Normalization Stages

- 1. Normal Form
- 2. Normal Form
- 3. Normal Form
- .....

Unnormal Form (UNF)

- According to the basic rule of the relational database model, the values taken by all attributes must be atomic (**single and simple**). Tables that do not comply with this rule are not normal.
- The **DISTRIBUTION** table below does not comply with this rule; therefore, it is not normal.

CustomerID	CityCode	CityName	ShipmentNum	Quantity
1	34	İstanbul	1,2,3,4,6	300,200,400,200,100
2	6	Ankara	1,2	300,400
3	6	Ankara	2	200
4	34	İstanbul	2,4,5	200,300,400

First Normal Form (1NF)

CustomerID	CityCode	CityName	ShipmentNum	Quantity
1	34	İstanbul	1	300
1	34	İstanbul	2	200
1	34	İstanbul	3	250
1	34	İstanbul	6	400
1	34	İstanbul	7	150
2	6	Ankara	1	300
2	6	Ankara	2	400
3	6	Ankara	2	200
4	34	İstanbul	2	200
4	34	İstanbul	4	300
4	34	İstanbul	5	400

- A table in the first form has repeated data in certain areas. For example; **CityCode** and **CityName** data are repeated for each customer. These repetitions will cause insert, delete, and update data problems.
- 1NF is mandatory because a relational query language like SQL cannot handle situations that are not in a 1NF relationship.

Problems of the First Normal Form

- Row Insertion Problem
  - In order to add the **CustomerID**, **CityCode,** and **CodeName** of a customer in another city, for example "**İZMİR** ", a distribution operation must be made for this customer and the values of **ShipmentNum** and **Quantity** must be determined. It is not possible to add records without these values.

CustomerID	CityCode	CityName	ShipmentNum	Quantity
1	34	İstanbul	1	300
…	…	…	…	…
4	34	İstanbul	5	400
5	35	İzmir		

Problems of the First Normal Form

- Row Deletion Problem
  - Suppose a distribution operation is canceled and the related record must be deleted.
  - When a record is deleted, not only the **ShipmentNum** and **Quantity** are deleted, but also the information about the customer is deleted.
  - For example, deleting the row related to the customer who has the «CustomerID = 3» will also delete the customer's **Citycode** and **CityName.**

CustomerID	CityCode	CityName	ShipmentNum	Quantity
1	34	İstanbul	1	300
1	34	İstanbul	2	200
…	…	…	…	…
3	6	Ankara	2	200
…	…	…	…	…

Problems of the First Normal Form

- Update Problem
  - If the location of customer who has the «CustomerID = 1» is moved from **ISTANBUL** to **ANKARA**, due to this new information, 2 records of in the table must be updated.
  - If the table is too large, just such a small change about a client would perhaps create the problem of updating thousands of records.

CustomerID	CityCode	CityName	ShipmentNum	Quantity
1	34	İstanbul	1	300
1	34	İstanbul	2	200
…	…	…	…	…
3	6	Ankara	2	200
…	…	…	…	…

Second Normal Form (2NF)

- When starting the second normal form, the columns that repeat the data are split up into new tables.
- In order to get rid of the problems in the first normal form, the partial dependencies between the attributes must be removed and the second normal form (2NF) is achieved by converting the first normal form (1NF) tables into more than one table.
- In the **second normal form 2NF**, each non-key column of the relational table should not be partially dependent on the primary key, but fully functionally dependent.
- The **CityCode** and **CityName** attributes are only fully functionally dependent on the **CustomerID** attribute of the composite key (**CustomerID, ShipmentNum**).
- Then **CityCode** and **CityName** attributes should form a separate table with **CustomerID.**

Second Normal Form (2NF)

CustomerID	CityCode	CityName	ShipmentNum	Quantity
1	34	İstanbul	1	300
1	34	İstanbul	2	200
1	34	İstanbul	3	250
1	34	İstanbul	6	400
1	34	İstanbul	7	150
2	6	Ankara	1	300
2	6	Ankara	2	400
3	6	Ankara	2	200
4	34	İstanbul	2	200
4	34	İstanbul	4	300
4	34	İstanbul	5	400

- Since the database shown in the figure is 1NF, we can do the following.
  - ***DISTRIBUTION*** *(CustomerID, CityCode, CityName, ShipmentNum, Quantity)*

    Is split into

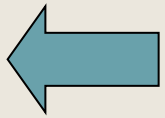
  - *CUSTOMERS(CustomerID, CityCode, CityName)*
  - *QUANTITIES(CustomerID, ShipmentNum, Quantity)*

Second Normal Form (2NF)

CustomerID	CityCode	CityName
1	34	İstanbul
2	6	Ankara
3	6	Ankara
4	34	İstanbul

CUSTOMERS

QUANTITIES



CustomerID	ShipmentNum	Quantity
1	1	300
1	2	200
1	3	400
1	4	200
1	6	100
2	1	300
2	2	400
3	2	200
4	2	200
4	4	300
4	5	400

Second Normal Form (2NF)

- Row Insertion Problem
  - It is not possible to enter a new **CityCode** and **CityName**, unless a new customer record is entered in the table. For example, in this table, a problem will be encountered when it is necessary to add new city with **Trabzon** for **CityName** and **61** for **CityCode** . To include this city in the table, a customer located in this province is needed. Unless there is such a customer, such information about the city cannot be added to the table.

