

Topic 2: Generating Design of other traditional database model

Aim:-

Creating Hierarchical (network) model of the database by enhancing the sound abstractly data by performing following tasks using rules of Inheritance:

a. Identify the specificity of each relationship, find and remove 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 constraints to the applicable.

d. Rename the relations

e. Perform SQL Relations using DDL, DDL, Constraints

a. Identifying the specificity of each relationship, find and remove surplus relations.

Entity Identification:

- Cricket Board - has multiple teams (play, manage) one - many
- 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

b. Surplus Relation (Associative table(s)) and removal

- team - player (Create 10, Player ID, Team ID)
- match - team (Create 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 tennis match crecer Board (TNCB) 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
Opponent

Attributes:

- The above entities have common attributes like First Name, Last Name, Date-of-Birth, age, contact No, and Email.

Potential Generalization:

Create 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

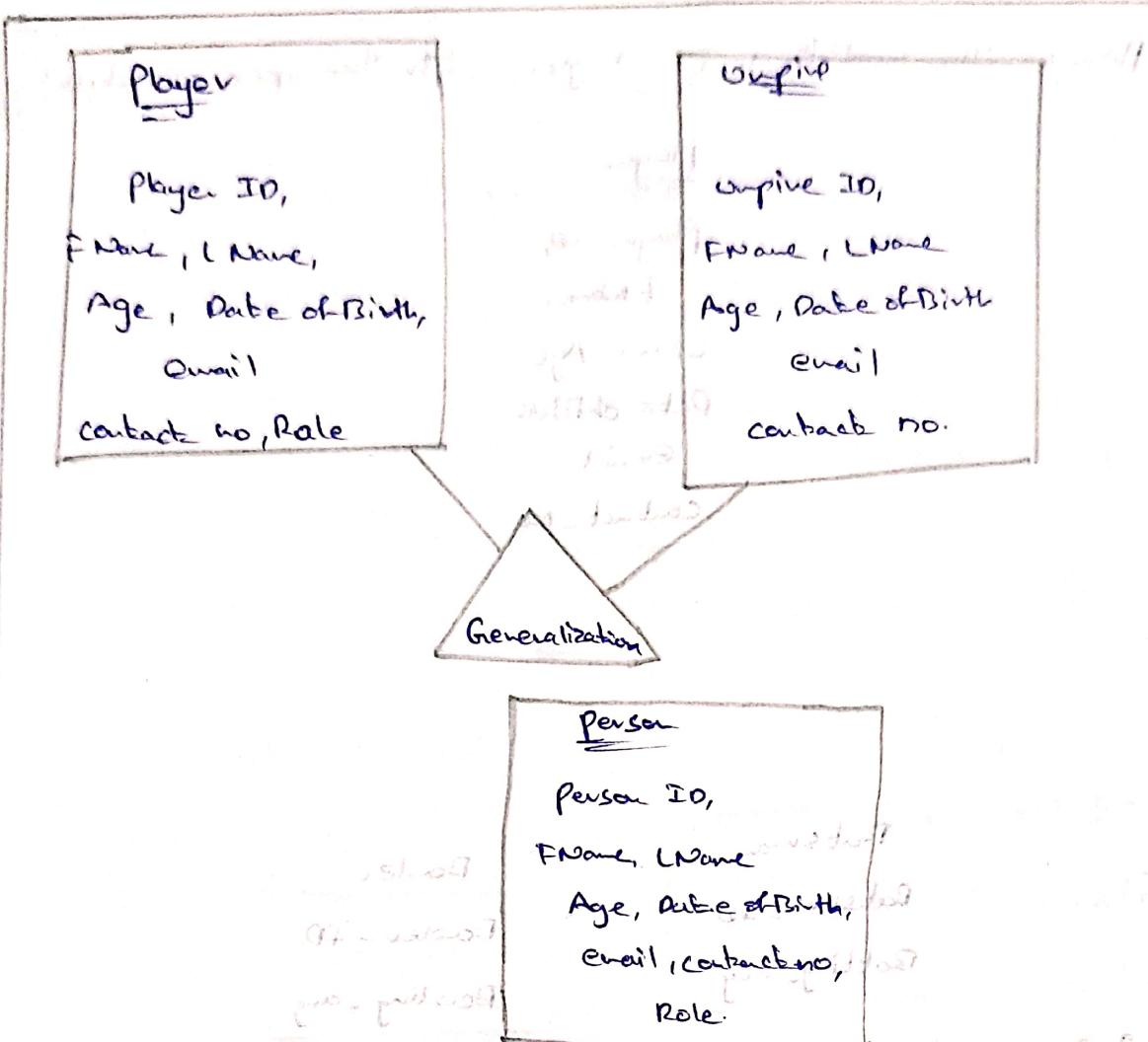
Email

Subclasses:

Player: Inherited attributes from "person" and add specific attributes like Umpire-ID.

