DATA 300
DATA MANAGEMENT

Aye Hninn Khine, Ph.D.,
Visiting Faculty

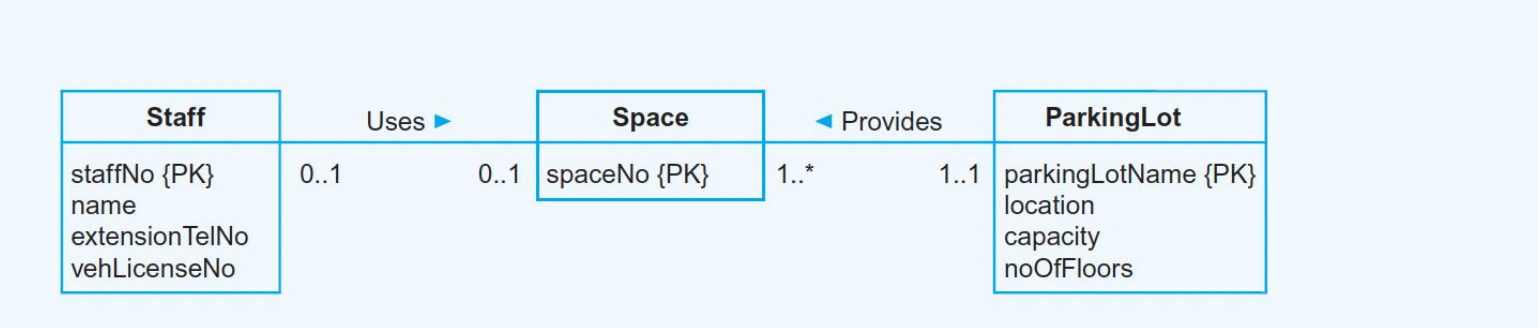
PARAMI
UNIVERSITY

Mid-Term Examination (Oct 30th, 2025 7:00 PM to 10:00 Thailand Time)

- MCQ - 10 questions
- Short Q & A OR drawing regarding ER database design
- Database DDL (with pgAdmin or psql)
- Database DML - 4 to 5 questions (Easy to medium)

- (a) A large organization has several parking lots, which are used by staff.
- (b) Each parking lot has a unique name, location, capacity, and number of floors (where appropriate).
- (c) Each parking lot has parking spaces, which are uniquely identified using a space number.
- (d) Members of staff can request the sole use of a single parking space. Each member of staff has a unique number, name, telephone extension number, and vehicle license number.
- (e) Represent all the ER models described in parts (a), (b), (c), and (d) as a single ER model. Provide any assumptions necessary to support your model.

ER Diagram Assignment Solution



Staff

staffNo {PK}
name
extensionTelNo
vehLicenseNo

Uses ►

0..1

0..1

Space

spaceNo {PK}

◄ Provides

1..*

1..1

ParkingLot

parkingLotName {PK}
location
capacity
noOfFloors

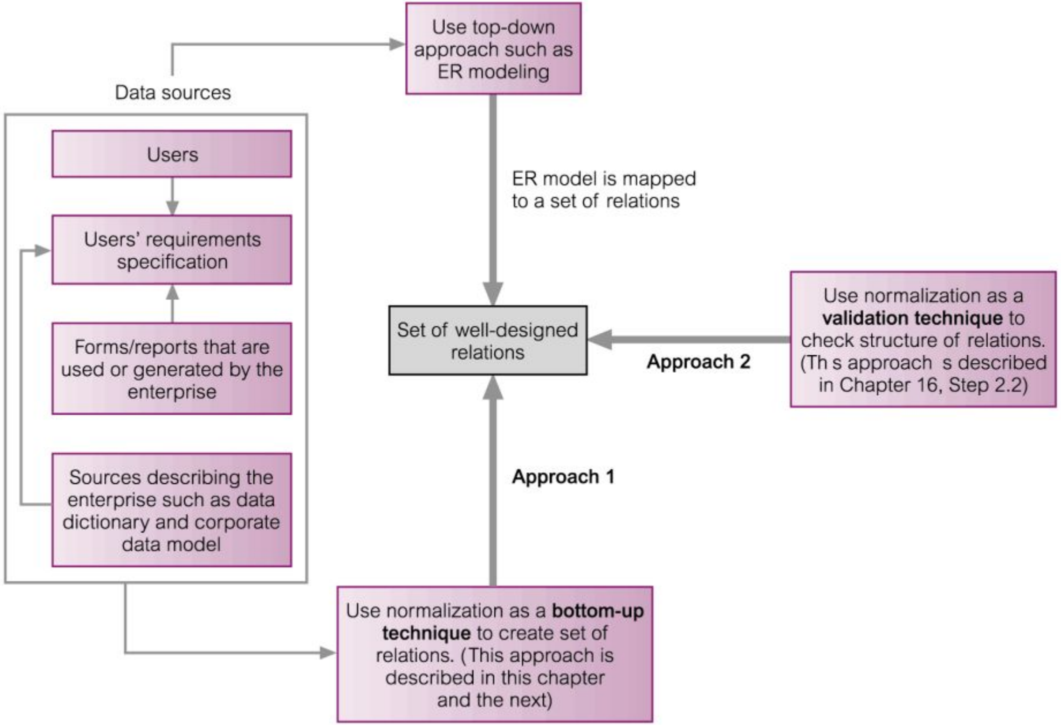
Database Normalization

- In database, normalization is **technique for producing a set of suitable relations** that support the data requirements of an enterprise
- Characteristics of a suitable set of relations includes:
  - **minimal number of attributes** necessary to support the data requirements
  - attributes with a close logical relationship are found in the same relation
  - **minimal redundancy** with each attribute **represented only once** (except attributes which forms a foreign key)

Why normalization?

- The benefits of using a database that is normalized includes:
  - Maintain **data integrity** by reducing duplication and inconsistencies
  - **Reduces the storage space** required for storing data
  - **Easier** for the user to **access and maintain** the data

Normalization in DB design



Data sources

Users

Users' requirements specification

Forms/reports that are used or generated by the enterprise

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

Use top-down approach such as ER modeling

ER model is mapped to a set of relations

Set of well-designed relations

Approach 2

Use normalization as a **validation technique** to check structure of relations. (Th s approach s described in Chapter 16, Step 2.2)

