# EIE 3112
# Normalization

T. Connolly and C. Begg, "*Database Systems: A Practical Approach to Design, Implementation, and Management*," 6th Edition, Chapter 14, Pearson, 2015. (5th Edition is also fine)

# Objectives

- The purpose of normalization.
- The potential problems associated with redundant data in base relations.
- The concept of functional dependency, which describes the relationship between attributes.
- How to identify functional dependencies for a given relation.
- How functional dependencies identify the primary key for a relation.
- How to undertake the process of normalization.
- How normalization uses functional dependencies to group attributes into relations that are in a known normal form.

# Objectives

- How to identify the most commonly used normal forms, namely First Normal Form (1NF), Second Normal Form (2NF), and Third Normal Form (3NF).
- The problems associated with relations that break the rules of 1NF, 2NF, or 3NF.
- How to represent attributes shown on a form as 3NF relations using normalization.

# Purpose of Normalization

- We have learned how to use ER modeling to design database
- Normalization is another database design technique
- It begins by examining the relationships (called functional dependencies) between attributes.
- It uses a series of tests (called normal forms) to identify the optimal grouping of the attributes.

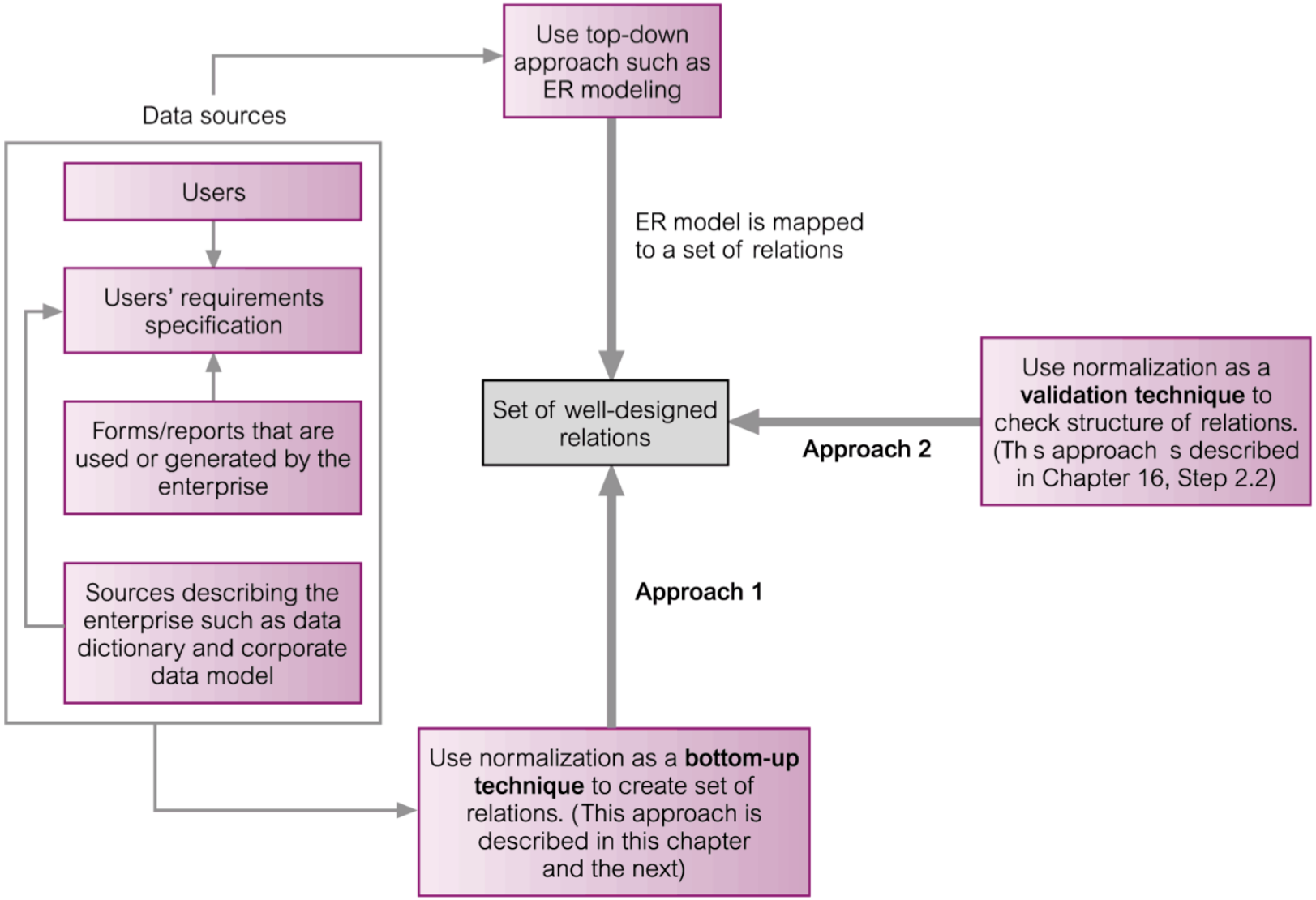
# Purpose of Normalization

- Characteristics of a suitable set of relations include:
  - attributes with a close logical relationship are found in the same relation;
  - *minimal* redundancy: each attribute is represented only once (except for the foreign keys).

