

# SWE 207

# Database Management Systems

~ Normalization in DBMS~

CELAL ÇEKEN,  
GÖZDE YOLCU ÖZTEL,  
VEYSEL HARUN ŞAHİN



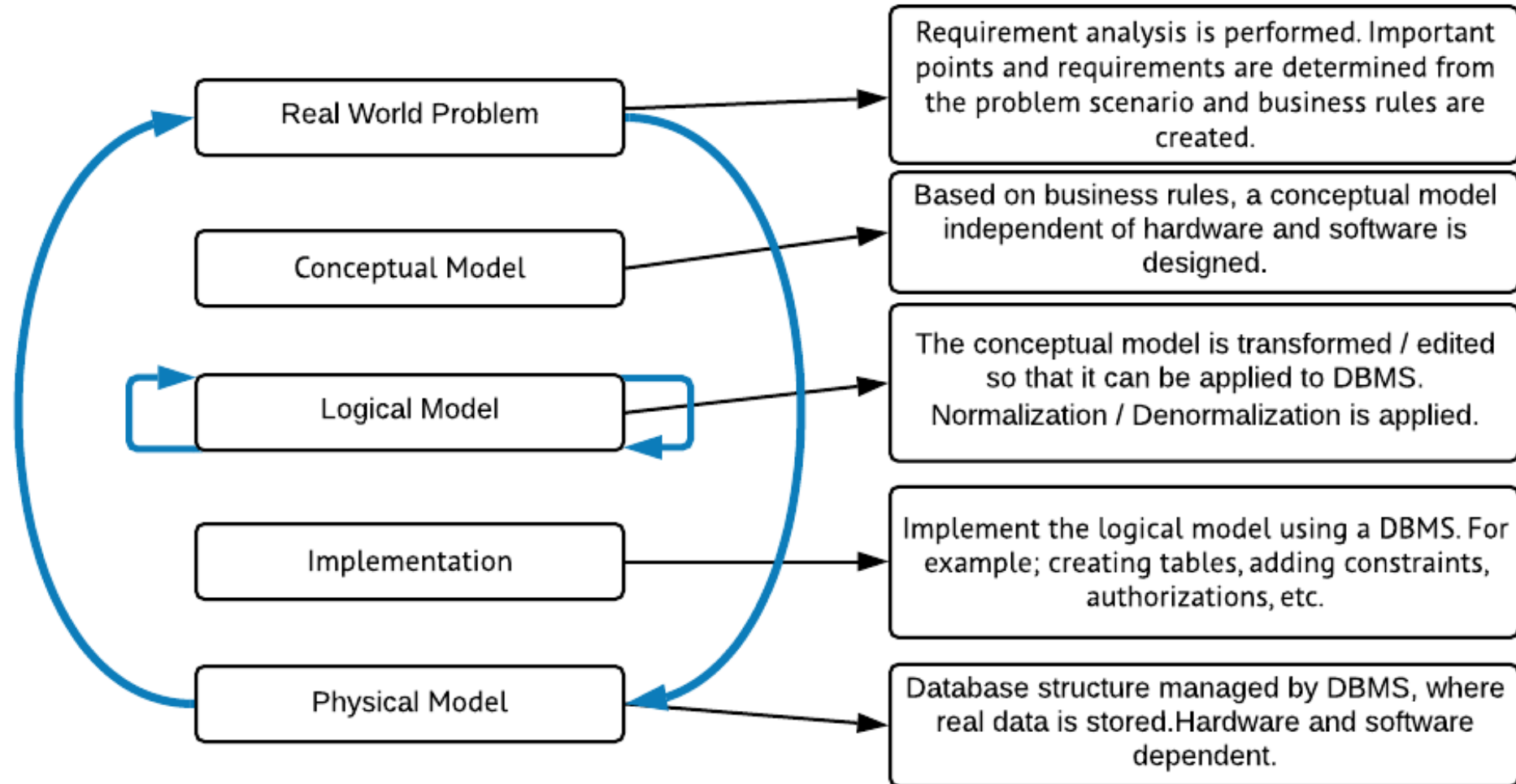
**SAKARYA**  
ÜNİVERSİTESİ

# Content

- Normalization
- Disadvantages of Data Redundancy
- First Normal Form (1NF)
- Functional Dependencies
- Second Normal Form (2NF)
- Third Normal Form (3NF)
- 1., 2. ve 3. Normal Forms
- Examples