Approach 1

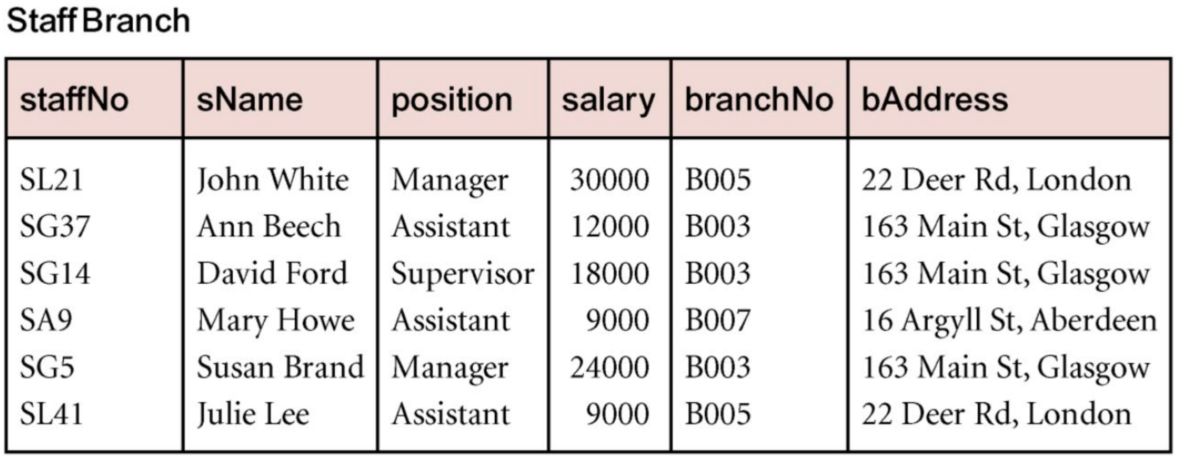
Use normalization as a **bottom-up technique** to create set of relations. (This approach is described in this chapter and the next)

Data Redundancy and Update Anomalies

- Major aim of relational database design is to **group attributes into relations** to **minimize data redundancy**
- Potential benefits for implemented database includes:
  - Updates to the data stored in 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

Consider this StaffBranch Relation



Staff Branch

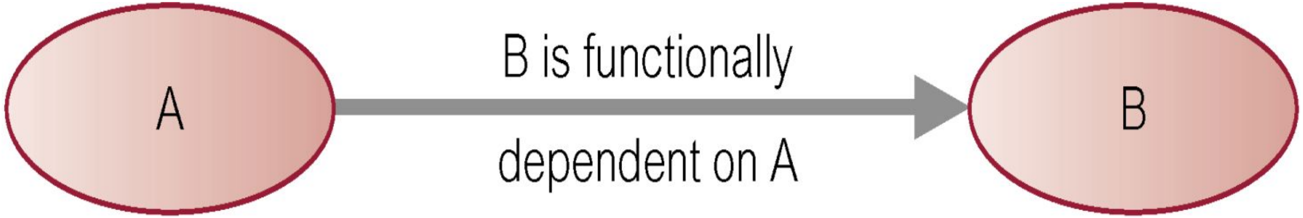
staffNo	sName	position	salary	branchNo	bAddress
SL21	John White	Manager	30000	B005	22 Deer Rd, London
SG37	Ann Beech	Assistant	12000	B003	163 Main St, Glasgow
SG14	David Ford	Supervisor	18000	B003	163 Main St, Glasgow
SA9	Mary Howe	Assistant	9000	B007	16 Argyll St, Aberdeen
SG5	Susan Brand	Manager	24000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

Data Redundancy and Update Anomalies

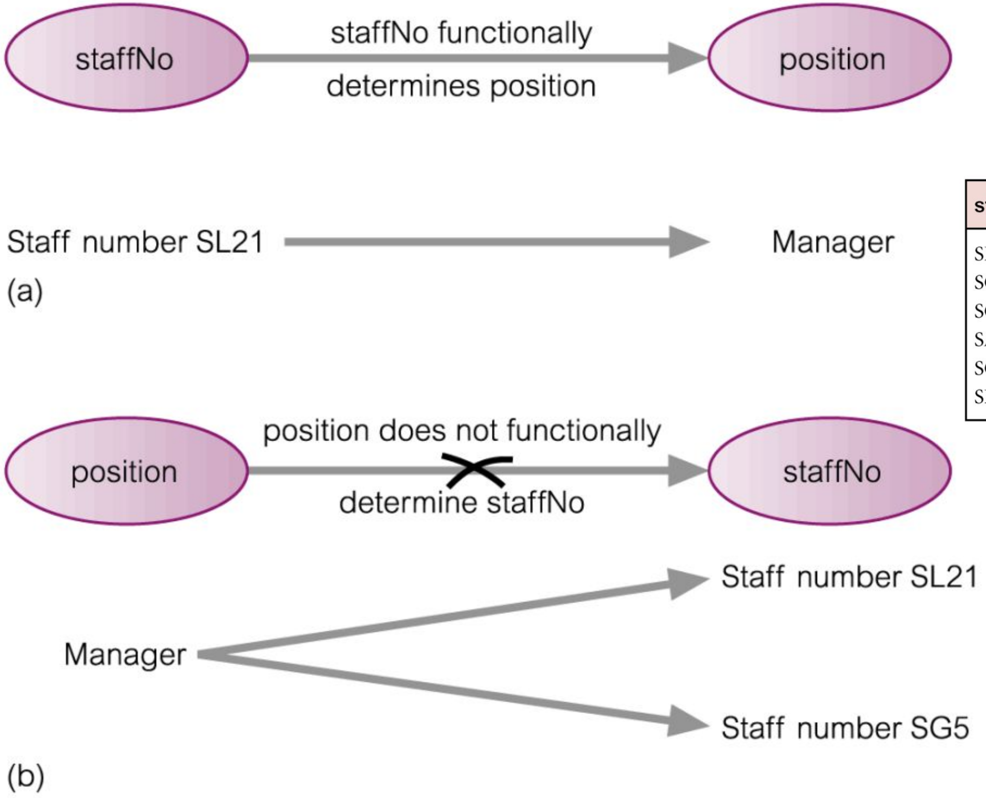
- StaffBranch relation has **redundant information**
  - The **details of a branch are repeated** for every member of staff
- Relations that contain redundant information may potentially suffer from **update anomalies**
  - Insertion
  - Deletion
  - Modification

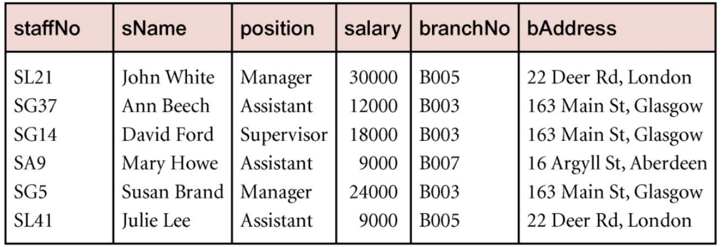
Functional Dependencies

- Functional dependency describes relationship between attributes
  - Let 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
  - The **determinant** of a functional dependency refers to the **attribute(s) on the left-hand side** of the arrow



