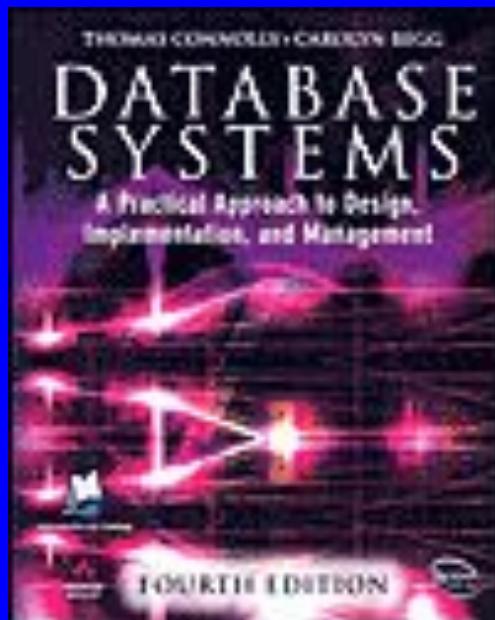


Lecture Thirteen

Normalization

Based on Chapter Thirteen of this book:



**Database Systems: A Practical Approach
to Design, Implementation and
Management**

International Computer Science S.

Carolyn Begg, Thomas Connolly

Lecture 13 - Objectives

- Purpose of normalization.
- Problems associated with redundant data.
- Identification of various types of update anomalies such as insertion, deletion, and modification anomalies.
- How to recognize appropriateness or quality of the design of relations.

Lecture 13 - Objectives

- How functional dependencies can be used to group attributes into relations that are in a known normal form.
- How to undertake process of normalization.
- How to identify most commonly used normal forms, namely 1NF, 2NF, 3NF, and Boyce–Codd normal form (BCNF).

Normalization

- **Main objective in developing a logical data model for relational database systems is to create an accurate representation of the data, its relationships, and constraints.**
- **To achieve this objective, must identify a suitable set of relations.**

Normalization

- Four most commonly used normal forms are first (1NF), second (2NF) and third (3NF) normal forms, and Boyce–Codd normal form (BCNF).
- Based on functional dependencies among the attributes of a relation.
- A relation can be normalized to a specific form to prevent possible occurrence of update anomalies.

Data Redundancy

- Major aim of relational database design is to group attributes into relations to minimize data redundancy and reduce file storage space required by base relations.
- Problems associated with data redundancy are illustrated by comparing the following Staff and Branch relations with the StaffBranch relation.

Data Redundancy

Staff

staffNo	sName	position	salary	branchNo
SL21	John White	Manager	30000	B005
SG37	Ann Beech	Assistant	12000	B003
SG14	David Ford	Supervisor	18000	B003
SA9	Mary Howe	Assistant	9000	B007
SG5	Susan Brand	Manager	24000	B003
SL41	Julie Lee	Assistant	9000	B005

Branch

branchNo	bAddress
B005	22 Deer Rd, London
B007	16 Argyll St, Aberdeen
B003	163 Main St, Glasgow

Staff Branch

staffNo	sName	position	salary	branchNo	bAddress
SL21	John White	Manager	30000	B005	22 Deer Rd, London
SG37	Ann Beech	Assistant	12000	B003	163 Main St, Glasgow
SG14	David Ford	Supervisor	18000	B003	163 Main St, Glasgow
SA9	Mary Howe	Assistant	9000	B007	16 Argyll St, Aberdeen
SG5	Susan Brand	Manager	24000	B003	163 Main St, Glasgow
SL41	Julie Lee	Assistant	9000	B005	22 Deer Rd, London

Data Redundancy

- StaffBranch relation has redundant data: details of a branch are repeated for every member of staff.
- In contrast, branch information appears only once for each branch in Branch relation and only branchNo is repeated in Staff relation, to represent where each member of staff works.

