



**IE220**

# Introduction to Database Systems

**Assist. Prof. Dr. Fırat YÜCEL**  
Akdeniz University  
Informatics Department



# Normalization of Databases

UNF, 1NF, 2NF, 3NF, BCNF

# Normalization of a Database

- **Database normalization**, or simply **normalization**, is the process of restructuring a relational database in accordance with a series of so-called normal forms in order to reduce data redundancy and improve data integrity.

# Unnormalized Form (UNF)

- Some attributes can be contained in one row.
- Example: *Customers buy the products.*

Orders

ID	Name	Products
1	Ayşe Kaya	HDD, Processor
2	Jack Hellmit	BIOS Battery, Processor, CD
3	Metin Aslan	HDD, Flash Memory

# First Normal Form (1NF)

- Each attributes can only have one single value in one row; no duplicated row; no row/column order
- Example: *Customers buy the products.*

Orders

ID	Name	Products
1	Ayşe Kaya	HDD
4	Ayşe Kaya	Processor
2	Jack Hellmit	BIOS Battery
5	Jack Hellmit	Processor
6	Jack Hellmit	CD
3	Metin Aslan	HDD
7	Metin Aslan	Flash Memory

# Second Normal Form (2NF)

- Example: *Customers buy the products.*
- We add to product ID and product properties fields to Orders table.

## Orders

ID	Name	Products	ProductID	Properties
1	Ayşe Kaya	HDD	101	1TB SATA
4	Ayşe Kaya	Processor	102	Intel i5 7400
2	Jack Hellmit	BIOS Battery	103	CR2032
5	Jack Hellmit	Processor	102	Intel i5 7400
6	Jack Hellmit	CD	104	CD-R 700MB
3	Metin Aslan	HDD	101	1TB SATA
7	Metin Aslan	Flash Memory	105	30GB USB3.0

# Second Normal Form (2NF) - *Continue*

- Example: *Customers buy the products.*
- We split Orders table as two tables, Orders and Products.

Orders

ID	Name	ProductID
1	Ayşe Kaya	101
4	Ayşe Kaya	102
2	Jack Hellmit	103
5	Jack Hellmit	102
6	Jack Hellmit	104
3	Metin Aslan	105
7	Metin Aslan	101

Products

ProductID	ProductName	Properties
101	HDD	1TB SATA
102	Processor	Intel i5 7400
103	BIOS Battery	CR2032
104	CD	CD-R 700MB
105	Flash Memory	30GB USB3.0

# Third Normal Form (3NF)

- Example: *Customers buy the products.*
- We also add to customer ID and customer city fields to Orders table.

## Orders

ID	Name	Products	ProductID	Properties	CustomerID	CustomerCity
1	Ayşe Kaya	HDD	101	1TB SATA	1001	Antalya
4	Ayşe Kaya	Processor	102	Intel i5 7400	1001	Antalya
2	Jack Hellmit	BIOS Battery	103	CR2032	1002	İzmir
5	Jack Hellmit	Processor	102	Intel i5 7400	1002	İzmir
6	Jack Hellmit	CD	104	CD-R 700MB	1002	İzmir
3	Metin Aslan	HDD	101	1TB SATA	1003	İstanbul
7	Metin Aslan	Flash Memory	105	30GB USB3.0	1003	İstanbul



# Third Normal Form (3NF) - *continue*

- Example: *Customers buy the products.*
- We split Orders table as 3 tables.

Orders

ID	CustomerID	ProductID
1	1001	101
4	1001	102
2	1002	103
5	1002	102
6	1002	104
3	1003	105
7	1003	101

Products

ProductID	ProductName	Properties
101	HDD	1TB SATA
102	Processor	Intel i5 7400
103	BIOS Battery	CR2032
104	CD	CD-R 700MB
105	Flash Memory	30GB USB3.0

Customers

CustomerID	CustomerCity
1001	Antalya
1002	İzmir
1003	İstanbul

# Boyce-Codd Normal Form (BCNF)

- We add Payment field to Orders table.

## Orders

ID	CustomerID	ProductID	Payment
1	1001	101	Credit card
4	1001	102	Credit card
2	1002	103	Cash
5	1002	102	Cash
6	1002	104	Cash
3	1003	105	Credit card
7	1003	101	Credit card

## Products

ProductID	ProductName	Properties
101	HDD	1TB SATA
102	Processor	Intel i5 7400
103	BIOS Battery	CR2032
104	CD	CD-R 700MB
105	Flash Memory	30GB USB3.0

## Customers

CustomerID	CustomerCity
1001	Antalya
1002	İzmir
1003	İstanbul

# Boyce-Codd Normal Form (BCNF) - *continue*

- We convert Payment field to Payment Type.
- We represent the payment types in a new table, Payments.

Orders

ID	CustomerID	ProductID	PaymentType
1	1001	101	1
4	1001	102	1
2	1002	103	2
5	1002	102	2
6	1002	104	2
3	1003	105	1
7	1003	101	1

Payments

PaymentID	Payment Type
1	Credit card
2	Cash

# Advantages of Normalization

- No Redundancy
- No Inconsistency - all changes can only be made at the same place and keep consistent (because of the key constraints), in DB terminology - get away with all update anomaly.
- Normalization is the process of decomposition, so all the business concepts can be modeled with clear logical relationships.
- The entire database system remains consistent over time as the database grows with least redundancy and much durability.