CustomerID	CityCode	CityName
1	34	İstanbul
2	6	Ankara
3	6	Ankara
4	34	İstanbul

	61	Trabzon

Second Normal Form (2NF)

- Row Deletion Problem
  - *When a Customer is deleted from the table, the **CityCode** and **CityName** information will also disappear. For example, when the customer who has «CustomerID= 5» is deleted from the table, the data related to* **Trabzon** city, *where that customer is located, disappears from the table.*

CustomerID	CityCode	CityName
1	34	İstanbul
2	6	Ankara
3	6	Ankara
4	34	İstanbul
5	61	Trabzon

CustomerID	CityCode	CityName
1	34	İstanbul
2	6	Ankara
3	6	Ankara
4	34	İstanbul

Transitive Dependency

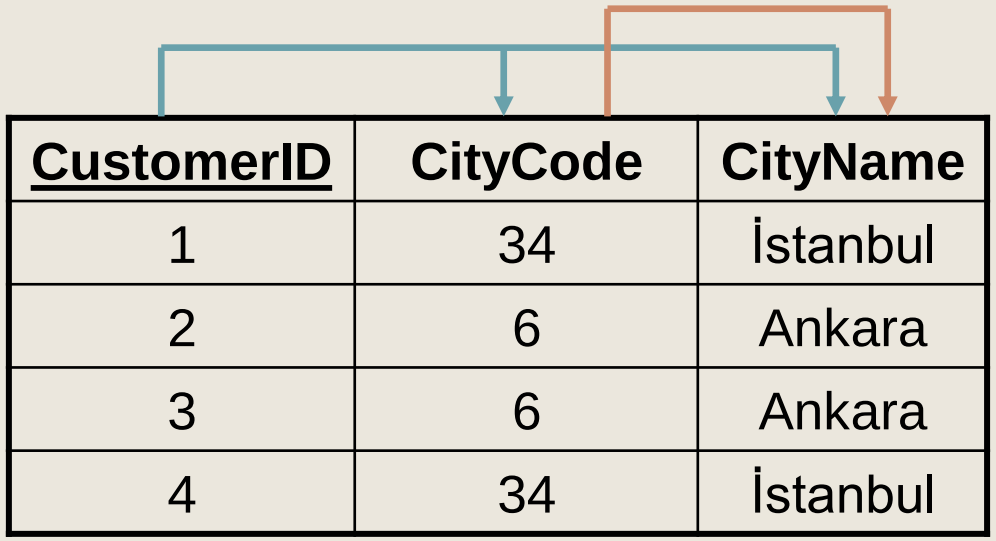
- A,B,C are atrributes in a table.
- For A, B, and C, If the following functional dependencies exist,
  - $A \rightarrow B$
  - $B \rightarrow C$
- Then, C is transitively dependent on A via B

Third Normal Form (3NF)

- In order to get rid of the problems in the first normal form, we removed the partial functional dependencies between the attributes.
- In order to get rid of the problems in the second normal form, we need to eliminate the transitive functional dependencies between the attributes.

Third Normal Form (3NF)

- In our example, the functional dependence of **CustomerID** and **CityName** is transitive because
  - CustomerID ->CityCode,CityName
  - CityCode->CityName



CustomerID	CityCode	CityName
1	34	İstanbul
2	6	Ankara
3	6	Ankara
4	34	İstanbul

- By converting this dependency, which is not connected to a key, into a separate table we can obtain the third normal form (3NF).

