Lecture # 13 Normalization

Normalization

- Normalization is a process of decomposing a relations so that we can reduce anomalies and different types of dependencies from the relations.
- Normalization is a staged process, each stage results in a normal form.
- Normalization results in well structured, small relations.

Why we do Normalization?

- There are some main reasons

- The first is to minimize duplicate data.
- The second is to minimize or avoid data modification issues.
- And the third is to simplify queries.
- To remove the anomalies

Steps in Normalization

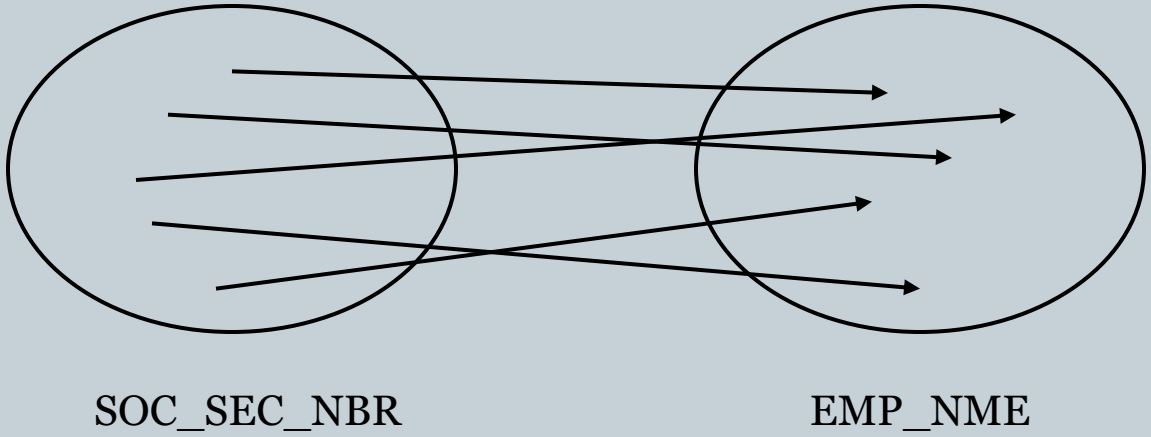
- **Normal Form :** A Normal form is a state that a relation undergoes after applying some sort of rules to eradicate the dependencies and anomalies. There are some normal forms in Normalization.
- **1st Normal Form :** A relation is said to be in 1NF or 1st Normal form if it does not have any repeating groups and multivalued attributes.
- **2nd Normal Form :** A relation is said to be 2NF if it does not have any partial dependency in it.
- **3rd Normal Form :** A relation is said to be 3NF if it does not have any transitive dependency in it.
- **4th Normal Form :** Any remaining multivalued attributes removed.
- **5th Normal Form :** Any remaining anomalies are removed.

Functional Dependency

- Relationship between columns X and Y such that, given the value of X, one can *determine* the value of Y. Written as X $\rightarrow$ Y
- X is called the *determinant* of Y

Functional Dependency

- Example
  - SOC_SEC_NBR → EMP_NME



-One and only one EMP_NME for a specific SOC_SEC_NBR

- SOC_SEC_NBR is the *determinant* of EMP_NME
- EMP_NME is functionally *dependent* on SOC_SEC_NBR