# Purpose of Normalization

- The benefits of having a normalized database with well-designed relations:
  - easier for users to access and maintain the data
  - take up minimal storage space on the computer (because of less data duplications)

# How Normalization Supports Database Design



# Data Redundancy and Update Anomalies

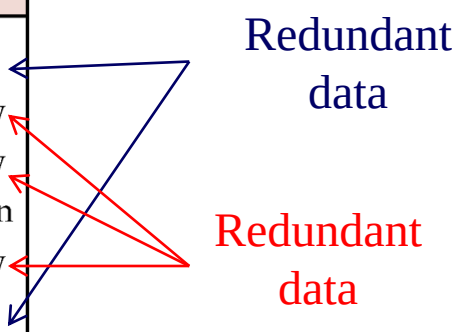
- Major aim of relational database design is to group attributes into relations to minimize data redundancy.
- Potential benefits:
  - Updates to the database are achieved with a minimal number of operations, thus **reducing** the opportunities for data **inconsistencies**.
  - Reduction in the file storage space required by the base relations thus **minimizing costs**.

# Data Redundancy and Update Anomalies

## With 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 |




## Without 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 |

# 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.

# Data Redundancy and Update Anomalies

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

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

# Data Redundancy and Update Anomalies

- Example of insertion anomaly:
  - Adding a new staff to StaffBranch requires adding the details of the branch where the new staff will be working.
  - Inserting details of a new branch that does not have any staff yet requires adding nulls to the entries corresponding to the staff. But it is not allowed to put null into the PK entries.

| 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 |

# Data Redundancy and Update Anomalies

- Example of deletion anomaly:
  - Deleting a row that represents the last member of staff in a branch will also remove all information of that branch in the database.

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 |

Delete this row →

Address of B007 will be lost

# Data Redundancy and Update Anomalies

- Example of modification anomaly:
  - Changing the branch address of a particular branch (e.g., B003) in StaffBranch requires changing the branch address of all staff working in B003.

StaffBranch

| 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 and Update Anomalies

- Another example: University database

| StdSSN | StdCity | StdClass | OfferNo | OffTerm | OffYear | EnrGrade | CourseNo | CrsDesc |
|---|---|---|---|---|---|---|---|---|
| S1 | SEATTLE | JUN | O1 | FALL | 2006 | 3.5 | C1 | DB |
| S1 | SEATTLE | JUN | O2 | FALL | 2006 | 3.3 | C2 | VB |
| S2 | BOTHELL | JUN | O3 | SPRING | 2007 | 3.1 | C3 | OO |
| S2 | BOTHELL | JUN | O2 | FALL | 2006 | 3.4 | C2 | VB |

Use one table for the entire database

Primary key: StdSSN, OfferNo

# Data Redundancy and Update Anomalies

- Insertion anomaly:
- Update (Modification) anomaly:
- Deletion anomaly:

| StdSSN | StdCity | StdClass | OfferNo | OffTerm | OffYear | EnrGrade | CourseNo | CrsDesc |
|---|---|---|---|---|---|---|---|---|
| S1 | SEATTLE | JUN | O1 | FALL | 2006 | 3.5 | C1 | DB |
| S1 | SEATTLE | JUN | O2 | FALL | 2006 | 3.3 | C2 | VB |
| S2 | BOTHELL | JUN | O3 | SPRING | 2007 | 3.1 | C3 | OO |
| S2 | BOTHELL | JUN | O2 | FALL | 2006 | 3.4 | C2 | VB |

# Data Redundancy and Update Anomalies

- Anomalies occur when a table contains facts about two or more different themes
- Normalization
  - Every normalized relation has a single theme
  - Normalization is to break up relation

# Functional Dependencies

- Functional dependency (FD) is an important concept associated with normalization.
- FD describes relationship between attributes.
- Definition of FD:
  - Assume that A and B are attributes of a relation
  - B is functionally dependent on A (denoted $A \rightarrow B$), if each value of A is associated with exactly one value of B.

A B

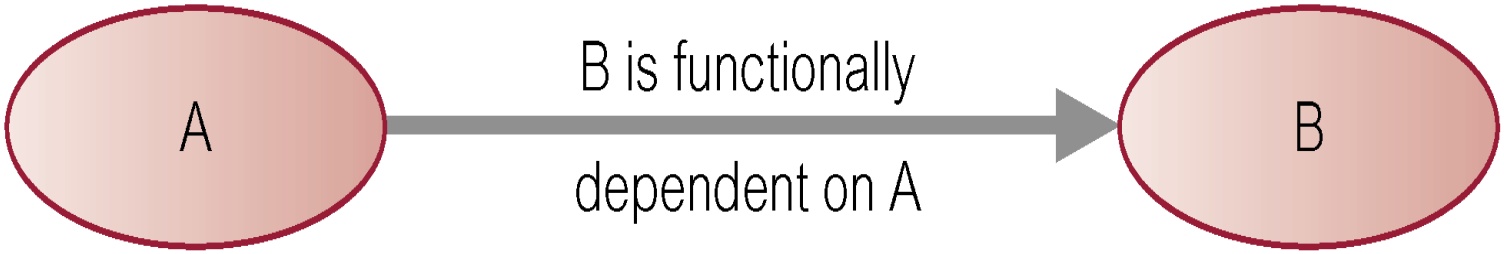
| staffNo | sName | position | salary | branchNo |
|---|---|---|---|---|
| SL21 | John White | Manager | 30000 | B005 |
| SG37 | Ann Beech | Assistant | 12000 | B003 |
| SG14 | David Ford | Supervisor | 18000 | B003 |
| SA9 | Mary Howe | Assistant | 9000 | B007 |
| SG5 | Susan Brand | Manager | 24000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

