T1	T2
READ (A, t)	
t := t + 100	
WRITE (A, t)	
	READ (A, s)
	s := s * 2
	WRITE (A, s)
READ (B, t)	
t := t + 100	
WRITE (B, t)	
	READ (B, s)
	s := s * 2
	WRITE (B, s)

T1	T2
READ (A, t)	
t := t + 100	
WRITE (A, t)	
READ (B, t)	
t := t + 100	
WRITE (B, t)	
	READ (A, s)
	s := s * 2
	WRITE (A, s)
	READ (B, s)
	s := s * 2
	WRITE (B, s)

T1	T2
	READ (A, s)
	s := s * 2
	WRITE (A, s)
	READ (B, s)
	s := s * 2
	WRITE (B, s)
READ (A, t)	
t := t + 100	
WRITE (A, t)	
READ (B, t)	
t := t + 100	
WRITE (B, t)	

0	0
200	**200**

100	**100**

T1	T2	A	B	t	s
READ (A, t)		0	0	0	---
t := t + 100		0	0	100	---
WRITE (A, t)		100	0	100	---
READ (B, t)		100	0	0	---
t := t + 100		100	0	100	---
WRITE (B, t)		100	100	100	---
	READ (A, s)	100	100	---	100
	s := s * 2			---	200
	WRITE (A, s)	200	100	---	200
	READ (B, s)	200	100	---	100
	s := s * 2	200		---	200
	WRITE (B, s)	200	200	---	200
FINAL STATE		**200**	**200**	---	---

T1	T2	A	B	t	s
READ (A, t)		0	1	0	---
t := t + 100		0	1	100	---
WRITE (A, t)		100	1	100	---
READ (B, t)		100	1	1	---
t := t + 100		100	1	101	---
WRITE (B, t)		100	101	101	---
	READ (A, s)	100	101	---	100
	s := s * 2		101	---	200
	WRITE (A, s)	200	101	---	200
	READ (B, s)	200	101	---	101
	s := s * 2	200	101	---	202
	WRITE (B, s)	200	202	---	202
FINAL STATE		**200**	**202**	---	---

T1	T2	A	B	t	s
	READ (A, s)	0	0	---	0
	s := s * 2			---	0
	WRITE (A, s)	0	0	---	0
	READ (B, s)	0	0	---	0
	s := s * 2			---	0
	WRITE (B, s)	0	0	---	0
READ (A, t)		0	0	0	---
t := t + 100				100	---
WRITE (A, t)		100	0	100	---
READ (B, t)		100	0	0	---
t := t + 100				100	---
WRITE (B, t)		100	100	100	---
FINAL STATE		**100**	**100**	---	---

T1	T2	A	B	t	s
READ (A, t)		0	0	0	
t := t + 100				100	
WRITE (A, t)		100	0	100	
	READ (A, s)	100	0		100
	s := s * 2				200
	WRITE (A, s)	200	0	100	200
READ (B, t)		200	0	0	
t := t + 100				100	
WRITE (B, t)		200	100	100	
	READ (B, s)	200	100		100
	s := s * 2				200
	WRITE (B, s)	200	200		200
FINAL STATE		**200**	**200**		

This MAY be serializable because the state of the DB at the end of the execution matches one of the serial schedule. However, based ONLY on one observation, we cannot derive a general conclusion.

T1	T2	A	B	t	s
READ (A, t)		0	1	0	
t := t + 100				100	
WRITE (A, t)		100	1	100	
	READ (A, s)	100	1		100
	s := s * 2				200
	WRITE (A, s)	200	1	100	200
READ (B, t)		200	1	1	
t := t + 100				101	
WRITE (B, t)		200	101	101	
	READ (B, s)	200	101		101
	s := s * 2				202
	WRITE (B, s)	200	202		202
FINAL STATE		**200**	202		

This MAY be serializable because the state of the DB at the end of the execution matches one of the serial schedule. However, based ONLY on one observation, we cannot derive a general conclusion.

T1	T2	A	B	t	s
READ (A, t)		0	0	0	
t := t + 100				100	
	READ (A, s)				0
	s := s * 2				0
WRITE (A, t)		100		100	
	WRITE (A, s)	0			0
READ (B, t)			0	0	
t := t + 100				100	
WRITE (B, t)			100	100	
	READ (B, s)		100		100
	s := s * 2				200
	WRITE (B, s)		200		200
FINAL STATE		0	200		

This MAY be serializable because the state of the DB at the end of the execution matches one of the serial schedule. However, based ONLY on one observation, we cannot derive a general conclusion.

