# Chapter 14

## Normalization

# Chapter 14 - Objectives

- The purpose of normalization.
- The potential problems associated with redundant data in base relations.
- The concept and characteristics of functional dependency, which describes the relationship between attributes.
- How inference rules can identify a set of *all* functional dependencies for a relation.
- How to undertake the process of normalization.
- How to identify 1st, 2nd, 3rd and BCNF Normal Forms.

# Purpose of Normalization

- **Normalization is a technique for producing a set of suitable relations that support the data requirements of an enterprise.**
- **The benefits of of Normalization:**
  - **easier for the user to access and maintain the data;**
  - **take up minimal storage space on the computer.**

# Characteristics of a suitable set of relations

- The *minimal* number of attributes necessary to support the data requirements of the enterprise;
- attributes with a close logical relationship are found in the same relation;
- minimal redundancy with each attribute represented only once with…
- exception for attributes that form all or part of foreign keys.

# Data Redundancy and Update Anomalies

## Data Redundancy and Update Anomalies

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 |

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 |

# Data Redundancy and Update Anomalies

- **StaffBranch relation has redundant data; the details of a branch are repeated for every member of staff.**

- **In contrast, the branch information appears only once for each branch in the Branch relation and only the branch number (branchNo) is repeated in the Staff relation, to represent where each member of staff is located.**

# Data Redundancy and Update Anomalies

- **Relations that contain redundant information may potentially suffer from update anomalies.**

- **Types of update anomalies include**
  - **Insertion**
  - **Deletion**
  - **Modification**

# Properties of decomposition

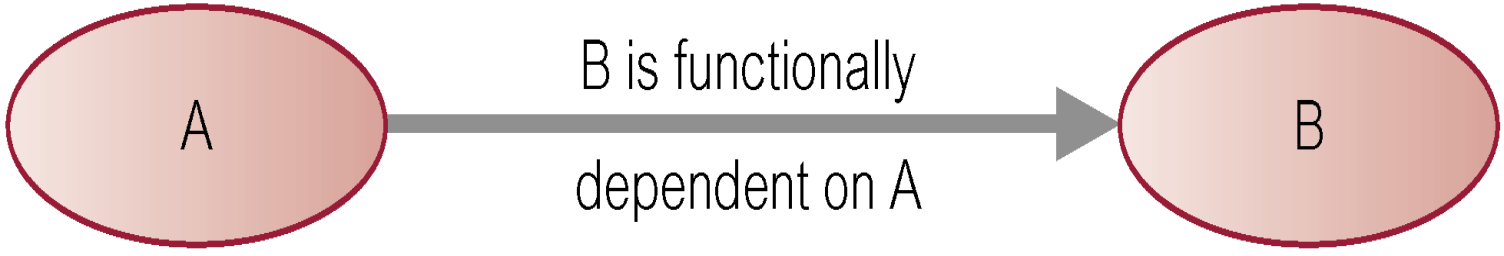
- **Two important properties of decomposition.**
  - ***Lossless-join property*** **enables us to find any instance of the original relation from corresponding instances in the smaller relations.**
  - ***Dependency preservation property*** **enables us to enforce a constraint on the original relation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

# Concept of Functional Dependencies

- **Important concept associated with normalization.**
- **Functional dependency describes relationship between attributes.**
- **For example, if A and B are attributes of relation R, B is functionally dependent on A (denoted $A \rightarrow B$), if each value of A in R is associated with exactly one value of B in R.**