A B

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

# Characteristics of Functional Dependencies

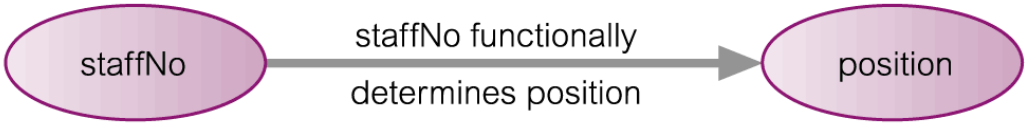
- A functional dependency (FD) is a constraint that specifies the relationship between two sets of attributes where one set can accurately determine the value of other sets.
- Diagrammatic representation.




- It is denoted as **A → B**, where **A** is a set of attributes that is capable of determining the value of **B**.
- The attribute set on the left side of the arrow, **A** is called **Determinant**, while on the right side, **B** is called the **Dependent**.

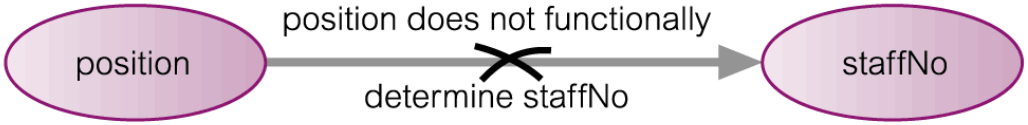
# An Example Functional Dependency

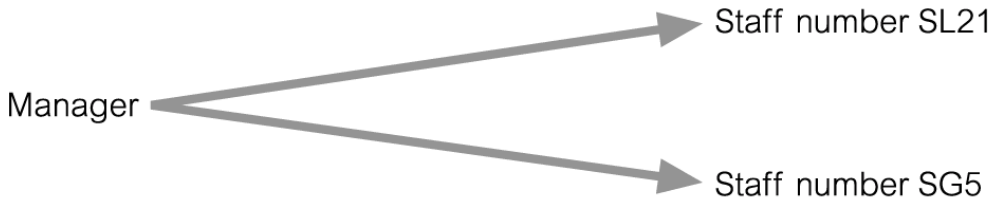
| staffNo | sName | position | salary | branchNo |
|---|---|---|---|---|
| SL21 | John White | Manager | 30000 | B005 |
| SG37 | Ann Beech | Assistant | 12000 | B003 |
| SG14 | David Ford | Supervisor | 18000 | B003 |
| SA9 | Mary Howe | Assistant | 9000 | B007 |
| SG5 | Susan Brand | Manager | 24000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





(a)




(b)

# Functional Dependency that Holds for All Time

- Consider the values shown in staffNo and sName attributes of the Staff relation.
- Based on sample data, the following functional dependencies appear to hold.
  - staffNo → sName
  - sName → staffNo ???

| 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 |

# 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:

$$staffNo \rightarrow sName$$

| 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 |

# Tutorial Q8

Given the following table

a) Identify the functional dependencies.

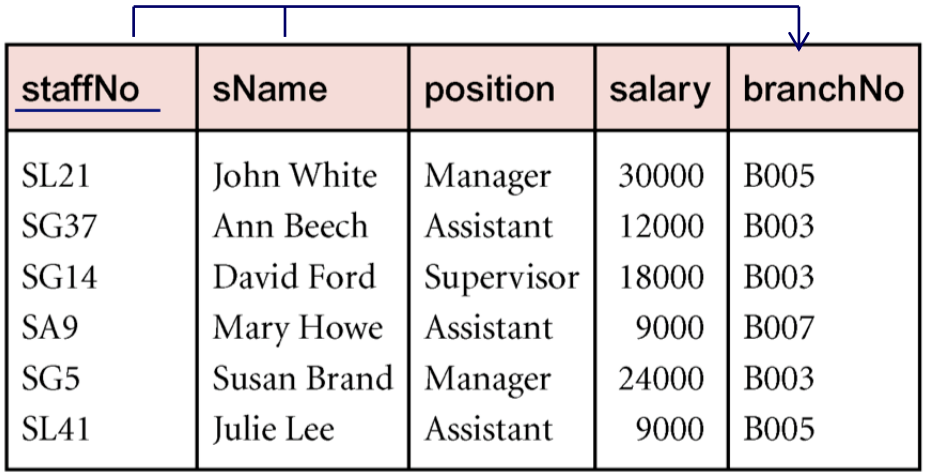
| CustNo | CustName | CustTel | ProdNo | ProdName | UnitCost | OrderNo | Qty |
|---|---|---|---|---|---|---|---|
| C1 | Peter | 1234567 | P1 | Shoes | 10 | O1 | 1 |
| C1 | Peter | 1234567 | P2 | Bottle | 20 | O1 | 2 |
| C1 | Peter | 1234567 | P1 | Shoes | 10 | O2 | 4 |
| C2 | Paul | 7654321 | P4 | Cup | 40 | O3 | 2 |
| C2 | Paul | 7654321 | P5 | Disk | 50 | O4 | 1 |
| C2 | Paul | 7654321 | P3 | Dress | 30 | O4 | 1 |

Functional Dependencies:

# Example of Partial Dependency

- Exists in the Staff relation

  staffNo, sName → branchNo
- True - 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. This is an example of ***partial dependency***.



