13 Normalization

Prerequisites for This Section

- **Readings:**
  - **Required:** Connolly and Begg, sections 13.1–13.4 (in third edition, sections 13.1–13.3).
  - **Required:** Connolly and Begg, sections 13.5–13.9 (in third edition, sections 13.4–13.8, and 13.10).
  - **Optional:** Connolly and Begg, sections 14.1–14.4 (in third edition, sections 13.9 and 13.11).

- **Assessments**
  - Multiple-Choice Quiz 6

Section Objectives

In this section you will learn:

① The purpose of normalization.

② The potential problems associated with redundant data in base relations.

③ The concept of functional dependency, which describes the relationship between attributes.

④ How to undertake the process of normalization.

⑤ How to identify 1NF, 2NF, 3NF .

⑥ The problems associated with relations that break the rules of 1NF, 2NF, or 3NF.

Agenda

1. Why Normalize?
2. Functional Dependencies
3. First Normal Form (1NF)
4. Second Normal Form (2NF)
5. Third Normal Form (3NF)
6. The Process of Normalization

An Example of Bad Database Design

- A member has borrowed many books
- Modify the SD from ‘CS’ to ‘ISS’ for Susan
- Insert the information of a new student
- Return the only book a student has borrowed

STUDENT_BOOK

S#	SN	SD	PHONE	B#	BN	DATE
200413001	Susan	CS	87654321	B001	DS	20070403
200413001	Susan	CS	87654321	B002	DBS	20070403
200413001	Susan	CS	87654321	B003	PL	20070406
200413001	Susan	CS	87654321	B004	OS	20070406
200413001	Susan	CS	87654321	B005	SE	20070420
200413001	Susan	CS	87654321	B006	AI	20070420
200512001	Avi	MA	12345678	C001	EL	20070508
200612002	Rajiv	MA	12345678	B001	DS	20070509

Solve the Problem

Solution ---- ：

S（S#,SN,SD,PHONE），B（B#,BN），SB（S#,B#,DATE）

S#	SN	SD	PHONE
200413001	Susan	CS	87654321
200512001	Avi	MA	12345678
200612002	Rajiv	MA	12345678

B#	BN
B001	DS
B002	DBS
B003	PL
B004	OS
B005	SE
B006	AI
C001	EL

S#	B#	DATE
200413001	B001	20070403
200413001	B002	20070403
200413001	B003	20070406
200413001	B004	20070406
200413001	B005	20070420
200413001	B006	20070420
200512001	C001	20070508
200812002	B001	20070509

Purpose of Normalization

- **Normalization** is a technique for producing a set of suitable relations that support the data requirements of an enterprise.

- Characteristics of a suitable set of relations include:
  - 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 the important exception of attributes that form all or part of foreign keys.

Benefits of Normalization

- Easier for the user to access and maintain the data;
- Takes up minimal storage space on the computer.

Data Redundancy and Update Anomalies

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

- Three kinds of *update anomalies* :

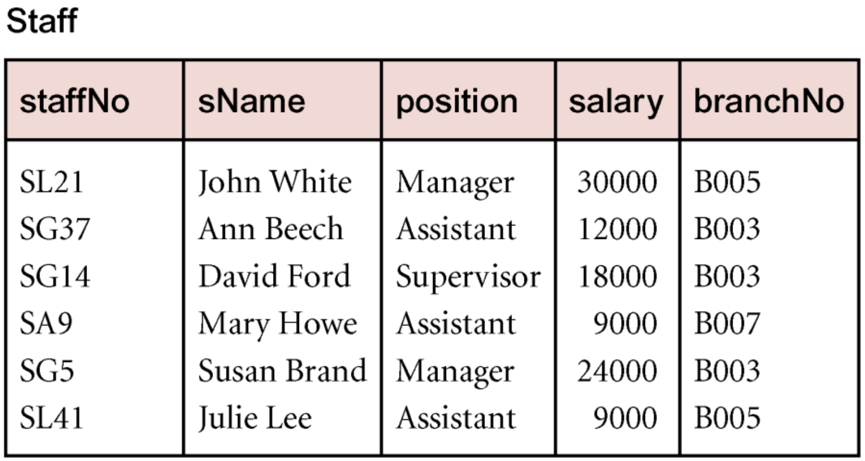
① An ***insertion anomaly*** occurs when we are prevented from inserting information that we want to keep track of.

② A ***deletion anomaly*** occurs when a deletion leads to an unintended drop of data.

③ A ***modification anomaly*** typically leads to inconsistencies because of missed updates.