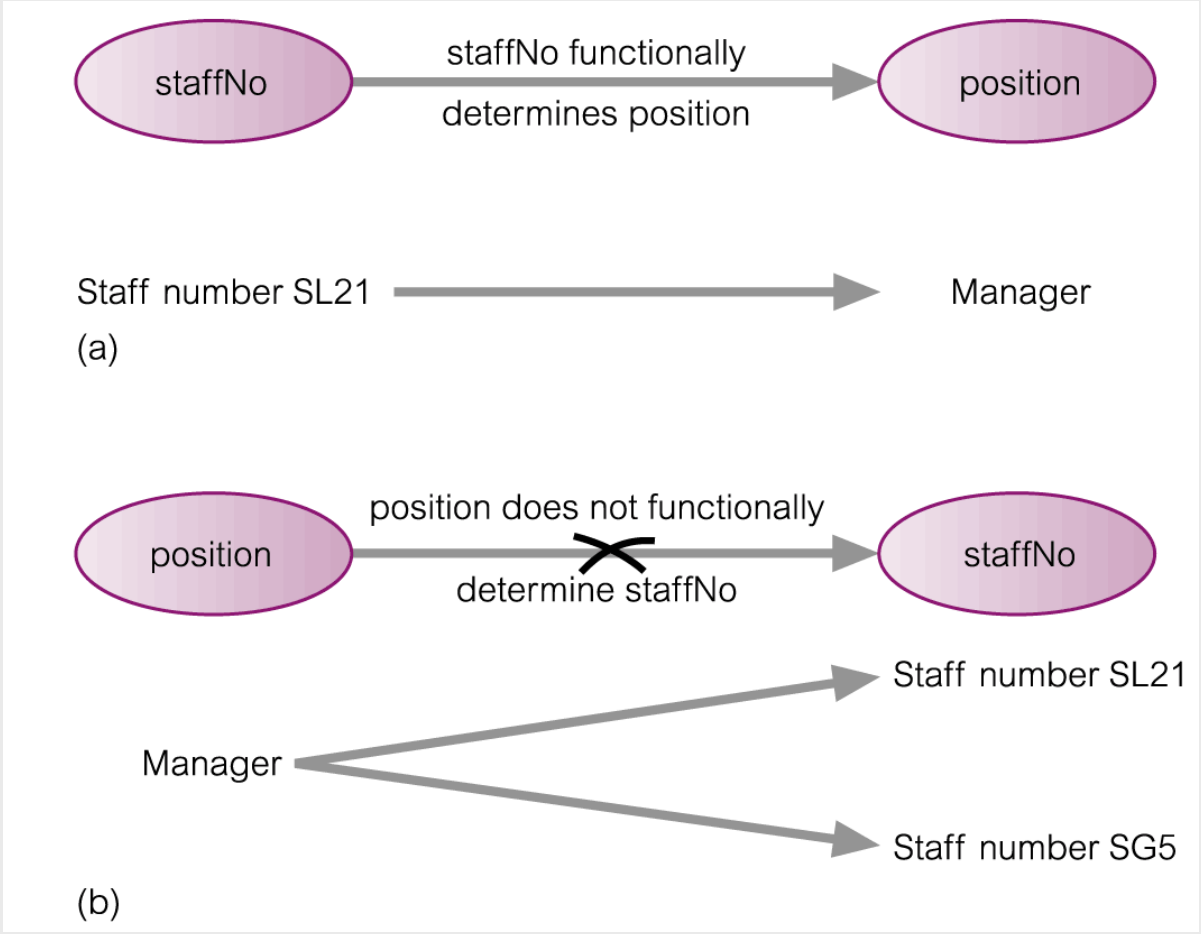
# Characteristics of Functional Dependencies

- **Property of the meaning or semantics of the attributes in a relation.**
- **Diagrammatic representation.**




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

# An Example Functional Dependency



# Example Functional Dependency that holds for all Time

- Consider the values shown in staffNo and sName attributes of the Staff relation (see Slide 5).

- Based on sample data, the following functional dependencies appear to hold.

staffNo → sName

sName → staffNo

# Example Functional Dependency that holds for all Time

- However, the only functional dependency that remains true for all possible values for the staffNo and sName attributes of the Staff relation is:

staffNo → sName

# Characteristics of Functional Dependencies

- **Determinants should have the minimal number of attributes necessary to maintain the functional dependency with the attribute(s) on the right hand-side.**

- **This requirement is called *full functional dependency*.**

# Characteristics of Functional Dependencies

- **Full functional dependency indicates that if A and B are attributes of a relation R,**
- **B is fully functionally dependent on A, if B is functionally dependent on A, but not on any proper subset of A.**

# Example Full Functional Dependency

- **In Staff relation (of Slide 5).**

  **staffNo, sName → branchNo**

- **Each value of (staffNo, sName) is associated with a single value of branchNo.**
- **However, branchNo is also functionally dependent on a subset of (staffNo, sName), namely staffNo.**
- **Example above is a *partial dependency*.**

# Characteristics of Functional Dependencies

- **Main characteristics of functional dependencies used in normalization:**
  - **There is a one-to-one relationship between the attribute(s) on the left-hand side (determinant) and those on the right-hand side of a functional dependency.**
  - **Holds for *all* time.**
  - **The determinant has the minimal number of attributes necessary to maintain the dependency with the attribute(s) on the right hand-side.**

