CSE3103 : Database
FALL 2020

Nazmus Sakib
Assistant Professor
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology

Why Normalization?

- If a table is not properly normalized and have data redundancy then it will not only eat up extra memory space but will also make it difficult to handle and update the database, without facing data loss.
- **Normalization is need to stop the data inconsistency. Three specific reasons that make the problem.**
  - Insert Anomaly
  - Delete Anomaly
  - Update Anomaly

Needs of Normalization

- To understand these anomalies let us take an example of a Student.

Std_id	name	dept	hod	office_tel
401	Adam	CSE	Mr. Tod	53337
402	Becky	CSE	Mr. Tod	53337
403	Ceara	CSE	Mr. Tod	53337
404	David	CSE	Mr. Tod	53337

In the table above, we have data of 4 Computer Sci. students. As we can see, data for the fields dept, hod(Head of Department) and office_tel is repeated for the students who are in the same department in university, this is **Data Redundancy**.

Insertion Anomaly

- **Suppose for a new admission, until and unless a student opts for a dept, data of the student cannot be inserted.**
- **Else we will have to set the branch information as NULL.**

Also, if we have to insert data of 100 students of same branch, then the branch information will be repeated for all those 100 students.

Needs of Normalization

- To understand these anomalies let us take an example of a Student.

Std_id	name	dept	hod	office_tel
401	Adam	CSE	Mr. Tod	53337
402	Becky	CSE	Mr. Tod	53337
403	Ceara	CSE	Mr. Tod	53337
404	David	CSE	Mr. Tod	53337

In the table above, we have data of 4 Computer Sci. students. As we can see, data for the fields dept, hod(Head of Department) and office_tel is repeated for the students who are in the same department in university, this is **Data Redundancy**.

Updation Anomaly

- What if Mr. Tod leaves the University? or is no longer the HOD of CSE department?
- In that case all the student records will have to be updated, and if by mistake we miss any record, it will lead to data inconsistency.

Needs of Normalization

- To understand these anomalies let us take an example of a Student.

Std_id	name	dept	hod	office_tel
401	Adam	CSE	Mr. Tod	53337
402	Becky	CSE	Mr. Tod	53337
403	Ceara	CSE	Mr. Tod	53337
404	David	CSE	Mr. Tod	53337

In the table above, we have data of 4 Computer Sci. students. As we can see, data for the fields dept, hod(Head of Department) and office_tel is repeated for the students who are in the same department in university, this is **Data Redundancy**.

Deletion Anomaly

- In Student table, two different information are kept together, Student information and department information.
- End of the academic year, if student records are deleted, we will also lose the branch information.

Normal Forms

- First Normal Form(1NF)
- Second Normal Form(2NF)
- Third Normal Form(3NF)
- Boyce & Codd Normal Form (BCNF)
- Forth Normal Form(4NF)
- Fifth Normal Norm(5NF)
- Sixth Normal Form(6NF)

Steps in Normalization

- Specify the **PRIMARY KEY** of the table.
- Specify the **Functional Dependency** of the table.
  - Determinant
  - Objects of Determinant
- Apply the rules of the each normal form (*starting from 1NF*).
- If the table fails to meet the rules and conditions of the normal form, **CHANGE** the table until it meets the rules and conditions.
- **RE-TEST** the **MODIFIED/NEW** tables to ensure they meet the rules and conditions of the each normal form.


DBMS
data
database
distributed
parallel
shared
system
architecture
Distributed
DBMSs
DDBMS
network
processing
performance
systems
processors
access
centralized
memory
multiple
sites
nothing
processor
Parallel
single
sometimes
local
may
fragments
typically
require
physically
users
site
applications
computers
number
disk
computer
stored
logically
technology
Shared
application
bottleneck
interconnection
Figure
organizational
management
transparent
provide
decentralized
large
islands
one
split
computing
also
replicated
make
directly
disks
remote
defined
controlled
based
structure
collection
provides
global
associated
partitioned
used
least
must
System
capable
nodes
definition
development
logical
distinction
improve
approach
scalability
control
Database
like
machines
within
reliability
across
fact
major
eliminates
distribution
communication
referred
operational
architectures
scalable
order
resources
com
Date
many
discuss
often
Management
makes
per
dis
user
incompatible
way
consists
administered
mode
databases
share