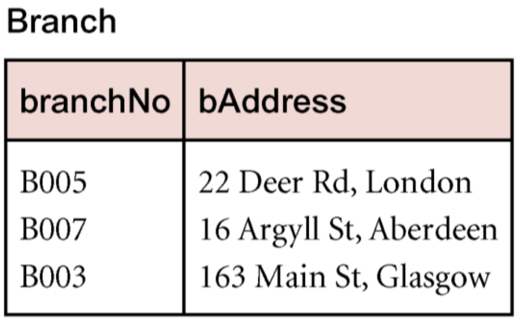
Example of Update Anomalies

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



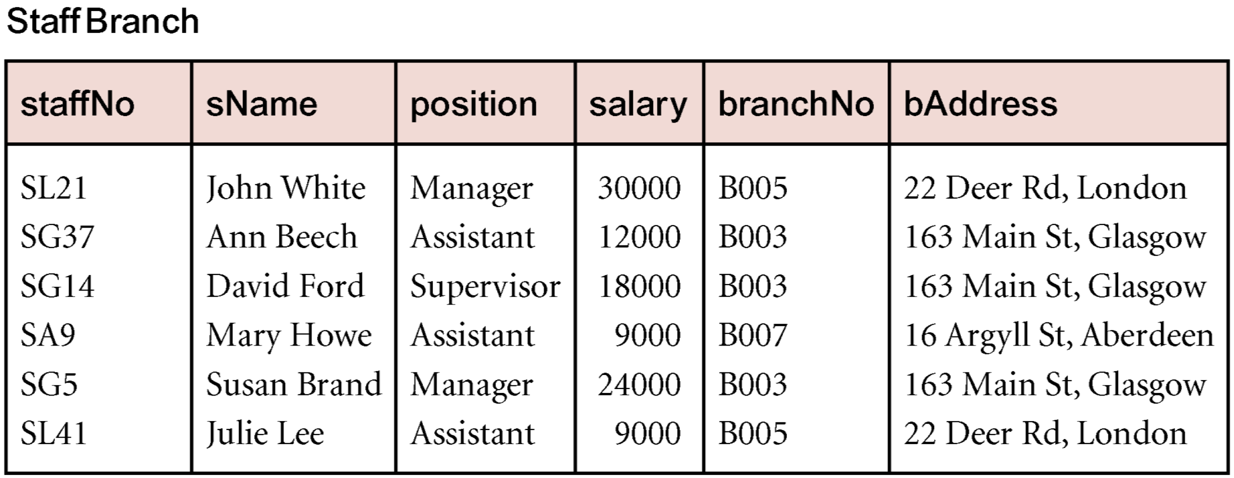
Staff

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

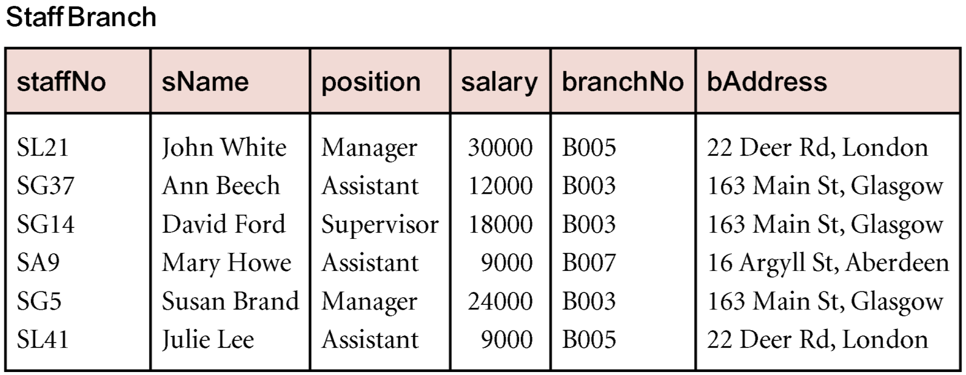
branchNo	bAddress
B005	22 Deer Rd, London
B007	16 Argyll St, Aberdeen
B003	163 Main St, Glasgow



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

Example of Update Anomalies



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

① Insertion anomalies

a) To insert the details of new members of staff, we must include the details of the branch at which the staff are to be located.

b) To insert the details of a new branch that currently has no members of staff

② Deletion anomalies

To delete the last member of staff located at a branch, the details about that branch are also lost from the database

③ Modification anomalies

To change the address for branch number B003, we must update the tuples of all staff located at that branch.

Causes of Update Anomalies

- Some dependent relationships between attributes within a single relation schema:
  - **Functional dependency**
    - Partial functional dependency
    - Transitive functional dependency
  - Multi-valued dependency
  - Join dependency

Agenda

1. Why Normalize?
2. Functional Dependencies
3. First Normal Form (1NF)
4. Second Normal Form (2NF)
5. Third Normal Form (3NF)
6. The Process of Normalization

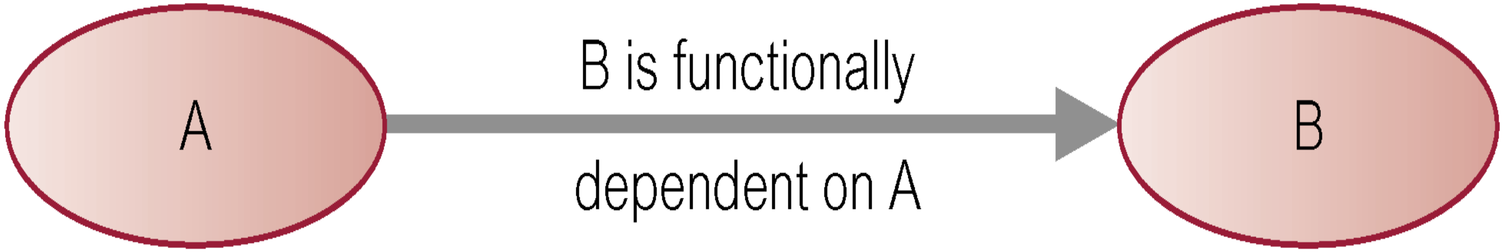
Functional Dependencies

- Functional dependency describes the relationship between attributes.

- if A and B are attributes of relation R, B is functionally dependent on A (denoted A → B), if each value of A in R is associated with exactly one value of B in R.