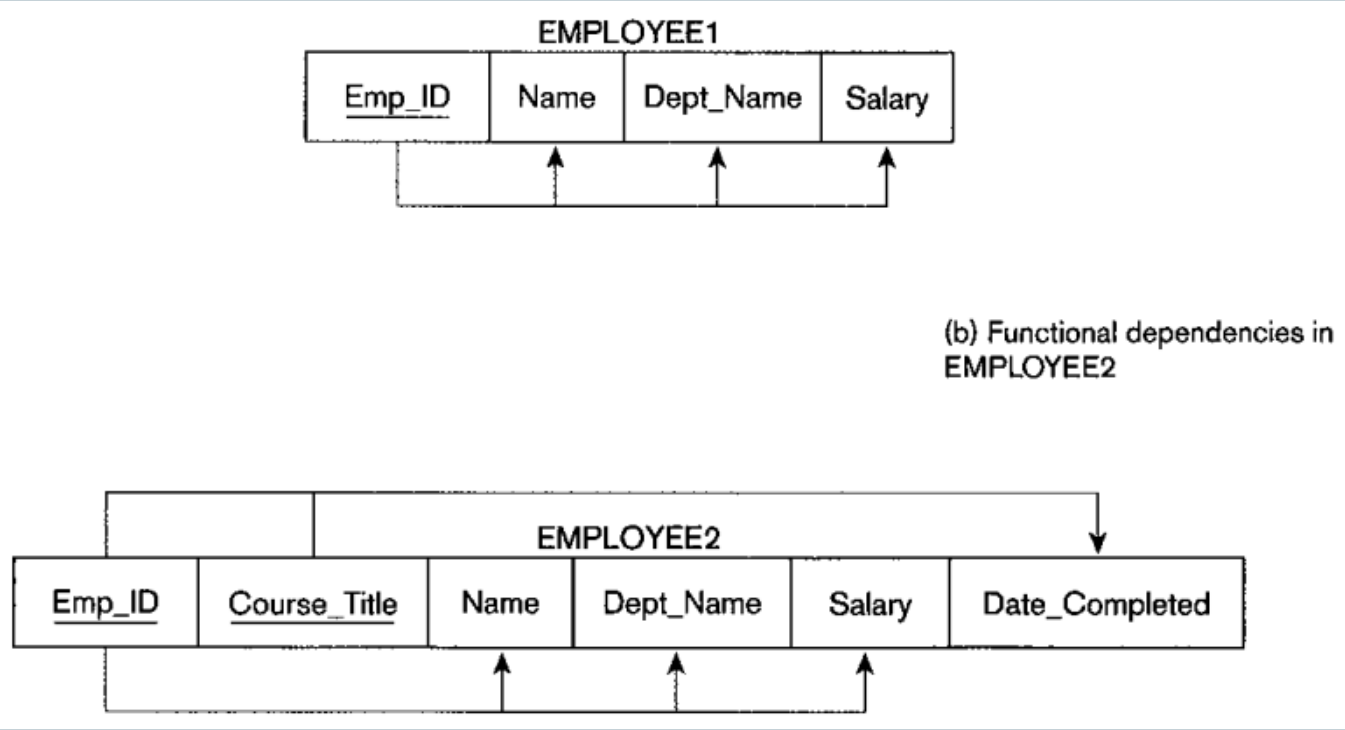
Dependencies (1)

- To understand first we should have sound knowledge of dependencies such as functional dependency, Partial Dependency, Transitive Dependency.
- **Functional Dependency:** Is A Relationship Between Or Among Attributes Such That The Values Of One Attribute Depend On, Or Are Determined By, The Values Of The Other Attribute(s). Such as if R is a relation B and A are attributes then B is functionally dependent on A if , A determines value of B. if is represented as
- $A \longrightarrow B$.
- Attributes on Left side of arrow is called determinants. There may be one or more determinants.
- Empid,CourseTitle $\longrightarrow$ DateCompleted

- **Partial Dependency:** Is A Relationship Between Attributes Such That The Values of non key Attribute Is Dependent on, Or Determined By, The Values of Another Attribute Which Is Part Of The Candidate Key.

Dependencies (2)

- Transitive Dependencies : If a Non-Key Attribute determines the value of another non key attribute then it is called transitive dependency.



1NF

But, didn't we just conclude that COURSE is a 'bad' table (the way it is structured) as it suffers from all the three anomalies we talked about?

So, what's the problem?

COURSE#	SECTION#	C_NAME
CIS564	072	Database Design
CIS564	073	Database Design
CIS570	072	Oracle Forms
CIS564	074	Database Design

Partial Dependency

- Occurs when a column in a table only depends on part of a concatenated key

Example

2NF

- C_Name only depends upon the Course# not the Section#. It is partially dependent upon the primary key.

- A table is in 2NF if it is in 1NF and has no partial dependencies.

