29-7-25

TASK 2

Generating Design of other tradition database model

Aim: Creating Hierarchical / Network model of the database by enhansing the sound abstract data by performing following tasks using forms of inheritence:

a. Identify specify of each relationship, find and form surplus relations.

b. check is-a hierarchy / has-a hierarchy and performs generalization and/or specilization

c. Find domain of attribute and perform check constraint to the applicable

d. Rename the Relations

e. Perform SQL Relations using DDL, DCL commands.

a. Identify specificity of each relationship, find and form ~~sets~~ surplus relations.

Entity Identification:

- Cricket Board ↔ Team → one-to-many
- Team ↔ player → many-to-many → team-player
- Match ↔ Team → Many-to-Many → many-team
- Match ↔ Orround → one-to-one

b. Check is-a hierarchy / has-a hierarchy and performs generalization and/or specialization relationship

Generalization

In the ER diagram for Tamilnadu Cricket Board (TNCA) described earlier, we can identify potential generalization based on common attributes or relationships among entities. Here's an example of possible generalization.

Entities:

Player

Umpire

Attributes:

The above entities have common attributes like First_Name, Last_Name, Date-of-Birth, age, Contact_NO, and Email.

Potential Generalization:

Create superclass called "Person" to represent common attributes shared by player and umpire. The "person" entity would have following attributes.

Person_ID (primary key)

First_Name

Last_name

Date-of-Birth

Age

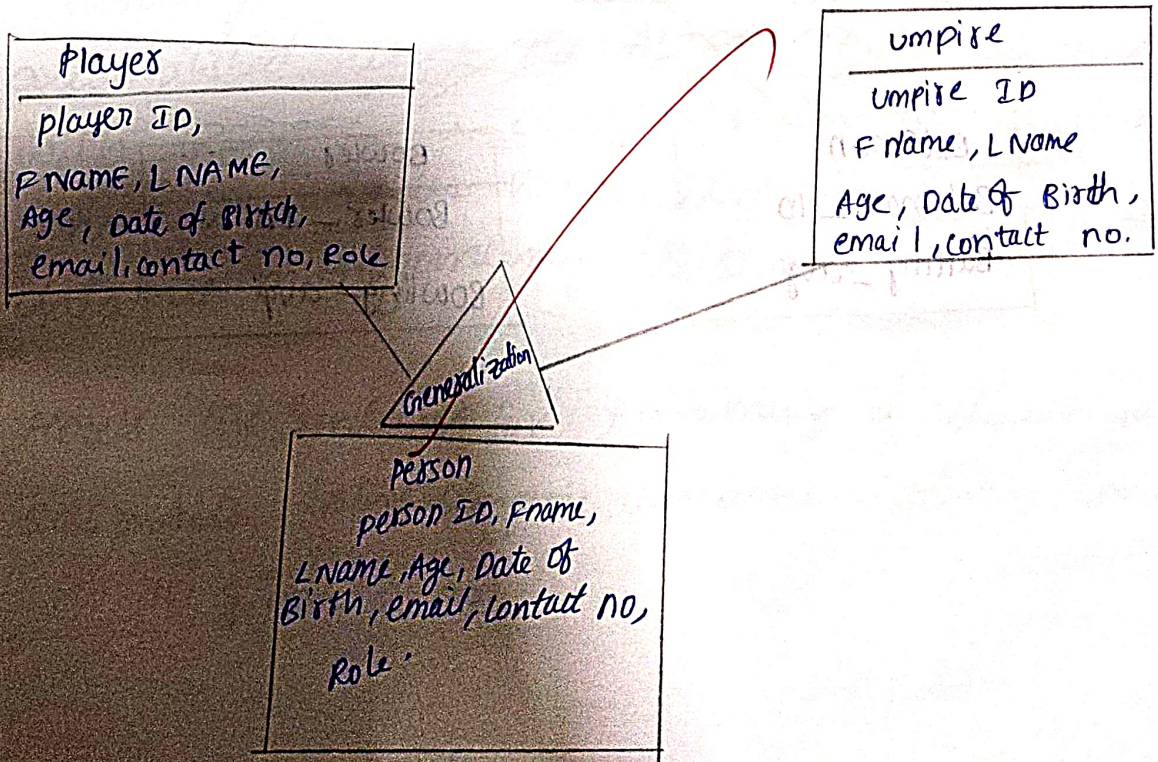
contact_number

Email

subclasses:

Player: Inherited attributes from "person" and add specific attributes like player_ID.

Umpire: Inherited attributes from "person" and add specific attributes like umpire_ID.