StudID	CourseID	StudName	CourseName	Grade	FacName	FacPhone
1	PROG2, DBSE2UE	Adams	Prog2, Database	1, 2	Dhungana, Gambi	1234, 1122
2	PROG2	Jones	Prog2	3	Dhungana	1234
3	PROG2	Smith	Prog2	1	Dhungana	1234
4	PROG2, DBSE2UE	Baker	Prog2, Database	3, 1	Dhungana, Gambi	1234, 1122
5	OS	Brown	OperSys	NULL	Torrubiano	1111
6	NULL	Gates	NULL	NULL	NULL	NULL
NULL	FORM	NULL	Formal Methods	NULL	Duck	12333

The above table is susceptible to update anomalies. Provide examples of insertion, deletion, and modification anomalies.

Deletion Anomalies:

Removing tuples causes students to disappear

Removing tuples causes course and faculties to disappear

StudID	CourseID	StudName	CourseName	Grade	FacName	FacPhone
1	PROG2, DBSE2UE	Adams	Prog2, Database	1, 2	Dhungana, Gambi	1234, 1122
2	PROG2	Jones	Prog2	3	Dhungana	1234
3	PROG2	Smith	Prog2	1	Dhungana	1234
4	PROG2, DBSE2UE	Baker	Prog2, Database	3, 1	Dhungana, Gambi	1234, 1122

Functional Dependencies

- StudID → StudName (OK)
- StudID, CourseID → Grade (OK)
- CourseID → CourseName (OK)
- CourseID → FacName (OK)
- FacName → FacPhone (OK)

StudID	CourseID	StudName	CourseName	Grade	FacName	FacPhone
1	PROG2, DBSE2UE	Adams	Prog2, Database	1, 2	Dhungana, Gambi	1234, 1122
2	PROG2	Jones	Prog2	3	Dhungana	1234
3	PROG2	Smith	Prog2	1	Dhungana	1234
4	PROG2, DBSE2UE	Baker	Prog2, Database	3, 1	Dhungana, Gambi	1234, 1122

- StudID → StudName
- StudID, CourseID → Grade
- CourseID → CourseName
- CourseID → FacName
- FacName → FacPhone

- A relation is in 1NF if it contains only atomic values
- Only single-value attributes are allowed

Is this 1NF? No, it is not

StudID	CourseID	StudName	CourseName	Grade	FacName	FacPhone
1	DBSE2UE	Adams	Database	2	Gambi	1122
4	DBSE2UE	Baker	Database	1	Gambi	1122
1	PROG2	Adams	Prog2	1	Dhungana	1234
2	PROG2	Jones	Prog2	3	Dhungana	1234
3	PROG2	Smith	Prog2	1	Dhungana	1234
4	PROG2	Baker	Prog2	3	Dhungana	1234

- StudID → StudName (Ok)
- StudID, CourseID → Grade (Ok)
- CourseID → CourseName (Ok)
- CourseID → FacName (Ok)
- FacName → FacPhone (Ok)

- A relation is in 1NF if it contains only atomic values
- Only single-value attributes are allowed

Is this 1NF? Yes!

StudID	CourseID	StudName	CourseName	Grade	FacName	FacPhone
1	DBSE2UE	Adams	Database	2	Gambi	1122
4	DBSE2UE	Baker	Database	1	Gambi	1122
1	PROG2	Adams	Prog2	1	Dhungana	1234
2	PROG2	Jones	Prog2	3	Dhungana	1234
3	PROG2	Smith	Prog2	1	Dhungana	1234
4	PROG2	Baker	Prog2	3	Dhungana	1234

- *StudID* → **StudName** (missing CourseID, partial dep.) KO
- *StudID, CourseID* → **Grade** (Ok)
- *CourseID* → **CourseName** (missing StudID, partial dep.) KO
- *CourseID* → **FacName** (missing StudID, partial dep.) KO
- **FacName** → **FacPhone**
- *CourseID* → **FacPhone** (derived FD, missing StudID, partial dep) KO

- PK = {StudID, CourseID}

Is this 2NF? NO

- All **non-key attributes** are **fully** functional dependent on the primary key
- There must be no **partial dependency** or **augmentation**
- (Note: if the PK is atomic, the relation is automatically in 2NF)

StudID	StudNam
1	Adams
2	Jones
3	Smith
4	Baker

2NF!