Third Normal Form (3NF)

CITIES

CityCode	CityName
6	Ankara
34	İstanbul
35	İzmir

CUSTOMERS

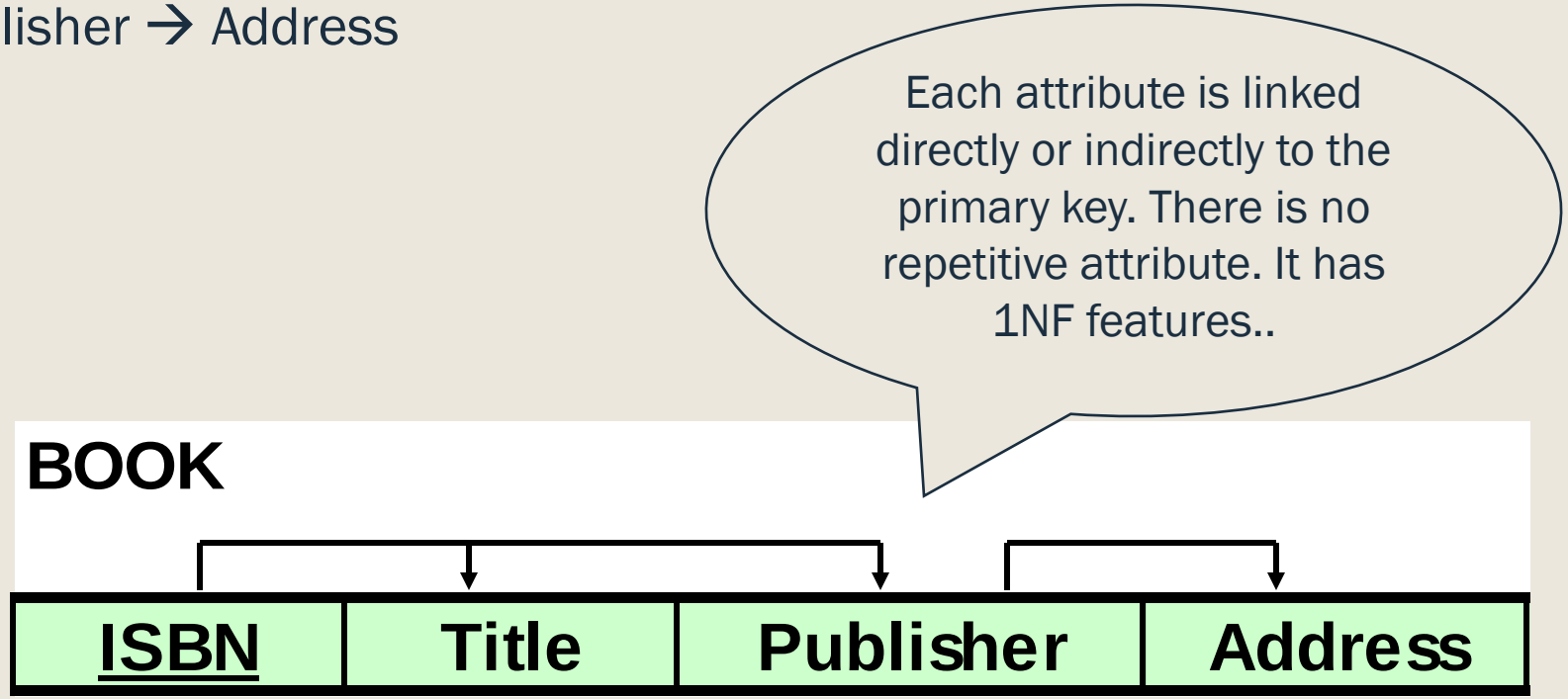
CustomerID	CityCode
1	34
2	6
3	6
4	34
5	35

QUANTITIES

CustomerID	ShipmentNum	Quntity
1	1	300
1	2	200
1	3	400
1	4	200
1	6	100
2	1	300
2	2	400
3	2	200
4	2	200
4	4	300
4	5	400

