

9/2/25 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.

Cricket Board (BoardID, Name, Address, Contact-No, TeamID, ATName, coach, captain, PlayerID, PFName, PLName, age, PDate of Birth, Playing Role, email, contact-No, Batting, Bowling, MatchID, Match-Role, Time1, Result, GroundID, GName, location, Capacity, UmpireID, UName, UAge, UDate of Birth, Country, Uemail, UContact-no).

- a) Apply the functional dependency, normalized to 1NF
- b) Normalize these relations using FD + constraint.
- 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 \rightarrow Name, Address, Contact-No

TeamID \rightarrow Name, coach, captain

~~Cricket~~ Player

PlayerID \rightarrow PFName, PLName, age, PDate of Birth, Playing Role, email, contact-no
Batting, Bowling

MatchID \rightarrow Match-Role, Time1, Result, GroundID

Ground ID → Ground Name, Location, Capacity.

Umpire ID → UName, UName, UAge, UDate of Birth, Country, UEmail, UContact - no

Note, we can create simplified tables:

Cricket Board (BoardID [PK], Name, Address, Contact - no)

Cricket Team (TeamID [PK], UName, Coach, Captain)

Cricket Player (PlayerID [PK], TeamID [FK], PName, Age, PDate of Birth, Playing Role, Email, Contact - no, Batting, Bowling)

Cricket Match (MatchID [PK], TeamID [FK], Match - Date, Time, Result, Ground ID [FK])

Cricket Ground (GroundID [PK], Ground Name, Location, Capacity)

Cricket Umpire (UmpireID [PK], UName, UName, UAge, UDate of Birth, Country, UEmail, UContact - no)

Create tables for all non-prime attributes using α+
α+ (Alpha Plus) allows to group attributes based on their functional dependency on a candidate key. The candidate keys in this case are Board ID, Team ID, Player ID, Match ID, and Umpire ID.

Board Table: BoardID (PK), Name, Address, Contact - no

Team Table: TeamID (PK), PName, coach, captain

Player Table: PlayerID (PK), PName, TeamID (FK), PLName, Age, PDate of Birth, Playing Role, Email, Contact - no, Batting, Bowling

Match Table: MatchID (PK), TeamID (FK), Match - Date, Time, Result

Ground Table: GroundID (PK), UName, Location, Capacity

Umpire Table: UmpireID (PK), UName, UName, UAge, UDate of Birth, Country, UEmail, UContact - no

First normal form:

To convert the given relation into the first normal form (1NF), we need to ensure that each attribute (column) contains atomic (indivisible) values, and there are no repeating group of attributes.

Second normal form:

To determine whether the given relation is in the second normal form (2NF), we need to check two conditions:

All non-prime attributes (attributes not part of any candidate key) must be fully functional dependent on the entire primary key. It appears that the potential candidate key could be

- 1) Board ID
- 2) Team ID
- 3) Player ID
- 4) Match ID
- 5) EmpID

Third normal form:

To determine whether the given relation is in the third normal form (3NF), we 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.

Board ID \rightarrow Name, Address, Contact-NO

There are no transitive dependencies in this case, as Name, Address, and Contact-NO are directly dependent on Board ID.

Team ID → Name, Coach, Captain

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

Player ID → Name, Player No, Date of Birth, Playing Role, email, Contact no, Batting, Bowling.

Ground ID → Name, Location, Capacity

There are no transitive dependencies for Ground ID, as Name, Location and Capacity are directly dependent on Ground ID with the introduction of the Match venue table to resolve the transitive dependency to the relation now satisfies the conditions of the third normal form (3NF).

VEL TECH - CSE	
EX NO.	18
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	5
TOTAL (20)	20
● N WITH DATE	✓

Result: Thus the normalization of the given relation is created the simplified tables with suitable constraints successfully.