Update Anomalies

- Relations that contain redundant information may potentially suffer from update anomalies.
- Types of update anomalies include:
 - Insertion
 - Deletion
 - Modification.

Data Redundancy

Staff Branch

staffNo	sName	position	salary	branchNo	bAddress
SL21	John White	Manager	30000	B005	22 Deer Rd, London
SG37	Ann Beech	Assistant	12000	B003	163 Main St, Glasgow
SG14	David Ford	Supervisor	18000	B003	163 Main St, Glasgow
SA9	Mary Howe	Assistant	9000	B007	16 Argyll St, Aberdeen
SG5	Susan Brand	Manager	24000	B003	163 Main St, Glasgow
SL41	Julie Lee	Assistant	9000	B005	22 Deer Rd, London

Insertion Anomalies

- **New member of staff joins branch B005**
 - Insert new row into StaffBranch table
 - Type wrong address: 163 Main St, Glasgow.
 - Database is now inconsistent!
- **Establish new branch with no members of staff**
 - B008, 57 Princes St, Edinburgh
 - No staff members, so staffNo must be NULL
 - But staffNo is the primary key of the StaffBranch table, so cannot be NULL!

Deletion Anomaly

- Mary Howe, staffNo SA9, leaves the company
 - Delete the appropriate row of StaffBranch
 - This also deletes details of branch B007 where Mary Howe works
 - But no-one else works at branch B007, so we no longer know the address of this branch!

Modification Anomaly

- Branch B003 has transferred to a new location
 - New address is 145 Main St, Glasgow
 - Must change *three* rows of the StaffBranch relation

Lossless-join and Dependency Preservation Properties

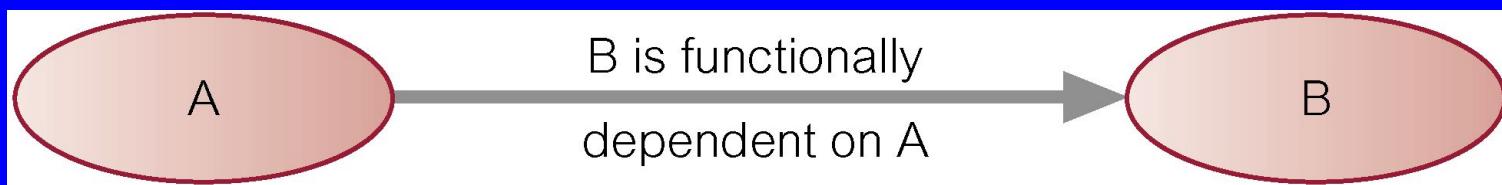
- Two important properties of decomposition:
 - *Lossless-join property* enables us to find any instance of original relation from corresponding instances in the smaller relations.
 - *Dependency preservation property* enables us to enforce a constraint on original relation by enforcing some constraint on each of the smaller relations.

Functional Dependency

- Main concept associated with normalization.
- Functional Dependency
 - Describes relationship between attributes in a relation.
 - If A and B are attributes of relation R, B is functionally dependent on A (denoted $A \square B$), if each value of A in R is associated with exactly one value of B in R.