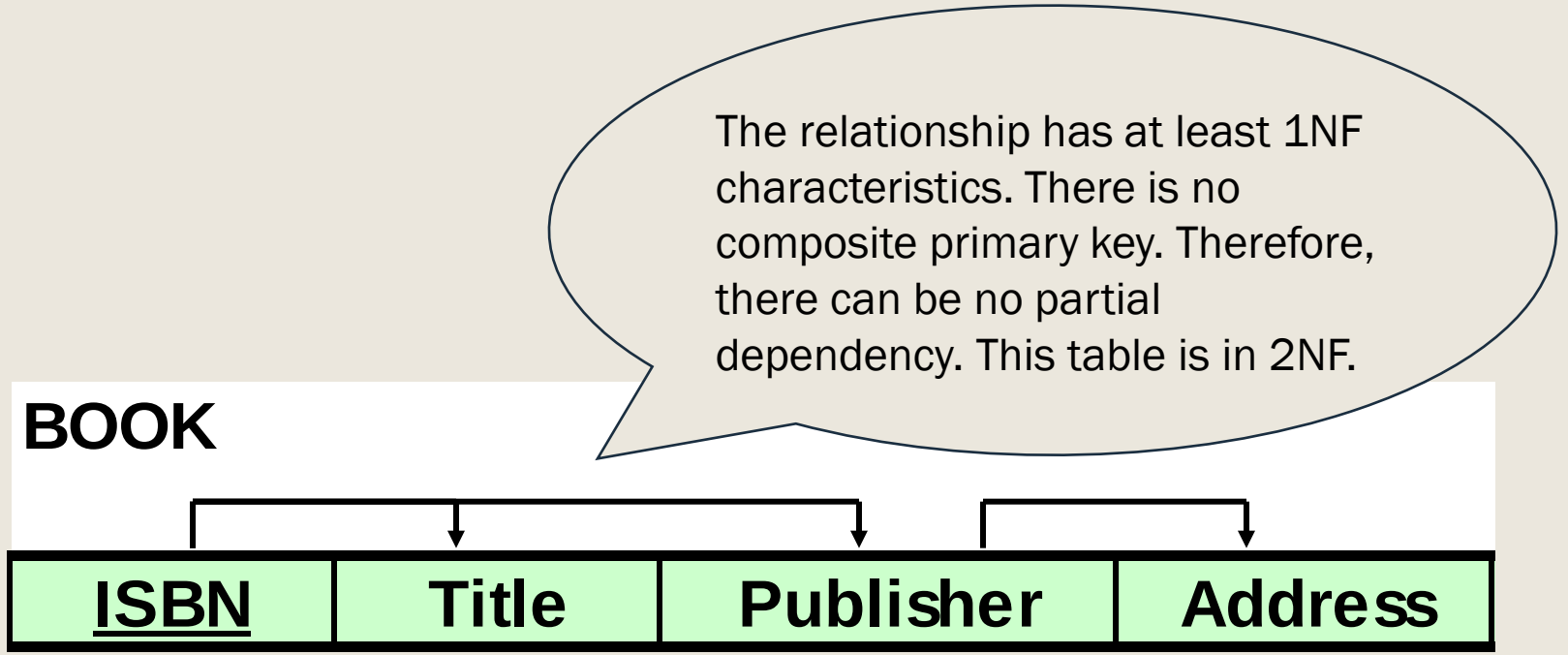
Example # 1

- ISBN → Title
- ISBN → Publisher
- Publisher → Address



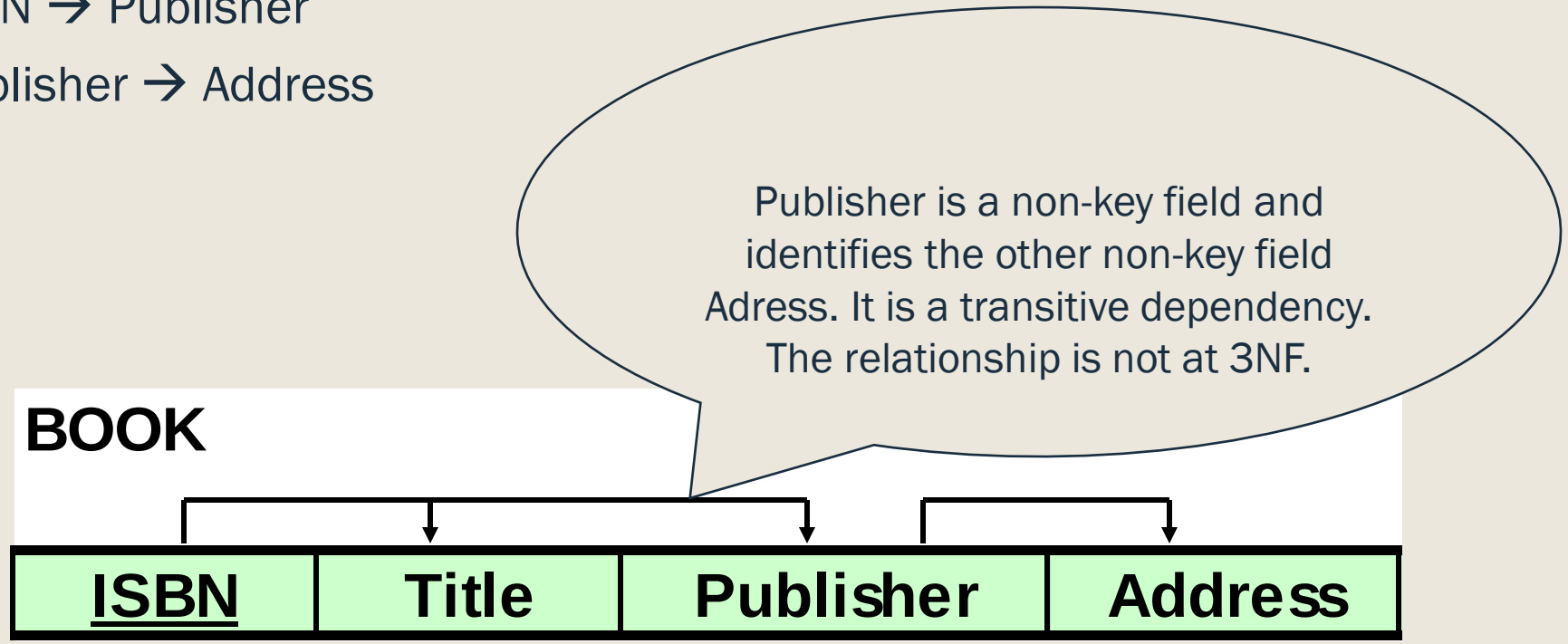
Example # 1

- ISBN → Title
- ISBN → Publisher
- Publisher → Address



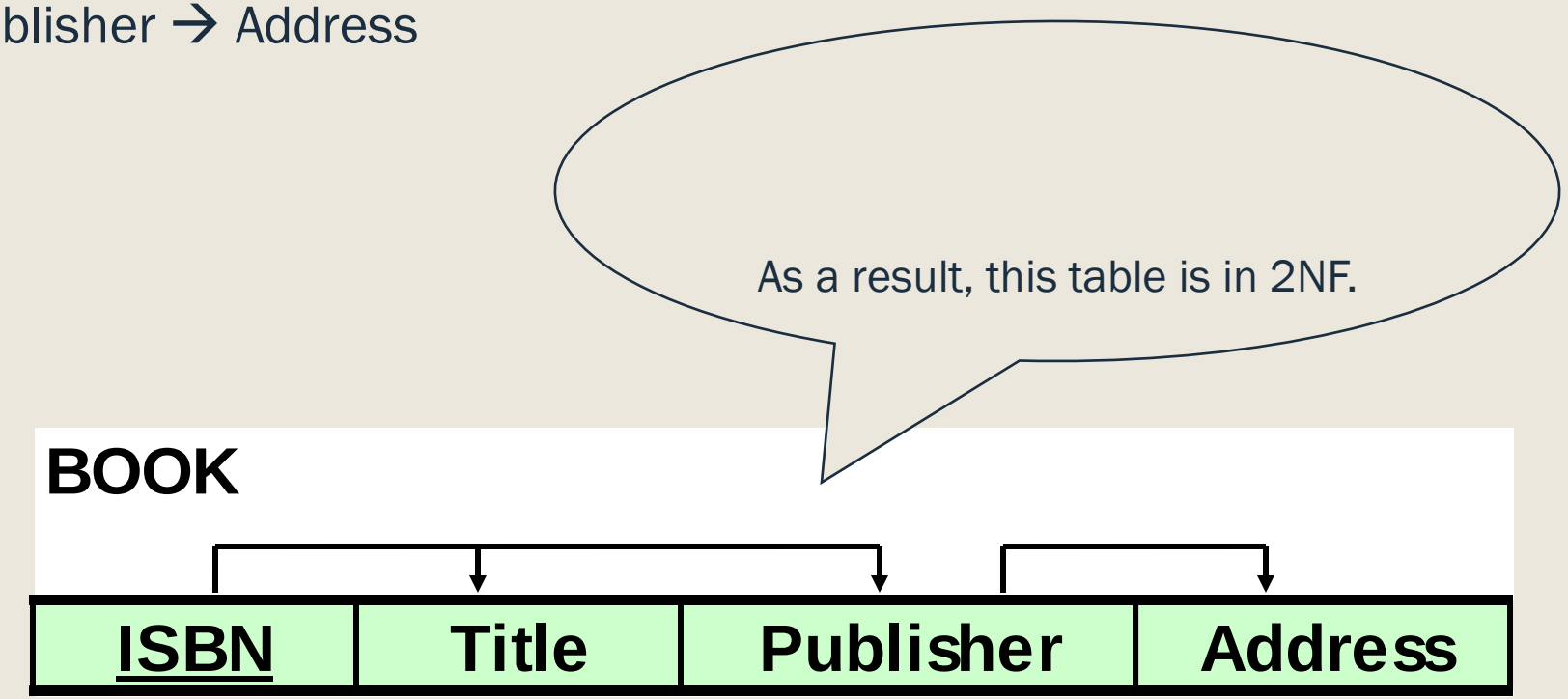
Example #1

- ISBN → Title
- ISBN → Publisher
- Publisher → Address



Example #1

- ISBN → Title
- ISBN → Publisher
- Publisher → Address



Example #2

SNum	SName	SLName	CourseNum	CourseName	Midterm	Final	TNum	TName	TLName
2001001	Ahmet	Solmaz	202	Math 2	70	60	11	Özlem	UÇAR
2001001	Ahmet	Solmaz	203	Physics 2	80	40	11	Özlem	UÇAR
2001001	Ahmet	Solmaz	204	Introduction to Computer Engineering 2	60	45	3	Aydın	CARUS
2001001	Ahmet	Solmaz	205	Ataturk's Principles and History of Turkish Revolution 2	90	95	9	Zeki	DURMUŞ
2001001	Ahmet	Solmaz	206	Turkish Language 2	70	75	12	Nebahat	YILDIZ
2001005	Seyhan	Gülmez	202	Math 2	80	95	11	Özlem	UÇAR
2001005	Seyhan	Gülmez	203	Physics 2	80	70	11	Özlem	UÇAR
2001005	Seyhan	Gülmez	204	Introduction to Computer Engineering 2	60	70	3	Aydın	CARUS
2001002	Selim	Solmaz	702	Database Management	60	50	6	Altan	MESUT
