Database Normalization

greenwich.edu.vn

UNIVERSITY of GREENWICH

Alliance with FPT Education

Chapter outcomes

- By the end of this chapter you will be able to:
  - Reasons for normalization: Anomalies
  - Understand about functional dependencies
  - Evaluate an entity against the first 3 normal forms
  - Remove all repeating lists or arrays (1st NF)
  - Remove all partial dependencies (2nd NF)
  - Remove all transitive dependencies (3rd NF)
  - Understand the importance of design review

THE DESIGN REVIEW

- We will need to review our DB and make sure
  - Entities are normalized
  - Relationships are correct
  - Diagram completely meets business requirements

Motivation

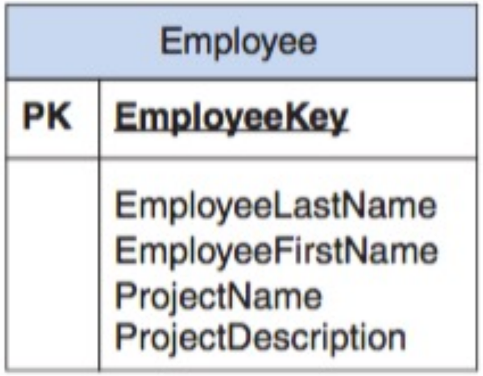
- One of the most difficult tasks in developing a DB is about designing it
- One simplest design approach is to use a big table and store everything
  - So no designing difficulty
- But what’s the problem with this?
  - Anomalies
  - Redundancies

Normalization

- Normalization is the process of removing **anomalies** and **redundancies** from DB
- Followings are anomalies in DB design
  - Insertion anomalies
  - Update anomalies
  - Deletion anomalies

Insertion Anomalies

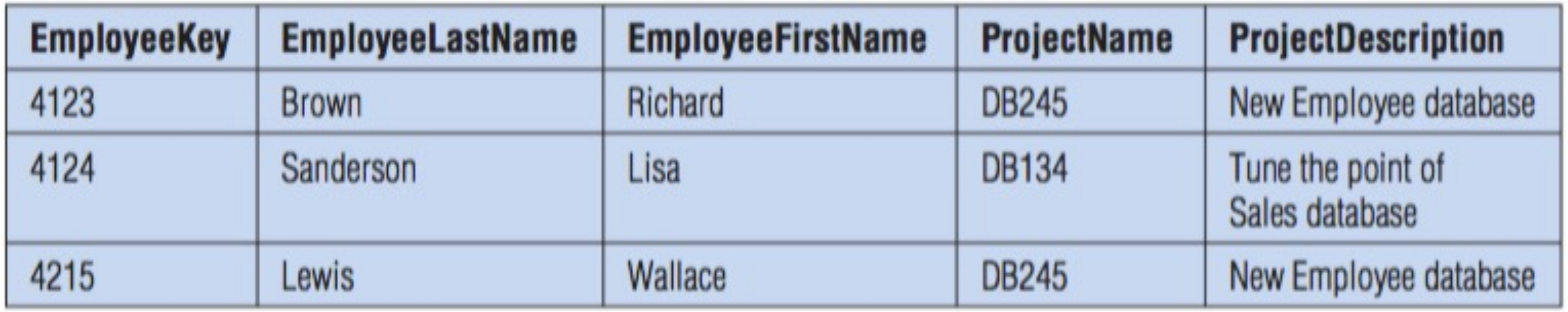
- This is when we can't insert data because some other data is missing



Employee	
PK	<u>EmployeeKey</u>
	EmployeeLastName EmployeeFirstName ProjectName ProjectDescription

E.g.,
We can't insert a new project if we don't have an employee assigned to it yet.

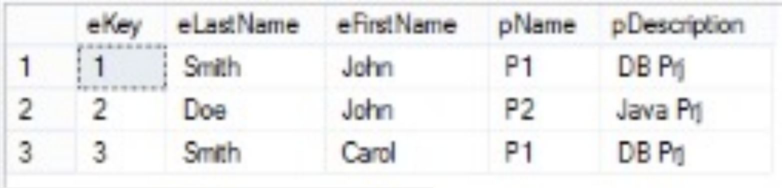
Note:
We cannot insert null value into PK attribute



EmployeeKey	EmployeeLastName	EmployeeFirstName	ProjectName	ProjectDescription
4123	Brown	Richard	DB245	New Employee database
4124	Sanderson	Lisa	DB134	Tune the point of Sales database
4215	Lewis	Wallace	DB245	New Employee database

Activity: Creating Employee Table

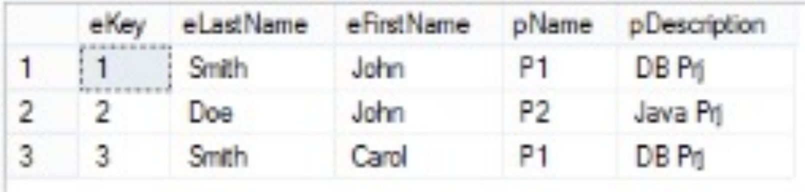
```
create database NormalizationTest;
go
use NormalizationTest;
go
create table Employee(
eKey int primary key,
eLastName varchar(50),
eFirstName varchar(50),
pName varchar(50),
pDescription varchar(50)
);
go
insert into Employee
values (1, 'Smith', 'John', 'P1', 'DB Prj'),
       (2, 'Doe', 'John', 'P2', 'Java Prj'),
       (3, 'Smith', 'Carol', 'P1', 'DB Prj');
```



	eKey	eLastName	eFirstName	pName	pDescription
1	1	Smith	John	P1	DB Prj
2	2	Doe	John	P2	Java Prj
3	3	Smith	Carol	P1	DB Prj