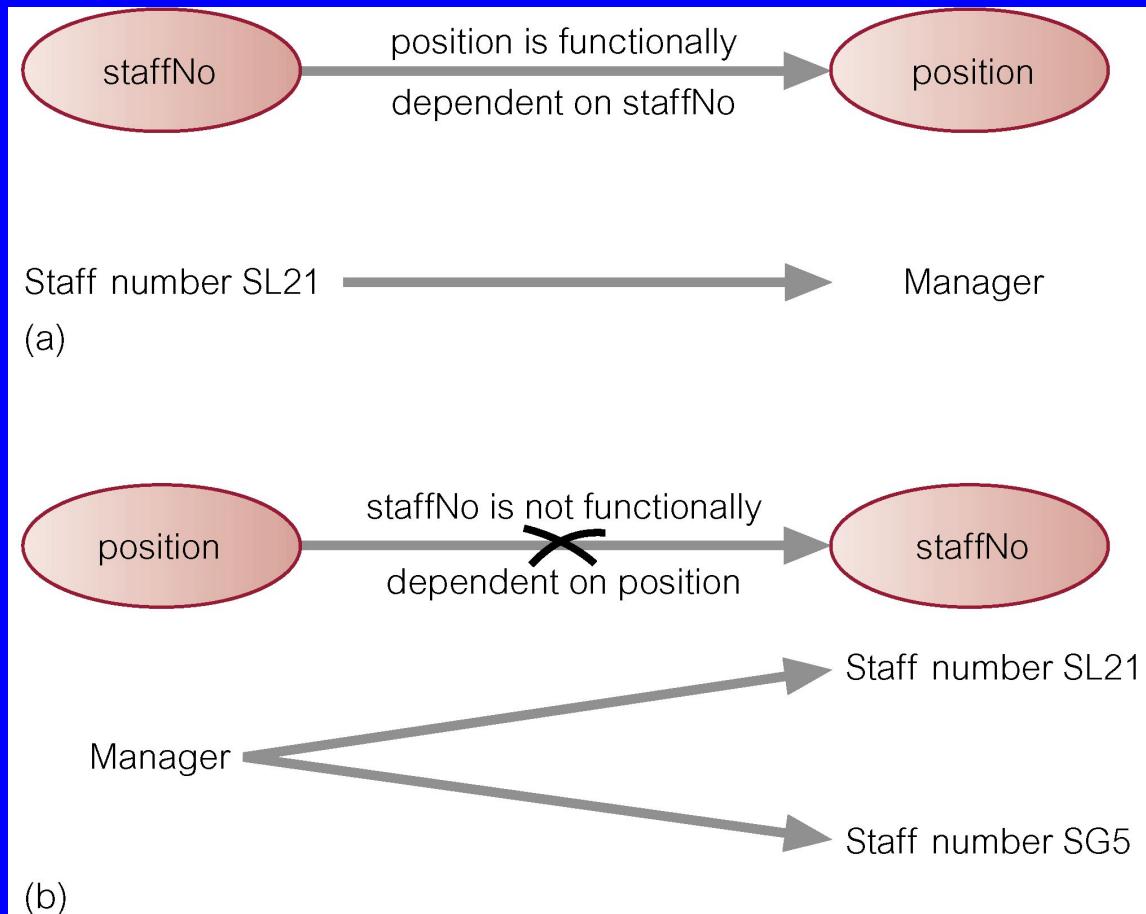
Functional Dependency

- Property of the meaning (or semantics) of the attributes in a relation.
- Diagrammatic representation:



- *Determinant* of a functional dependency refers to attribute or group of attributes on left-hand side of the arrow.

Example - Functional Dependency



Identifying Candidate Keys

- A candidate key is an attribute, or set of attributes, that uniquely identifies a row
 - Must be *irreducible*
 - *No part of a candidate can ever be NULL*
- An attribute A that functionally determines *every other* attribute of the relation is a candidate key
 - For each value of A there is exactly one value of each of the other attributes
 - So each value of A must identify a single row

Identifying Primary Keys

- A primary key is a candidate key chosen to identify rows uniquely within a table
 - Other candidate keys called *alternate keys*
- Some guidelines on choosing the primary key
 - Pick the candidate key with fewest attributes
 - Pick the candidate key with shortest length
 - Pick the candidate key that makes most sense for the business!

Functional Dependency

- **Main characteristics of functional dependencies used in normalization:**
 - have a 1:1 relationship between attribute(s) on left and right-hand side of a dependency;
 - hold for all time;
 - are nontrivial.

Functional Dependency

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

Functional Dependency

- Set of all functional dependencies implied by a given set of functional dependencies X called closure of X (written X^+).
- Set of inference rules, called Armstrong's axioms, specifies how new functional dependencies can be inferred from given ones.