2001003	Ahmet	Vardar	702	Database Management	60	60	6	Altan	MESUT
2001004	Sezai	Kantar	702	Database Management	65	55	6	Altan	MESUT

- Which attribute(s) represent the primary key?
- What are functional dependencies that exist in this relation?
- In which normal form does this relation reside?

Example #2

SNum	SName	SLName	CourseNum	CourseName	Midterm	Final	TNum	TName	TLName
2001001	Ahmet	Solmaz	202	Math 2	70	60	11	Özlem	UÇAR
2001001	Ahmet	Solmaz	203	Physics 2	80	40	11	Özlem	UÇAR
2001001	Ahmet	Solmaz	204	Introduction to Computer Engineering 2	60	45	3	Aydın	CARUS
2001001	Ahmet	Solmaz	205	Ataturk's Principles and History of Turkish Revolution 2	90	95	9	Zeki	DURMUŞ
2001001	Ahmet	Solmaz	206	Turkish Language 2	70	75	12	Nebahat	YILDIZ
2001005	Seyhan	Gülmez	202	Math 2	80	95	11	Özlem	UÇAR
2001005	Seyhan	Gülmez	203	Physics 2	80	70	11	Özlem	UÇAR
2001005	Seyhan	Gülmez	204	Introduction to Computer Engineering 2	60	70	3	Aydın	CARUS
2001002	Selim	Solmaz	702	Database Management	60	50	6	Altan	MESUT
2001003	Ahmet	Vardar	702	Database Management	60	60	6	Altan	MESUT
2001004	Sezai	Kantar	702	Database Management	65	55	6	Altan	MESUT