Activity: Insertion Anomaly

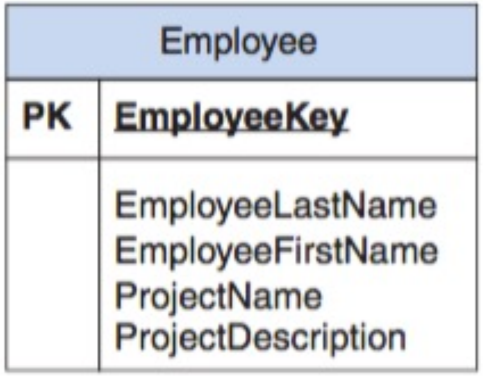
```
insert into Employee
values (null, null, null, 'P3', 'Good project');

Error:
Cannot insert the value NULL into column 'eKey'
```



	eKey	eLastName	eFirstName	pName	pDescription
1	1	Smith	John	P1	DB Prj
2	2	Doe	John	P2	Java Prj
3	3	Smith	Carol	P1	DB Prj

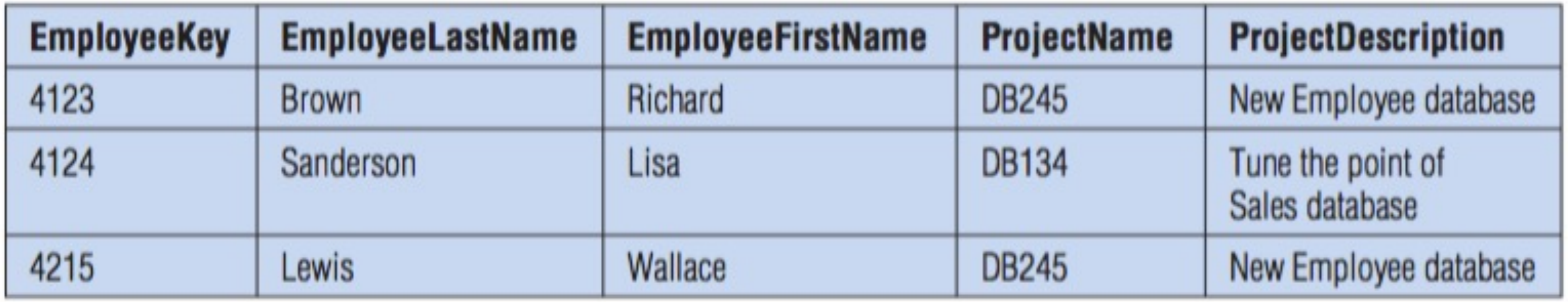
Update anomalies

- An instance where the same information must be updated in several different places



Employee	
PK	EmployeeKey
	EmployeeLastName EmployeeFirstName ProjectName ProjectDescription

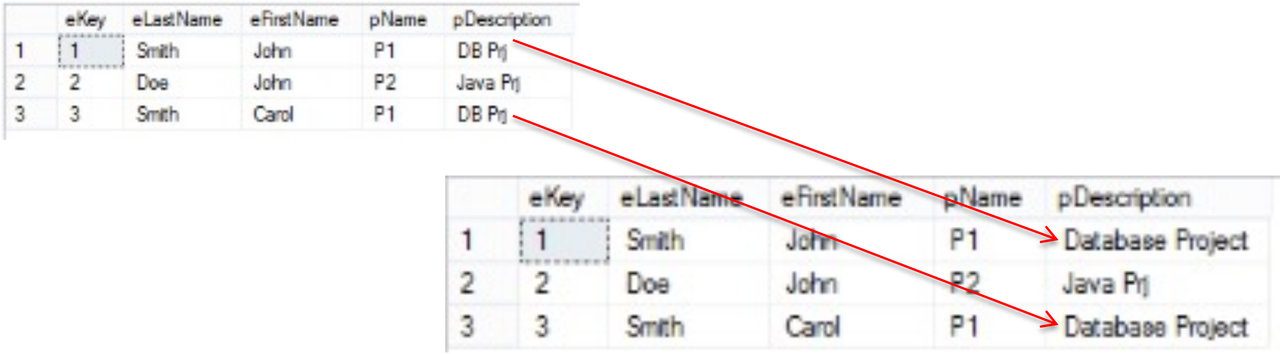
E.g.,
If you update the project DB245 name, you would need to update in two different places (not efficient)



EmployeeKey	EmployeeLastName	EmployeeFirstName	ProjectName	ProjectDescription
4123	Brown	Richard	DB245	New Employee database
4124	Sanderson	Lisa	DB134	Tune the point of Sales database
4215	Lewis	Wallace	DB245	New Employee database

