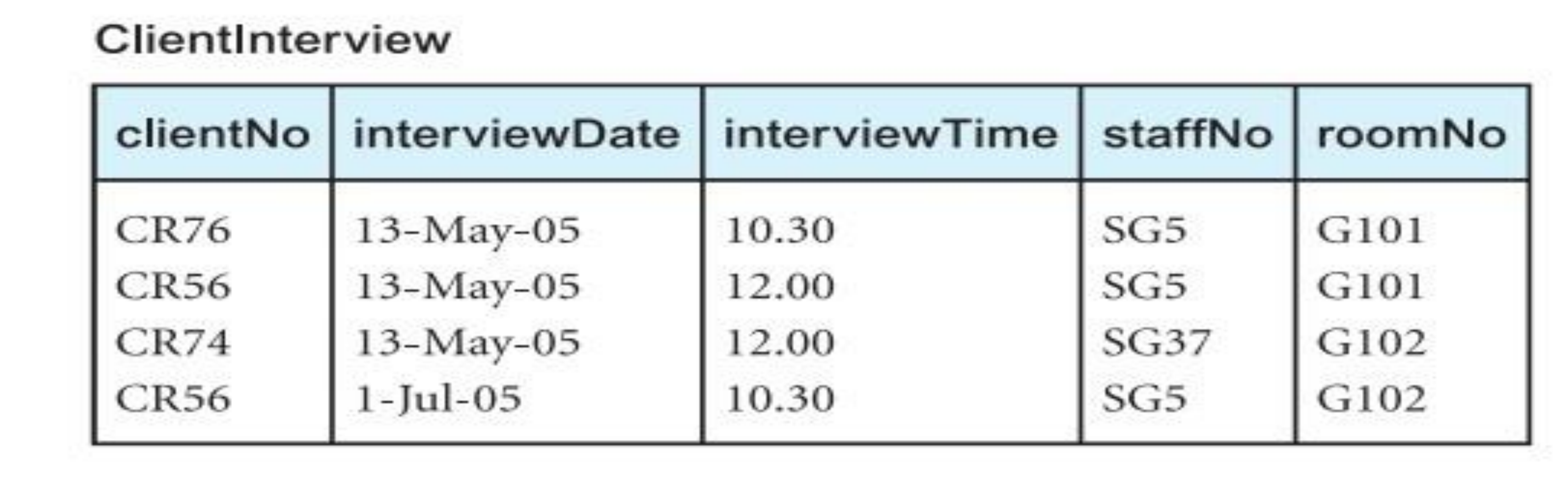
Boyce–Codd Normal Form (BCNF)

- Difference between 3NF and BCNF is that for a functional dependency A → B,
  - 3NF allows this dependency in a relation if B is a primary-key attribute and A is not a candidate key
  - Whereas, BCNF insists that for this dependency to remain in a relation, A must be a candidate key
- Every relation in BCNF is also in 3NF
  - However, a relation in 3NF is not necessarily in BCNF

Boyce–Codd Normal Form (BCNF)

- Violation of BCNF is quite rare
- The potential to violate BCNF may occur in a relation that:
  - Contains two (or more) composite candidate keys
  - The candidate keys overlap, that is have at least one attribute in common

Example: (BCNF)



ClientInterview

clientNo	interviewDate	interviewTime	staffNo	roomNo
CR76	13-May-05	10.30	SG5	G101
CR56	13-May-05	12.00	SG5	G101
CR74	13-May-05	12.00	SG37	G102
CR56	1-Jul-05	10.30	SG5	G102

