

## Task 2:- Generating Design of other traditional database model

Aim:-

Creating Hierarchical (Network model) of the database by enhancing the second abstract data by performing following tasks using tools of Inheritance:

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

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

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

d. Rename the relations.

e. Perform SQL Relations using PDL, DDL, Commands.

a. Identify the specificity of each relationship, find and form 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 supervises the match

Specificity Analysis:

- Cricket Board  $\Leftrightarrow$  Team  $\rightarrow$  one-to-many

- Team  $\Leftrightarrow$  Player  $\rightarrow$  many-to-many  $\rightarrow$  Team - Player

- Match  $\Leftrightarrow$  Team  $\rightarrow$  many-to-many  $\rightarrow$  Match - Team

- Match  $\Leftrightarrow$  Ground  $\rightarrow$  one-to-one

Surplus Relation (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 generalizations based on common attributes or relationships among entities.

Here's an example of a 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 a superclass 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

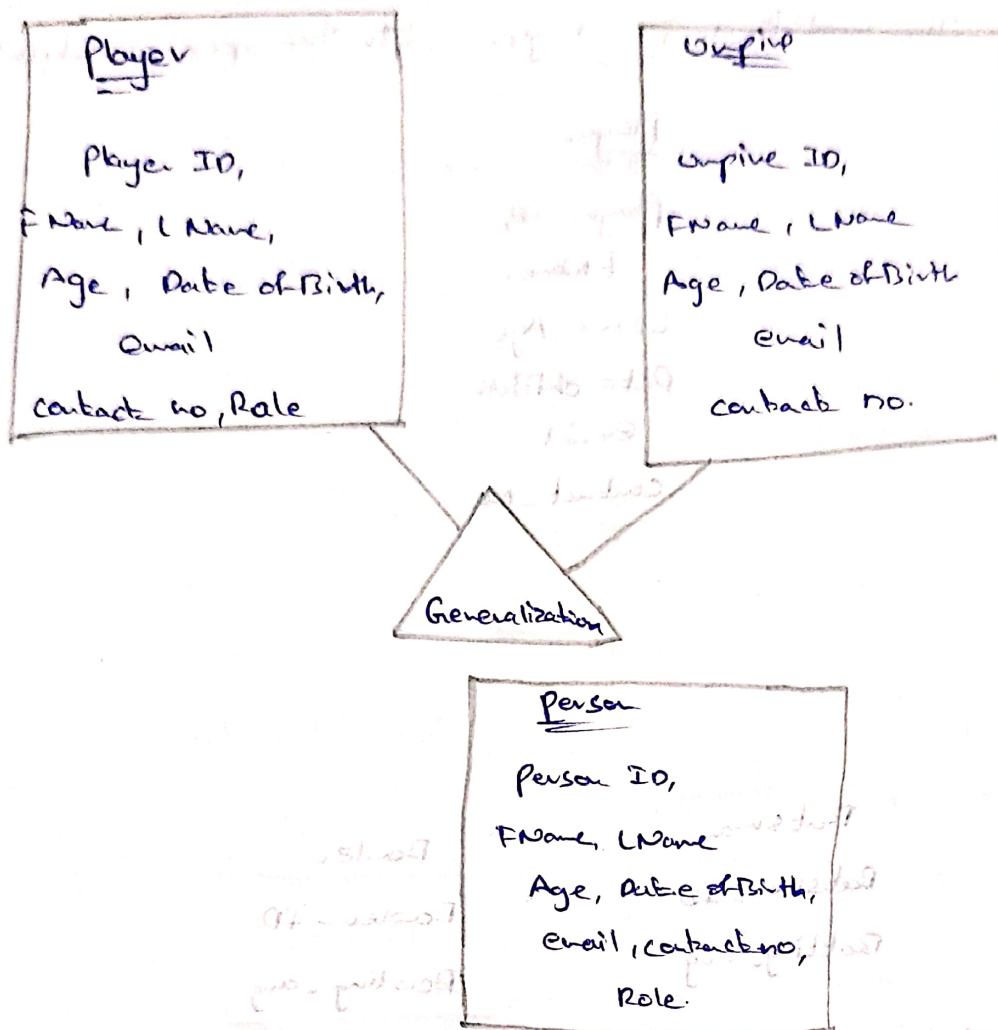
Contact Number

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 update, as changes made to the attributes shared by all "person" entities will be automatically reflected in the subclasses.

