30/9/25

TASK 10

NORMALIZING DATABASES USING FUNCTIONAL DEPENDENCIES UPTO THIRD NORMAL FORM

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

Cricket Board (BoardID, Name, Address, Contact_No, TeamID, TName, Coach, Captain, PlayerID, PFName, PLName, Age, pDate of Birth, Time, Result, GroundID, GName, Location, Capacity, UmpireID, UFName, ULName, UAge, UDate of Birth, Country, Uemail, Ucontact_no)

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

Procedure:

Normalize given relation and create simplified tables suitable constraints, we need to identify functional dependencies seprate them into different tables.
Normalization involves breaking down into smaller.
Let's identify 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_Data, Time, Result, Ground ID
GroundID → GName, Location, Capacity
UmpireID → UFName, ULName, UAge, UDate of Birth, Country, Uemail, Ucontact_no

Now, we can create simplified tables:

Crick Board (Board ID[PK], Name, Address, Contact-No)

Cricket Team (Team ID[PK], T name, Coach, Captain)

Cricket player (player ID[PK], Team ID [FK], PF Name, PL name, Age, P Date of Birth, Playing Role, email, Contact-no, Batting, Bowling)

Cricket match (Match ID [PK], Team ID [FK], Match-Date, Time 1, Result, Ground ID

Ground ID → G Name, Location, capacity

Umpire ID → UF Name, U Age, U Date of Birth; Coun U email, U contact no

Now, we can create simplified tables:

Cricket Board (Board ID[PK], Name, Address, contact-No

Cricket Team (Team ID [PK], Tname, Coach, Captain)

Cricket player (player ID [PK], Team ID [FK], PF Name, PL Name, Age, P Date of Birth, PL Name, Age. PDate of Birth, Playing Role, email, contact-no, Batting, Bowling)

Cricket match (Match ID [PK], Team ID [FK], Match-Date, Time 1, Result, Ground ID [FK])

Cricket Ground (Ground ID [PK], GName, Location, capacity)

Cricket umpire (umpire ID [PK], UF Name, UAge, U Date of Birth

CREATE TABLE: BOARD ID (PK), Name, Address, contact

Team table: Team ID (PK), TName, Coach, captain player

Table: player ID (PK), Team ID (FK), PF Name, Age, pDate of Birth, Playing Role, Email, Contact_NO, Batting, Bowling

Match table: Match ID (PK), Team ID (FK), Match_Dat, Time 1, Result

Ground table: Ground ID (PK), G Name, Location, Capacity.

Umpire table: Umpire ID (PK), UF Name, UL Name, Uge, U Date of Birth, Country, Uemail, Ucontact-no.

Create table for all non-prime attributes using α^+

α^+ (Alpha plus) allows group attributes based on their function dependencies candidate keys, and create tables for each set of attributes functionally depend on candidate key. candidate keys in case are Board ID, Team ID, player ID, Match ID and umpire ID.

Create additional tables to represent transitive dependencies Already addressed transitive dependencies previous normalization steps by introducing match venue table for transtive dependency Match ID and ground ID through Result attribute.

Match venue: Match ID (PK, FK), Ground ID (FK)

First normal form:

The given Relation into First normal forms (1NF), need to ensure each attribute contains atomic (indivisible values, and there are no Repeating groups or arrays.

Based provided Relation, it appears each attributes already contains atomic values, so there no Repeating groups to eliminate

SECOND Normal Form:

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

The Relation must already be in 1NF (First normal form).

All non-prime attributes must be fully functionally dependent on entire primary key

First identify potential candidate key(s) from given relation based on functional dependencies

It appears that potential candidate keys could

1. Board ID
2. Team ID
3. Player ID
4. Match ID
5. Umpire ID

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

Third Normal Form:

To determine whether given relation is third normal form (3NF) need to check two conditions

1. The relation must already in second normal form (2NF)
2. There should be transitive dependencies between non prime attributes and Candidate keys

The given Relation satisfied condition second normal form

Now check transitive dependencies.

Board ID → Name, Address, Contact_No.

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

TeamID → TName, Coach, Captain

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

Match ID → Match_Date, Time1, Result, Ground ID.

There are no transitive dependencies between match ID and Ground ID through Result attribute.

Match venue (match ID [PK], Ground ID [FK])

There are no transtive dependencies for umpire ID as UF Name, UL Name, U Age, U Date of Birth, Country, Uemail, and U contact-no are directly dependencies on umpire ID.

with introduction of match venue table to resolve the transtive dependancy, the Relation now satisfies the condition of third Normal form (3NF)

There are no transtive dependencies for groundID, as G name Location, and capatainy are directly dependent on Ground ID

Umpire ID → UF-Name, ULName, UAge, UDate of Birth, Country, Uemail, U Contact-no.

VEL TECH	
[illegible]	
PERFORMANCE (5)	
RESULT AND ANALYSIS (5)	
VIVA VOCE (5)	
RECORD (5)	
TOTAL (25)	
SIGNATURE DATE	

Result: Thus the normalization of given relation is created the simplied tables with suitable constrain successfully.