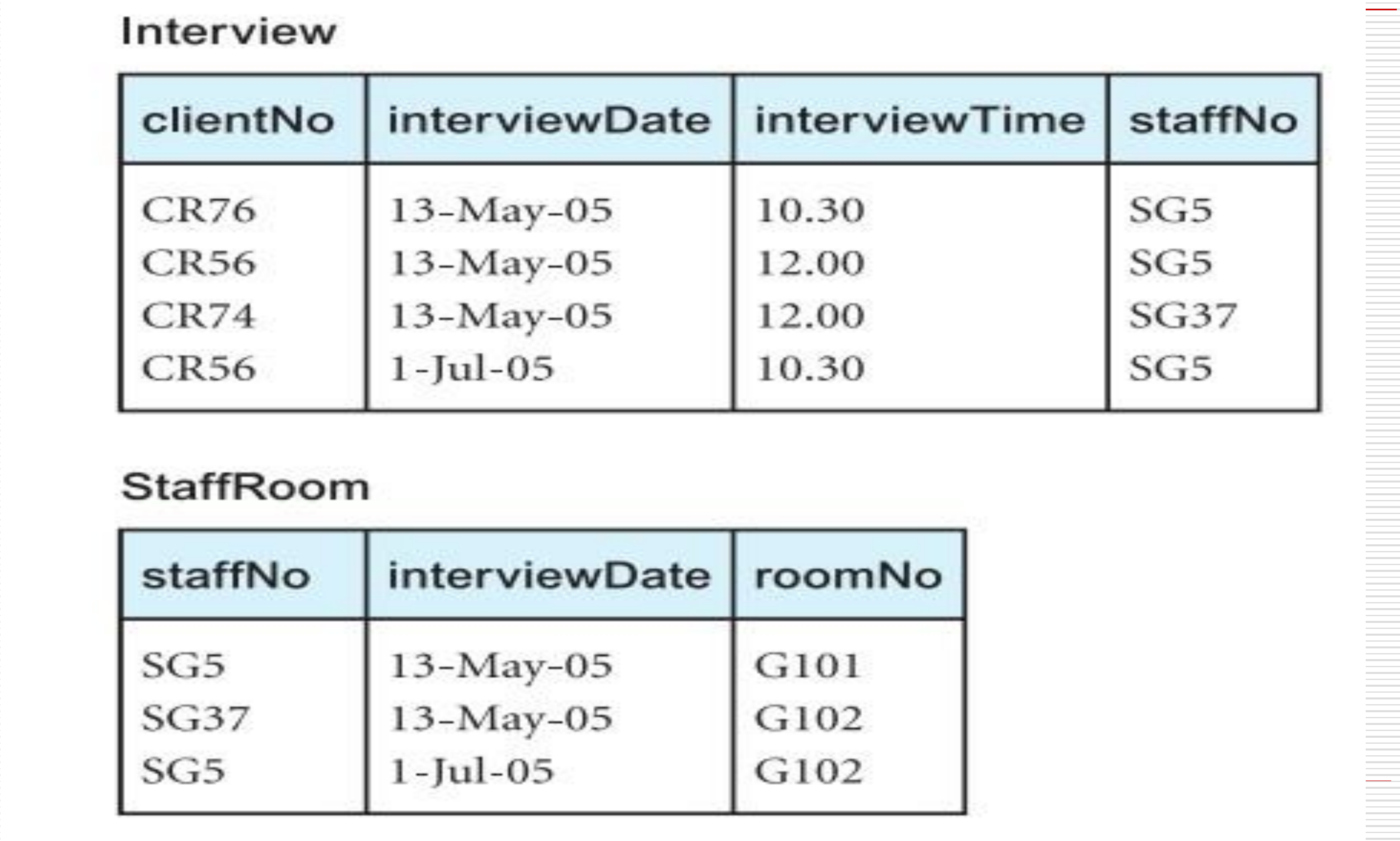
fd1 clientNo, interviewDate → interviewTime, staffNo, roomNo **(PK)**

fd2 staffNo, interviewDate, interviewTime → clientNo **(CK)**

fd3 roomNo, interviewDate, interviewTime → staffNo, clientNo **(CK)**

fd4 staffNo, interviewDate → roomNo

Example: (BCNF)



Interview

clientNo	interviewDate	interviewTime	staffNo
CR76	13-May-05	10.30	SG5
CR56	13-May-05	12.00	SG5
CR74	13-May-05	12.00	SG37
CR56	1-Jul-05	10.30	SG5

StaffRoom

staffNo	interviewDate	roomNo
SG5	13-May-05	G101
SG37	13-May-05	G102
SG5	1-Jul-05	G102

3NF or BCNF

- May not always be desirable to transform a relation into BCNF e.g. if there is a FD that is not preserved when we perform the decomposition (i.e. the determinant and the attributes it determines are placed in different relations)
  - Loss of fd3 when ClientInterview relation was transformed to BCNF
  - Decision about to stop at 3NF or progress to BCNF is dependent on the amount of redundancy resulting from the presence of fd4 and the significance of the 'loss' of fd3

Summary

- The normalization process
  - 1NF, 2NF, 3NF
  - Inference rules for FDs
  - BCNF

References

- All the material (slides, diagrams etc.) presented in this lecture is taken (with modifications) from the Pearson Education website :
  - http://www.booksites.net/connbegg

THANK YOU