### Specialization

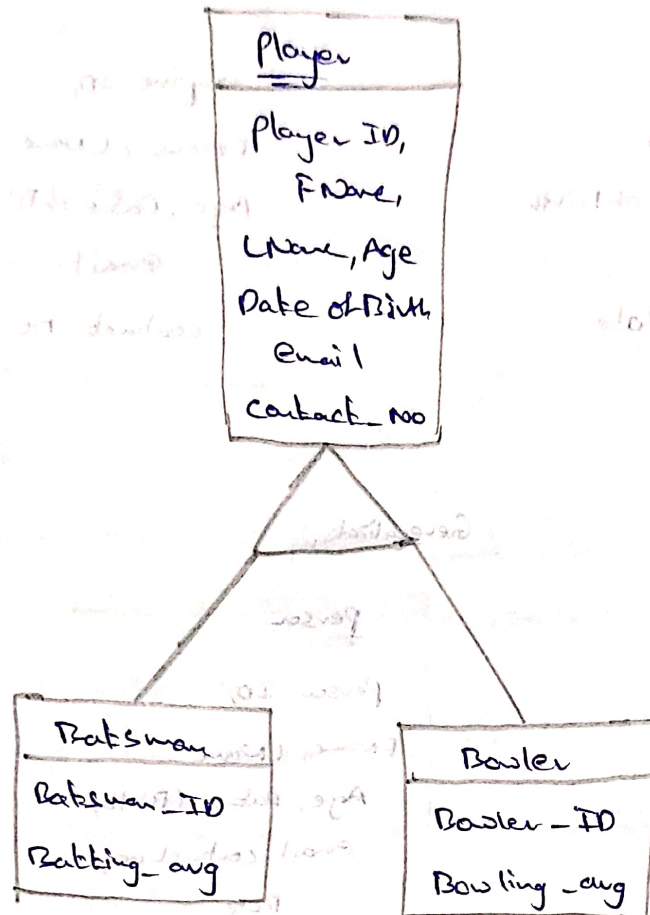
In the context of Entity-Relationship (ER) diagram, specialization refers to the process of defining subtypes with an entity type. It allows to represent entities that have specific attributes or relationship distinct from the general attributes or relationship, of the party. For entity.

In the case of the Tamil Nadu cricket Board Association let's

consider the specialization of the "player" entity into two subtypes: "Batsman" and Bowler." This specialization is based on the



Here's the modified ER diagram with the specialization!



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

Attribute	Domain	check constraint Example
Age	Integer	check (Age >= 18)
Contact_no	VARCHAR(10)	check (length (Contact_no) between 10 and 15)
Email	VARCHAR	check (Email like '%@%.%')
Capacity	Integer	check (Capacity > 0)
Playing Role	VARCHAR	check (Playing Role in (Batsman, Bowler, All- Rounder, wicket-keeper))

```
SOL> ALTER TABLE Player ADD CONSTRAINT check_col CHECK  
(Age >= 18);
```

Table altered.

2nd Rename the relations:

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

Here's the syntax for renaming a column in the table:

```
SOL> ALTER table Umpire RENAME column contact_no  
TO phone_no;
```

Table altered

```
SOL> DESC Umpire
```

<u>Name</u>	<u>Null? Type</u>
UMPIREID	VARCHAR2(10)
FNAME	VARCHAR2(30)
LNAME	VARCHAR2(30)
AGE	NUMBER(5,1)
DATE OF BIRTH	DATE
COUNTRY	VARCHAR2(30)
EMAIL	VARCHAR2(40)
PHONE_NO	NUMBER

2.e Perform SQL Relation 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 permission, 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 user or roles, allowing them to perform certain actions on database objects (e.g. tables, views, procedures). Privileges may include SELECT, INSERT, UPDATE, DELETE, EXECUTE, and more.

`SOL> create user Raj identified by Kumar;`

User Created

`SOL> grant resource to Raj;`

Grant succeeded.

`SOL> conn`

Enter user-name: Raj

Enter password: ABHIRAM

Connected.

`SOL> create table emp (eno number, ename varchar(10));`

Table Created.

`SOL> conn system / manager`

Connected.

`SOL> grant all on emp to Raj;`

Grant succeeded.

VEL TECH - CSE	
EX NO.	2
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	3
RECORD (5)	5
TOTAL (20)	18
SIGN WITH DATE	✓

### Results:-

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