# Task-2: Generating Design for other traditional database Model.

Aim: Creating Hierarchial / network model of the database by Enhancing the sound abstract data by Performing following tasks using forms of inheritance.

A. Identify the specificity of Each relationship, find and form the surplus relations.

Entity identification:

— Cricket Board has multiple teams.

— Team consists of multiple players.

— Match involves multiple teams and is played on a ground

— Umpire supervise the match.

Specificity Analysis:

— Cricket Board ⟷ team → one-to-many

— team ⟷ player → Many-to-Many → team - Player.

— Match ⟷ team → may-to-may → Match - Team

— Match ⟷ Ground → one-to-one.

Surplus Relations (Associative tables):

— Team - Player (Team ID, Player ID)

— Match - Team (match ID, Team ID)

B. check is a hierarchy has-a hierarchy and Performs generalization and /or specialization relationship.

Generalization: In the ER diagram for the tamil Nadu cricket Board (TNCA) described Earlier, we can identify Potential of the generalizations based on common attributes or relationships among entities. Here's an Example of a Possible generalization.

Entities:

Player

Umpire.

Attributes:

The above Entites have common attributes like first_Name Last_Name, last_Name, Date_of_Birth, age, contact_No and Email.

## Potential Generalization:

creates a super class called "Person" to represent the common attributes shared by player and umpire. The "Person" entity would have the following attributes:

Person_ID (Primary key)

First_Name

Last_Name

Date_of_Birth

Age

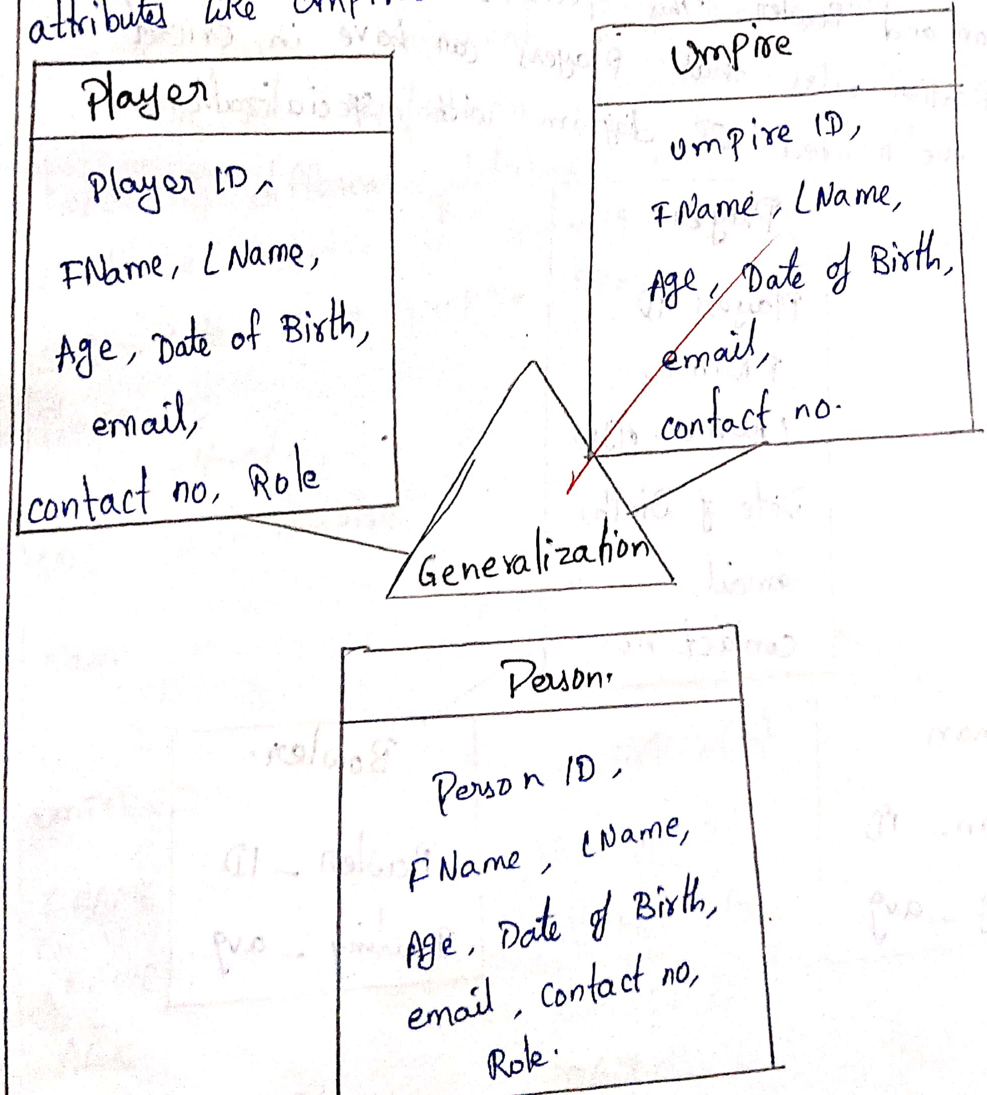
Contact_Number

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.