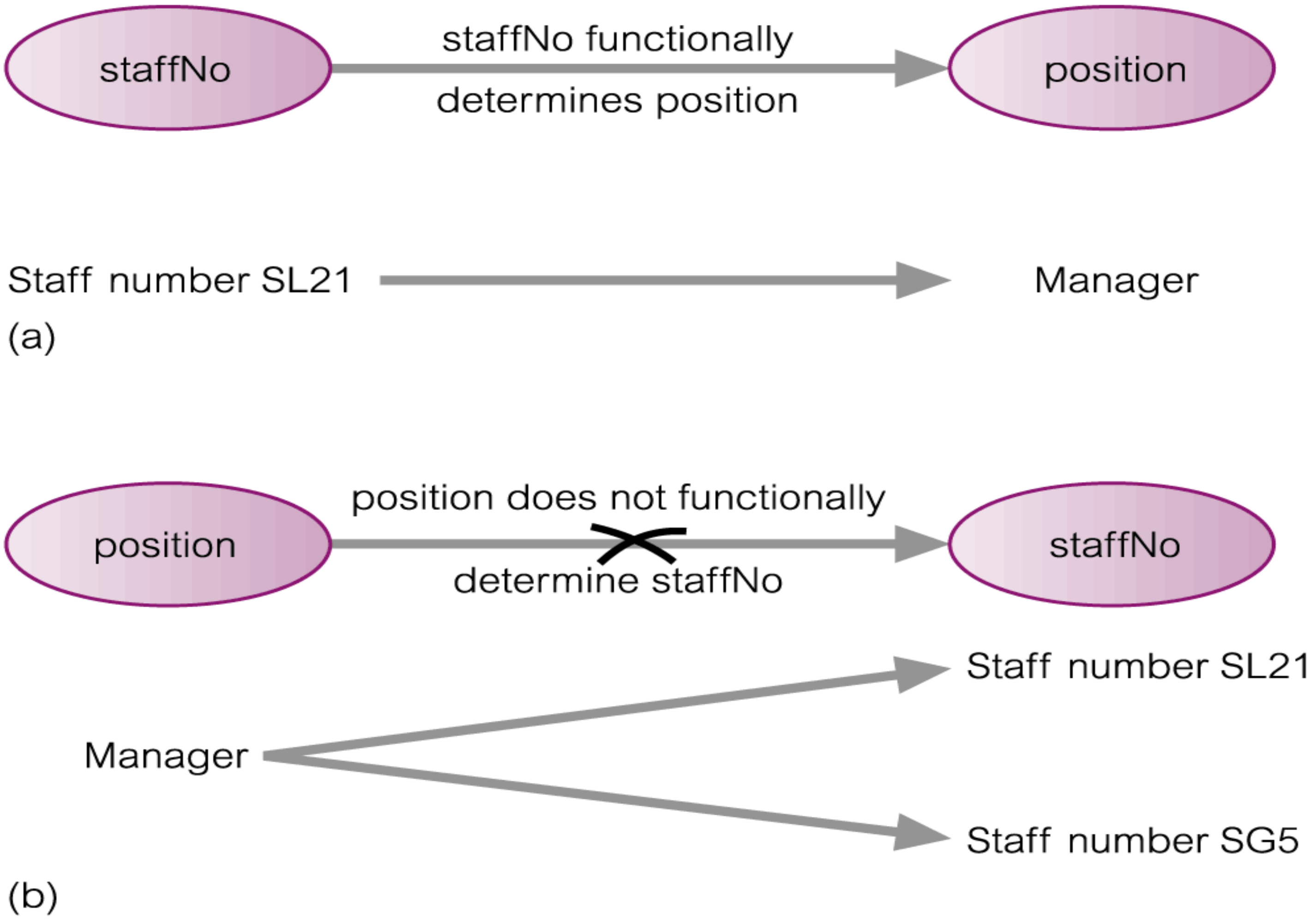
Determinant of a Functional Dependency

- Diagrammatic representation.

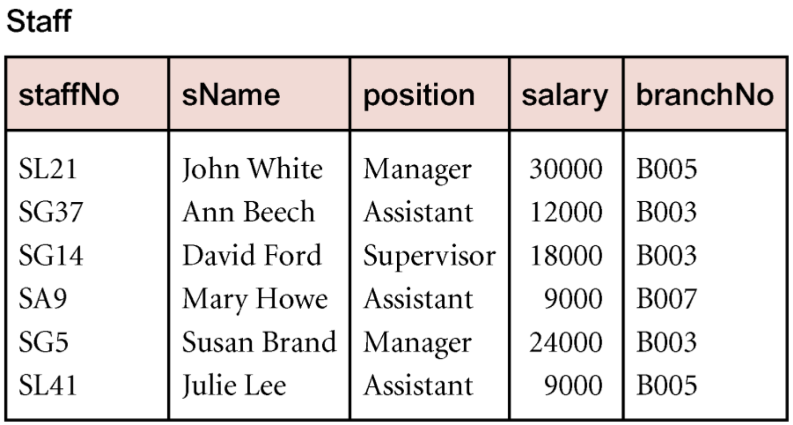


- The ***determinant*** of a functional dependency refers to the attribute or group of attributes on the left-hand side of the arrow.