Functional Dependency

- Let A, B, and C be subsets of the attributes of relation R. Armstrong's axioms are as follows:
 1. Reflexivity
If B is a subset of A, then $A \rightarrow B$
 2. Augmentation
If $A \rightarrow B$, then $A,C \rightarrow B,C$
 3. Transitivity
If $A \rightarrow B$ and $B \rightarrow C$, then $A \rightarrow C$

The Process of Normalization

- **Formal technique for analyzing a relation based on its primary key and functional dependencies between its attributes.**
- Often executed as a series of steps. Each step corresponds to a specific normal form, which has known properties.
- As normalization proceeds, relations become progressively more restricted (stronger) in format and also less vulnerable to update anomalies.

Unnormalized Form (UNF)

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

First Normal Form (1NF)

- A relation in which 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 repeating group(s) in unnormalized table which repeats for the key attribute(s).

UNF to 1NF

- Remove repeating group by:
 - entering appropriate data into the empty columns of rows containing repeating data ('flattening' the table).

Or by

- placing repeating data along with copy of the original key attribute(s) into a separate relation.

ClientRental UNF To 1NF By Flattening

clientNo	cName	propertyNo	pAddress	rentStart	rentFinish	rent	ownerNo	oName
CR76	John Kay	PG4	6 Lawrence St, Glasgow	1-Jul-00	31-Aug-01	350	CO40	Tina Murphy
		PG16	5 Novar Dr, Glasgow	1-Sep-01	1-Sep-02	450	CO93	Tony shaw
CR56	Aline Stewart	PG4	6 Lawrence St, Glasgow	1-Sep-99	10-June-00	350	CO40	Tina Murphy
		PG36	2 Manor Rd, Glasgow	10-Oct-00	1-Dec-01	375	CO93	Tony shaw
		PG16	5 Novar Dr, Glasgow	1-Nov-02	10-Aug-03	450	CO93	Tony shaw

clientNo	cName	propertyNo	pAddress	rentStart	rentFinish	rent	ownerNo	oName
CR76	John Kay	PG4	6 Lawrence St, Glasgow	1-Jul-00	31-Aug-01	350	CO40	Tina Murphy
CR76	John Kay	PG16	5 Novar Dr, Glasgow	1-Sep-01	1-Sep-02	450	CO93	Tony shaw
CR56	Aline Stewart	PG4	6 Lawrence St, Glasgow	1-Sep-99	10-June-00	350	CO40	Tina Murphy
CR56	Aline Stewart	PG36	2 Manor Rd, Glasgow	10-Oct-00	1-Dec-01	375	CO93	Tony shaw
CR56	Aline Stewart	PG16	5 Novar Dr, Glasgow	1-Nov-02	10-Aug-03	450	CO93	Tony shaw

ClientRental Functional Dependencies

- On the ClientRental relation:
 - $\text{clientNo} \rightarrow \text{cName}$
 - $\text{propertyNo} \rightarrow \text{pAddress}, \text{rent}, \text{ownerNo}, \text{oName}$
 - $\text{ownerNo} \rightarrow \text{oName}$
 - $\text{clientNo}, \text{propertyNo} \rightarrow \text{cName}, \text{pAddress}, \text{rentStart}, \text{rentFinish}, \text{rent}, \text{ownerNo}, \text{oName}$
 - $\text{clientNo}, \text{rentStart} \rightarrow \text{cName}, \text{propertyNo}, \text{pAddress}, \text{rentFinish}, \text{rent}, \text{ownerNo}, \text{oName}$
 - $\text{propertyNo}, \text{rentStart} \rightarrow \text{clientNo}, \text{cName}, \text{pAddress}, \text{rentFinish}, \text{rent}, \text{ownerNo}, \text{oName}$