Functional Dependencies

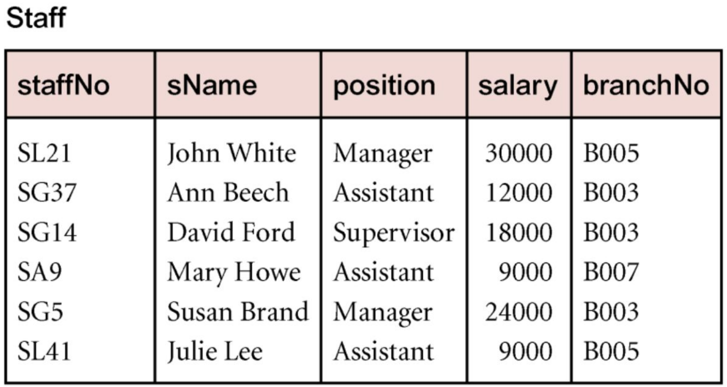




staffNo	sName	position	salary	branchNo	bAddress
SL21	John White	Manager	30000	B005	22 Deer Rd, London
SG37	Ann Beech	Assistant	12000	B003	163 Main St, Glasgow
SG14	David Ford	Supervisor	18000	B003	163 Main St, Glasgow
SA9	Mary Howe	Assistant	9000	B007	16 Argyll St, Aberdeen
SG5	Susan Brand	Manager	24000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

Functional Dependencies

- Consider the values shown in **Staff** table
- Based on sample data, the following functional dependencies appear to hold
  - staffNo → sName
  - sName → staffNo
- 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**



Staff

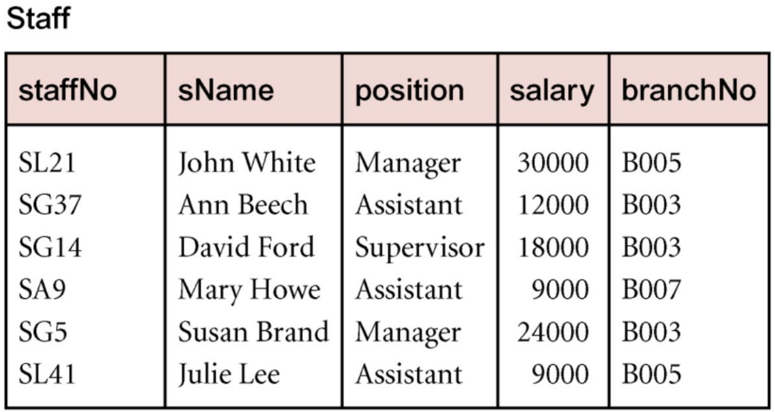
staffNo	sName	position	salary	branchNo
SL21	John White	Manager	30000	B005
SG37	Ann Beech	Assistant	12000	B003
SG14	David Ford	Supervisor	18000	B003
SA9	Mary Howe	Assistant	9000	B007
SG5	Susan Brand	Manager	24000	B003
SL41	Julie Lee	Assistant	9000	B005

Full 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
- 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 any subset of A**

Partial Functional Dependencies

- (staffNo, sName) → branchNo
- Each value of (staffNo, sName) is associated with a single value of branchNo



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

- However, branchNo is also functionally dependent on a subset of (staffNo, sName), namely staffNo. This is called **partial dependency**

Student grade

Student_ID	Course_ID	GPA
P001	C001	3.0
P001	C002	4.0
P002	C001	3.5
P002	C002	2.0
P003	C001	3.2
P003	C002	3.8

PKs ⇒ composite keys ⇒ Student_ID, Course_ID

Student_ID → GPA (not functionally dependent)
Course_ID → GPA (not functionally dependent)
(Student_ID, Course_ID) → GPA ⇒ full functional dependency

Student Information

SID	SName	Total_GPA
P001	Aye	3.0
P002	Hninn	3.0
P003	Khine	3.0

SID → Total_GPA

SID → SName, Total_GPA (Full functional dependencies)

Characteristics of Full 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 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

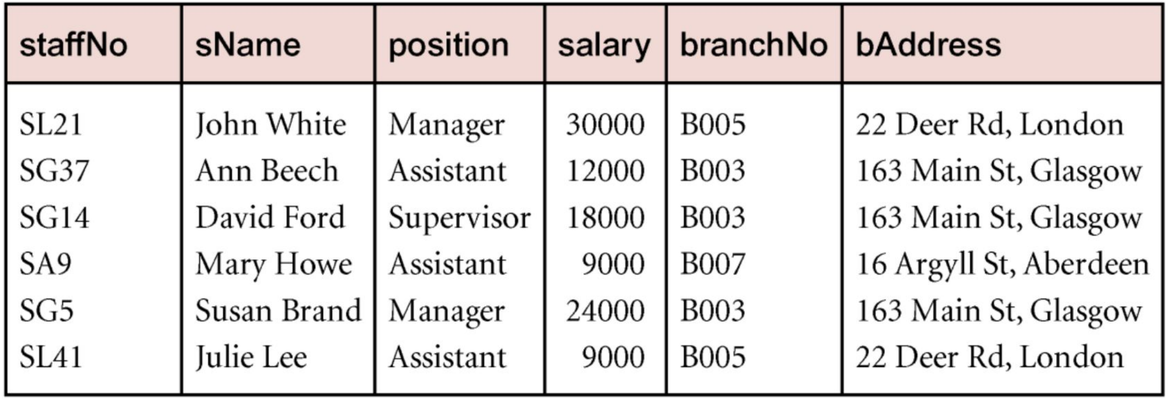
Transitive Dependencies

- Transitive dependencies can **potentially cause update anomalies**
- It 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** (if A is not functionally dependent on B or C)

Transitive Dependencies

- Consider functional dependencies in the **StaffBranch** table
  - staffNo → sName, position, salary, branchNo, bAddress
  - branchNo → bAddress
- Transitive dependency, **branchNo → bAddress** exists on **staffNo** via **branchNo**

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

Transitive Dependency (Student_Course)

SID	SName	CourseID (FK)	CourseName
P001	Aye	C001	Database
P002	Hninn	C001	Database
P003	Khine	C003	Machine Learning

SID as PK

SID → SName, CourseID, CourseName (Full Dependency)

CourseID → CourseName (Transitive Dependency)