# Transitive Dependencies

- **Important to recognize a transitive dependency because its existence in a relation can potentially cause update anomalies.**

- **Transitive dependency describes a condition where A, B, and C are attributes of a relation such that if A → B and B → C, then C is transitively dependent on A via B (provided that A is not functionally dependent on B or C).**

# Example Transitive Dependency

- **Consider functional dependencies in the StaffBranch relation (see Slide 6).**

**staffNo → sName, position, salary, branchNo, bAddress**

**branchNo → bAddress**

- **Transitive dependency, branchNo → bAddress exists on staffNo via branchNo.**

# Identifying Functional Dependencies

- Identifying all functional dependencies between a set of attributes is relatively simple if the meaning of each attribute and the relationships between the attributes are well understood.
- This information should be provided by the enterprise in the form of discussions with users and/or documentation such as the users' requirements specification.
- If unavailable then use common sense and/or experience

# More on Functional Dependencies

- **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.**

# Inference Rules for Functional Dependencies

- **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 Functional Dependencies

- **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 AC → BC**

  **(3)** ***Transitivity***

  **If A → B and B → C, then A → C**

# Inference Rules for Functional Dependencies

- **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***

  $A \rightarrow A$

  **(5)** ***Decomposition***

  **If $A \rightarrow BC$, then $A \rightarrow B$ and $A \rightarrow C$**

# Inference Rules for Functional Dependencies

**(6) *Union***

**If A → B and A → C, then A → BC**

**(7) *Composition***

**If A → B and C → D then AC → BD**

**A set of functional dependencies Y is covered by a set of functional dependencies X, if every functional dependency in Y is also in X+; that is, every dependency in Y can be inferred from X.**

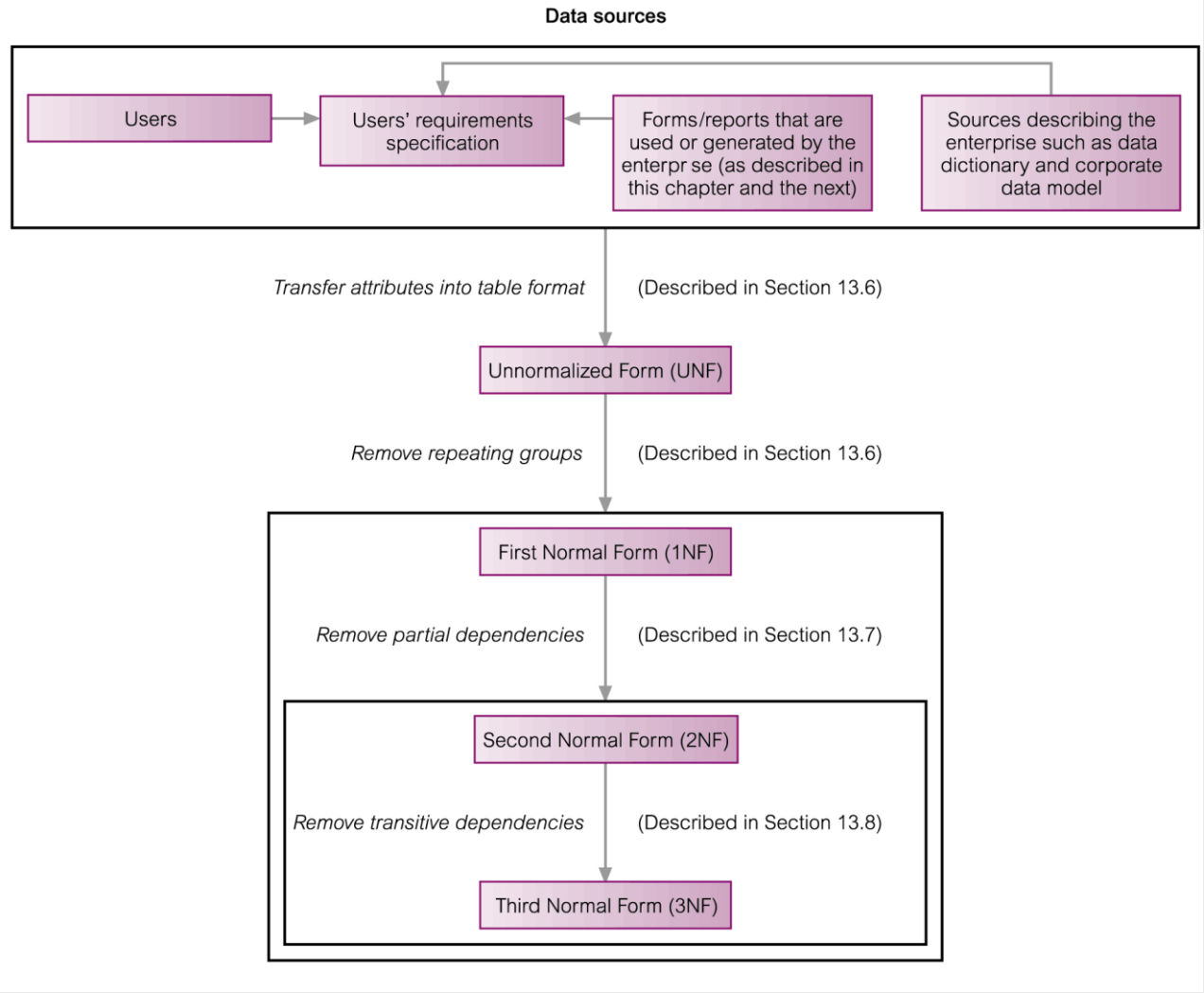
# Minimal Sets of Functional Dependencies