Activity: Update Anomaly

```
update Employee set pDescription = 'Database Project'
where pName = 'P1';

Result
2 row(s) affected.
```

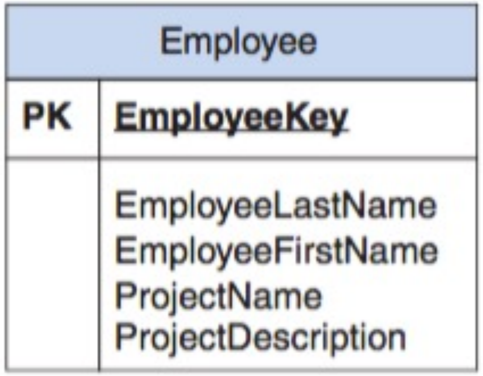


	eKey	eLastName	eFirstName	pName	pDescription
1	1	Smith	John	P1	DB Prj
2	2	Doe	John	P2	Java Prj
3	3	Smith	Carol	P1	DB Prj

	eKey	eLastName	eFirstName	pName	pDescription
1	1	Smith	John	P1	Database Project
2	2	Doe	John	P2	Java Prj
3	3	Smith	Carol	P1	Database Project

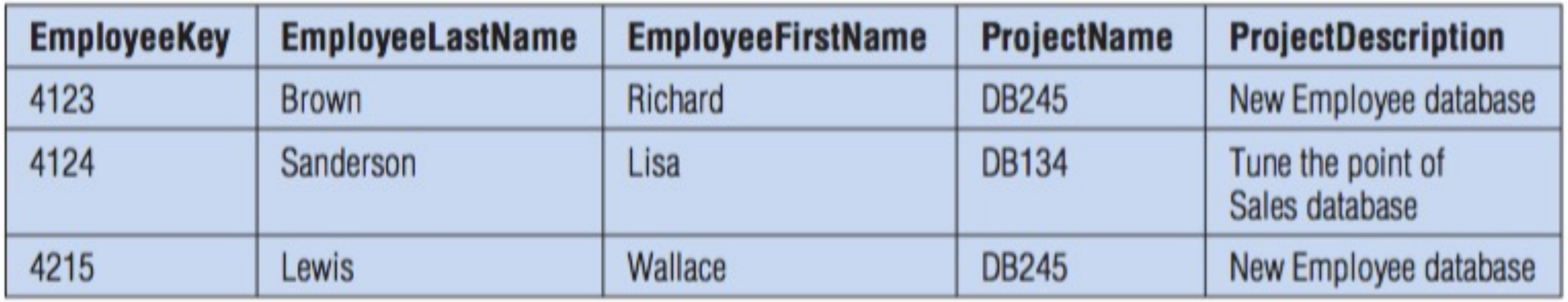
Deletion Anomalies

- Where deleting one piece of data inadvertently causes other data to be lost



Employee	
PK	EmployeeKey
	EmployeeLastName EmployeeFirstName ProjectName ProjectDescription

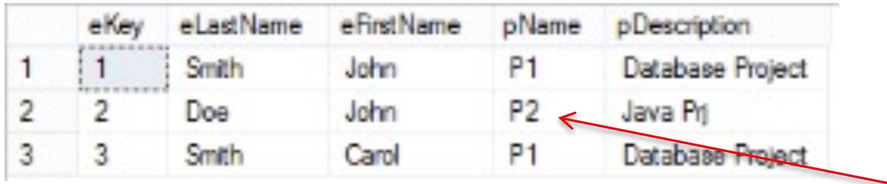
E.g.,
If we delete employee Sanderson Lisa (e.g., she quit),
then we will lose information about Project DB134



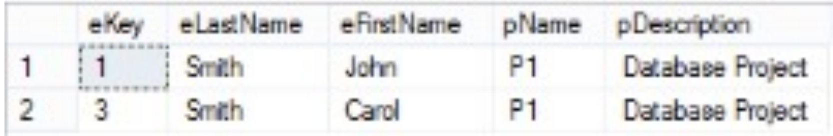
EmployeeKey	EmployeeLastName	EmployeeFirstName	ProjectName	ProjectDescription
4123	Brown	Richard	DB245	New Employee database
4124	Sanderson	Lisa	DB134	Tune the point of Sales database
4215	Lewis	Wallace	DB245	New Employee database

Activity: Delete Anomaly

```
delete Employee where eKey = 2;
```



	eKey	eLastName	eFirstName	pName	pDescription
1	1	Smith	John	P1	Database Project
2	2	Doe	John	P2	Java Prj
3	3	Smith	Carol	P1	Database Project



	eKey	eLastName	eFirstName	pName	pDescription
1	1	Smith	John	P1	Database Project
2	3	Smith	Carol	P1	Database Project

Loose Project 2 information
When delete employee 2

Functional dependencies

- Functional dependencies
  - Describe relationship among attributes in a relation
- We write A -> B
  - Read as: A determines B or B is functionally dependent on A
  - Means given a **value** of A, we can find **one and exactly one value** of B