| 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 |

# Characteristics of Functional Dependencies

- 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

- 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

  A→B, B→C, then A→C

- It is important to recognize a transitive dependency because its existence in a relation can potentially cause update anomalies

# Example Transitive Dependency

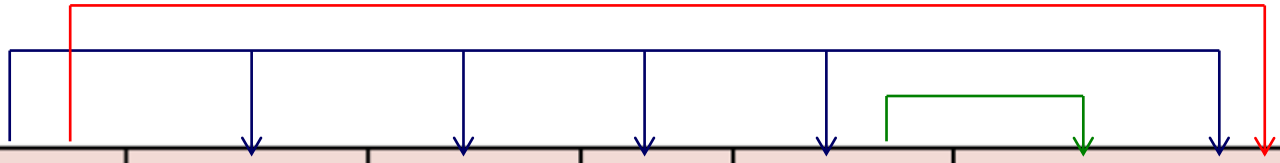
- Consider functional dependencies in the StaffBranch relation.

  staffNo → sName, position, salary, branchNo, bAddress

  branchNo → bAddress

- Transitive dependency:

  staffNo → bAddress via branchNo.



| 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 |

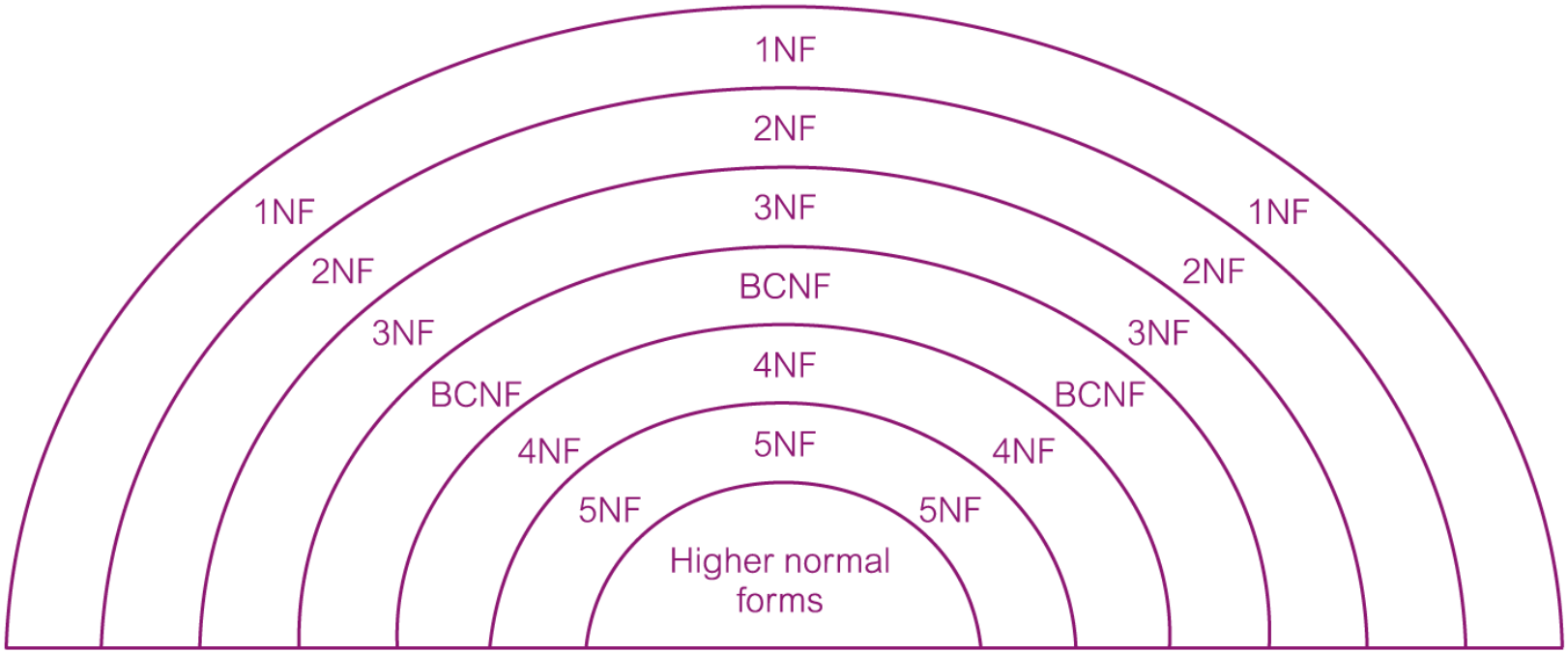
# The Process of Normalization

- Formal technique for analyzing a relation based on its primary key and the functional dependencies between the attributes of that relation.