# Database development lifecycle



# Normalization

- Good database design requires regular and robust table designs.
- Normalization can be defined as the process of evaluating and correcting table structures to prevent data anomalies by minimizing data redundancy.
- The normalization process consists of serial operations called normal form.  
1NF, 2NF, 3NF, 4NF
- 2NF is better than 1NF, 3NF is better than 2NF, and 4NF is better than 3NF.

# Normalization

- The highest NF for each design may not perform better results.
- In some cases where high performance is required, it may be necessary to lower the normal form (NF) (denormalization).
- For example, for high speed, denormalization can be applied considering the data redundancy.
- The main benefit of normalization is that it helps us avoid data anomalies by minimizing data redundancy.

## Disadvantages of Data Redundancy

- Data redundancy means that a table is embedded in another table.

<u>personelId</u>	name	surname	departmentId	departmentName	staffNumber
1	James	White	1	information technologies	12
2	John	Lever	1	information technologies	12
3	Mary	Murphy	2	quality control	8
4	Jennifer	Fraser	2	quality control	8

# Disadvantages of Data Redundancy

- In data repetition, the same information is recorded repeatedly.
- Data repetition causes unnecessary resource usage.
- Data duplication can cause data anomalies.
  - Data anomalies in data inserting
  - Data anomalies in data updating
  - Data anomalies in data deletion

# Disadvantages of Data Redundancy

- Data anomalies in data inserting
  - If the primary key value is repeated in tables with duplicate data, new records may not be added.
  - In the table, no value can be written in the personellId field and departmentId is 3 cannot be added.

<u>personellId</u>	name	surname	departmentId	departmentName	staffNumber
1	James	White	1	information technologies	12
2	John	Lever	1	information technologies	12
3	Mary	Murphy	2	quality control	8
4	Jennifer	Fraser	2	quality control	8
1	James	White	3	research and development	20



# Disadvantages of Data Redundancy

- Data anomalies in data updating
  - While updating data, modifications are required at many points, and data anomalies may occur in this case.
  - A data anomalies occurred while updating the staffNumber in the table.

<u>personelId</u>	name	surname	departmentId	departmentName	staffNumber
1	James	White	1	information technologies	15
2	John	Lever	1	information technologies	12
3	Mary	Murphy	2	quality control	8
4	Jennifer	Fraser	2	quality control	8

# Disadvantages of Data Redundancy

- Data anomalies in data deletion
  - Some data may be lost while some data is being deleted.
  - When the staff of departmentId is 1 in the table are deleted, the information of the relevant department is also lost.

<u>personelId</u>	name	surname	departmentId	departmentName	staffNumber
3	Mary	Murphy	2	quality control	8
4	Jennifer	Fraser	2	quality control	8

# Disadvantages of Data Redundancy

## Wrong design

<u>personelId</u>	name	surname	departmentId	departmentName	staffNumber
1	James	White	1	information technologies	12
2	John	Lever	1	information technologies	12
3	Mary	Murphy	2	quality control	8
4	Jennifer	Fraser	2	quality control	8

## Improved Design

<u>personelId</u>	name	surname
1	James	White
2	John	Lever
3	Mary	Murphy
4	Jennifer	Fraser

<u>personelId</u>	<u>departmentId</u>
1	1
2	1
3	2
4	2

<u>departmentId</u>	departmentName	staffNumber
1	information technologies	12
2	quality control	8

# First Normal Form (1NF)

- For a table to be in first normal form (1NF), the following conditions must be met.
  - All fields must be determined by the primary key.
  - All fields must be single-valued.