ClientRental Primary Key

- Candidate keys are
 - (clientNo, propertyNo)
 - (clientNo, rentStart)
 - (propertyNo, rentStart)
- Choose (clientNo, propertyNo) as Primary key

clientNo	propertyNo	cName	pAddress	rentStart	rentFinish	rent	ownerNo	oName
CR76	PG4	John Kay	6 Lawrence St, Glasgow	1-Jul-00	31-Aug-01	350	CO40	Tina Murphy
CR76	PG16	John Kay	5 Novar Dr, Glasgow	1-Sep-01	1-Sep-02	450	CO93	Tony shaw
CR56	PG4	Aline Stewart	6 Lawrence St, Glasgow	1-Sep-99	10-June-00	350	CO40	Tina Murphy
CR56	PG36	Aline Stewart	2 Manor Rd, Glasgow	10-Oct-00	1-Dec-01	375	CO93	Tony shaw
CR56	PG16	Aline Stewart	5 Novar Dr, Glasgow	1-Nov-02	10-Aug-03	450	CO93	Tony shaw

Client rental 1NF relation

- Write down the ClientRental relation using standard notation

ClientRental (clientNo, propertyNo, cName, pAddress,
rentStart, rentFinish, rent, ownerNo, oName)

Second Normal Form (2NF)

- Based on concept of full functional dependency:
 - A_1, \dots, A_n and B are attributes of a relation,
 - B is fully dependent on A_1, \dots, A_n if B is functionally dependent on A_1, \dots, A_n but not on any proper subset of A_1, \dots, A_n .
- 2NF - A relation that is in 1NF and every non-primary-key attribute is fully functionally dependent on the primary key.

1NF to 2NF

- Identify primary key for the 1NF relation.
- Identify functional dependencies in the relation.
- If partial dependencies exist on the primary key remove them by placing them in a new relation along with copy of their determinant.

ClientRental Example: 1NF to 2NF

clientNo	propertyNo	cName	pAddress	rentStart	rentFinish	rent	ownerNo	oName
CR76	PG4	John Kay	6 Lawrence St, Glasgow	1-Jul-00	31-Aug-01	350	CO40	Tina Murphy
CR76	PG16	John Kay	5 Novar Dr, Glasgow	1-Sep-01	1-Sep-02	450	CO93	Tony shaw
CR56	PG4	Aline Stewart	6 Lawrence St, Glasgow	1-Sep-99	10-June-00	350	CO40	Tina Murphy
CR56	PG36	Aline Stewart	2 Manor Rd, Glasgow	10-Oct-00	1-Dec-01	375	CO93	Tony shaw
CR56	PG16	Aline Stewart	5 Novar Dr, Glasgow	1-Nov-02	10-Aug-03	450	CO93	Tony shaw

- Partial dependencies are:
 - $\text{clientNo} \rightarrow \text{cName}$
 - $\text{propertyNo} \rightarrow \text{pAddress}, \text{rent}, \text{ownerNo}, \text{oName}$

ClientRental Example: clientNo → cName

- Create new relation Client, with primary key clientNo

clientNo	cName
CR76	John Kay
CR56	Aline Stewart

- Remove cName from the ClientRental relation

clientNo	propertyNo	pAddress	rentStart	rentFinish	rent	ownerNo	oName
CR76	PG4	6 Lawrence St, Glasgow	1-Jul-00	31-Aug-01	350	CO40	Tina Murphy
CR76	PG16	5 Novar Dr, Glasgow	1-Sep-01	1-Sep-02	450	CO93	Tony shaw
CR56	PG4	6 Lawrence St, Glasgow	1-Sep-99	10-June-00	350	CO40	Tina Murphy
CR56	PG36	2 Manor Rd, Glasgow	10-Oct-00	1-Dec-01	375	CO93	Tony shaw
CR56	PG16	5 Novar Dr, Glasgow	1-Nov-02	10-Aug-03	450	CO93	Tony shaw