Course
Course ID Coursename

Normalization Process

- 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

Identifying Functional Dependencies

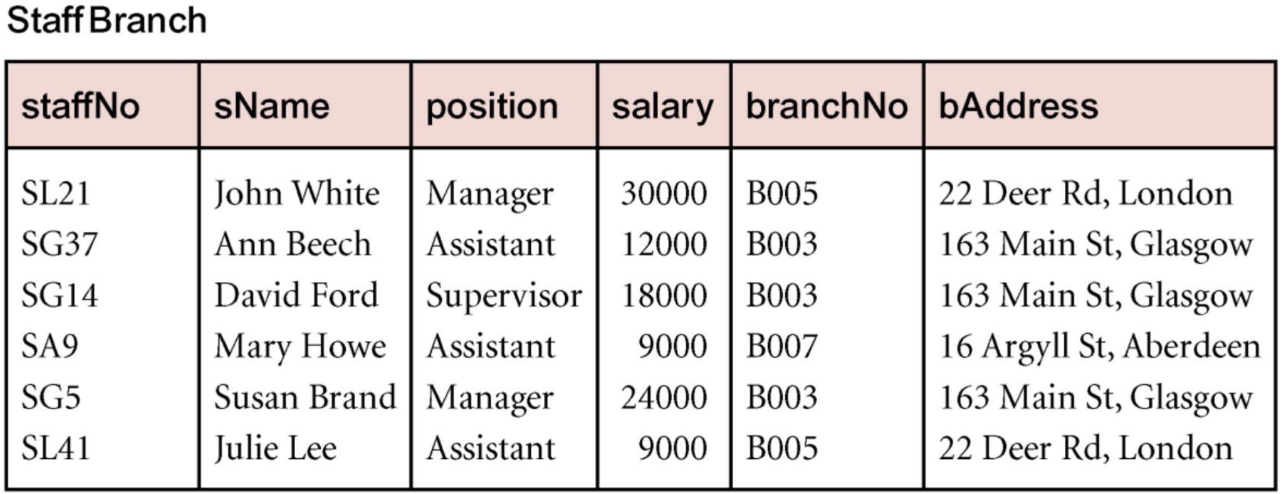
- Identifying all functional dependencies between a set of attributes is relatively simple if the meaning of each attribute and the relationships between the attributes are well understood
- This information should be provided by the enterprise in the form of discussions with users and/or documentation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

Identifying Functional Dependencies

- However, if the users are unavailable for consultation and/or the documentation is incomplete, **it may be necessary for the database designer to use their common sense and/or experience** to provide the missing information

Identifying Functional Dependencies

- Examine semantics of attributes in **StaffBranch** relation



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

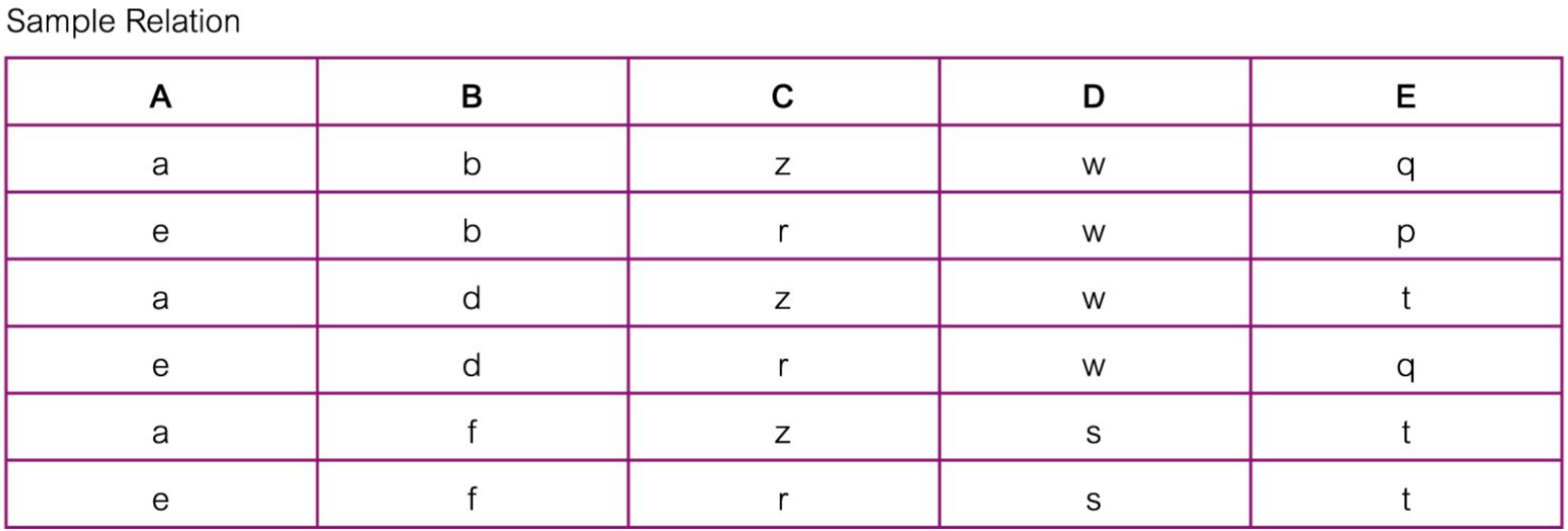
Identifying Functional Dependencies

- With **sufficient information** available, the functional dependencies of the **StaffBranch** relation includes:
  - staffNo → sName, position, salary, branchNo, bAddress
  - branchNo → bAddress
  - bAddress → branchNo
  - branchNo, position → salary
  - bAddress, position → salary

Using samples to identify functional dependencies

- Consider the data for attributes denoted A, B, C, D, and E in the Sample relation
- Important to establish that sample data values shown in relation are representative of all possible values that can be held by attributes A, B, C, D, and E

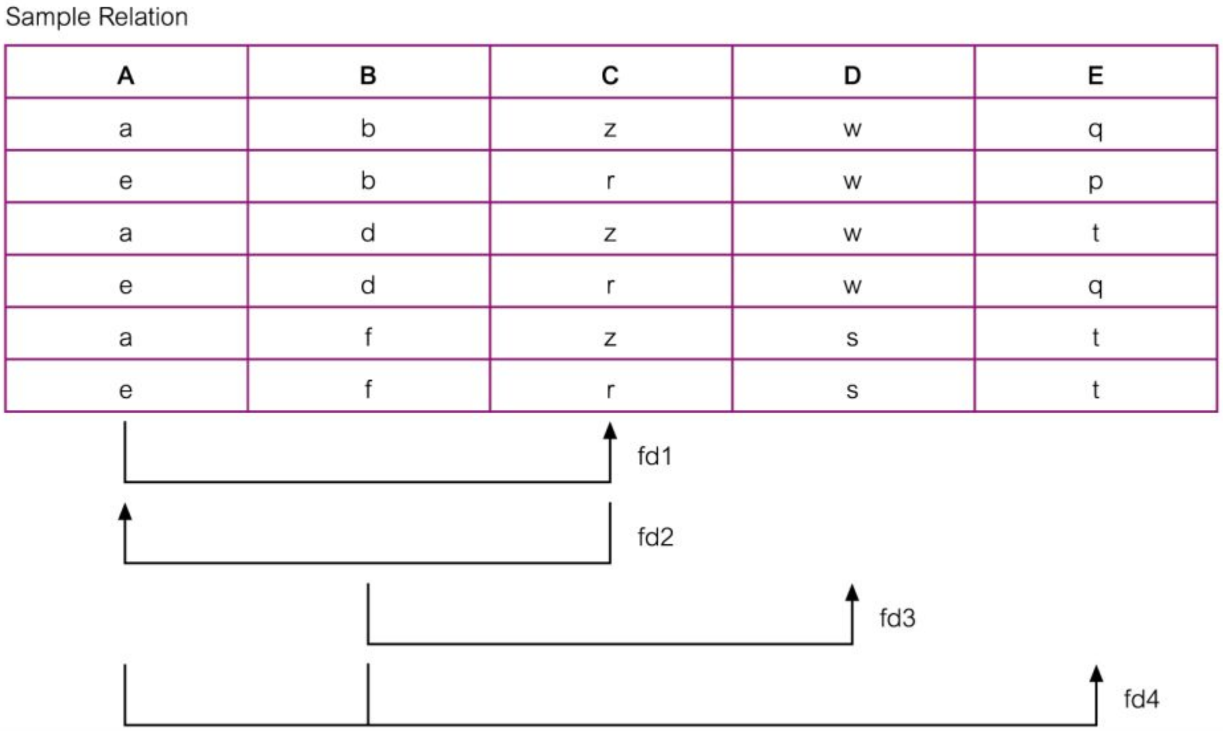
Using samples to identify functional dependencies



