09/9/24

Task 10

Normalizing databases using functional dependacies 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, TName, Coach, Captain, PlayerID, PFName, PLName, Age, Date of Birth, Playing Role, email, Contact_no, Batting, Bowling, MatchID, Match-Dat, Time1, Result, GroundID, GName, Location, Capacity, UmpireID, UFName, ULName, UAge, UDateofBirth, Country, Uemail, Ucontact, Uemail, Ucontact_no).

Procedure:

Normalize the given relation and create simplified table with suitable constraints, we need to identify the function dependencies and seperate them into diffent tables. Normalization involves breaking down the data into smaller relating tables to minimize data redundancy and maintain an data integrity. Let's identify the functional dependencies.

Functional Dependency:

BoardID → Name, Address, Contact_NO

TeamID → TName, Coach, Captain

PlayerID → PFName, PLName, Age, PDate of Birth, Playing Role, email, Contact_no, Batting, Bowling.

MatchID → Match-Dat, 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)

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

Cricket Player (PlayerID[PK], TeamID[FK], PFName, PLName, Age, PDate of Birth, Playing Role, email, Contact_no, Batting, Bowling.

uDateofBirth, Country, Uemail, uContact_no)

Create tables for all non-prime attributes using 4+.
4+ (Alpha plus) allows to group attributes based on their on their function dependencies and candidate keys. And create table for each set of attributes that functionally depend on candidate key. The candidate key in this case are BoardID, TeamID, PlayerID, MatchID, and umpireID

BoardTable: Board(PK), Name, Address, Contact_No
TeamTable: TeamID(PK), TName, Coach, Captain.
PlayerTable: PlayerID(PK), TeamID(FK), PFName, PLName, PAge, PDateofBirth, PlayingRole, Email, Contact_no, Batting, Bowling
MatchTable: MatchID(PK), TeamID(PK), Match_Date, Time1, 1
Result:
GroundTable: GroundID(PK), GName, Location, Capacity
umpireTable: UmpireID(PK), UFName, ULName, UAge, UDateofBirth
PlayingRul.

Email, Contact_no, Batts, Bowling

UDateofBirth, Country, Uemail, Contact_no

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 is in the First Normal Form (1NF), to need two conditions:

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

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

First, Lets 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 (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, Lets check for transitive dependencies:

Now, Let's analyze each functional dependence and check for transitive dependency:

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 - No are directly dependent on BoardID.

TeamID → TName, Coach, Captain.

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 transitive dependencies for GroundID, as GName, Location, and Capacity are directly dependent on 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 Match Venue table to resolve the transitive dependency, the relation now satisfies the conditions of the Third Normal Form(3NF).

VEL TECH CSE	
EX NO	10
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	
RECORD (5)	
TOTAL (20)	20

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