ClientRental Example:

propertyNo -> pAddress, rent, ownerNo, oName

- Create new relation PropertyOwner, with primary key propertyNo

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

- Remove attributes pAddress, rent, ownerNo, oName from the ClientRental relation

clientNo	propertyNo			rentStart	rentFinish			
CR76	PG4			1-Jul-00	31-Aug-01			
CR76	PG16			1-Sep-01	1-Sep-02			
CR56	PG4			1-Sep-99	10-June-00			
CR56	PG36			10-Oct-00	1-Dec-01			
CR56	PG16			1-Nov-02	10-Aug-03			

ClientRental Example: 2NF relations

- Tidy up, and re-name the ClientRental relation “Rental”

clientNo	propertyNo	rentStart	rentFinish
CR76	PG4	1-Jul-00	31-Aug-01
CR76	PG16	1-Sep-01	1-Sep-02
CR56	PG4	1-Sep-99	10-June-00
CR56	PG36	10-Oct-00	1-Dec-01
CR56	PG16	1-Nov-02	10-Aug-03

- Write down the 2NF relations:

Client (clientNo, cName)

PropertyOwner (propertyNo, pAddress, rent, ownerNo, oName)

Rental (clientNo, propertyNo, rentStart, rentFinish)

Third Normal Form (3NF)

- Based on concept of transitive dependency:
 - A, B and C are attributes of a relation such that $A \sqsubset B$ and $B \sqsubset C$,
 - then C is transitively dependent on A through B. (Provided that A is not functionally dependent on B or C).
- 3NF - A relation that is in 1NF and 2NF and in which no non-primary-key attribute is transitively dependent on the primary key.

2NF to 3NF

- Identify the primary key in the 2NF relation.
- Identify functional dependencies in the relation.
- If transitive dependencies exist on the primary key remove them by placing them in a new relation along with copy of their determinant.

ClientRental Example: 2NF to 3NF

- Consider the relation:

PropertyOwner (propertyNo, pAddress, rent, ownerNo, oName)

- We have functional dependencies
 - $\text{propertyNo} \rightarrow \text{ownerNo}$
 - $\text{ownerNo} \rightarrow \text{oName}$
- So oName is transitively dependent on propertyNo, the primary key

ClientRental Example: Remove Transitive Dependency On Primary Key

- Create new relation Owner, with primary key ownerNo and attribute oName

ownerNo	oName
CO40	Tina Murphy
CO93	Tony shaw

- Remove oName from PropertyOwner relation

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

ClientRental Example: 3NF Relations

- Tidy up, and re-name PropertyOwner relation “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

- Write down the 3NF relations:
Client (clientNo, cName)
Rental (clientNo, propertyNo, rentStart, rentFinish)
PropertyOwner (propertyNo, pAddress, rent, ownerNo)
Owner (ownerNo, oName)

General Definitions of 2NF and 3NF

- Second normal form (2NF)
 - A relation that is in 1NF and every non-primary-key attribute is fully functionally dependent on *any candidate key*.
- Third normal form (3NF)
 - A relation that is in 1NF and 2NF and in which no non-primary-key attribute is transitively dependent on *any candidate key*.

Boyce–Codd Normal Form (BCNF)

- Based on functional dependencies that take into account all candidate keys in a relation, however BCNF also has additional constraints compared with general definition of 3NF.
- 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 $A \rightarrow B$, 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, relation in 3NF may not be in BCNF.

Boyce–Codd normal form (BCNF)

- Violation of BCNF is quite rare.
- Potential to violate BCNF may occur in a relation that:
 - contains two (or more) composite candidate keys;
 - the candidate keys overlap (ie. have at least one attribute in common).