Examples of Functional Dependency



Examples of Functional Dependency



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

- In the Staff relation, there are functional dependencies:

(staffNo, sName) → branchNo

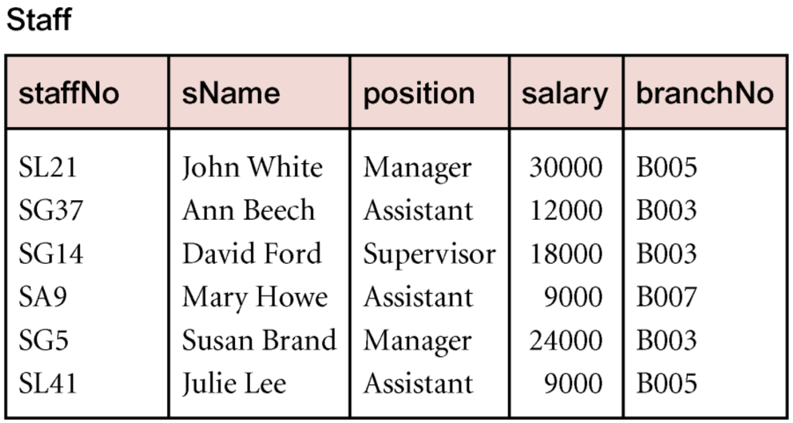
staffNo → branchNo

  - each value of (staffNo, sName) is associated with a single value of branchNo.
  - branchNo is also functionally dependent on a subset of (staffNo, sName), namely staffNo.

Full and Partial Functional Dependency

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

Full and Partial Functional Dependency



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

- In the Staff relation, there are functional dependencies:

(staffNo, sName) → branchNo

staffNo → branchNo

  - (staffNo, sName) → branchNo is a partial functional dependency
  - staffNo → branchNo is a full functional dependency

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 (provided that A is not functionally dependent on B or C).

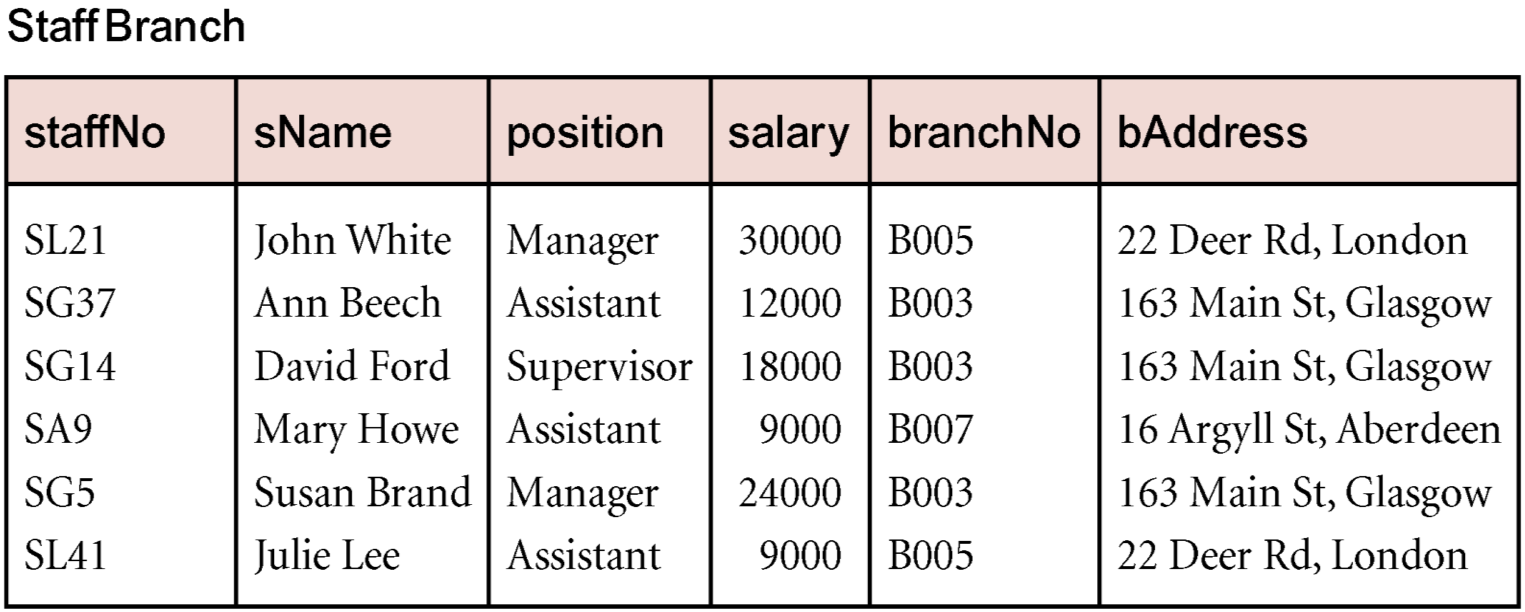
Example of Transitive Dependency

- Consider functional dependencies in the StaffBranch

  staffNo → branchNo

  branchNo → bAddress

  - The transitive dependency staffNo → bAddress exists via the branchNo attribute.



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

Agenda

1. Why Normalize?
2. Functional Dependencies
3. First Normal Form (1NF)
4. Second Normal Form (2NF)
5. Third Normal Form (3NF)
6. The Process of Normalization

UNF and 1NF

- **Unnormalized Form (UNF)**: A table that contains one or more repeating groups.

EMPLOYEES

eid	ename		position	dependents
	lname	fname		
e001	Smith	John	Agent	Michael J.
				Susan R.
e002	Andrews	David	Superintendent	David M. Jr.
e003	Jones	Franklin	Agent	Andrew K.
				Mark W.
				Louisa M.