Functional dependencies examples

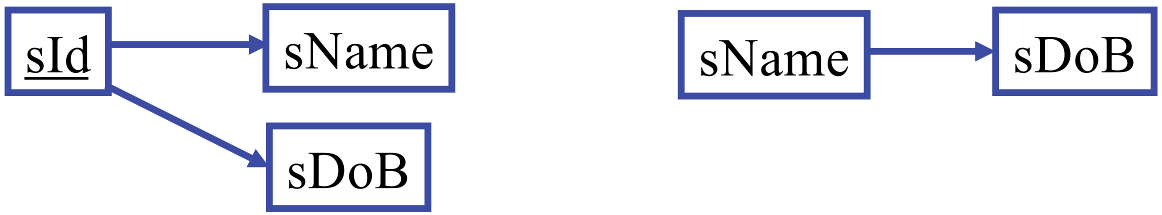
Student

sId	sName	sDoB
1	John Smith	1999-01-12
2	Remesh Shah	1998-02-28
3	Susan Black	1999-08-10
4	John Smith	1999-01-12
5	John Doe	1998-02-28

Functional dependency
sId -> {sName, sDob}
sName -> sDoB

Note: Keys (super key, primary key, candidate key) determine all the attributes in a relation

Functional Dependency Diagram



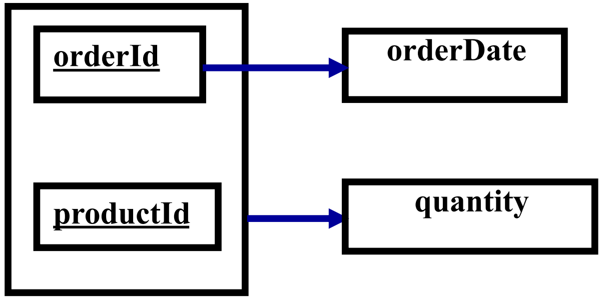
Full & Partial Dependency

orderId	orderDate	productId	quantity
O1	2016-01-01	P1	2
O1	2016-01-01	P2	3
O2	2016-01-01	P1	1
O2	2016-01-01	P3	2

Key: (orderId, productId)

Full Key Dependency:
{orderId, productId} -> quantity

Partial Key Dependency:
orderId -> orderDate



orderId

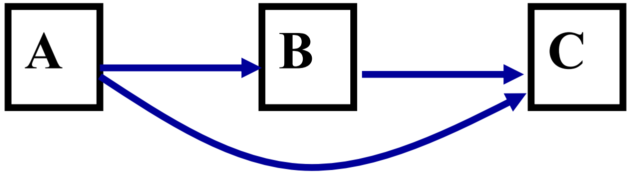
orderDate

productId

quantity

Transitive dependency

- If A->B and B->C
  - Attribute A must be the determinant of C.
  - Attribute A transitively determines attribute C or
  - C is transitively dependent on A



Normal Forms

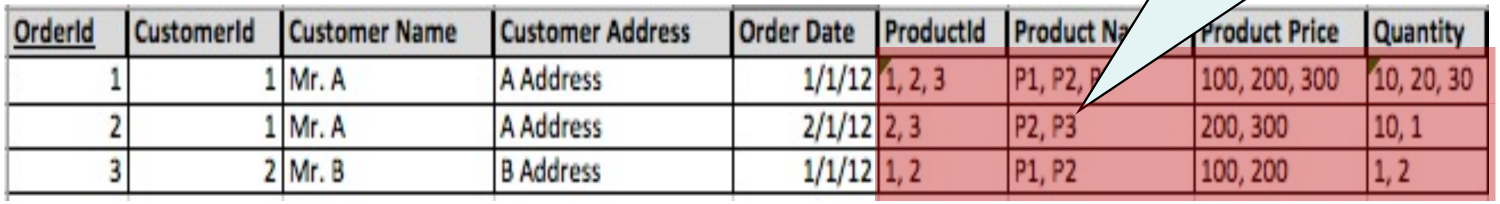
- Each form was designed to eliminate one or more of these anomalies
  - First Normal Form
  - Second Normal Form
  - Third Normal Form
- The above 3 are most critical and the followings are refinements
  - Boyce Codd Normal Form
  - Fourth Normal Form
  - Fifth Normal Form
  - Domain Key Normal Form

Unnormalised Form (UNF)

- A table that contains one or more repeating groups. I.e., its cell may contain multiple values