- A set of functional dependencies X is minimal if it satisfies the following conditions:
  - Every dependency in X has a single attribute on its right-hand side.
  - We cannot replace any dependency $A \rightarrow B$ in X with dependency $C \rightarrow B$, where C is a proper subset of A, and still have a set of dependencies that is equivalent to X.
  - We cannot remove any dependency from X and still have a set of dependencies that is equivalent to X.

# The Process of Normalization

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

# The Process of Normalization



# 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.

# 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 group by
  - Entering appropriate data into the empty columns of rows containing the repeating data ('flattening' the table).

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

# Second Normal Form (2NF)

- Based on the concept of full functional dependency.

- Full functional dependency indicates that if
  - A and B are attributes of a relation,
  - B is fully dependent on A if B is functionally dependent on A but not on any proper subset of A.

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

# Third Normal Form (3NF)

- Based on the concept of transitive dependency.

- Transitive Dependency is a condition where
  - A, B and C are attributes of a relation such that if $A \rightarrow B$ and $B \rightarrow C$,
  - then C is transitively dependent on A through B. (Provided that A is not functionally dependent on B or C).

# 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.**

# 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 a copy of their dominant.

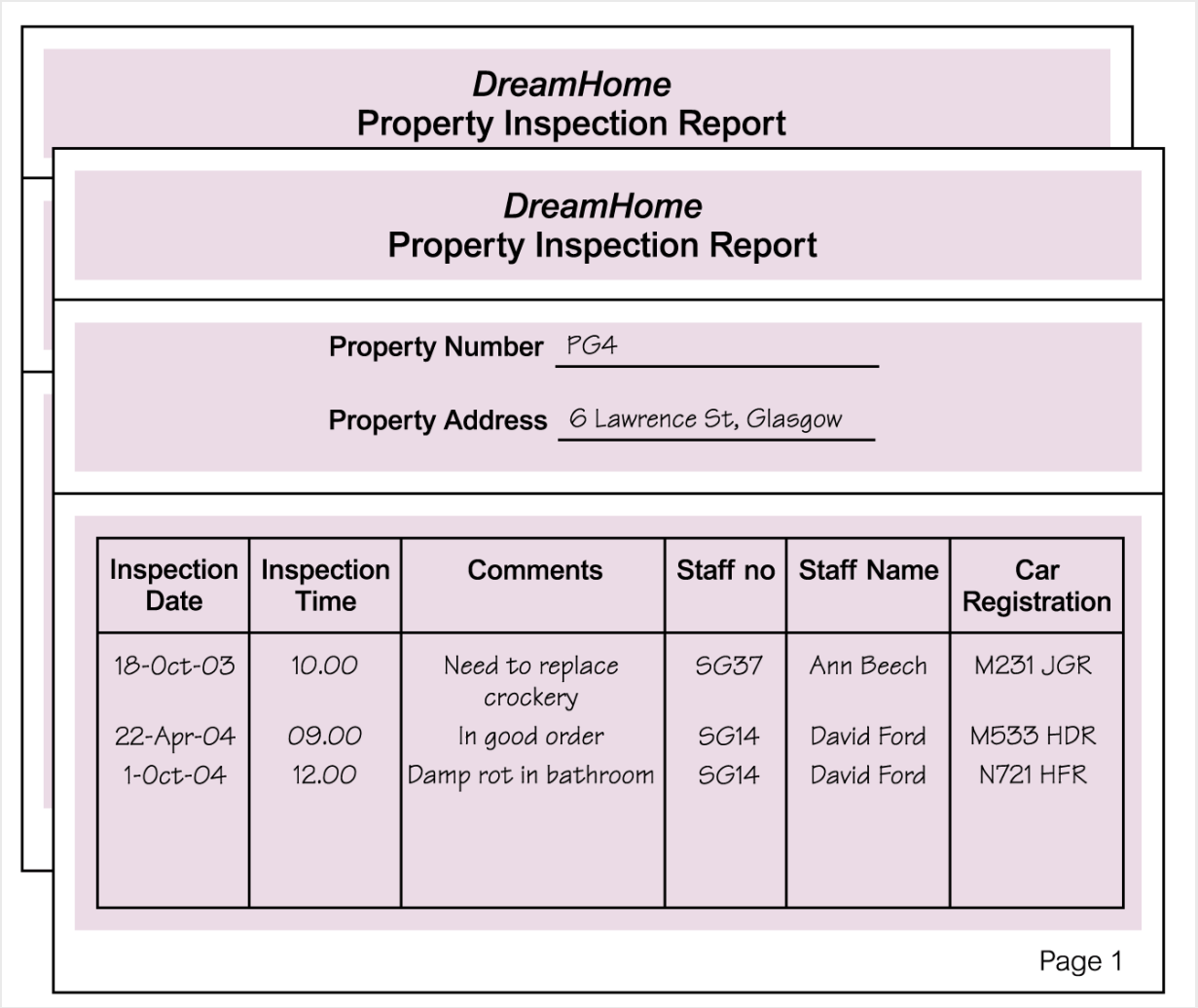
# Boyce–Codd Normal Form (BCNF)