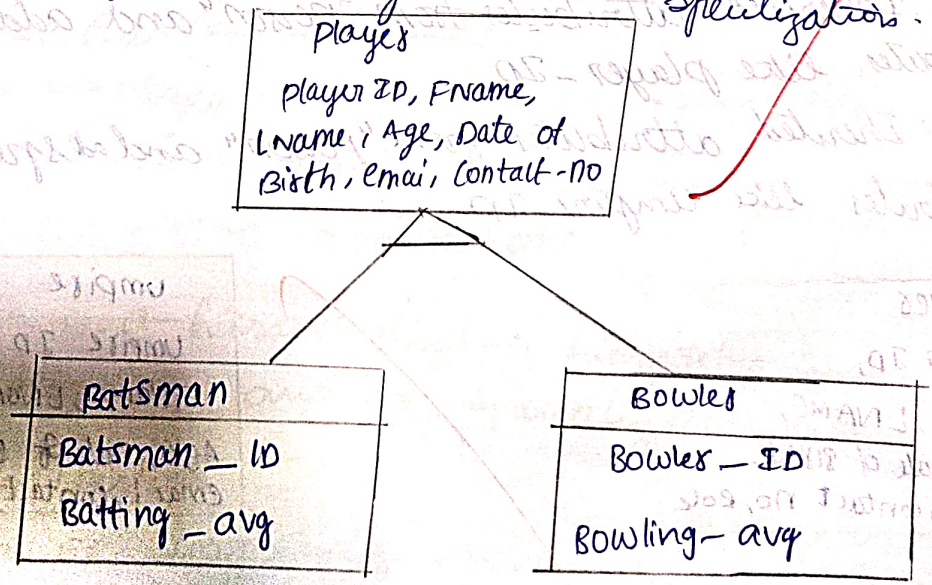
By using generalization, we can reduce data redundancy, enprove data intigrity, and simplify the structure of ER diagram. This approach also allows for easier maintanence and updates as changes made to attribute shared by all "person" entities will be automatically reflected in the subclasses.

Specialization

In the context of Entity-Relationship (ER) diagrams, Specialization refers the prous of defining subtypes within entitily type. It allows to represent entities that have specific attributes or relationships distinct from general attributes or relationships of parent entity.

In case of Tamilnadu cricket Board Association. let's Consider the specialization of "player" entity into two subtypes: "Batsman" and "Bowler." This specialization is based on specific roles that players can have in cricket

Here's modified ER diagram with specilization.



2.c) Find domain of attribute and perform check constraint to the applicable.

Attribute	Domain	check Constraint Example
Age	Integer	Check (Age>=18) ~~check constraint Example~~
contact_NO	varchar(10-15)	check (length (contact_NO) Between 10 and 15)
Email	varchar	check (Email like '%.@ %, %.')
Capacity	Integer	check (capacity>70)
playing Role	VARCHAR	Check (playingRole IN ('Batsman', 'Bowler', 'All-Rounder', 'cricket keeper'))

SQL> Alter table ~~Umpire~~ player ADD CONSTRAINT Check_con CHCK (age >=18);

Table altered.

2d Rename the Relations:

Renaming a table (relation) in SQL can be accomplished using the ALTER TABLE statement with RENAME TO clause. The specific syntax for renaming tables varies slightly between different database management systems.

Here's the syntax for renaming a column in table

SQL> Alter table umpire Rename column contact_no TO phone_no;

Table altered

SQL>DESC umpire

Name	NULL.?	TYPE
Umpireid		VARCHAR2(10)
FName		VARCHAR2(30)
LName		VARCHAR2(30)
Age		Number(5,2)
Date of Birth		Date
Country		VARCHAR2(30)
Email		VARCHAR2(40)
Phone_NO		Number

2.e perform SQL Relations Using DDL, DCL commands

DCL stands for "Data control language," which is a subset of SQL (structured QUERY language) used to control access to data in database. DCL commands are responsible for managing user permissions, granting privileges, and controlling data security within a database system. There are two primary DCL commands

1. Grant
2. Revoke

GRANT:

The GRANT Command is used to provide specific privileges to users or roles, allowing them to perform certain action on database objects (eg tables, views, procedures) privileges may include SELECT, INSERT, UPDATE, DELET, EXECUTE, and more.

SQL>create user raj identified by kumar;
User created.
SQL>grant resource to raj;
Grant succeeded.
SQL>grant create session to raj;
Grant succeeded.
SQL>conn
Enter user-name: raj
Enter password:
Connected.
SQL>create table emp(eno number, ename varchar(10));
Table created.
SQL>conn system/manager
Connected.
SQL>grant all on empire to raj;
Grant succeeded.

VEL TECH

EX NO	2
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	5
TOTAL (20)	15
SIGN WITH DATE	29/7/25

Result: Thus the Hierarchical model and Network model has been succesfully created.