Sample Relation

A	B	C	D	E
a	b	z	w	q
e	b	r	w	p
a	d	z	w	t
e	d	r	w	q
a	f	z	s	t
e	f	r	s	t

Using samples to identify functional dependencies



Sample Relation

A	B	C	D	E
a	b	z	w	q
e	b	r	w	p
a	d	z	w	t
e	d	r	w	q
a	f	z	s	t
e	f	r	s	t

$A \rightarrow C$ (fd1)

$C \rightarrow A$ (fd2)

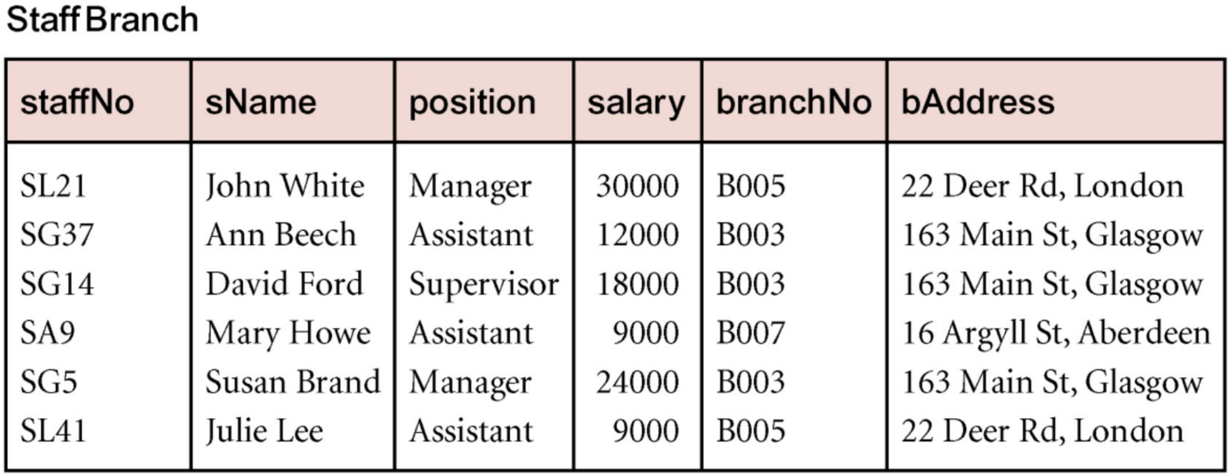
$B \rightarrow D$ (fd3)

$A,B \rightarrow E$ (fd4)

Identifying Primary Key using Functional Dependencies

- Main purpose of identifying a set of functional dependencies for a relation is to **specify the set of integrity constraints that must hold** on a relation
- An important integrity constraint to consider first is the identification of **candidate keys**, one of which is **selected to be the primary key** for the relation

Identifying Primary Key using Functional Dependencies



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

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

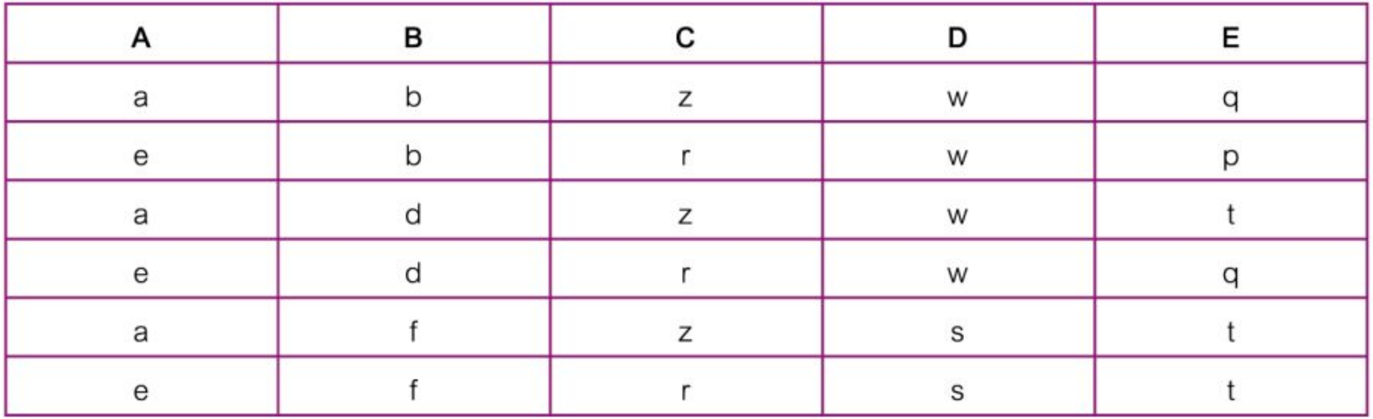
Identifying Primary Key using Functional Dependencies

- **StaffBranch** relation has five functional dependencies
- The **determinants** are staffNo, branchNo, bAddress, (branchNo, position), and (bAddress, position)
- To identify all candidate key(s), identify the attribute(s) that uniquely identifies each tuple in this relation

Identifying Primary Key using Functional Dependencies

- All attributes that are not part of a candidate key should be **functionally dependent** on the key
- The only candidate key and therefore primary key for StaffBranch relation, is **staffNo**, as all other attributes of the relation are functionally dependent on staffNo

Identifying Primary Key using Functional Dependencies



Sample Relation

A	B	C	D	E
a	b	z	w	q
e	b	r	w	p
a	d	z	w	t
e	d	r	w	q
a	f	z	s	t
e	f	r	s	t

A → C (fd1)

C → A (fd2)

B → D (fd3)