- A relation is in BCNF if and only if every determinant is a candidate key.
- The potential to violate BCNF may occur in a relation that:
  - contains two (or more) composite candidate keys; and the candidate keys overlap, that is have at least one attribute in common.
- Every relation in BCNF is also in 3NF. However, a relation in 3NF is not necessarily in BCNF.

# 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.
- Violation of BCNF is quite rare.

# Review of Normalization (UNF to BCNF)



*DreamHome*
Property Inspection Report

*DreamHome*
Property Inspection Report

**Property Number** PG4

**Property Address** 6 Lawrence St, Glasgow

| Inspection Date | Inspection Time | Comments | Staff no | Staff Name | Car Registration |
|---|---|---|---|---|---|
| 18-Oct-03 | 10.00 | Need to replace crockery | SG37 | Ann Beech | M231 JGR |
| 22-Apr-04 | 09.00 | In good order | SG14 | David Ford | M533 HDR |
| 1-Oct-04 | 12.00 | Damp rot in bathroom | SG14 | David Ford | N721 HFR |

Page 1

# Review of Normalization (UNF to BCNF)

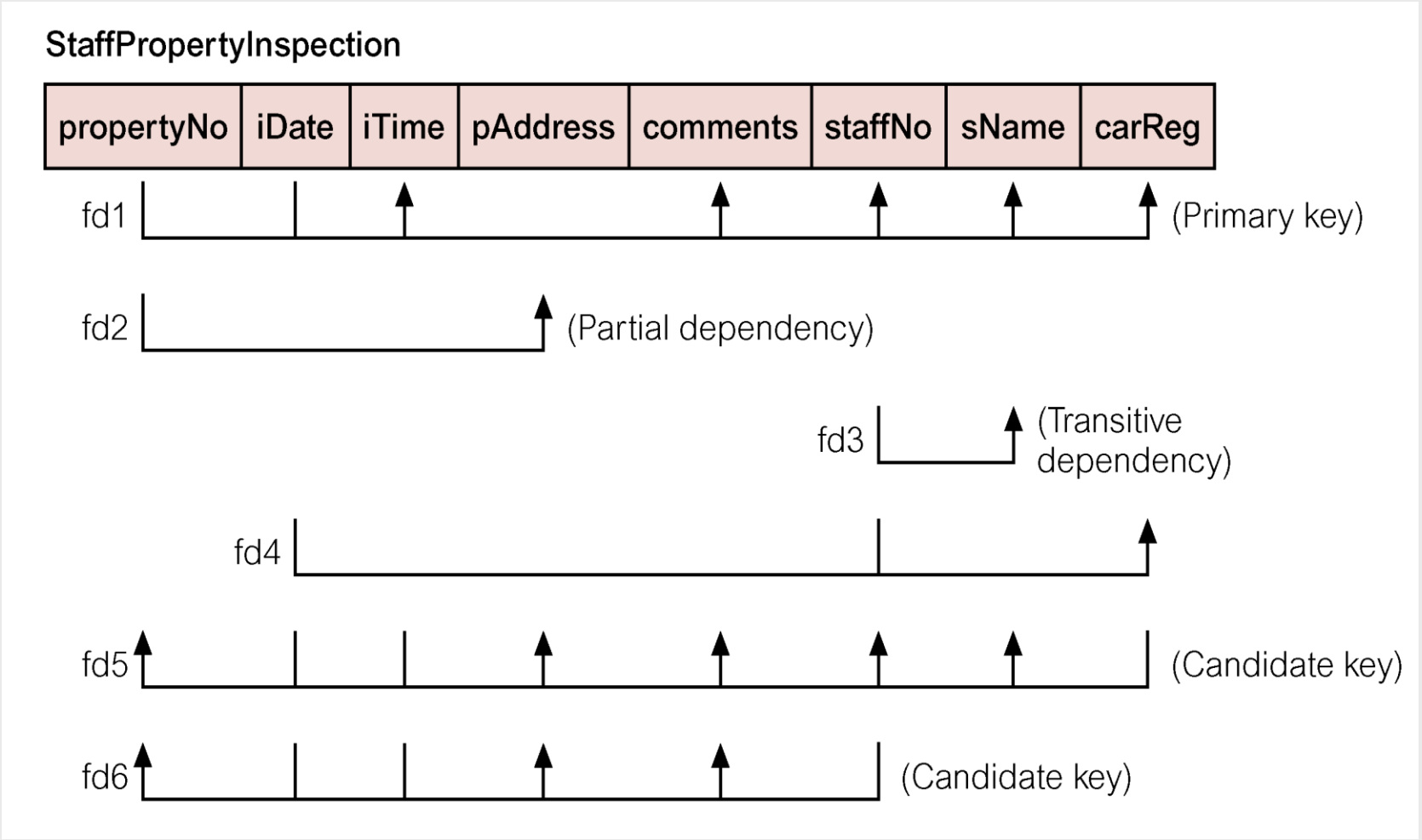
StaffPropertyInspection

| propertyNo | pAddress | iDate | iTime | comments | staffNo | sName | carReg |
|---|---|---|---|---|---|---|---|