All kind of anomalies
Especially: Delete product P3 out of Order 1

Multi Value
Or repeating groups



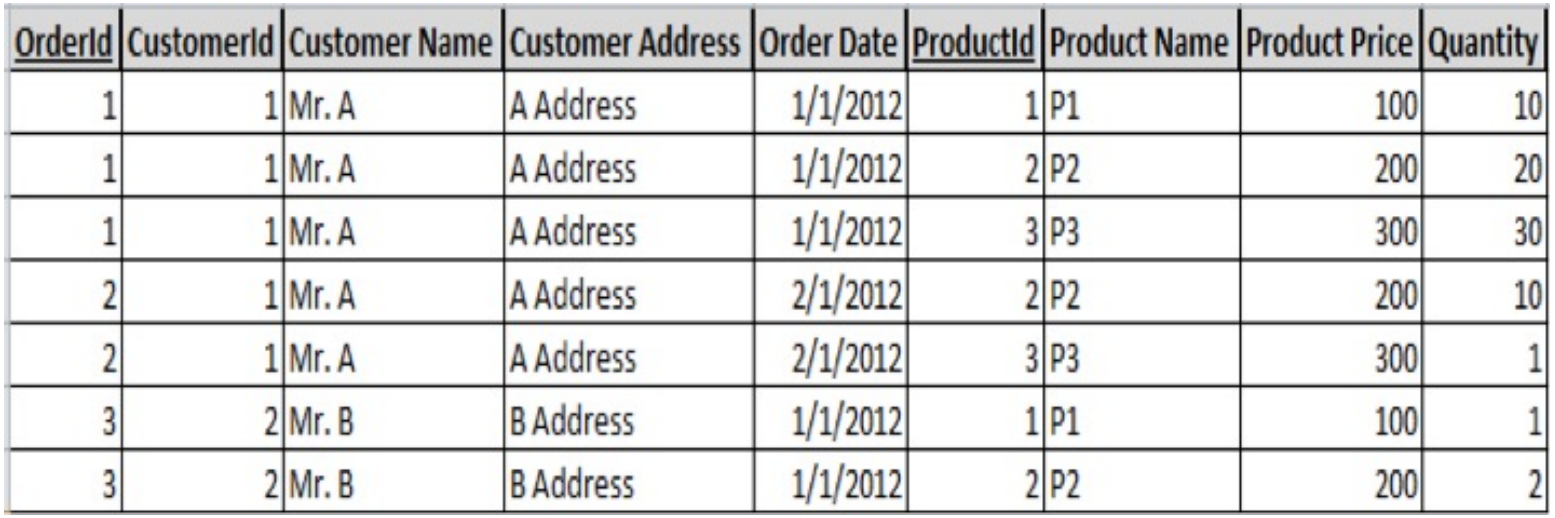
OrderId	CustomerId	Customer Name	Customer Address	Order Date	ProductId	Product Na	Product Price	Quantity
1	1	Mr. A	A Address	1/1/12	1, 2, 3	P1, P2, P	100, 200, 300	10, 20, 30
2	1	Mr. A	A Address	2/1/12	2, 3	P2, P3	200, 300	10, 1
3	2	Mr. B	B Address	1/1/12	1, 2	P1, P2	100, 200	1, 2

First Normal Form (1NF)

- A cell in a relation contains one and only one value.
  - Disallows composite attributes, multivalued attributes or nested relations
- UNF to 1NF
  - Nominate an attribute or group of attributes to act as the key for the unnormalized table.
  - Identify repeating group(s) in unnormalized table which repeats for the key attribute(s).

UNF to 1NF/1

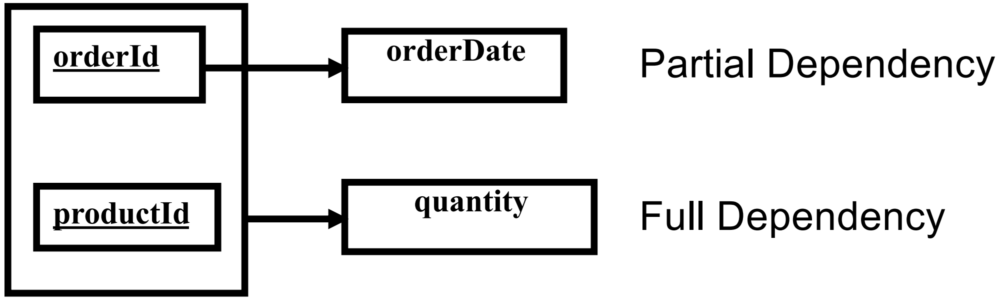
- Remove repeating group by:
  - – Entering data into the empty columns of rows containing repeating data (‘flattening’ the table)



OrderId	CustomerId	Customer Name	Customer Address	Order Date	ProductId	Product Name	Product Price	Quantity
1	1	Mr. A	A Address	1/1/2012	1	P1	100	10
1	1	Mr. A	A Address	1/1/2012	2	P2	200	20
1	1	Mr. A	A Address	1/1/2012	3	P3	300	30
2	1	Mr. A	A Address	2/1/2012	2	P2	200	10
2	1	Mr. A	A Address	2/1/2012	3	P3	300	1
3	2	Mr. B	B Address	1/1/2012	1	P1	100	1
3	2	Mr. B	B Address	1/1/2012	2	P2	200	2

Second Normal Form (2NF)