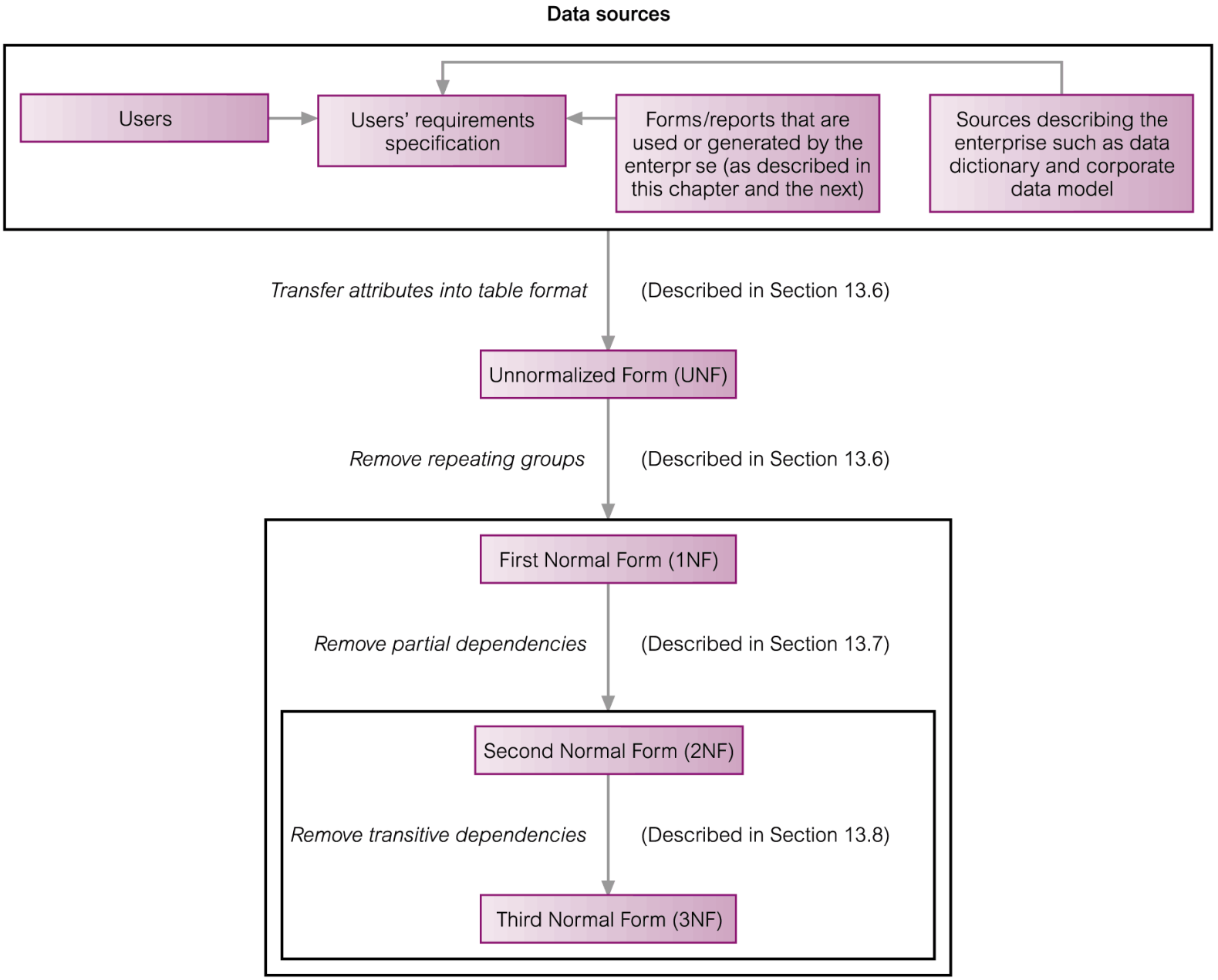
- Often executed as a series of steps.  Each step corresponds to a specific normal form, which has known properties.

# The Process of Normalization

- 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 un-normalized table
  - Transform the data from the information source (e.g. forms) into table format with columns and rows.

# Unnormalized Form (UNF)

Example of UNF

SALESPERSON/PRODUCT table

| Salesperson Number | Product Number | Salesperson Name | Commission Percentage | Year of Hire | Department Number | Manager Name | Product Name | Unit Price | Quantity |
|---|---|---|---|---|---|---|---|---|---|
| 137 | 19440 | Baker | 10 | 1995 | 73 | Scott | Hammer | 17.50 | 473 |
| | 24013 | | | | | | Saw | 26.25 | 170 |
| | 26722 | | | | | | Pliers | 11.50 | 688 |
| 186 | 16386 | Adams | 15 | 2001 | 59 | Lopez | Wrench | 12.95 | 1745 |
| | 19440 | | | | | | Hammer | 17.50 | 2529 |
| | 21765 | | | | | | Drill | 32.99 | 1962 |
| | 24013 | | | | | | Saw | 26.25 | 3071 |
| 204 | 21765 | Dickens | 10 | 1998 | 73 | Scott | Drill | 32.99 | 809 |
| | 26722 | | | | | | Pliers | 11.50 | 734 |
| 361 | 16386 | Carlyle | 20 | 2001 | 73 | Scott | Wrench | 12.95 | 3729 |
| | 21765 | | | | | | Drill | 32.99 | 3110 |
| | 26722 | | | | | | Pliers | 11.50 | 2738 |