UNF

- **First Normal Form (1NF)**: A relation in which the intersection of each tuple and attribute contains one and only one value.

Change UNF to 1NF

a) Nominate an attribute or group of attributes to act as the key for the unnormalized table.

b) Identify the repeating group(s) in the unnormalized table which repeats for the key attribute(s).

c) Remove the repeating group by

① Entering appropriate data into the empty columns of rows containing the repeating data (‘flattening’ the table).

Or

② Placing the repeating data along with a copy of the original key attribute(s) into a separate relation.

① Example of Change UNF to 1NF

EMPLOYEES

eid	ename		position	dependents
	lname	fname		
e001	Smith	John	Agent	Michael J.
				Susan R.
e002	Andrews	David	Superintendent	David M. Jr.
e003	Jones	Franklin	Agent	Andrew K.
				Mark W.
				Louisa M.

UNF

eid	ename		position	dependents
	lname	fname		
e001	Smith	John	Agent	Michael J.
e001	Smith	John	Agent	Susan R.
e002	Andrews	David	Superintendent	David M. Jr.
e003	Jones	Franklin	Agent	Andrew K.
e003	Jones	Franklin	Agent	Mark W.
e003	Jones	Franklin	Agent	Louisa M.

INF

② Example of change UNF to 1NF

EMPLOYEES

eid	ename		position	dependents
	lname	fname		
e001	Smith	John	Agent	Michael J.
				Susan R.
e002	Andrews	David	Superintendent	David M. Jr.
e003	Jones	Franklin	Agent	Andrew K.
				Mark W.
				Louisa M.

UNF

EMPLOYEES

eid	lname	fname	position
e001	Smith	John	Agent
e002	Andrews	David	Superintendent
e003	Jones	Franklin	Agent

DEPENDENTS

eid	dependent
e001	Michael J.
e001	Susan R.
e002	David M. Jr.
e003	Andrew K.
e003	Mark W.
e003	Louisa M.

INF

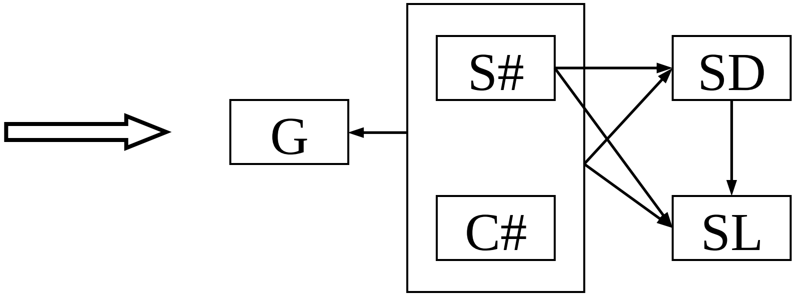
Example of 1NF

- Relational Model: S-L-C(S#，SD，SL，C#，G),
- Functional dependencies in S-L-C

$S\# \rightarrow SD,$ $\quad (S\#, C\#) \xrightarrow{F} G$

$S\# \rightarrow SL,$ $\quad (S\#, C\#) \xrightarrow{P} SD$

$SD \rightarrow SL,$ $\quad (S\#, C\#) \xrightarrow{P} SL$



- Key is（S#，C#）
- S-L-C(S#,SD,SL,C#,G)∈1NF

Agenda

1. Why Normalize?
2. Functional Dependencies
3. First Normal Form (1NF)
4. Second Normal Form (2NF)
5. Third Normal Form (3NF)
6. The Process of Normalization

Second Normal Form (2NF)

- Based on the concept of full functional dependency.

- A relation that is in 1NF and every non-primary-key attribute is fully functionally dependent on the primary key.