- The scheme given, is in First Normal Form (atomic values)
  - COLLEGE (SNum, SName, SLName, CourseNum, CourseName, Midterm, Final, TNum, TName, TLName)

Example #2

SNum	SName	SLName	CourseNum	CourseName	Midterm	Final	TNum	TName	TLName
2001001	Ahmet	Solmaz	202	Math 2	70	60	11	Özlem	UÇAR
2001001	Ahmet	Solmaz	203	Physics 2	80	40	11	Özlem	UÇAR
2001001	Ahmet	Solmaz	204	Introduction to Computer Engineering 2	60	45	3	Aydın	CARUS
2001001	Ahmet	Solmaz	205	Ataturk's Principles and History of Turkish Revolution 2	90	95	9	Zeki	DURMUŞ
2001001	Ahmet	Solmaz	206	Turkish Language 2	70	75	12	Nebahat	YILDIZ
2001005	Seyhan	Gülmez	202	Math 2	80	95	11	Özlem	UÇAR
2001005	Seyhan	Gülmez	203	Physics 2	80	70	11	Özlem	UÇAR
2001005	Seyhan	Gülmez	204	Introduction to Computer Engineering 2	60	70	3	Aydın	CARUS
2001002	Selim	Solmaz	702	Database Management	60	50	6	Altan	MESUT
2001003	Ahmet	Vardar	702	Database Management	60	60	6	Altan	MESUT
2001004	Sezai	Kantar	702	Database Management	65	55	6	Altan	MESUT

- The scheme given, is in First Normal Form (atomic values)
  - COLLEGE (SNum, SName, SLName, CourseNum, CourseName, Midterm, Final, TNum, TName, TLName)