204 21765,26722 10 1998 73 Scott Drill,Pliers 32.99,115.0 809,734

Repeating group of attribute

# First Normal Form (1NF)

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

SALESPERSON/PRODUCT table

| Salesperson Number | Product Number | Salesperson Name | Commission Percentage | Year of Hire | Department Number | Manager Name | Product Name | Unit Price | Quantity |
|---|---|---|---|---|---|---|---|---|---|
| 137 | 19440 | Baker | 10 | 1995 | 73 | Scott | Hammer | 17.50 | 473 |
| 137 | 24013 | Baker | 10 | 1995 | 73 | Scott | Saw | 26.25 | 170 |
| 137 | 26722 | Baker | 10 | 1995 | 73 | Scott | Pliers | 11.50 | 688 |
| 186 | 16386 | Adams | 15 | 2001 | 59 | Lopez | Wrench | 12.95 | 1475 |
| 186 | 19440 | Adams | 15 | 2001 | 59 | Lopez | Hammer | 17.50 | 2529 |
| 186 | 21765 | Adams | 15 | 2001 | 59 | Lopez | Drill | 32.99 | 1962 |
| 186 | 24013 | Adams | 15 | 2001 | 59 | Lopez | Saw | 26.25 | 3071 |
| 204 | 21765 | Dickens | 10 | 1998 | 73 | Scott | Drill | 32.99 | 809 |
| 204 | 26722 | Dickens | 10 | 1998 | 73 | Scott | Pliers | 11.50 | 734 |
| 361 | 16386 | Carlyle | 20 | 2001 | 73 | Scott | Wrench | 12.95 | 3729 |
| 361 | 21765 | Carlyle | 20 | 2001 | 73 | Scott | Drill | 32.99 | 3110 |
| 361 | 26722 | Carlyle | 20 | 2001 | 73 | Scott | Pliers | 11.50 | 2738 |

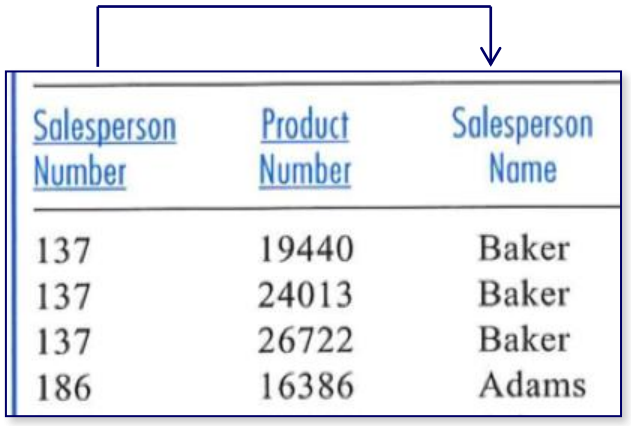
Table in 1NF

# 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).
- Remove repeating groups by entering appropriate data into the empty rows.

# Second Normal Form (2NF)

- For a table in 2NF, it must be in 1NF and each non-key attribute must be **dependent on the whole key**
- Table with single-column key is in 2NF.
- Violations
  - Part of the key → nonkey, e.g., SalepersonNumber →SalepersonName
- Solution:
  - Split the table



| Salesperson Number | Product Number | Salesperson Name |
|---|---|---|
| 137 | 19440 | Baker |
| 137 | 24013 | Baker |
| 137 | 26722 | Baker |
| 186 | 16386 | Adams |

# 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.

  (Keep a copy of determinant in the original table!)

# 1NF to 2NF

SALESPERSON/PRODUCT table

| Salesperson Number | Product Number | Salesperson Name | Commission Percentage | Year of Hire | Department Number | Manager Name | Product Name | Unit Price | Quantity |
|---|---|---|---|---|---|---|---|---|---|

Primary Key

Table in 1NF

Salesperson Number → Salesperson Name
Salesperson Number → Commission Percentage
Salesperson Number → Year of Hire
Salesperson Number → Department Number
Salesperson Number → Manager Name
Product Number → Product Name
Product Number → Unit Price
Department Number → Manager Name
Salesperson Number, Product Number → Quantity

Violation of 2NF

Functional Dependence

# 1NF to 2NF

**Solution:** Split the 1NF table into several 2NF tables to remove the partial dependencies


Tables in 2NF

# Relations in 2NF

SALESPERSON table

| Salesperson Number | Salesperson Name | Commission Percentage | Year of Hire | Department Number | Manager Name |
|---|---|---|---|---|---|
| 137 | Baker | 10 | 1995 | 73 | Scott |
| 186 | Adams | 15 | 2001 | 59 | Lopez |
| 204 | Dickens | 10 | 1998 | 73 | Scott |
| 361 | Carlyle | 20 | 2001 | 73 | Scott |

PRODUCT table

| Product Number | Product Name | Unit Price |
|---|---|---|
| 16386 | Wrench | 12.95 |
| 19440 | Hammer | 17.50 |
| 21765 | Drill | 32.99 |
| 24013 | Saw | 26.25 |
| 26722 | Pliers | 11.50 |

QUANTITY table

| Salesperson Number | Product Number | Quantity |
|---|---|---|
| 137 | 19440 | 473 |
| 137 | 24013 | 170 |
| 137 | 26722 | 688 |
| 186 | 16386 | 1745 |
| 186 | 19440 | 2529 |
| 186 | 21765 | 1962 |
| 186 | 24013 | 3071 |
| 204 | 21765 | 809 |
| 204 | 26722 | 734 |
| 361 | 16386 | 3729 |
| 361 | 21765 | 3110 |
| 361 | 26722 | 2738 |