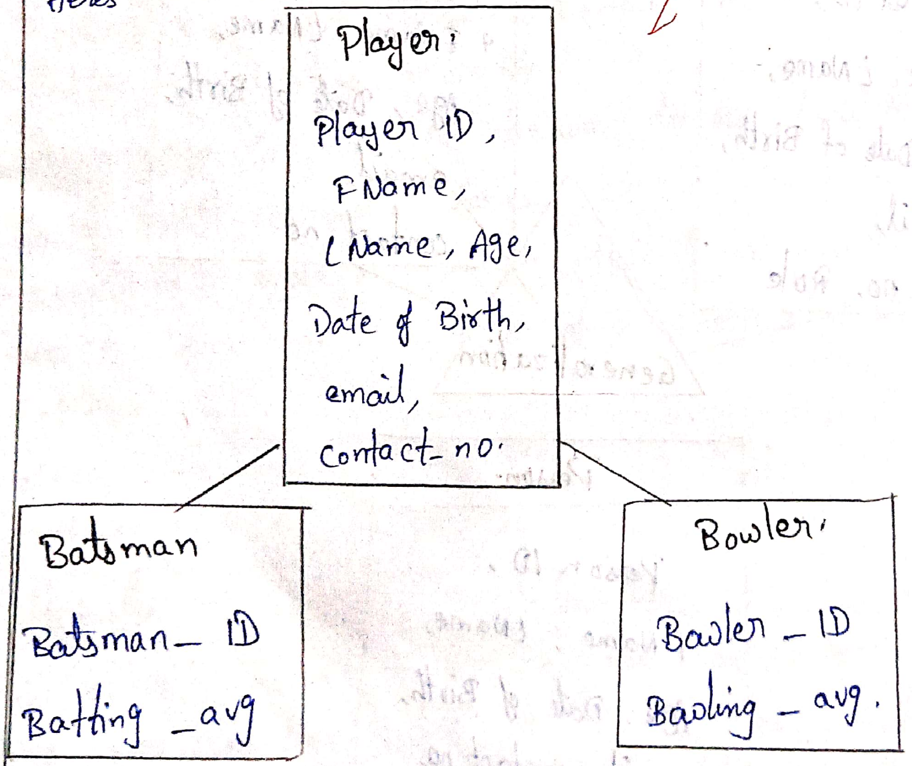


By using generalization, we can reduce data redundancy, improve data integrity, and simplify the structure of the ER diagram. This approach also allows for easier maintenance and updates, as changes made to the attributes shared by all "Person" entities will be automatically reflected in the subclasses.

Specialization:

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

In the case of the Tamilnadu cricket Board Association, let's consider specialization of "Player" entity into two subtypes; "Batsman" and "Bowler." This specialization is based on the specification roles that players can have in cricket. Here's the modified ER diagram with specialization.



2c) Find the domain of the attribute and Perform check constraint to the applicab le

| Attribute | Domain | check constraint Example. |
|---|---|---|
| Age | Integer | CHECK (Age >==18) |
| Contact_No | VARCHAR(10-15) | check(Length(contact_No) between 10&15) |
| Email | Integer | CHECK(Email like "%@%.%") |
| capacity | VARCHAR | CHECK (capacity >0) |
| Playing Role | | CHECK(Playing Role IN('Batsman', "Bowler", "All Rounder", "wicket-keeper")) |

```
SQL> ALTER TABLE Player ADD Constraint check_con
CHECK (age>=18);
Table altered.
```

2d) Rename the relations:

Renaming a table in SQL can be accomplished using ALTER Table statement with the 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 ompire RENAME column contact_no To
Phone_no;
Table altered.
SQL> DESC ompire

Name              NULL ?    Type.

UMPIREID                    VARCHAR 2(10)
FNAME                       VARCHAR 2(30)
LNAME                       VARCHAR 2(30)
AGE                         NUMBER (5,2)
DATE OF BIRTH               DATE.
```

| | |
|---|---|
| COUNTRY | VARCHAR 2(30) |
| EMAIL | VARCHAR 2(40) |
| PHONE_NO | NUMBER. |

2e): 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 a 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 actions on database objects. Privileges may include SELECT, INSERT, UPDATE, DELETE, 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.
```

```
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 umpire to Raj;
Grant suceeded.
```

| VEL TECH - CSE | |
|---|---|
| EX NO: | ✓ |
| PERFORMANCE (5) | 5 |
| RESULT AND ANALYSIS (5) | 5 |
| VIVA VOCE (5) | 5 |
| RECORD (5) | 5 |
| TOTAL (20) | 20 |
| SIGN WITH DATE | |

**Result:**

Thus the Hierarchical model and Network model has been successfully created.