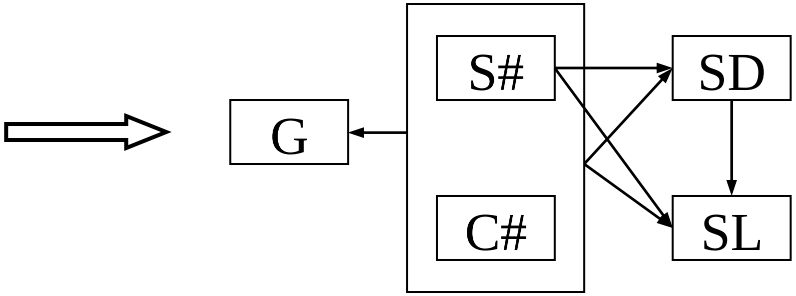
Example of Not 2NF

- S-L-C(S#，SD，SL，C#，G) ∈1NF

S# → SD,　　$(S\#，C\#) \xrightarrow{F} G$

S# → SL,　　$(S\#，C\#) \xrightarrow{P} SD$

SD → SL,　　$(S\#，C\#) \xrightarrow{P} SL$



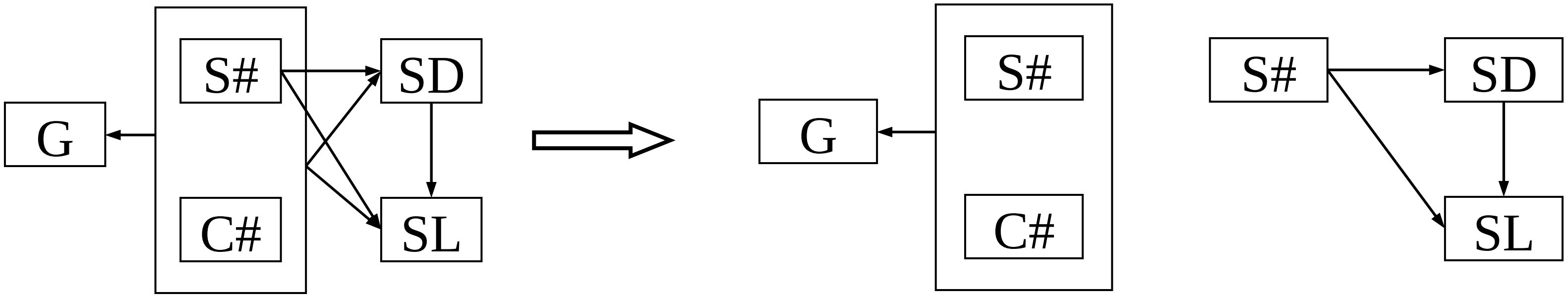
- Key is（S#，C#）
- Exists partial functional dependencies:（S#，C#)⟶ SD, (S#，C#)⟶ SL
- S-L-C(S#,SD,SL,C#,G) ∉ 2NF

Change 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.

Example of Change 1NF to 2NF

- S-L-C(S#, SD, SL, C#, G) ∈1NF



- (S#, C#) is the key of S-C
- S# is the key of S-L
- No partial functional dependencies in S-C or S-L
- S-C ∈ 2NF ; S-L ∈ 2NF

Agenda

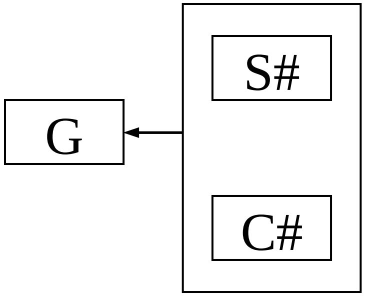
1. Why Normalize?
2. Functional Dependencies
3. First Normal Form (1NF)
4. Second Normal Form (2NF)
5. Third Normal Form (3NF)
6. The Process of Normalization

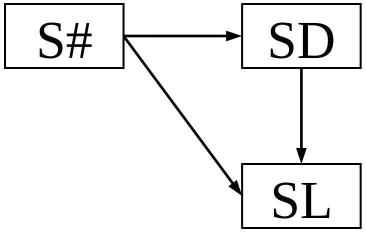
Third Normal Form (3NF)

- Based on the concept of transitive dependency.
- A relation that is in 1NF and 2NF and in which no non-primary-key attribute is transitively dependent on the primary key.

Example of Not 3NF

- S-C ∈ 2NF ; S-L ∈ 2NF





- (S#, C#) is the key of S-C ,
- S# is the key of S-L
- No transitive dependency in S-C, then S-C ∈ 3NF ;
- Transitive dependency S# → SL in S-L, then S-L ∉ 3NF

Change 2NF to 3NF

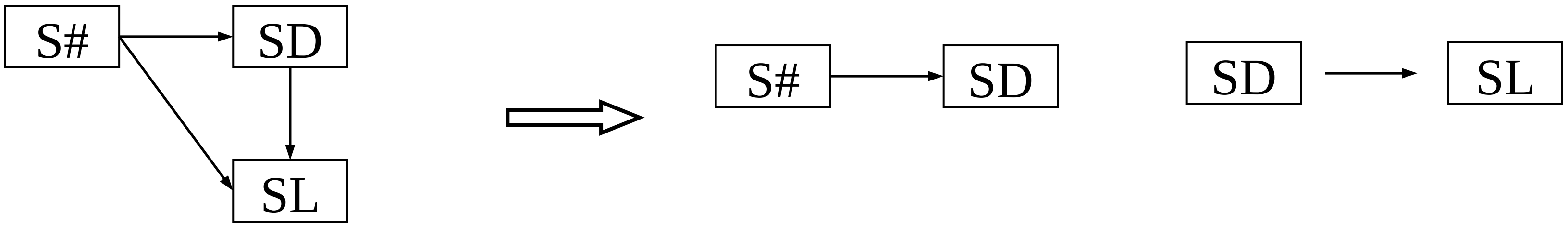
- Identify the primary key in the 2NF relation.

- 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.

Example of Change 2NF to 3NF

- S-L ∈ 2NF, but S-L ∉ 3NF



- S－D （S#， SD）∈3NF ； D－L（SD， SL）∈3NF.