2NF

- How do you resolve partial dependency?
  - Decompose the problematic table into smaller tables.
  - Must be a 'loss-less' decomposition. That is, you must be able to put the decomposed tables back together again to arrive at the original information.
  - Remember *Foreign Keys*!

2NF

OFFERED_COURSE

COURSE#	SECTION#
CIS564	072
CIS564	073
CIS564	074
CIS570	072

COURSE

COURSE#	C_NAME
CIS564	Database Design
CIS570	Oracle Forms

2NF

- Are the two (decomposed) tables COURSE and OFFEERED_COURSE are 2NF?
- Do these two tables have any modification anomalies?
  - Can you now readily enter the info that a new approved course CIS563?
  - Can you now delete the section# 072 for CIS570 without losing the info tat CIS570 exists?
  - How many times do you have to change the name of a given course?

Transitive Dependency
Table: Student-Subj-Fee

SID	SUBJ	FEE
101	Oracle	1000
102	Oracle	1000
103	DB2	800
104	DB2	800
105	Sybase	500

Transitive Dependency

- Is the table Student-Dorm-Fee in 2NF?
- Does this table have any modification anomalies?
  - Insertion?
  - Deletion?
  - Update?

Transitive Dependency

•Occurs when a non-key attribute is functionally dependent on
one or more non-key attributes.

Example: HOUSING (SID, SUBJ, FEE)
PRIMARY KEY: SID
FUNCTIONAL DEPENDENCIES:

SID	→	SUBJ
SID	→	FEE
SUBJ	→	FEE

• A table is in 3NF if it is in 2NF and has no transitive dependencies

3NF

- Besides SID, FEE is also functionally dependent on SUBJ which is a non-key attribute.

- A table is in 3NF if it is in 2NF and has no transitive Dependencies.

3NF

- How do you resolve transitive dependency?
  - Decompose the problematic table into smaller tables.
  - Must be a 'loss-less' decomposition. That is, you must be able to put the decomposed tables back together again to arrive at the original information.
  - Remember *Foreign Keys*!

3NF

STUDENT_SUBJ

SID	SUBJ
101	Oracle
102	Oracle
103	DB2
104	DB2
105	Sybase

SUBJ_FEE

SUBJ	FEE
Oracle	1000
DB2	800
Sybase	500

3NF

- Are the two (decomposed) tables STUDENT_SUBJ and SUBJ_FEE in 2NF?
- Are they in 3NF?
- Do these two tables have any modification anomalies?

Repeating Groups

- A repeating group is a domain or set of domains, directly relating to the key, that repeat data across tuples in order to cater for other domains where the data is different for each tuple.
- It is a common problem organizations face, as the same set of information being present in different areas can cause data redundancy and data inconsistency.

Repeating Groups of Data

StudentId	StudentName	Year	Semester	UnitCode	UnitName
0023765	John Doe	2009	2	UG45783	Advance Database
0023765	John Doe	2009	2	UG45832	Network Systems
0023765	John Doe	2009	2	UG45734	Multi-User Operating Systems
0035643	Ann Smith	2009	2	UG45832	Network Systems
0035643	Ann Smith	2009	2	UG45951	Project
0061234	Peter Wolfe	2009	2	UG45783	Advance Database