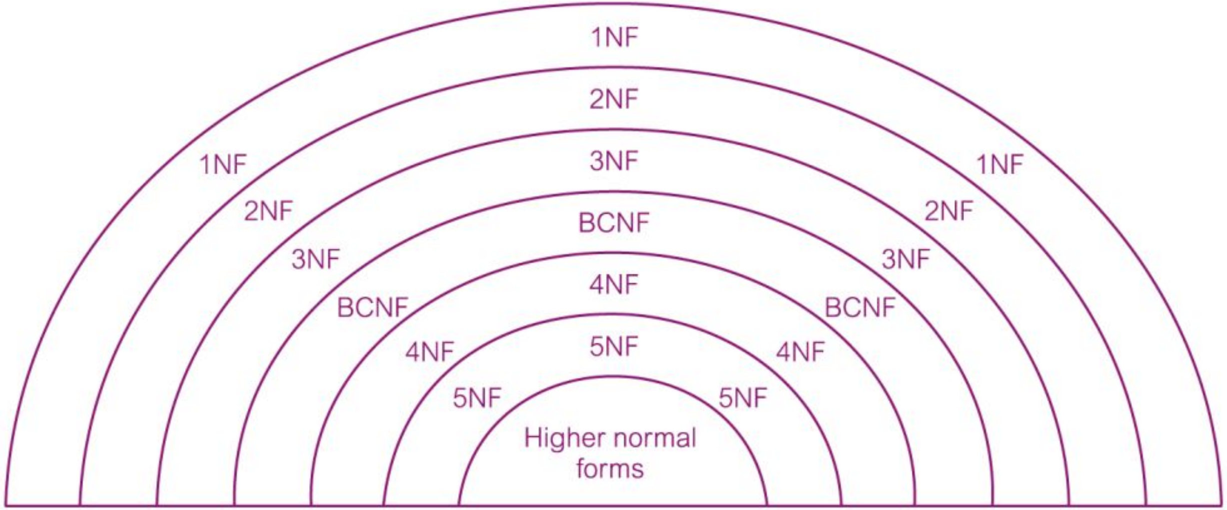
A,B → E (fd4)

Identifying Primary Key using Functional Dependencies

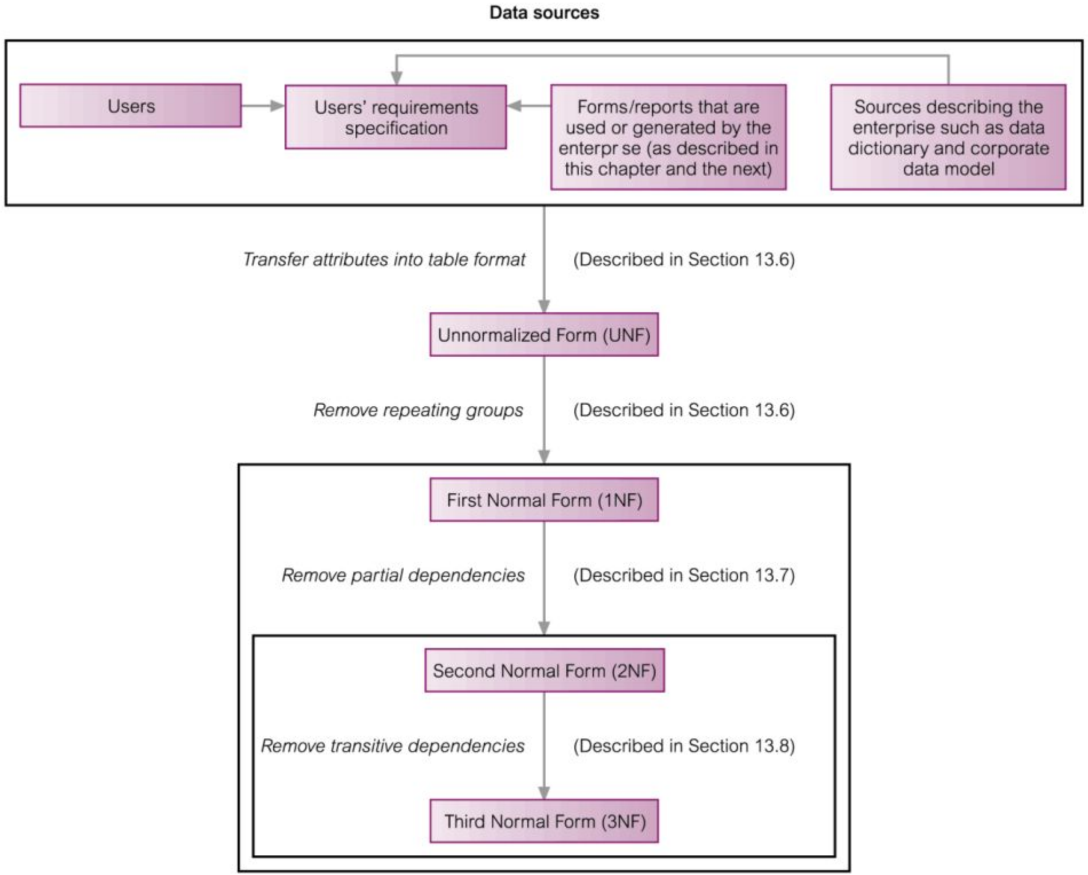
- The determinants in the Sample relation are A, B, C, and (A, B). However, the only **determinant that functionally determines all the other attributes** of the relation is **(A, B)**
- **(A, B)** is identified as the **primary key** for this relation

Normalization Process

- As normalization proceeds, the relations become **progressively more restricted (stronger)** in format and **less vulnerable to update anomalies**



Normalization Process



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

clientNo	cName	propertyNo	pAddress	rentStart	rentFinish	rent	ownerNo	oName
CR76	Alan Flynn	PG04	6 Lawrence St, Glasgow	1-Jul-15	30-Jun-16	350	CO40	Alicia Nelson
		PG16	5 Novar Dr, Glasgow	30-Jul-16	10-Aug-17	450	CO19	Tony Harper
CR87	Casey Palmer	PG04	6 Lawrence St, Glasgow	5-Jul-16	31-Aug-17	350	CO40	Alicia Nelson
		PG36	9 High St, Aberdeen	1-Sep-17	31-Oct-18	425	CO19	Tony Harper
		PG16	5 Novar Dr, Glasgow	31-Oct-18	15-Mar-20	450	CO19	Tony Harper

First Normal Form (INF)

- A relation in which the intersection of each row and column contains **one and only one value**
- Normalizing UNF to 1NF
  - **Nominate an attribute(s) 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 the repeating group** by
    - Entering appropriate data into the empty columns of rows containing the repeating data (‘**flattening**’ the table)
    - Placing the repeating data along with a copy of the original key attribute(s) into a **separate relation**

