CSE3103 : Database
FALL 2020

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

Normalization

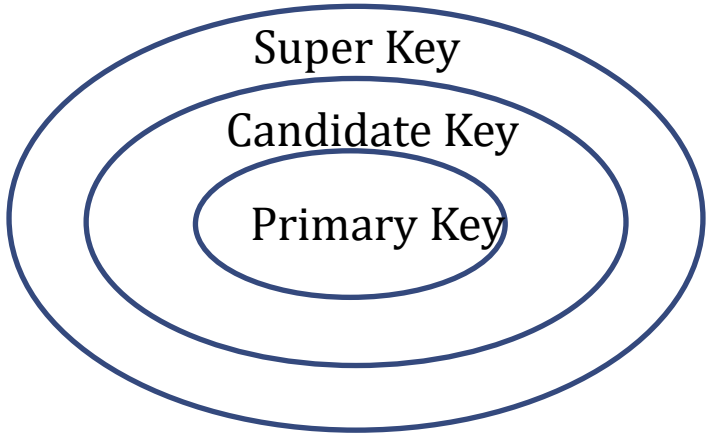
- Database Normalization is a technique of organizing the data in the database.
- Normalization is a systematic approach of decomposing tables to eliminate data redundancy(repetition) and undesirable characteristics like Insertion, Update and Deletion Anomalies.
- Normalization is used for mainly two purposes
  - Eliminating redundant(useless) data.
  - Ensuring data dependencies make sense i.e data is logically stored.

Candidate Key

- A Candidate Key is an attribute (possible composite) that can be used to uniquely identify each tuple in a relation.
- A relation may have more than one candidate key.
- If so, one candidate key is nominated as the primary key for that relation.
- Recall chapter ERD for the more details.

Keys:

1. Super Keys: All the attributes can for super keys.
    1. Suppose, A = { 1, 2, 3, 4}
    2. So, P(A) = {{1},{2},{3},{4},{1,2},{2,3},{3,4} ……{1,2,3,4}, {}}
2. Candidate Key: Super keys which value can be minimal.
3. Primary Key: Candidate key which value is unique.
4. Alternative Key: Keys which are candidate key but not in primary key.
5. Composite Key: More than one attribute jointly declare as primary key is the composite key.
6. Foreign Key: Primary key which is used another table as reference is foreign key.



Functional Dependency

- Functional dependency (FD) is a set of constraints between two attributes in a relation.
- Functional dependency says that if two tuples have same values for attributes A1, A2,..., An, then those two tuples must have to have same values for attributes B1, B2, ..., Bn.
- Functional dependency is represented by an arrow sign (→) that is, X→Y, where X functionally determines Y.
- **The left-hand side attributes determine the values of attributes on the right-hand side.**

Functional Dependency

- **Determinant**
- **Objects of Determinant**
- Example:

  - Student_ID → {First_Name, Last_Name}

  - Course_No → Course_Name, Faculty_Name

  - Course_No, Course_Name → Faculty_Name

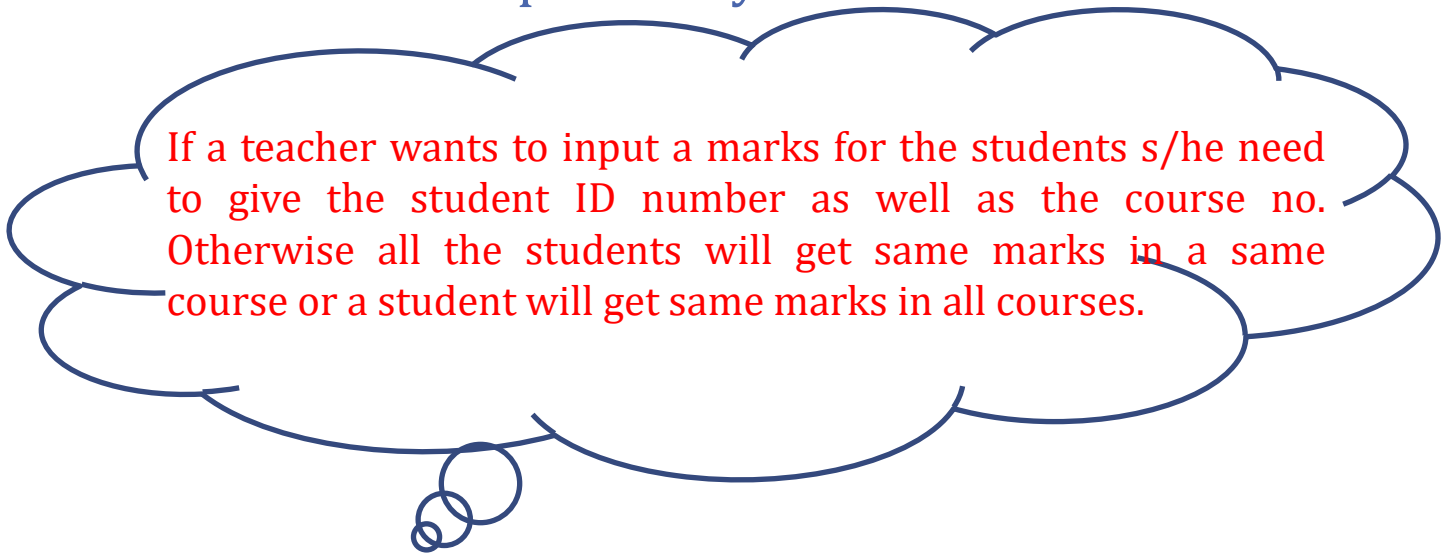
  - Marks → Grade

Partial Functional Dependency

- If an attribute is removed from the composite determinant but the dependency is not impacted, then it will be Partial Functional dependency.
- Example:
  - Course_No, Course_Name → Faculty_Name

Full Functional Dependency.

- If an attribute is removed from the composite determinant but the dependency is impacted, then it will be Full Functional dependency.
- Example:



- **Student_ID, Course_No → Marks**

Transitive Functional Dependency

- If some attributes is created a chain of dependency or a cycle among them then that is created a transitive dependency.
- Example:

R is a relation with attribute A, B, C

If, $A \rightarrow B$

$B \rightarrow C$

Then, $C \rightarrow A$


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