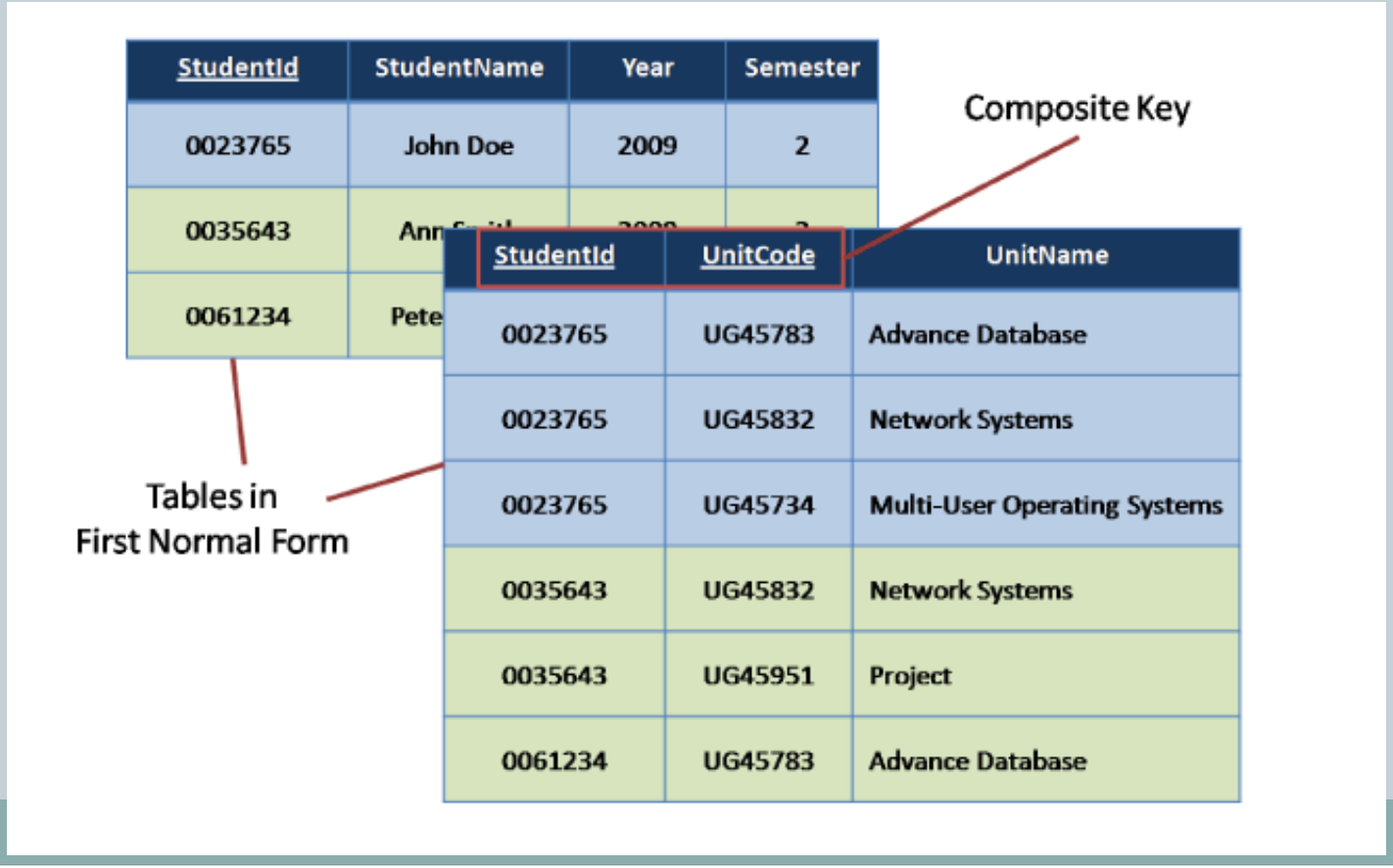
- In this example we see the three domains, StudentName, Year and Semester repeat themselves across the tuples .

- The rules of First Normal Form break this relation into two and relate them to each other so the information needed can be found without storing unneeded data.

- So from our example we would have one table with the student information and another with the Unit Information with the two relations linked by a domain common to both, in this case, the StudentId.

Removal of Repeating Group

- Taking our original example once we have followed these simple steps we have relations that looks like this



Example of Repeating Group

- Suppose that a multinational company has many employees.
- Roger Davis, Tina Martins and Josh Turner are three employees who all work with both the IT and finance departments.
- And their records such as Employee Code, Employee First Name and Employee Last Name are maintained in the records of both the IT and finance departments.
- Therefore, since the records are maintained in the databases of multiple departments, it is a case of repeating groups.

Problems Caused Due to Repeating Groups

- This can cause problems when records need to be updated.
- For example, if Tina Martins gets married and changes her last name to Elton, this means that the records would need to be updated in the databases of both the IT and finance departments.
- This not only results in a bigger effort, but also is fraught with risks, as mistakes in any record updating could cause major problems.
- This thing increases Data Redundancy, which is cause of many other problems.