Tables in 2NF

# 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)

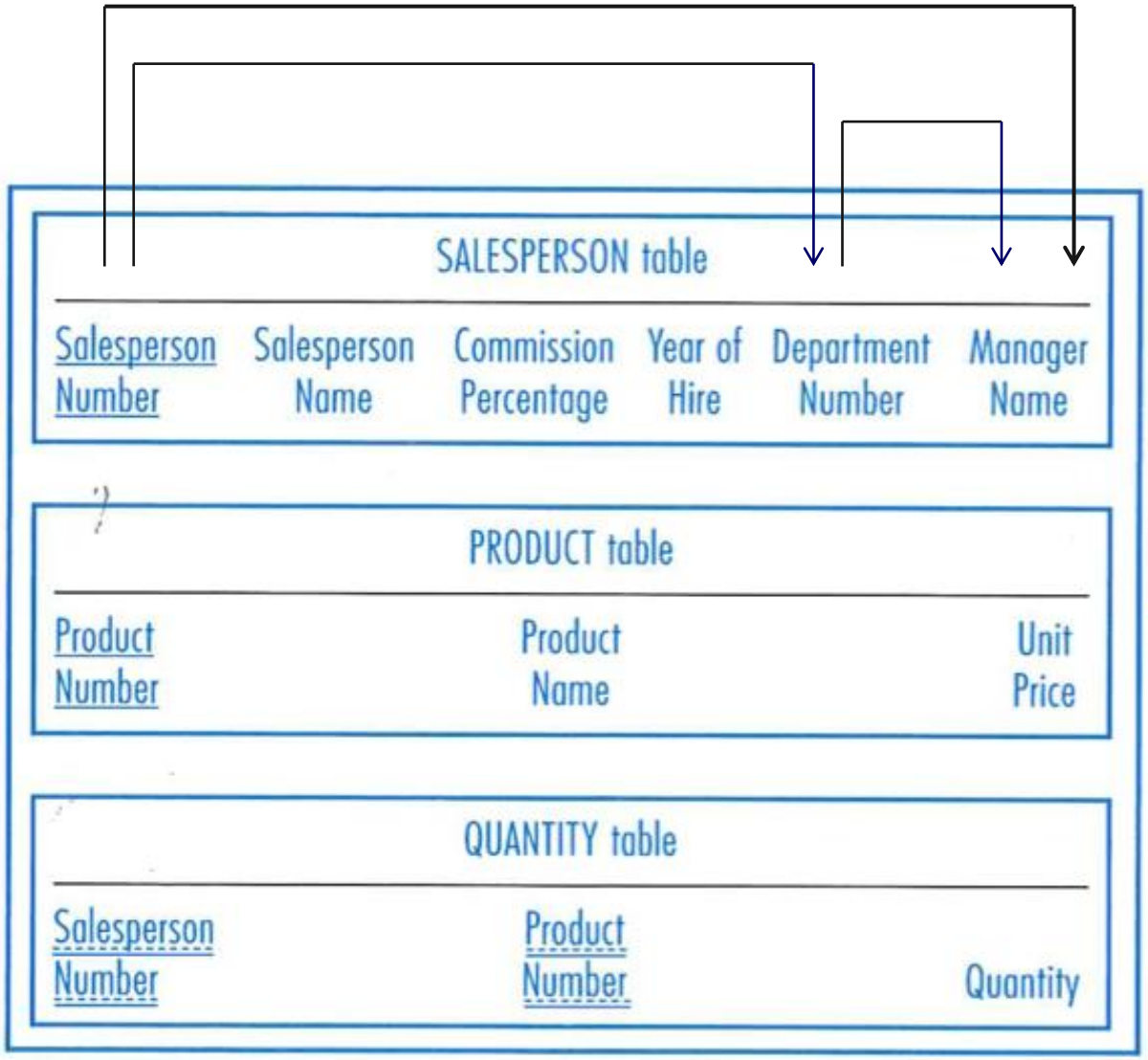
- A relation is in 3NF if it is
  - in 1NF and 2NF and
  - none of the non-primary-key attribute is transitively dependent on the primary key.
- Therefore, there is no nonkey → nonkey via transitive dependence

# 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 determinants.

  (Keep a copy of determinant in the original table!)