- StudID → StudName (missing CourseID, partial dep.) KO

StudID	CourseID	CourseNa	Grade	FacName	FacPho
1	DBSE2UE	Database	2	Gambi	1122
4	DBSE2UE	Database	1	Gambi	1122
1	PROG2	Prog2	1	Dhungana	1234
2	PROG2	Prog2	3	Dhungana	1234
3	PROG2	Prog2	1	Dhungana	1234
4	PROG2	Prog2	3	Dhungana	1234

2NF? No!

- StudID, CourseID → Grade (Ok)
- CourseID → CourseName, FacName, FacPhone (KO)
- FacName → FacPhone

- PK = {StudID, CourseID}

- All **non-key attributes** are **fully** functional dependent on the primary key
- There must be no **partial dependency** or **augmentation**
- (Note: if the PK is **atomic**, the relation is automatically in 2NF)

StudI	CourseID	Grad
1	DBSE2U	2
4	DBSE2U	1
1	PROG2	1
2	PROG2	3
3	PROG2	1
4	PROG2	3

- StudID, CourseID → Grade

2NF!

CourseID	CourseName	FacName	FacPhone
DBSE2UE	Database	Gambi	1122
PROG2	Prog2	Dhungana	1234

- CourseID → CourseName, FacName, FacPhone (KO)
- FacName → FacPhone

2NF!

- PK = {StudID, CourseID}

- All **non-key attributes** are **fully** functional dependent on the primary key
- There must be no **partial dependency** or **augmentation**
- (Note: if the PK is **atomic**, the relation is automatically in 2NF)

StudI	CourseID	Grad
1	DBSE2U	2
4	DBSE2U	1
1	PROG2	1
2	PROG2	3
3	PROG2	1
4	PROG2	3

CourseID	CourseName	FacName	FacPhone
DBSE2UE	Database	Gambi	1122
PROG2	Prog2	Dhungana	1234

StudID	StudNam
1	Adams
2	Jones
3	Smith
4	Baker

2NF!

- PK = {StudID, CourseID}

StudI	CourseID	Grad
1	DBSE2U	2
4	DBSE2U	1
1	PROG2	1
2	PROG2	3
3	PROG2	1
4	PROG2	3

CourseID	CourseName	FacName	FacPhone
DBSE2UE	Database	Gambi	1122
PROG2	Prog2	Dhungana	1234

- StudID → StudName
- StudID, CourseID → Grade
- CourseID → CourseName
- CourseID → FacName
- FacName → FacPhone
- CourseID → FacPhone

StudID	StudNam
1	Adams
2	Jones
3	Smith
4	Baker

- a non-key attribute cannot be functionally dependent on another non-key attribute

Is this 3NF? Yay/nay?

StudID	CourseID	Grad
1	DBSE2UE	2
4	DBSE2UE	1
1	PROG2	1
2	PROG2	3
3	PROG2	1
4	PROG2	3

3NF!

StudID	StudNam
1	Adams
2	Jones
3	Smith
4	Baker

3NF!

CourseID	CourseName	FacName	FacPhone
DBSE2UE	Database	Gambi	1122
PROG2	Prog2	Dhungana	1234

3NF? No!

- StudID → StudName
- StudID, CourseID → Grade
- CourseID → CourseName
- CourseID → FacName
- **FacName → FacPhone**
- CourseID → FacPhone

- a non-key attribute cannot be functionally dependent on another non-key attribute

CourseID	**CourseName**	**FacName**
DBSE2UE	Database	Gambi
PROG2	Prog2	Dhungana

- CourseID → CourseName
- CourseID → FacName

3NF!

FacName	**FacPhone**
Gambi	1122
Dhungana	1234

- FacName → FacPhone

3NF!

- a non-key attribute cannot be functionally dependent on another non-key attribute

StudI	CourseID	Grad
1	DBSE2U	2
4	DBSE2U	1
1	PROG2	1
2	PROG2	3
3	PROG2	1
4	PROG2	3

Exam

CourseID	CourseName	FacName
DBSE2UE	Database	Gambi
PROG2	Prog2	Dhungana

Course

FacName	FacPhone
Gambi	1122
Dhungana	1234

Faculty

Student

StudID	StudNam
1	Adams
2	Jones
3	Smith
4	Baker

Final Version of the DB