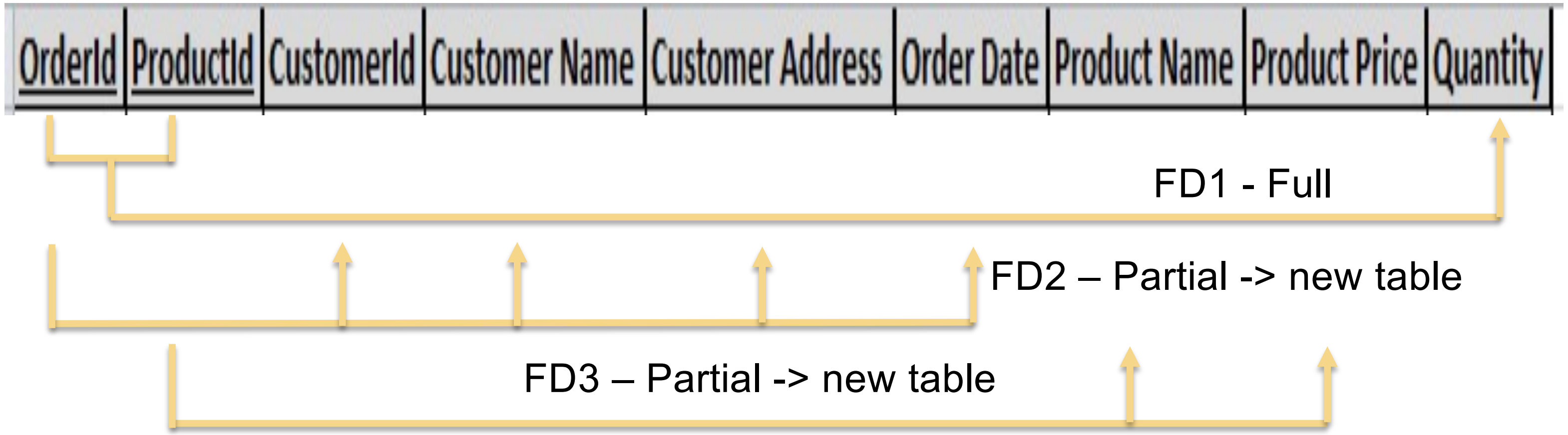
- Based on concept of full functional dependency
- Prime attribute
  - – It is an attribute that is member of the PK
- 2NF Relation is
  - – in 1NF and every non-primary-key attribute is fully functionally dependent on the primary key



1NF to 2NF

- Identify primary key for the 1NF relation
- Identify functional dependencies in the relation
- If partial dependencies exist on the primary key
    - Remove them by placing them in a new relation
    - Leave a copy of their determinant in the original relation to keep the relationship

Transformation into 2NF



OrderId	ProductId	CustomerId	Customer Name	Customer Address	Order Date	Product Name	Product Price	Quantity