First Normal Form (INF)

clientNo	cName	propertyNo	pAddress	rentStart	rentFinish	rent	ownerNo	oName
CR76	Alan Flynn	PG04	6 Lawrence St, Glasgow	1-Jul-15	30-Jun-16	350	CO40	Alicia Nelson
CR76	Alan Flynn	PG16	5 Novar Dr, Glasgow	30-Jul-16	10-Aug-17	450	CO19	Tony Harper
CR87	Casey Palmer	PG04	6 Lawrence St, Glasgow	5-Jul-16	31-Aug-17	350	CO40	Alicia Nelson
CR87	Casey Palmer	PG36	9 High St, Aberdeen	1-Sep-17	31-Oct-18	425	CO19	Tony Harper
CR87	Casey Palmer	PG16	5 Novar Dr, Glasgow	31-Oct-18	15-Mar-20	450	CO19	Tony Harper

Second Normal Form (2NF)

- Based on the concept of **full functional dependency**
- A relation that is in 1NF and **every nonprimary-key** attribute is **fully functionally dependent on the primary key**
- Normalizing 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

Second Normal Form (2NF)

clientNo	cName	propertyNo	pAddress	rentStart	rentFinish	rent	ownerNo	oName
CR76	Alan Flynn	PG04	6 Lawrence St, Glasgow	1-Jul-15	30-Jun-16	350	CO40	Alicia Nelson
CR76	Alan Flynn	PG16	5 Novar Dr, Glasgow	30-Jul-16	10-Aug-17	450	CO19	Tony Harper
CR87	Casey Palmer	PG04	6 Lawrence St, Glasgow	5-Jul-16	31-Aug-17	350	CO40	Alicia Nelson
CR87	Casey Palmer	PG36	9 High St, Aberdeen	1-Sep-17	31-Oct-18	425	CO19	Tony Harper
CR87	Casey Palmer	PG16	5 Novar Dr, Glasgow	31-Oct-18	15-Mar-20	450	CO19	Tony Harper

clientNo, propertyNo → cName, pAddress, rentStart, rentFinish, rent, ownerNo, oName

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

Functional Dependencies

fd1 clientNo, propertyNo ® rentStart, rentFinish (Primary key)

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)

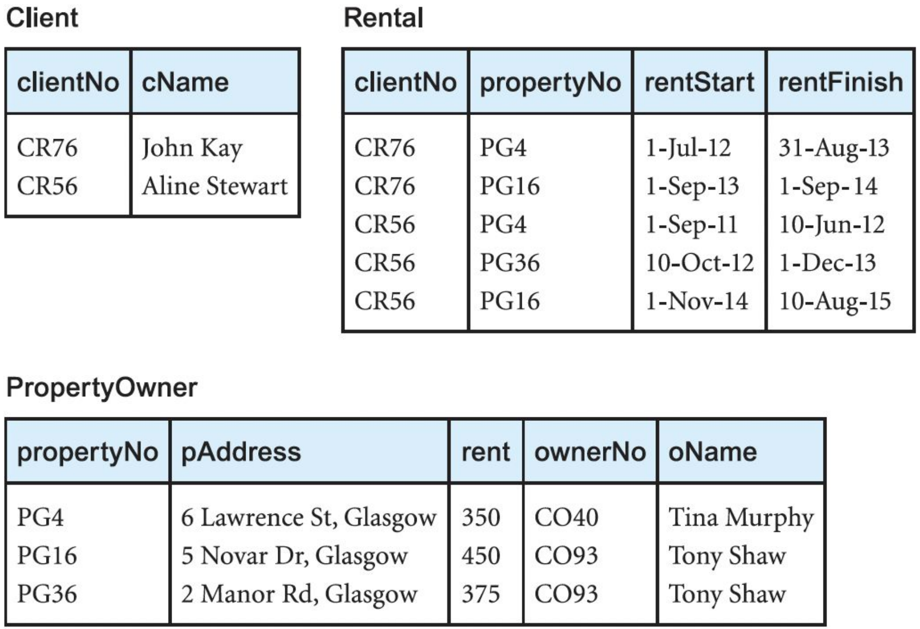
fd6 propertyNo, rentStart ® clientNo, cName, rentFinish (Candidate key)

Second Normal Form (2NF)

Client (**clientNo**, cName)

Rental (**clientNo, propertyNo**, rentStart, rentFinish)

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



Client

clientNo	cName
CR76	John Kay
CR56	Aline Stewart

Rental

clientNo	propertyNo	rentStart	rentFinish
CR76	PG4	1-Jul-12	31-Aug-13
CR76	PG16	1-Sep-13	1-Sep-14
CR56	PG4	1-Sep-11	10-Jun-12
CR56	PG36	10-Oct-12	1-Dec-13
CR56	PG16	1-Nov-14	10-Aug-15

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

Figure 14.14 Second normal form relations derived from the ClientRental relation.

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**
- Normalizing 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 determinant

Functional Dependencies

Client

fd2 clientNo ® cName (Primary key - Full FD)

Rental

fd1 clientNo, propertyNo ® rentStart, rentFinish (Primary key - Full FD)

fd5' clientNo, rentStart ® propertyNo, rentFinish (Candidate key)

fd6' propertyNo, rentStart ® clientNo, rentFinish (Candidate key)

PropertyOwner

fd3 propertyNo ® pAddress, rent, ownerNo, oName (Primary key)

fd4 ownerNo ® oName (Transitive dependency)

Third Normal Form (3NF)

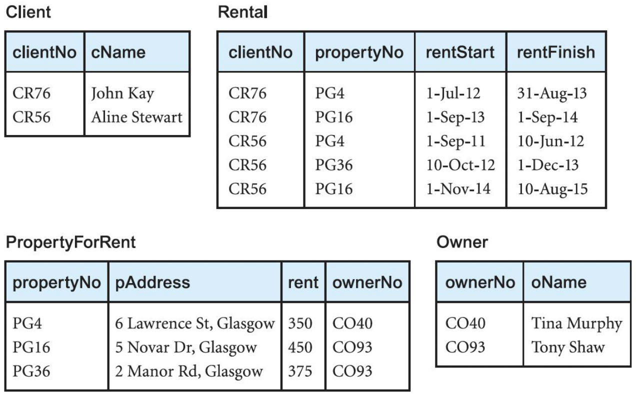
Client (clientNo, cName)

Rental (clientNo, propertyNo, rentStart, rentFinish)

PropertyForRent (propertyNo, pAddress, rent, ownerNo)

Owner (ownerNo, oName)

Figure 14.17 A summary of the 3NF relations derived from the ClientRental relation.