Example #2

SNum	SName	SLName	CourseNum	CourseName	Midterm	Final	TNum	TName	TLName
2001001	Ahmet	Solmaz	202	Math 2	70	60	11	Özlem	UÇAR
2001001	Ahmet	Solmaz	203	Physics 2	80	40	11	Özlem	UÇAR
2001001	Ahmet	Solmaz	204	Introduction to Computer Engineering 2	60	45	3	Aydın	CARUS
2001001	Ahmet	Solmaz	205	Ataturk's Principles and History of Turkish Revolution 2	90	95	9	Zeki	DURMUŞ
2001001	Ahmet	Solmaz	206	Turkish Language 2	70	75	12	Nebahat	YILDIZ
2001005	Seyhan	Gülmez	202	Math 2	80	95	11	Özlem	UÇAR
2001005	Seyhan	Gülmez	203	Physics 2	80	70	11	Özlem	UÇAR
2001005	Seyhan	Gülmez	204	Introduction to Computer Engineering 2	60	70	3	Aydın	CARUS
2001002	Selim	Solmaz	702	Database Management	60	50	6	Altan	MESUT
2001003	Ahmet	Vardar	702	Database Management	60	60	6	Altan	MESUT
2001004	Sezai	Kantar	702	Database Management	65	55	6	Altan	MESUT

- The functional dependencies are:
  - SNum, CourseNum → SName, SLName, CourseName, Midterm, Final, TNum, TName, TLName) (PK)
  - SNum → SName, SLName (Partial dependency)
  - CourseNum → CourseName, TNum, TName, TLName (Partial dependency)

Example #2

SNum	SName	SLName	CourseNum	CourseName	Midterm	Final	TNum	TName	TLName
2001001	Ahmet	Solmaz	202	Math 2	70	60	11	Özlem	UÇAR
2001001	Ahmet	Solmaz	203	Physics 2	80	40	11	Özlem	UÇAR
2001001	Ahmet	Solmaz	204	Introduction to Computer Engineering 2	60	45	3	Aydın	CARUS
2001001	Ahmet	Solmaz	205	Ataturk's Principles and History of Turkish Revolution 2	90	95	9	Zeki	DURMUŞ
2001001	Ahmet	Solmaz	206	Turkish Language 2	70	75	12	Nebahat	YILDIZ
2001005	Seyhan	Gülmez	202	Math 2	80	95	11	Özlem	UÇAR
2001005	Seyhan	Gülmez	203	Physics 2	80	70	11	Özlem	UÇAR
2001005	Seyhan	Gülmez	204	Introduction to Computer Engineering 2	60	70	3	Aydın	CARUS
2001002	Selim	Solmaz	702	Database Management	60	50	6	Altan	MESUT
2001003	Ahmet	Vardar	702	Database Management	60	60	6	Altan	MESUT
2001004	Sezai	Kantar	702	Database Management	65	55	6	Altan	MESUT

- The functional dependencies are:
  - SNum, CourseNum→ SName, SLName, CourseName, Midterm, Final, TNum, TName, TLName) (PK)
  - SNum → SName, SLName (Partial dependency)
  - CourseNum → CourseName, TNum, TName, TLName (Partial dependency)

1NF → Remove partial dependencies → 2NF

Example #2

SCORES

SNum	CourseNum	Midterm	Final
2001001	202	70	60
2001001	203	80	40
2001001	204	60	45
2001001	205	90	95
2001001	206	70	75
2001005	202	80	95
2001005	203	80	70
2001005	204	60	70
2001002	702	60	50
2001003	702	60	60
2001004	702	65	55

STUDENTS

SNum	SName	SLName
2001001	Ahmet	Solmaz
2001005	Seyhan	Gülmez
2001002	Selim	Solmaz
2001003	Ahmet	Vardar
2001004	Sezai	Kantar

COURSES

CourseNum	CourseName	TNum	TName	TLName
202	Math 2	11	Özlem	UÇAR
203	Physics 2	11	Özlem	UÇAR
204	Introduction to Computer Engineering 2	3	Aydın	CARUS
205	Ataturk's Principles and History of Turkish Revolution 2	9	Zeki	DURMUŞ
206	Turkish Language 2	12	Nebahat	YILDIZ
702	Database Management	6	Altan	MESUT

- This results in the creation of three new relations called SCORES, STUDENTS, and COURSES.
  - SCORES(SNum, CourseNum, Midterm, Final)
  - STUDENTS(SNum, SName, SLName)
  - COURSES(CourseNum, CourseName, TNum, TName, TLName)

Example #2

SCORES

SNum	CourseNum	Midterm	Final
2001001	202	70	60
2001001	203	80	40
2001001	204	60	45
2001001	205	90	95
2001001	206	70	75
2001005	202	80	95
2001005	203	80	70
2001005	204	60	70
2001002	702	60	50
2001003	702	60	60
2001004	702	65	55

STUDENTS

SNum	SName	SLName
2001001	Ahmet	Solmaz
2001005	Seyhan	Gülmez
2001002	Selim	Solmaz
2001003	Ahmet	Vardar
2001004	Sezai	Kantar