## First Normal Form (1NF)

- The table below is not in first normal form (1NF).
  - Because the eMail column is not single-valued.
  - Normalization should be done.

<u>studentId</u>	name	surname	e-mail
112	James	White	james@xmail.com, james@ymail.com
115	John	Lever	john@xmail.com
119	Mary	Murphy	mary@xmail.com
121	Jennifer	Fraser	jennifer@xmail.com

# First Normal Form (1NF)

<u>studentId</u>	name	surname	e-mail
112	James	White	james@xmail.com, james@ymail.com
115	John	Lever	john@xmail.com
119	Mary	Murphy	mary@xmail.com
121	Jennifer	Fraser	jennifer@xmail.com



<u>studentId</u>	name	surname
112	James	White
115	John	Lever
119	Mary	Murphy
121	Jennifer	Fraser

<u>id</u>	e-mail	studentId
1	james@xmail.com	112
2	james@ymail.com	112
3	john@xmail.com	115
4	mary@xmail.com	119
5	jennifer@xmail.com	121

# First Normal Form (1NF)

- Repeated Data Sets
  - The table below is in first normal form.
    - The table has a primary key.
    - Each column has only one value.
  - However, there is data repetition.

<u>studentId</u>	courseId	courseName	credit	mark
113	M1	Math	5	80
117	H1	History	3	85
120	P1	Physics	4	75
127	M1	Math	5	95
153	P1	Physics	4	90

## Second Normal Form (2NF)

- Functional Dependence
  - It allows the records to be distinguished.
  - It allows to establish relationships between tables.
  - Functional dependency should be taken into account when determining the keys.
  - In the table below, the name field can be specified using the studentID attribute. (the reverse is not true)
  - In this case:
    - The studentID field determines the name field.
    - The name field is functionally dependent on the studentID field ( studentId → studentName).



## Second Normal Form (2NF)

- The studentID field determines the name field.
- The name field is functionally dependent on the studentID field ( studentId → studentName).

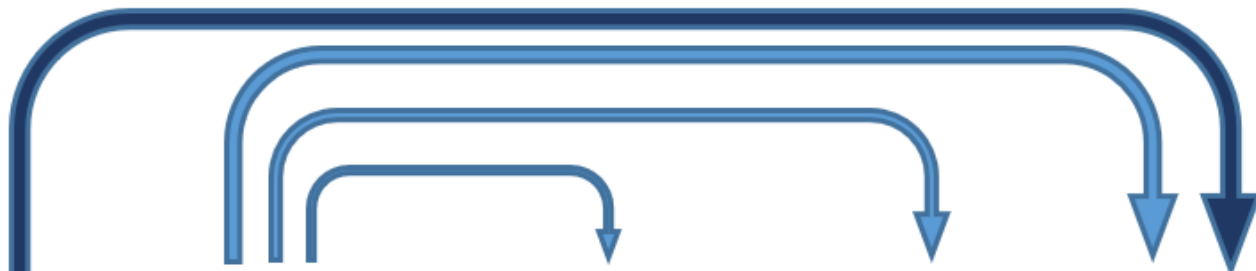
<u>studentId</u>	name	surname	phone	e-mail	address	birthDate
112	James	White		james@ymail.com	unknown	1980
115	John	Lever		john@xmail.com	unknown	<i>NULL</i>
119	Mary	Murphy		mary@xmail.com	Coumbia	2000
121	Jennifer	Fraser	3332331223	jennifer@xmail.com	Kansas	2001

## Second Normal Form (2NF)

- Full Functional Dependence
  - Attributes can be functionally dependent on the combination of more than one field.
    - studentId, courseId → mark Full functional dependency
    - studentId, courseId → courseName Partial functional dependency
- Conditions of Second Normal Form
  - The table must be in first normal form.
  - If the primary key is a composite primary key, the other fields must be fully dependent on the primary key.
  - If the primary key consists of a single field and the table is in 1NF, then 2NF is also provided.