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





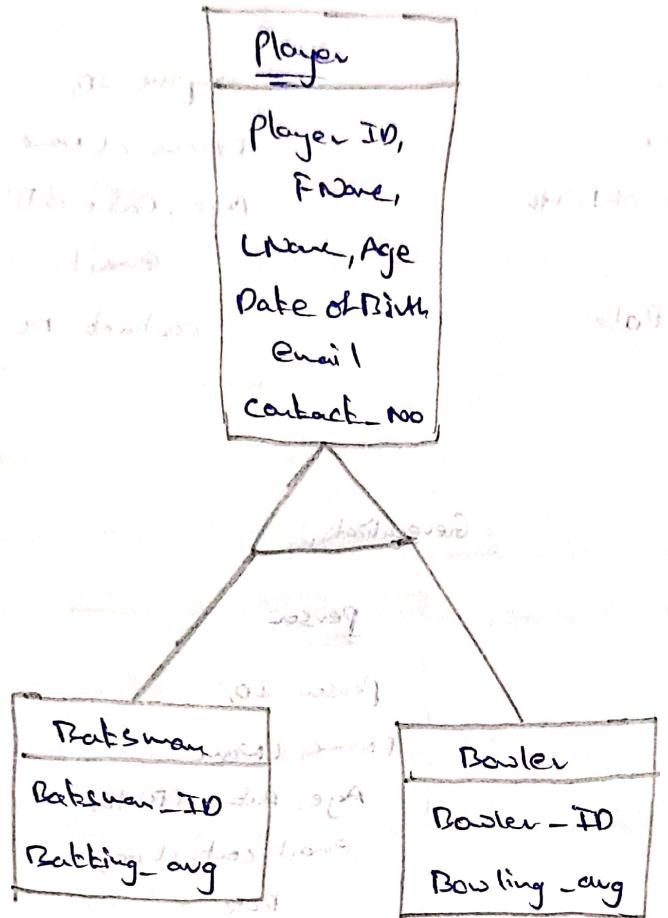
By Using generalization, we can reduce data redundancy, improve data integrity, and simplify the structure of the ER diagram. This approach also allows for easier maintenance and update, as changes made to the attributes shared by all "Person" entries will be automatically reflected in the subclasses.

Specialization

In the context of Entity-Relationship (ER) diagram, specialization refers to the process of defining subtypes with their own type. It allows to represent entities that have specific attributes or relationships distinct from the general attributes or relationships of the parent 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 constraint to the applicable.

Attribute	Domain	Check constraint Example
Age	Integer	Check (Age >= 10)
Contact_no	VARCHAR(10s)	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'))

SQL> ALTER TABLE Player ADD CONSTRAINT check-age CHECK
(Age >= 10);

Table altered.

2-d 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 tables varies slightly between different database management systems.

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

SQL> Alter table Umpire RENAME Column Contact_no
TO phone_no;

Table altered

SQL> DESC Umpire

Name	Null?	Type
UMPIREID	NO	VARCHAR2(10)
F_NAME	NO	VARCHAR2(20)
L_NAME	NO	VARCHAR2(30)
AGE	NO	NUMBER (5,1)
DATE_OF_BIRTH	NO	DATE
COUNTRY	NO	VARCHAR2(20)
EMAIL	NO	VARCHAR2(40)
PHONE_NO	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 SQL commands in DBMS.

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 (e.g. tables, views, procedures). 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 > conn raj/kumar

Enter user-name: raj

Enter password: kumar

Connected.

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

Table created.

SQL > conn system / manager

Connected.

SQL > grant all on emp to Raj;

Grant succeeded.

SQL > conn Raj/kumar

SQL > select * from emp

SQL > drop table emp

SQL > drop user Raj;



VEL TECH - CSB	
TX NO.	2
PERFORMANCE (5)	3
RESULT AND ANALYSIS (5)	3
VIVA VOCE (5)	3
RECORD (5)	3
TOTAL (20)	18
ENTER WITH DATE	18

Results:-

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