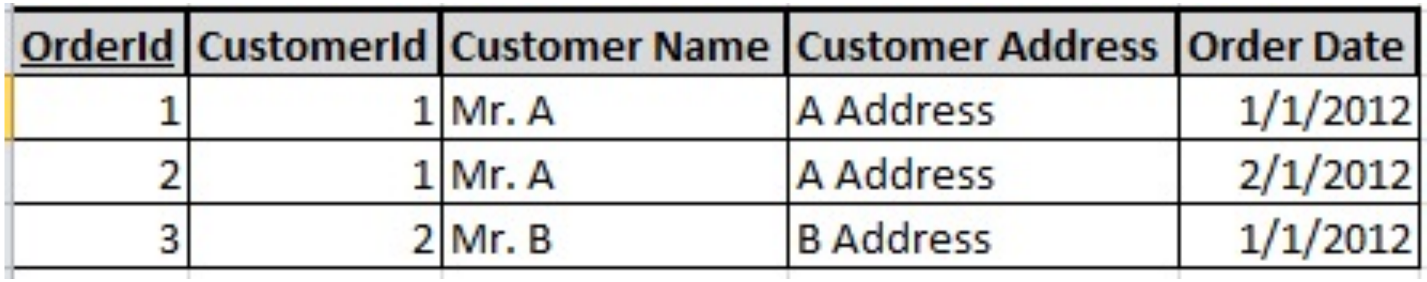
FD1 - Full

FD2 – Partial -> new table

FD3 – Partial -> new table

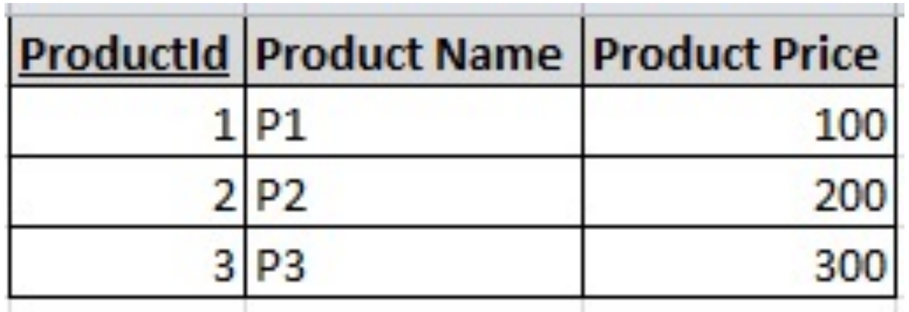
Transformation into 2NF

FD2 – Partial -> new table



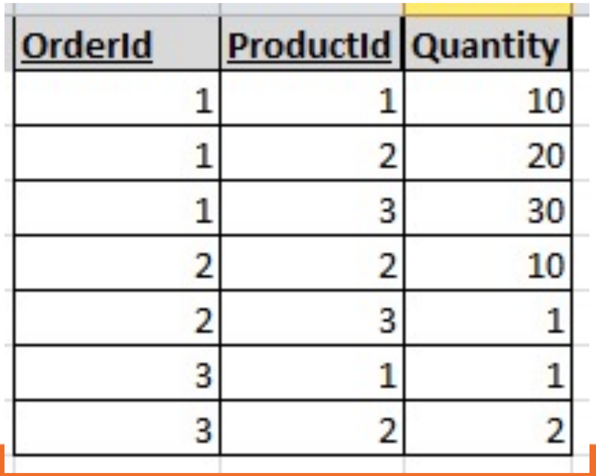
OrderId	CustomerId	Customer Name	Customer Address	Order Date
1	1	Mr. A	A Address	1/1/2012
2	1	Mr. A	A Address	2/1/2012
3	2	Mr. B	B Address	1/1/2012

FD3 – Partial -> new table



ProductId	Product Name	Product Price
1	P1	100
2	P2	200
3	P3	300

FD1 - Full



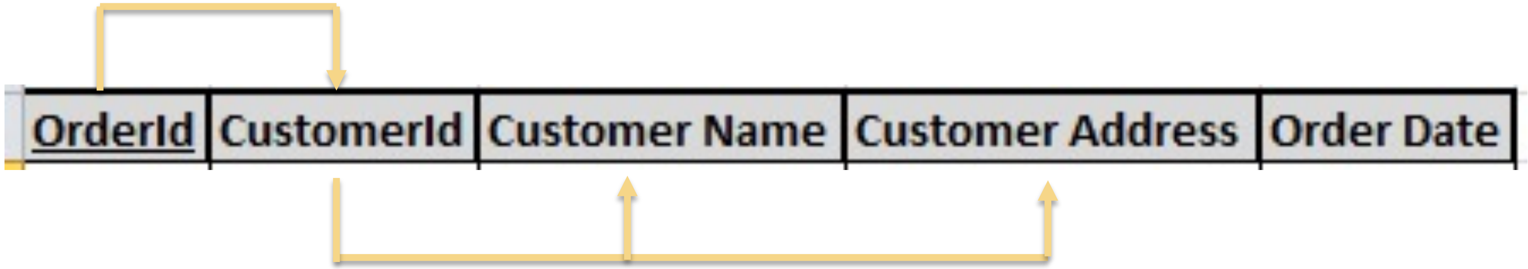
OrderId	ProductId	Quantity
1	1	10
1	2	20
1	3	30
2	2	10
2	3	1
3	1	1
3	2	2

Third Normal Form (3NF)

- 3NF - A relation that is
  - In 2NF and in which no non-primary-key attribute is transitively dependent on the primary key
  - I.e, all non-prime attributes are fully & directly dependent on the PK.

2NF to 3NF

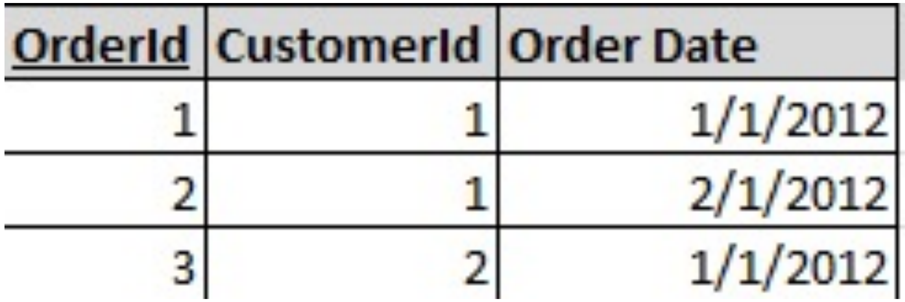
- If transitive dependencies exist
  - – Remove them by placing them in a new relation
  - – Leave a copy of their determinant in the original relation to keep the relationship



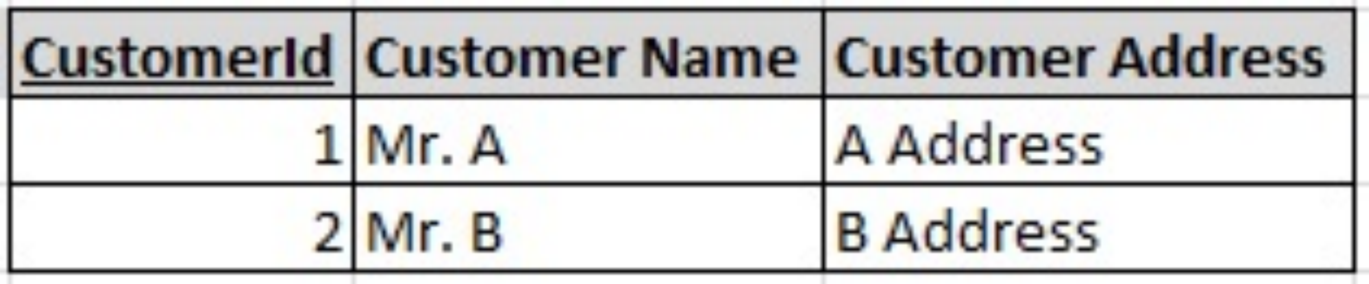
OrderId	CustomerId	Customer Name	Customer Address	Order Date

FD1 - transitively dependent on non-key

Transformation into 3NF



OrderId	CustomerId	Order Date
1	1	1/1/2012
2	1	2/1/2012
3	2	1/1/2012



CustomerId	Customer Name	Customer Address
1	Mr. A	A Address
2	Mr. B	B Address

General Definitions of 2NF & 3NF

- Second normal form (2NF)
  - A relation that is in 1NF and every non-primary-key attribute is fully functionally dependent on any candidate key.
- Third normal form (3NF)
  - A relation that is in 1NF and 2NF and in which no non-primary-key attribute is transitively dependent on any candidate key.