## First Normal Form (1NF)

- The inconsistency in the table is shown below.

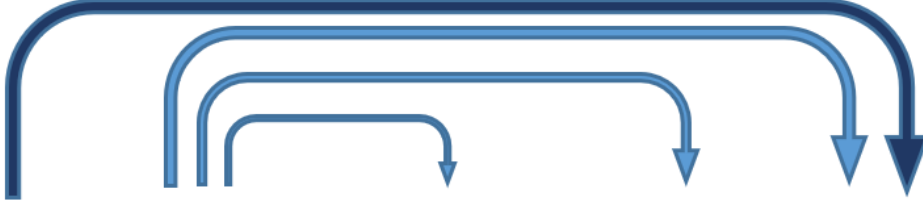


<u>studentId</u>	<u>courseId</u>	courseName	credit	mark
113	M1	Math	5	80
117	H1	History	3	85
120	P1	Physics	4	75
127	M1	Math	5	95
153	P1	Physics	4	90

## Second Normal Form (2NF)

- Example

- The primary key is a composite primary key consisting of studentId and courseId fields.
- The mark field is fully functional dependent on the primary key.
- The courseName and credit fields are partially functionally dependent on the primary key, and therefore the table is not in second normal form (2NF). Data duplication has occurred.



<u>studentId</u>	<u>courseId</u>	courseName	credit	mark
113	M1	Math	5	80
117	H1	History	3	85
120	P1	Physics	4	75
127	M1	Math	5	95
153	P1	Physics	4	90

## Second Normal Form (2NF)

<u>studentId</u>	<u>courseId</u>	courseName	credit	mark
113	M1	Math	5	80
117	H1	History	3	85
120	P1	Physics	4	75
127	M1	Math	5	95
153	P1	Physics	4	90

<u>studentId</u>	<u>courseId</u>	mark
113	M1	80
117	H1	85
120	P1	75
127	M1	95
153	P1	90

<u>courseId</u>	courseName	credit
M1	Math	5
H1	History	3
P1	Physics	4

# Third Normal Form (3NF)

- Transitive Dependency
  - If  $A \rightarrow B$  and  $B \rightarrow C$  then  $A \rightarrow B \rightarrow C$ 
    - A determines C over B.
    - C is transitively dependent on A.
  - The following table has a transitive dependency.

## Third Normal Form (3NF)

- The following table has a transitive dependency.
  - borrowId → bookId → bookName
  - borrowId → bookId → releaseYear
  - The borrowId field determines the bookName field over the bookId field.
  - The bookName field is transitively dependent on the borrowId field.

<u>borrowId</u>	memberId	bookId	bookName	releaseYear	borrowDate	deliveryDate
1	103	1	math	2000	1.01.2010	10.01.2010
2	107	1	math	2000	5.05.2011	15.05.2011
3	105	2	history	2001	3.01.2010	13.01.2010
4	112	3	physics	2005	8.05.2011	18.05.2011
5	140	3	physics	2005	1.01.2010	11.01.2010

## Third Normal Form (3NF)



<u>borrowId</u>	memberId	bookId	bookName	releaseYear	borrowDate	deliveryDate
1	103	1	math	2000	1.01.2010	10.01.2010
2	107	1	math	2000	5.05.2011	15.05.2011
3	105	2	history	2001	3.01.2010	13.01.2010
4	112	3	physics	2005	8.05.2011	18.05.2011
5	140	3	physics	2005	1.01.2010	11.01.2010



# Third Normal Form (3NF)

- Conditions of Third Normal Form
  - If the table is in 2NF and
  - If there is no transitive dependency, it is in 3NF.