BCNF Example

- Consider the relation:

ClientInterview (clientNo, interviewdate, interviewTime, staffNo, roomNo)

clientNo	interviewDate	interviewTime	staffNo	roomNo
CR76	13-May-02	10:30	SG5	G101
CR56	13-May-02	12:00	SG5	G101
CR74	13-May-02	12:00	SG37	G102
CR56	1-Jul-02	10:30	SG5	G102

BCNF Example: Functional dependencies

- Functional dependencies are

fd1: **clientNo, interviewdate → interviewTime, staffNo, roomNo**

fd2: **staffNo, interviewdate, interviewTime → clientNo**

fd3: **roomNo, interviewdate, interviewTime → staffNo, clientNo**

fd4: **staffNo, interviewdate → roomNo**

BCNF Example

- ClientInterview relation is in 3NF
- Is ClientInterview relation in BCNF?
 - fd1: the determinant, (clientNo, interviewdate), is the primary key
 - fd2 and fd3: both determinants are candidate keys
 - fd4: its determinant is NOT a candidate key

BCNF Example

- Create new relation StaffRoom with the attributes from fd4:
 - Determinant attributes form the primary key of the new relation
 - Include dependent attributes in the relation

staffNo	interviewDate	roomNo
SG5	13-May-02	G101
SG5	13-May-02	G101
SG37	13-May-02	G102
SG5	1-Jul-02	G102

BCNF Example

- Remove the dependent attributes of fd4 from the ClientInterview relation and rename it Interview.

clientNo	interviewDate	interviewTime	staffNo
CR76	13-May-02	10:30	SG5
CR56	13-May-02	12:00	SG5
CR74	13-May-02	12:00	SG37
CR56	1-Jul-02	10:30	SG5

- Write down the BCNF relations:
StaffRoom (staffNo, interviewdate, roomNo)
Interview (clientNo, interviewdate, interviewTime, staffNo)

Review of Normalization (UNF to BCNF)

<i>DreamHome</i> Property Inspection Report					
<i>DreamHome</i> Property Inspection Report					
Property Number <u>PG4</u>					
Property Address <u>6 Lawrence St, Glasgow</u>					
Inspection Date	Inspection Time	Comments	Staff no	Staff Name	Car Registration
18-Oct-00	10.00	Need to replace crockery	SG37	Ann Beech	M231 JGR
22-Apr-01	09.00	In good order	SG14	David Ford	M533 HDR
1-Oct-01	12.00	Damp rot in bathroom	SG14	David Ford	N721 HFR

Page 1

Review of Normalization (UNF to BCNF)