| PG4 | 6 Lawrence St, Glasgow | 18-Oct-03<br>22-Apr-04<br>1-Oct-04 | 10.00<br>09.00<br>12.00 | Need to replace crockery<br>In good order<br>Damp rot in bathroom | SG37<br>SG14<br>SG14 | Ann Beech<br>David Ford<br>David Ford | M231 JGR<br>M533 HDR<br>N721 HFR |
| PG16 | 5 Novar Dr, Glasgow | 22-Apr-04<br>24-Oct-04 | 13.00<br>14.00 | Replace living room carpet<br>Good condition | SG14<br>SG37 | David Ford<br>Ann Beech | M533 HDR<br>N721 HFR |

StaffPropertyInspection

| propertyNo | iDate | iTime | pAddress | comments | staffNo | sName | carReg |
|---|---|---|---|---|---|---|---|
| PG4 | 18-Oct-03 | 10.00 | 6 Lawrence St, Glasgow | Need to replace crockery | SG37 | Ann Beech | M231 JGR |
| PG4 | 22-Apr-04 | 09.00 | 6 Lawrence St, Glasgow | In good order | SG14 | David Ford | M533 HDR |
| PG4 | 1-Oct-04 | 12.00 | 6 Lawrence St, Glasgow | Damp rot in bathroom | SG14 | David Ford | N721 HFR |
| PG16 | 22-Apr-04 | 13.00 | 5 Novar Dr, Glasgow | Replace living room carpet | SG14 | David Ford | M533 HDR |
| PG16 | 24-Oct-04 | 14.00 | 5 Novar Dr, Glasgow | Good condition | SG37 | Ann Beech | N721 HFR |

# Review of Normalization (UNF to BCNF)



# Review of Normalization (UNF to BCNF)