<u>borrowId</u>	memberId	bookId	bookName	releaseYear	borrowDate	deliveryDate
1	103	1	math	2000	1.01.2010	10.01.2010
2	107	1	math	2000	5.05.2011	15.05.2011
3	105	2	history	2001	3.01.2010	13.01.2010
4	112	3	physics	2005	8.05.2011	18.05.2011
5	140	3	physics	2005	1.01.2010	11.01.2010

<u>borrowId</u>	memberId	bookId	borrowDate	deliveryDate
1	103	1	1.01.2010	10.01.2010
2	107	1	5.05.2011	15.05.2011
3	105	2	3.01.2010	13.01.2010
4	112	3	8.05.2011	18.05.2011
5	140	3	1.01.2010	11.01.2010

<u>bookId</u>	bookName	releaseYear
1	math	2000
2	history	2001
3	physics	2005

## summary

- 1NF: Primary key must be and single-valued fields must be.
- 2NF: It must in the first normal form and there must be no partial dependency.
- 3NF: It must in the second normal form and there must be no transitive dependency.

## Example-1

- Explain the reason for data repetition in the table and how it can be corrected.

<u>courseId</u>	coordinatorId	courseName	coordinatorName	coordinatorSurname	credit
1	1	Math	James	White	3
2	2	History	John	Lever	3
3	1	Physics	James	White	5
4	2	Chemistry	John	Lever	5
5	3	Geography	Robert	Black	3

## Example-1

<u>courseId</u>	coordinatorId	courseName	coordinatorName	coordinatorSurname	credit
1	1	Math	James	White	3
2	2	History	John	Lever	3
3	1	Physics	James	White	5
4	2	Chemistry	John	Lever	5
5	3	Geography	Robert	Black	3

Answer:

The table is in the 1st normal form. Because there is a primary key and each field is single-valued.

The table is in the 2nd normal form. Because there is not a composite primary key and it is in the 1st normal form. Thus it is in the 2nd normal form.

The table is not in the 3rd normal form. Let's remember the rules:

It must be in the second normal form (this is provided).

and there must be no transitive dependency. (this rule is not provided. Because course id defines coordinatorId. If we know coordinatorId we can reach the coordinatorName and coordinatorSurname information. So; for coordinator and for course we must create 2 tables for avoiding data redundancy.

# Example-1

<u>courseId</u>	coordinatorId	courseName	coordinatorName	coordinatorSurname	credit
1	1	Math	James	White	3
2	2	History	John	Lever	3
3	1	Physics	James	White	5
4	2	Chemistry	John	Lever	5
5	3	Geography	Robert	Black	3

Improved design

<u>courseId</u>	coordinatorId	courseName	credit
1	1	Math	3
2	2	History	3
3	1	Physics	5
4	2	Chemistry	5
5	3	Geography	3

<u>coordinatorId</u>	coordinatorName	coordinatorSurname
1	James	White
2	John	Lever
3	Robert	Black

## Example-2

- Explain the reason for data repetition in the table and how it can be corrected.

<u>productId</u>	<u>orderId</u>	productName	amount
1	1	HDD	2
1	2	HDD	1
2	1	SDD	2
3	2	RAM	2

## Example-?

<u>productId</u>	<u>orderId</u>	productName	amount
1	1	HDD	2
1	2	HDD	1
2	1	SDD	2
3	2	RAM	2

The table is in the 1st normal form. Because there is a primary key and each field is single-valued.

The table is not in the 2nd normal form. Let's remember the rules:

It must be in the first normal form (this is provided).

There must be no partial dependency (this is not provided, because productName partially depends on the primary key. If we know productId, we can reach productName information. We don't need orderId information for this purpose. Thus, we need to create two new tables)

## Example-2

<u>productId</u>	<u>orderId</u>	productName	amount
1	1	HDD	2
1	2	HDD	1
2	1	SDD	2
3	2	RAM	2



Improved design

<u>productId</u>	<u>orderId</u>	amount
1	1	2
2	1	2
3	2	2

<u>productId</u>	productName
1	HDD
2	SDD
3	RAM