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

- May represent the matriculation number
-

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)

This table is now in 1NF

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

This is not yet in 2NF (despite Grade is OK)

- StudID → StudName (Violate 2NF)
- StudID, CourseID → Grade (OK)
- CourseID → CourseName (Violate 2NF)
- CourseID → FacName (Violate 2NF)
- FacName → FacPhone (I do not care, the left-hand side is not part of the PK)
- *CourseID* → *FacPhone* (Violate 2NF)

- Candidate Key: {StudID, CourseID}
  - Primary Key: {StudID, CourseID}

StudID	StudName
1	Adams
2	Jones
3	Smith
4	Baker

This is yet in 2NF

StudID	CourseID	CourseName	Grade	FacName	FacPhone
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

STEP 2.1

This is not yet in 2NF (despite Grade is OK)

- StudID → StudName (Violate 2NF)
- StudID, CourseID → Grade (OK)
- CourseID → CourseName (Violate 2NF)
- CourseID → FacName (Violate 2NF)
- FacName → FacPhone (I do not care, the left-hand side is not part of the PK)
- *CourseID* → *FacPhone* (Violate 2NF)

- Candidate Key: {StudID, CourseID}
  - Primary Key: {StudID, CourseID}

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

This is in 2NF

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

This is in 2NF

STEP 2.2

- StudID → StudName (Violate 2NF)
- StudID, CourseID → Grade (OK)
- CourseID → CourseName (Violate 2NF)
- CourseID → FacName (Violate 2NF)
- FacName → FacPhone (I do not care, the left-hand side is not part of the PK)
- *CourseID* → *FacPhone* (Violate 2NF)

- Candidate Key: {StudID, CourseID}
  - Primary Key: {StudID, CourseID}

StudID	CourseID	Grade
1	DBSE2UE	2
4	DBSE2UE	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
1	Adams
2	Jones
3	Smith
4	Baker

- StudID → StudName (Violate 2NF)
- StudID, CourseID → Grade (OK)
- CourseID → CourseName (Violate 2NF)
- CourseID → FacName (Violate 2NF)
- FacName → FacPhone (I do not care, the left-hand side is not part of the PK)
- *CourseID* → *FacPhone* (Violate 2NF)

- Candidate Key: {StudID, CourseID}
  - Primary Key: {StudID, CourseID}

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

This is in 3NF

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

This is in not yet in 3NF

StudID	StudName
1	Adams
2	Jones
3	Smith
4	Baker

This is in 3NF

- StudID → StudName (Violate 2NF)
- StudID, CourseID → Grade (OK)
- CourseID → CourseName (Violate 2NF)
- CourseID → FacName (Violate 2NF)
- FacName → FacPhone (I do not care, the left-hand side is not part of the PK)
- *CourseID* → *FacPhone* (Violate 2NF)

- Primary Key: {StudID, CourseID}

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

CourseID	CourseName	FacName
DBSE2UE	Database	Gambi
PROG2	Prog2	Dhungana

FacName	FacPhone
Gambi	1122
Dhungana	1234

StudID	StudName
1	Adams
2	Jones
3	Smith
4	Baker

All of them are in 3NF

- StudID → StudName (Violate 2NF)
- StudID, CourseID → Grade (OK)
- CourseID → CourseName (Violate 2NF)
- CourseID → FacName (Violate 2NF)
- FacName → FacPhone (I do not care, the left-hand side is not part of the PK)
- *CourseID* → *FacPhone* (Violate 2NF)

- Primary Key: {StudID, CourseID}

Exam

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

Course

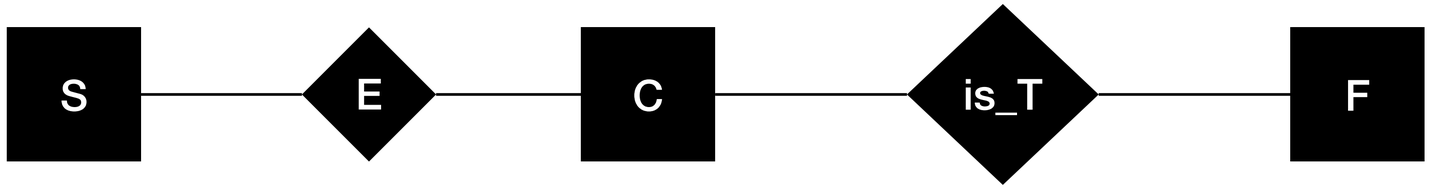
CourseID	CourseName	FacName
DBSE2UE	Database	Gambi
PROG2	Prog2	Dhungana

Faculty

FacName	FacPhone
Gambi	1122
Dhungana	1234

Student

StudID	StudName
1	Adams
2	Jones
3	Smith
4	Baker



A schedule is **serializable** if it is equivalent to a **serial** schedule, which executes all the operations comprising a query before executing the operations that belong to the next query.

Schedule 1

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

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)	

Serial Schedule 1 = (A=200, B=200)

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	100	100		200
	WRITE (A, s)	200	100		200
	READ (B, s)	200	100		100
	s := s * 2	200	100		200
	WRITE (B, s)	200	200		200

Serial Schedule 2 = (A=100, B=100)

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

Target Schedule: Is this serializable?

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			
	READ (B, s)	200	0		0
	s := s * 2				0
	WRITE (B, s)	200	0		
READ (B, t)		200	0	100	
t := t + 100				200	
WRITE (B, t)		200	200		

This is the state of a serial schedule, so the target schedule may be serizaliable