COURSES

CourseNum	CourseName	TNum	TName	TLName
202	Math 2	11	Özlem	UÇAR
203	Physics 2	11	Özlem	UÇAR
204	Introduction to Computer Engineering 2	3	Aydın	CARUS
205	Ataturk's Principles and History of Turkish Revolution 2	9	Zeki	DURMUŞ
206	Turkish Language 2	12	Nebahat	YILDIZ
702	Database Management	6	Altan	MESUT

- The functional dependencies are as follows.
  - For SCORES : SNum, CourseNum → Midterm, Final (PK)
  - For STUDENTS : SNum → SName, SLName (PK)
  - For COURSES :
    - CourseNum → CourseName, TNum, TName, TLName (PK)
    - TNum → TName, TLName (Transitive dependency)

Example #2

SCORES

SNum	CourseNum	Midterm	Final
2001001	202	70	60
2001001	203	80	40
2001001	204	60	45
2001001	205	90	95
2001001	206	70	75
2001005	202	80	95
2001005	203	80	70
2001005	204	60	70
2001002	702	60	50
2001003	702	60	60
2001004	702	65	55

STUDENTS

SNum	SName	SLName
2001001	Ahmet	Solmaz
2001005	Seyhan	Gülmez
2001002	Selim	Solmaz
2001003	Ahmet	Vardar
2001004	Sezai	Kantar

COURSES

CourseNum	CourseName	TNum	TName	TLName
202	Math 2	11	Özlem	UÇAR
203	Physics 2	11	Özlem	UÇAR
204	Introduction to Computer Engineering 2	3	Aydın	CARUS
205	Ataturk's Principles and History of Turkish Revolution 2	9	Zeki	DURMUŞ
206	Turkish Language 2	12	Nebahat	YILDIZ
702	Database Management	6	Altan	MESUT

- The functional dependencies are as follows.
  - For SCORES : SNum, CourseNum → Midterm, Final (PK)
  - For STUDENTS : SNum → SName, SLName (PK)
  - For COURSES :
    - CourseNum → CourseName, TNum, TName, TLName (PK)
    - TNum → TName, TLName (Transitive dependency)

2NF → Remove transitive dependencies → 3NF

Example #2

SCORES

SNum	CourseNum	Midterm	Final
2001001	202	70	60
2001001	203	80	40
2001001	204	60	45
2001001	205	90	95
2001001	206	70	75
2001005	202	80	95
2001005	203	80	70
2001005	204	60	70
2001002	702	60	50
2001003	702	60	60
2001004	702	65	55

STUDENTS

SNum	SName	SLName
2001001	Ahmet	Solmaz
2001005	Seyhan	Gülmez
2001002	Selim	Solmaz
2001003	Ahmet	Vardar
2001004	Sezai	Kantar

COURSES

CourseNum	CourseName	TNum
202	Math 2	11
203	Physics 2	11
204	Introduction to Computer Engineering 2	3
205	Ataturk's Principles and History of Turkish Revolution 2	9
206	Turkish Language 2	12
702	Database Management	6

TEACHERS

TNum	TName	TLName
11	Özlem	UÇAR
3	Aydın	CARUS
9	Zeki	DURMUŞ
12	Nebahat	YILDIZ
6	Altan	MESUT

- This results in the creation of two new relations called COURSES and TEACHERS.
  - COURSES(CourseNum, CourseName, TNum)
  - TEACHERS(TNum, TName, TLName)

Example #2

SCORES

SNum	CourseNum	Midterm	Final
2001001	202	70	60
2001001	203	80	40
2001001	204	60	45
2001001	205	90	95
2001001	206	70	75
2001005	202	80	95
2001005	203	80	70
2001005	204	60	70
2001002	702	60	50
2001003	702	60	60
2001004	702	65	55

STUDENTS

SNum	SName	SLName
2001001	Ahmet	Solmaz
2001005	Seyhan	Gülmez
2001002	Selim	Solmaz
2001003	Ahmet	Vardar
2001004	Sezai	Kantar

COURSES

CourseNum	CourseName	TNum
202	Math 2	11
203	Physics 2	11
204	Introduction to Computer Engineering 2	3
205	Ataturk's Principles and History of Turkish Revolution 2	9
206	Turkish Language 2	12
702	Database Management	6

TEACHERS

TNum	TName	TLName
11	Özlem	UÇAR
3	Aydın	CARUS
9	Zeki	DURMUŞ
12	Nebahat	YILDIZ
6	Altan	MESUT

- The full set of relations that are in 3NF is:
  - SCORES(SNum, CourseNum, Midterm, Final)
  - STUDENTS(SNum, SName, SLName)
  - COURSES(CourseNum, CourseName, TNum)
  - TEACHERS(TNum, TName, TLName)