# 2NF to 3NF



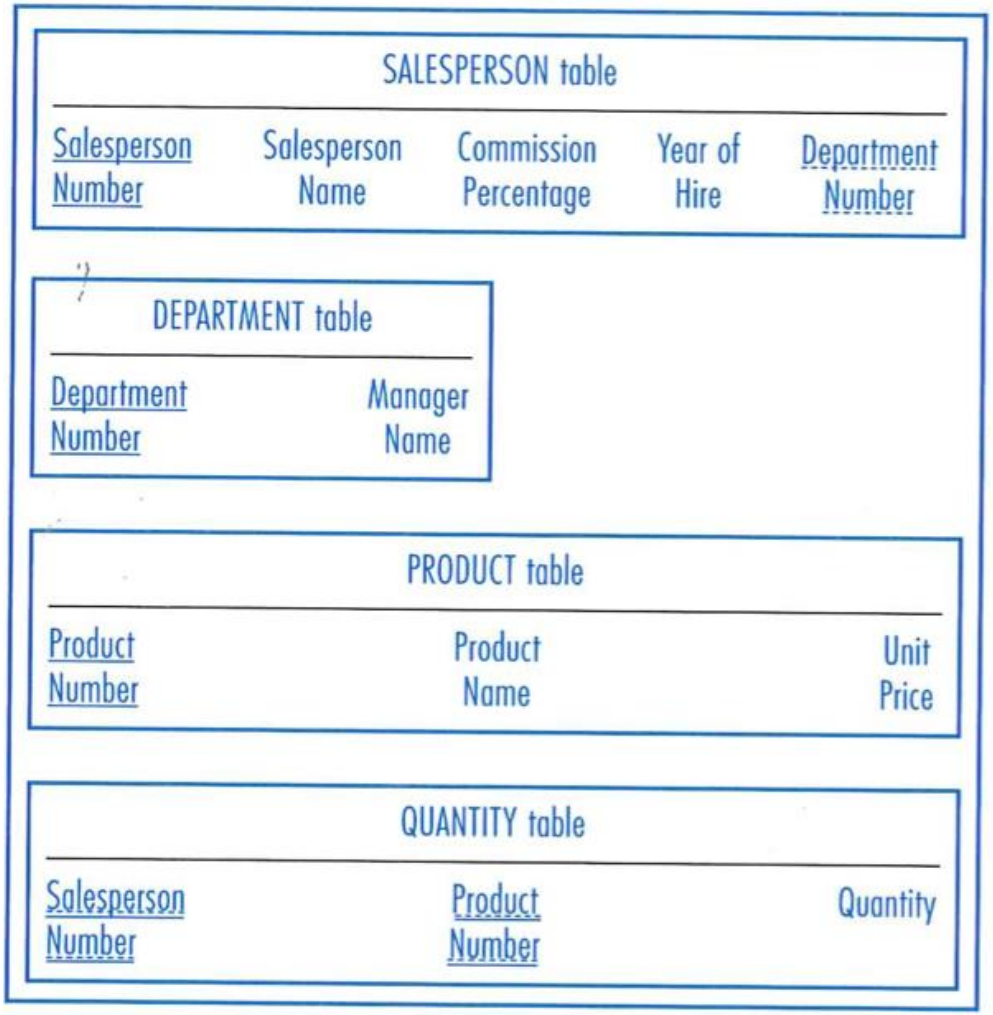

Why violating 3NF?

1. ManagerName is transitively dependent on SalespersonNumber
2. ManagerName depends on DepartmentNumber and SalespersonNumber

Salesperson Number → Salesperson Name
Salesperson Number → Commission Percentage
Salesperson Number → Year of Hire
Salesperson Number → Department Number
Salesperson Number → Manager Name
Product Number → Product Name
Product Number → Unit Price
Department Number → Manager Name
Salesperson Number, Product Number → Quantity

# 2NF to 3NF

Solution: Split the SALEPERSON table to move the ManagerName to a new table called DEPARTMENT




Salesperson Number → Salesperson Name
Salesperson Number → Commission Percentage
Salesperson Number → Year of Hire
Salesperson Number → Department Number
Salesperson Number → Manager Name
Product Number → Product Name
Product Number → Unit Price
Department Number → Manager Name
Salesperson Number, Product Number → Quantity

# Relations in 3NF

SALESPERSON table

| Salesperson Number | Salesperson Name | Commission Percentage | Year of Hire | Department Number |
|---|---|---|---|---|
| 137 | Baker | 10 | 1995 | 73 |
| 186 | Adams | 15 | 2001 | 59 |
| 204 | Dickens | 10 | 1998 | 73 |
| 361 | Carlyle | 20 | 2001 | 73 |

DEPARTMENT table

| Department Number | Manager Name |
|---|---|
| 59 | Lopez |
| 73 | Scott |

PRODUCT table

| Product Number | Product Name | Unit Price |
|---|---|---|
| 16386 | Wrench | 12.95 |
| 19440 | Hammer | 17.50 |
| 21765 | Drill | 32.99 |
| 24013 | Saw | 26.25 |
| 26722 | Pliers | 11.50 |

QUANTITY Table

| Salesperson Number | Product Number | Quantity |
|---|---|---|
| 137 | 19440 | 473 |
| 137 | 24013 | 170 |
| 137 | 26722 | 688 |
| 186 | 16386 | 1745 |
| 186 | 19440 | 2529 |
| 186 | 21765 | 1962 |
| 186 | 24013 | 3071 |
| 204 | 21765 | 809 |
| 204 | 26722 | 734 |
| 361 | 16386 | 3729 |
| 361 | 21765 | 3110 |
| 361 | 26722 | 2738 |

# Normal Forms: Review

Unnormalized – There are multivalued attributes or repeating groups

1 NF – No multivalued attributes or repeating groups.

2 NF – 1 NF plus no partial dependencies

3 NF – 2 NF plus no transitive dependencies

More precisely, every non-key (including non-candidate key) attribute depends on the key (1NF), the whole key (2NF) and nothing but the key (3NF).

# Tutorial Q8

b) Normalize the table to 3NF (Hints: you may need to use 4 tables and provide proper names for these tables).

| CustNo | CustName | CustTel | ProdNo | ProdName | UnitCost | OrderNo | Qty |
|---|---|---|---|---|---|---|---|
| C1 | Peter | 1234567 | P1 | Shoes | 10 | O1 | 1 |
| C1 | Peter | 1234567 | P2 | Bottle | 20 | O1 | 2 |
| C1 | Peter | 1234567 | P1 | Shoes | 10 | O2 | 4 |
| C2 | Paul | 7654321 | P4 | Cup | 40 | O3 | 2 |
| C2 | Paul | 7654321 | P5 | Disk | 50 | O4 | 1 |
| C2 | Paul | 7654321 | P3 | Dress | 30 | O4 | 1 |

# Tutorial Q8 solution

1NF: (No multivalued attributes or repeating groups)

2NF: (No partial dependencies)

3NF: (No transitive dependencies)