Client

clientNo	cName
CR76	John Kay
CR56	Aline Stewart

Rental

clientNo	propertyNo	rentStart	rentFinish
CR76	PG4	1-Jul-12	31-Aug-13
CR76	PG16	1-Sep-13	1-Sep-14
CR56	PG4	1-Sep-11	10-Jun-12
CR56	PG36	10-Oct-12	1-Dec-13
CR56	PG16	1-Nov-14	10-Aug-15

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

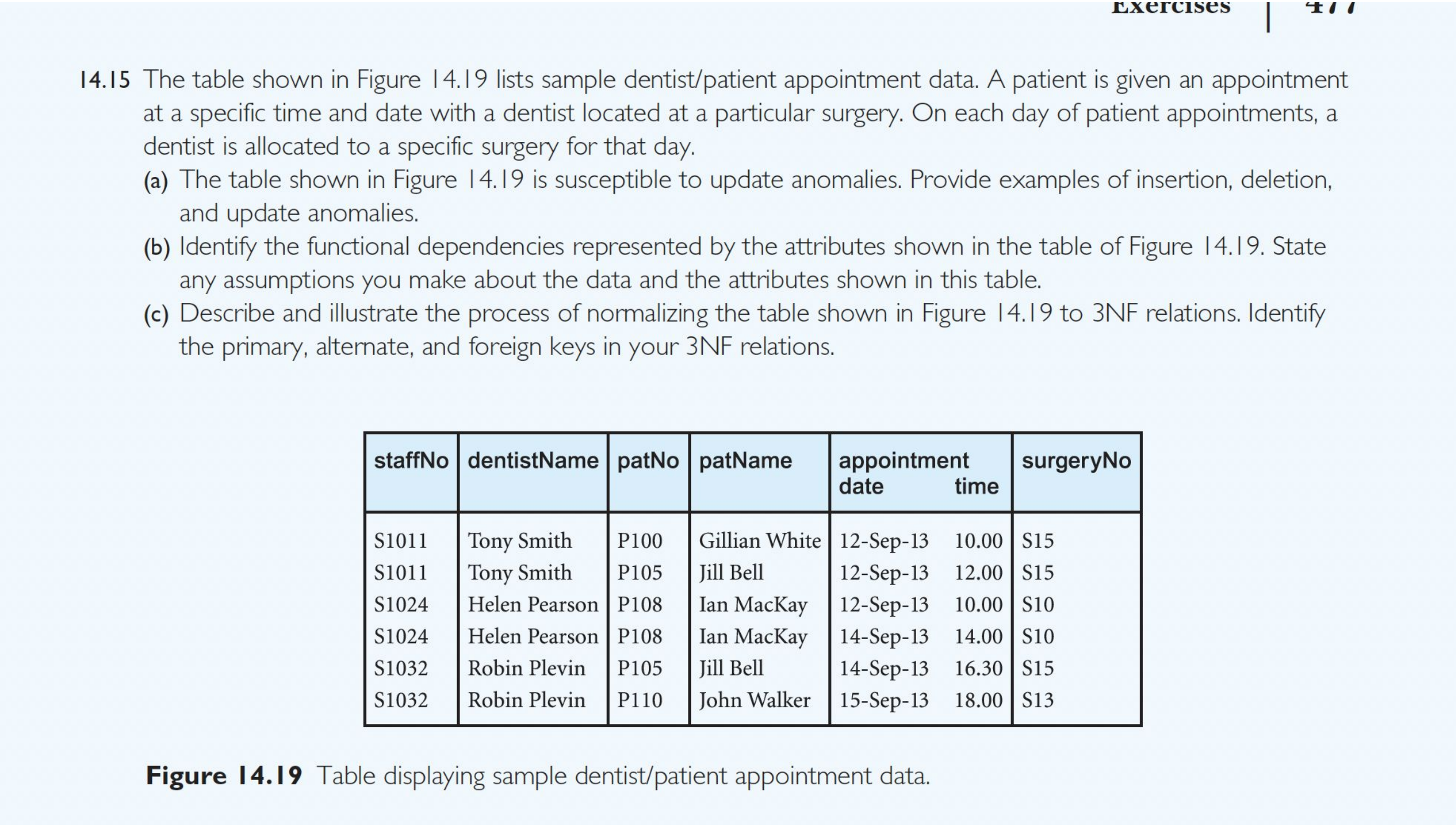
Owner

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**

Exercise 14.15 (Page 477)



14.15 The table shown in Figure 14.19 lists sample dentist/patient appointment data. A patient is given an appointment at a specific time and date with a dentist located at a particular surgery. On each day of patient appointments, a dentist is allocated to a specific surgery for that day.

(a) The table shown in Figure 14.19 is susceptible to update anomalies. Provide examples of insertion, deletion, and update anomalies.

(b) Identify the functional dependencies represented by the attributes shown in the table of Figure 14.19. State any assumptions you make about the data and the attributes shown in this table.

(c) Describe and illustrate the process of normalizing the table shown in Figure 14.19 to 3NF relations. Identify the primary, alternate, and foreign keys in your 3NF relations.

staffNo	dentistName	patNo	patName	appointment		surgeryNo
				date	time	
S1011	Tony Smith	P100	Gillian White	12-Sep-13	10.00	S15
S1011	Tony Smith	P105	Jill Bell	12-Sep-13	12.00	S15
S1024	Helen Pearson	P108	Ian MacKay	12-Sep-13	10.00	S10
S1024	Helen Pearson	P108	Ian MacKay	14-Sep-13	14.00	S10
S1032	Robin Plevin	P105	Jill Bell	14-Sep-13	16.30	S15
S1032	Robin Plevin	P110	John Walker	15-Sep-13	18.00	S13

Figure 14.19 Table displaying sample dentist/patient appointment data.

Exercise 14.15 (Page 477)

(a) Update Anomalies

Insertion Anomaly: If a new patient wants to book an appointment but doesn't have a dentist assigned yet, we can't add them to the table because all fields are required.

Deletion Anomaly: If a patient cancels their appointment, deleting that record would also remove the information about the dentist's surgery location for that day.

Update Anomaly: If a dentist changes their surgery location for a specific day, we would need to update multiple records. If we miss any, the data becomes inconsistent.

(b) Functional Dependencies

Assumptions:

- A dentist is assigned to one surgery per day.
- A patient can have multiple appointments with different dentists, but only one appointment per dentist per day.

Functional dependencies:

- `staffNo -> dentistName, surgeryNo`
- `patNo -> patName`
- `staffNo, patNo, appointment date -> time`

Exercise 14.15 (Page 477)

1NF

Appointment (staffNo, appointmentDate, appointmentTime, dentistName, patNo, patName, surgeryNo)

2NF

Appointment (staffNo, appointmentDate, appointmentTime, patNo, patName)

SurgeryReg (staffNo, appointmentDate, surgeryNo)

Dentist (staffNo, dentistName)

3NF

Appointment (staffNo, appointmentDate, appointmentTime, patNo)

SurgeryReg (staffNo, appointmentDate, surgeryNo)

Dentist (staffNo, dentistName)

Patient (patNo, patName)

Thank You

ayehninnkhine@parami.edu.mm