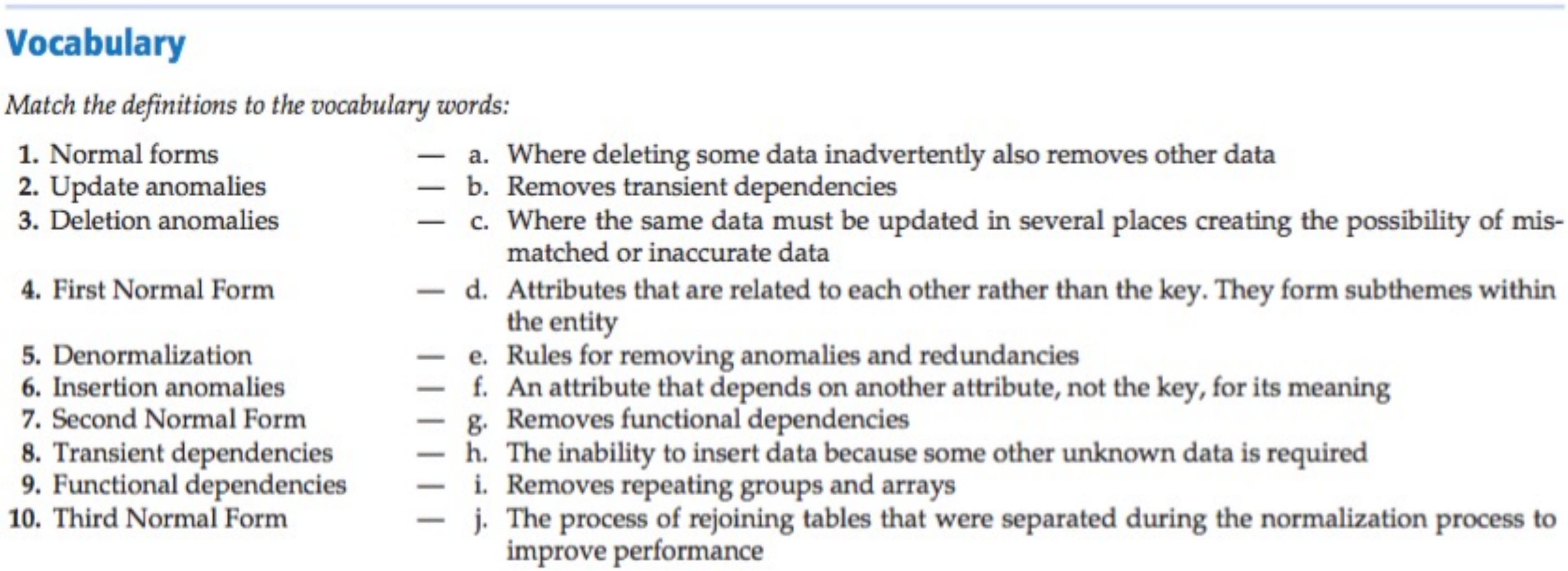
Documentation

- It's useful to keep multiple versions of the ERD
- One change that is often made is about "denormalization" step
  - Some entities that are separated at the normalization now are rejoined
  - This is the trade off between performance and anomalies and data storage
- A DB Design should always be normalized up until 3rd NF
  - Otherwise, designer should have a strong reason for it

Things We Have Done

- Looked at three types of DB anomalies
  - Insert, update, and delete
- Introduced functional dependencies
- Introduced normal forms
- Reviewed DB design for 1NF, 2NF, 3NF
- Reviewed completed DB design

Vocabularies



Vocabulary

Match the definitions to the vocabulary words:

1. Normal forms — a. Where deleting some data inadvertently also removes other data
2. Update anomalies — b. Removes transient dependencies
3. Deletion anomalies — c. Where the same data must be updated in several places creating the possibility of mismatched or inaccurate data
4. First Normal Form — d. Attributes that are related to each other rather than the key. They form subthemes within the entity
5. Denormalization — e. Rules for removing anomalies and redundancies
6. Insertion anomalies — f. An attribute that depends on another attribute, not the key, for its meaning
7. Second Normal Form — g. Removes functional dependencies
8. Transient dependencies — h. The inability to insert data because some other unknown data is required
9. Functional dependencies — i. Removes repeating groups and arrays
10. Third Normal Form — j. The process of rejoining tables that were separated during the normalization process to improve performance

References

- Cogner, S., 2012. *Hands-on Database: An Introduction to Database Design and Development*. Prentice Hall.