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.***

Agenda

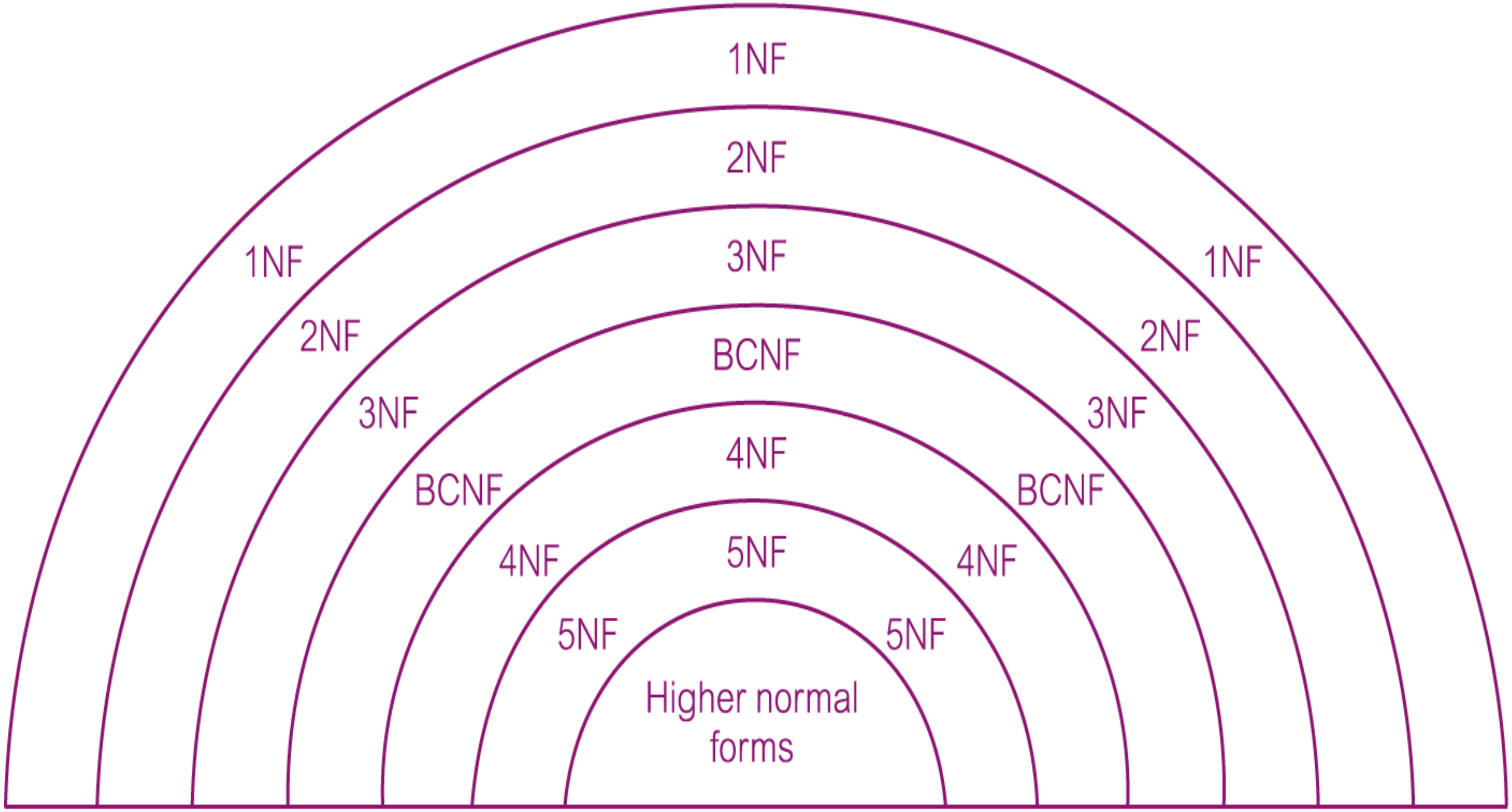
1. Why Normalize?
2. Functional Dependencies
3. First Normal Form (1NF)
4. Second Normal Form (2NF)
5. Third Normal Form (3NF)
6. The Process of Normalization

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.