StaffPropertyInspection

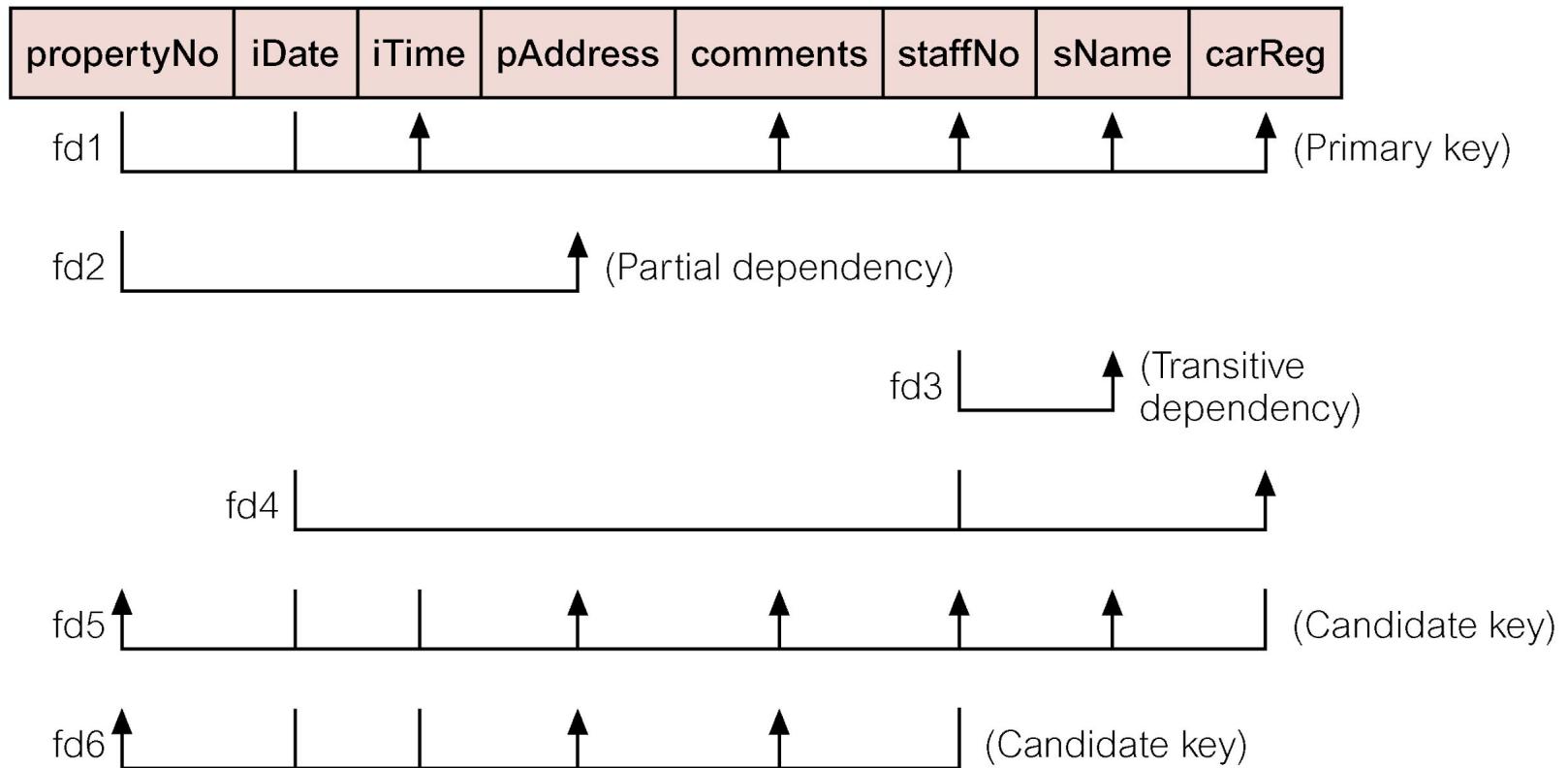
propertyNo	pAddress	iDate	iTime	comments	staffNo	sName	carReg
PG4	6 Lawrence St, Glasgow	18-Oct-00	10.00	Need to replace crockery	SG37	Ann Beech	M231 JGR
		22-Apr-01	09.00	In good order	SG14	David Ford	M533 HDR
		1-Oct-01	12.00	Damp rot in bathroom	SG14	David Ford	N721 HFR
PG16	5 Novar Dr, Glasgow	22-Apr-01	13.00	Replace living room carpet	SG14	David Ford	M533 HDR
		24-Oct-01	14.00	Good condition	SG37	Ann Beech	N721 HFR

StaffPropertyInspection

propertyNo	iDate	iTime	pAddress	comments	staffNo	sName	carReg
PG4	18-Oct-00	10.00	6 Lawrence St, Glasgow	Need to replace crockery	SG37	Ann Beech	M231 JGR
PG4	22-Apr-01	09.00	6 Lawrence St, Glasgow	In good order	SG14	David Ford	M533 HDR
PG4	1-Oct-01	12.00	6 Lawrence St, Glasgow	Damp rot in bathroom	SG14	David Ford	N721 HFR
PG16	22-Apr-01	13.00	5 Novar Dr, Glasgow	Replace living room carpet	SG14	David Ford	M533 HDR
PG16	24-Oct-01	14.00	5 Novar Dr, Glasgow	Good condition	SG37	Ann Beech	N721 HFR

Review of Normalization (UNF to BCNF)

StaffPropertyInspection



Review of Normalization (UNF to BCNF)

