Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology
(Deemed to be University Estd. u/s 3 of UGC Act, 1956)
School of Computing
B.Tech. – Computer Science and Engineering

<div style="text-align:center">**VTR UGE2021- (CBCS)**</div>

Academic Year: 2025–2026

SUMMER SEMESTER -SS2526

Course Code   : 10211CS207

Course Name : Database Management Systems

Slot No   : S1L4

DBMS TASK - 10 REPORT

Title: Normalizing databases using functional dependencies upto Third Normal Form

Submitted by:

VTUNO	REGISTER NUMBER	STUDENT NAME
VTU29773	24UECS0556	G.Shasikala

Task 10

Normalizing databases using functional dependencies upto Third Normal Form

Aim: To normalize the below relation and create the simplified table with suitable constraint.

CricketBoard(BoardID, Name, Address, Contact_No, TeamID, TName, Coach, Captain, PlayerID, PFName, PLName, Age, PDateofBirth, PlayingRole, email, contact_no, Batting, Bowling, MatchID, Match_Date, Time1, Result, GroundID, GName, Location, Capacity, UmpireID, UFName, ULName, UAge, UDateofBirth, Country, Uemail, Ucontact_no).

   a) Apply the functional dependency, normalize to 1NF
   b) Normalize the relations using FD+ and α+.
   c) Find the minimal cover, canonical cover.
   d) Normalize to 2NF, add/alter constraints if necessary.
   e) Normalize to 3NF, add/alter constraints if necessary.

Procedure:

Normalize the given relation and create simplified tables with suitable constraints, we need to identify the functional dependencies and separate them into different tables. Normalization involves breaking down the data into smaller, related tables to minimize data redundancy and maintain data integrity. Let's identify the functional dependencies:

Functional Dependency:

BoardID → Name, Address, Contact_No

TeamID → TName, Coach, Captain

PlayerID → PFName, PLName, Age, PDateofBirth, PlayingRole, email, contact_no, Batting, Bowling

MatchID → Match_Date, Time1, Result, GroundID

GroundID → GName, Location, Capacity

UmpireID → UFName, ULName, UAge, UDateofBirth, Country, Uemail, Ucontact_no

Now, we can create simplified tables:

CricketBoard (BoardID [PK], Name, Address, Contact_No)
CricketTeam (TeamID [PK], TName, Coach, Captain)
CricketPlayer (PlayerID [PK], TeamID [FK], PFName, PLName, Age, PDateofBirth,
PlayingRole, email, contact_no, Batting, Bowling)
CricketMatch (MatchID [PK], TeamID [FK], Match_Date, Time1, Result, GroundID
[FK])
CricketGround (GroundID [PK], GName, Location, Capacity)
CricketUmpire (UmpireID [PK], UFName, ULName, UAge, UDateofBirth, Country,
Uemail, Ucontact_no)

In these tables, [PK] denotes the primary key, [FK] denotes the foreign key, and suitable
constraints should be added to maintain data integrity.

Create tables for all non-prime attributes using α+

α+ (Alpha Plus) allows to group attributes based on their functional dependencies and
candidate keys. And create tables for each set of attributes that functionally depend on a
candidate key. The candidate keys in this case are BoardID, TeamID, PlayerID, MatchID,
and UmpireID.

CricketBoard Table: BoardID (PK), Name, Address, Contact_No
Team Table: TeamID (PK), TName, Coach, Captain
Player Table: PlayerID (PK), TeamID (FK), PFName, PLName, Age, PDateofBirth,
PlayingRole,
Email, contact_no, Batting, Bowling
Match Table: MatchID (PK), TeamID (FK), Match_Date, Time1, Result
Ground Table: GroundID (PK), GName, Location, Capacity
Umpire Table: UmpireID (PK), UFName, ULName, UAge, UDateofBirth, Country,
Uemail, Ucontact_no

Create additional tables to represent transitive dependencies.

Already addressed transitive dependencies in previous normalization steps by introducing
the MatchVenue table for the transitive dependency between MatchID and GroundID
through the Result attribute.

MatchVenue Table: MatchID (PK, FK), GroundID (FK)

First Normal Form:

The given relation into the First Normal Form (1NF), to need to ensure that each attribute (column) contains atomic (indivisible) values, and there are no repeating groups or arrays.

Based on the provided relation, it appears that each attribute already contains atomic values, so there are no repeating groups to eliminate.

Second Normal Form:

To determine whether the given relation is in the Second Normal Form (2NF), we need to check two conditions:

The relation must already be in 1NF (First Normal Form).

All non-prime attributes (attributes not part of any candidate key) must be fully functionally dependent on the entire primary key.

First, let's identify the potential candidate key(s) from the given relation based on functional dependencies:

It appears that the potential candidate keys could be:

1. BoardID
2. TeamID
3. PlayerID
4. MatchID
5. UmpireID

Next, we need to check if all non-prime attributes are fully functionally dependent on their respective candidate key(s).

Third Normal Form:

To determine whether the given relation is in the Third Normal Form (3NF), need to check two conditions:

1. The relation must already be in the Second Normal Form (2NF).
2. There should be no transitive dependencies between non-prime attributes and candidate keys.

The given relation satisfies the conditions of the Second Normal Form (2NF). Now, let's check for transitive dependencies:

Now, let's analyze each functional dependency and check for transitive dependencies:

BoardID → Name, Address, Contact_No

There are no transitive dependencies in this case, as Name, Address, and Contact_No are directly dependent on BoardID.

TeamID → TName, Coach, Captain

There are no transitive dependencies here either, as TName, Coach, and Captain are directly dependent on TeamID.

PlayerID → PFName, PLName, Age, PDateofBirth, PlayingRole, email, contact_no, Batting, Bowling

There are no transitive dependencies for PlayerID, as all the mentioned attributes are directly dependent on PlayerID.

MatchID → Match_Date, Time1, Result, GroundID

There is a transitive dependency between MatchID and GroundID through the Result attribute. To resolve this, we create a new table called MatchVenue:
MatchVenue (MatchID [PK], GroundID [FK])

GroundID → GName, Location, Capacity

There are no transitive dependencies for GroundID, as GName, Location, and Capacity are directly dependent on GroundID.

UmpireID → UFName, ULName, UAge, UDateofBirth, Country, Uemail, Ucontact_no

There are no transitive dependencies for UmpireID, as UFName, ULName, UAge, UDateofBirth, Country, Uemail, and Ucontact_no are directly dependent on UmpireID.

With the introduction of the MatchVenue table to resolve the transitive dependency, the relation now satisfies the conditions of the Third Normal Form (3NF).

Result:

Thus the normalization of the given relation is created the simplified tables with suitable constraint successfully.