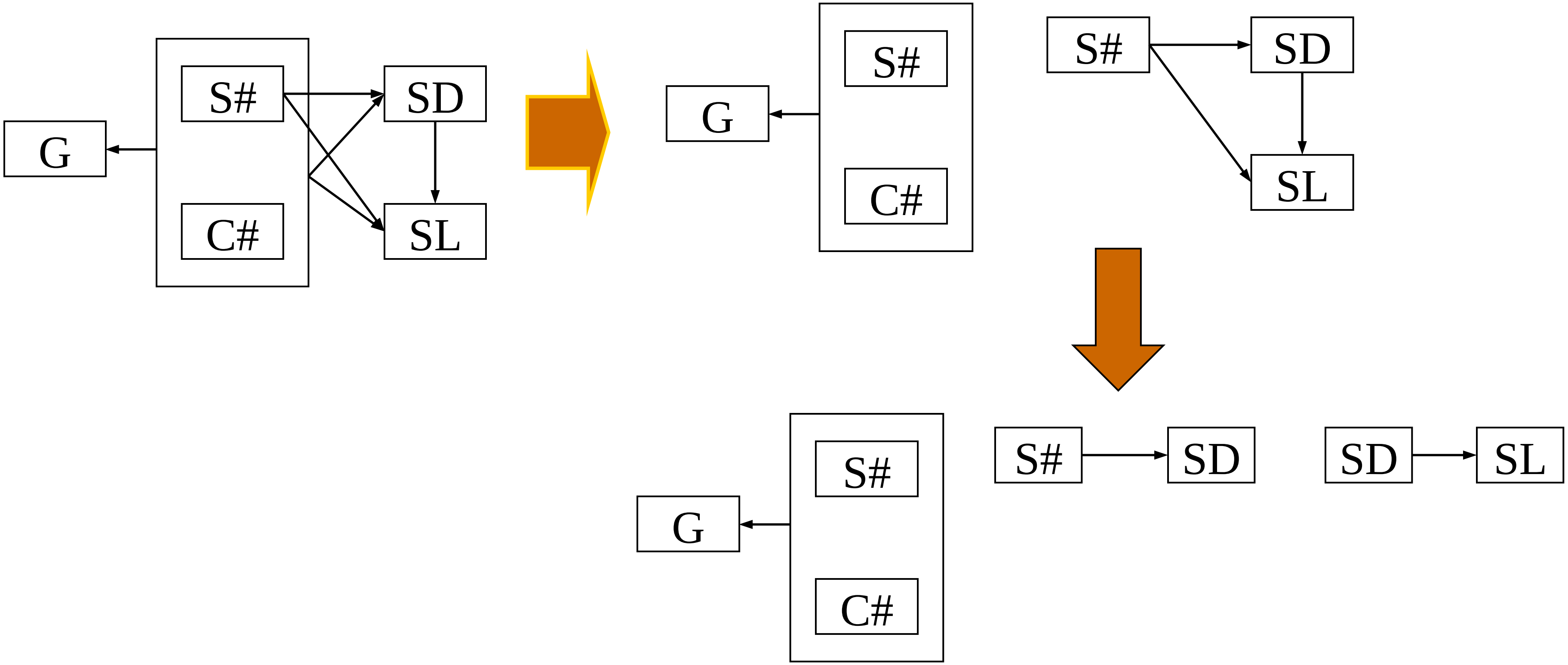
The Process of Normalization



An Example of the Process of Normalization

1NF

2NF



3NF

Properties of Decomposition

Assumed R(A, F) is decomposed to $R_1(A_{1,}\ F_1)$, $R_2(A_{2,}\ F_2)$, … , $R_n(A_{n,}\ F_n)$:

① Attributes equivalent

$$A= A_1 \cup A_2 \cup \ldots \cup A_n$$

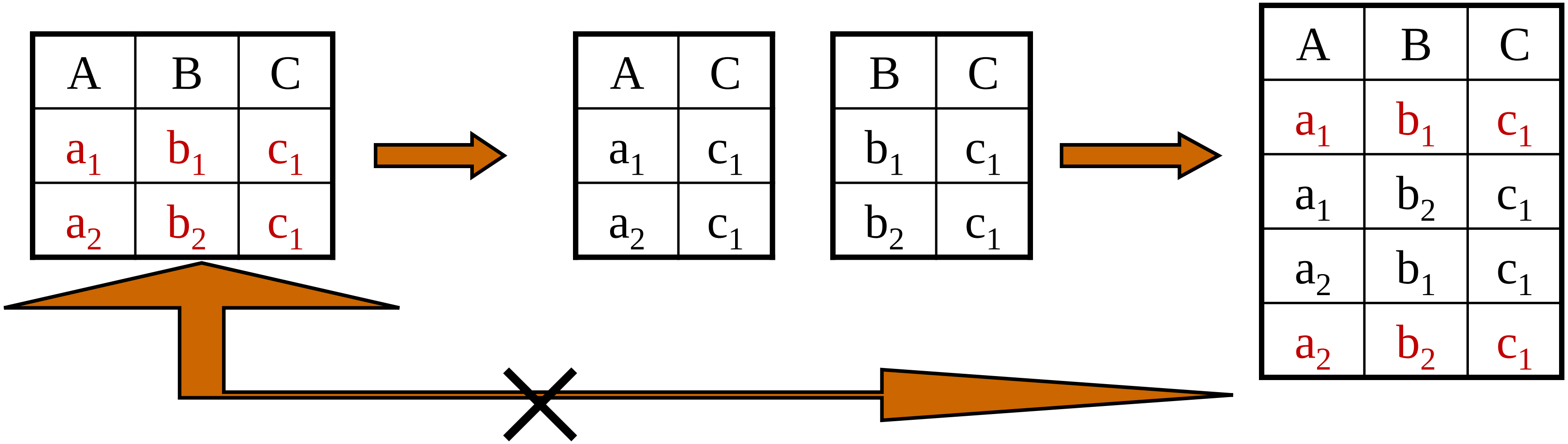
② Lossless-join property

$$R= R_1 \bowtie R_2 \bowtie \ \ldots \bowtie R_n$$

③ Dependency preservation

enables us to enforce a constraint on the original relation by enforcing some constraint on each of the smaller relations.

Lossless-join Property



A	B	C
a_1	b_1	c_1
a_2	b_2	c_1

A	C
a_1	c_1
a_2	c_1

B	C
b_1	c_1
b_2	c_1

A	B	C
a_1	b_1	c_1
a_1	b_2	c_1
a_2	b_1	c_1
a_2	b_2	c_1

Section Objectives

In this section you will learn:

① The purpose of normalization.

② The potential problems associated with redundant data in base relations.

③ The concept of functional dependency, which describes the relationship between attributes.

④ How to undertake the process of normalization.

⑤ How to identify 1NF, 2NF, 3NF .

⑥ The problems associated with relations that break the rules of 1NF, 2NF, or 3NF.

Questions?

Assignments

- Multiple-Choice Quiz 6
- Exercise 6

Prerequisites for Next Section

- **Readings:**
  - **Required:** Connolly and Begg, sections 20.1, 20.2.1, and 20.2.2 (in third edition, sections 19.1, 19.2.1, and 19.2.2).
  - **Required:** Connolly and Begg, sections 20.2.3 and 20.2.4 (in third edition, sections 19.2.3 and 19.2.4).
  - **Required:** Connolly and Begg, section 20.3.1–20.3.4 (in third edition, section 19.3).
- **Assessments**
  - Multiple-Choice Quiz 7