

18125 TASK 2:- Generating Design of other -traditional database model

AIM:-

Creating hierarchical / 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 relationships, find and form surplus relations.
 - b. check is hierarchy has-a hierarchy and performs generalization and/or specialization.
- Find domain of the attribute and perform check constraint to the applicable.
- rename the Relations.
- c. Perform SQL Relation using DDL, DCL commands.
 - d. 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 Relations (Associate tables):

- Team-Player (TeamID, PlayerID)
- Match-Team (MatchID, TeamID)

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

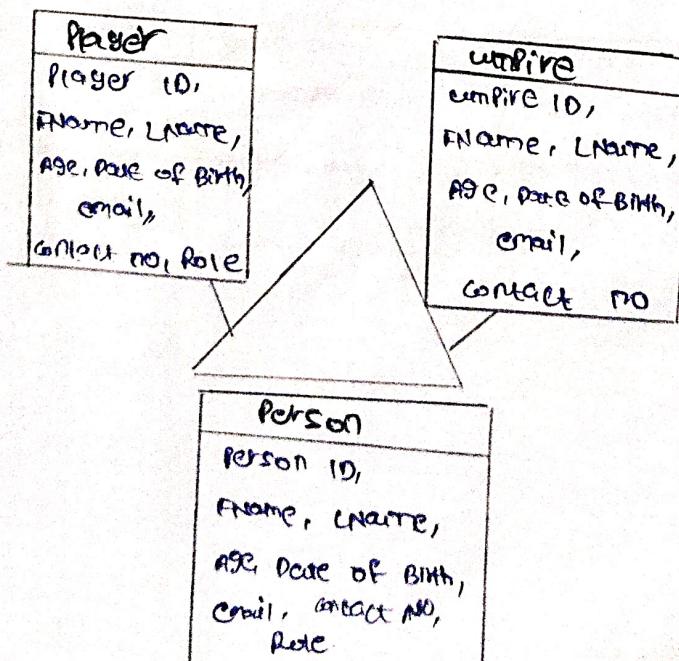
Create a superclass called "Person" to represent the common attributes shared by Player and umpire. The "Person" entity would have the following attributes:

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, FNname, LNname, Age, Date of Birth, Email, Contact no, Role.



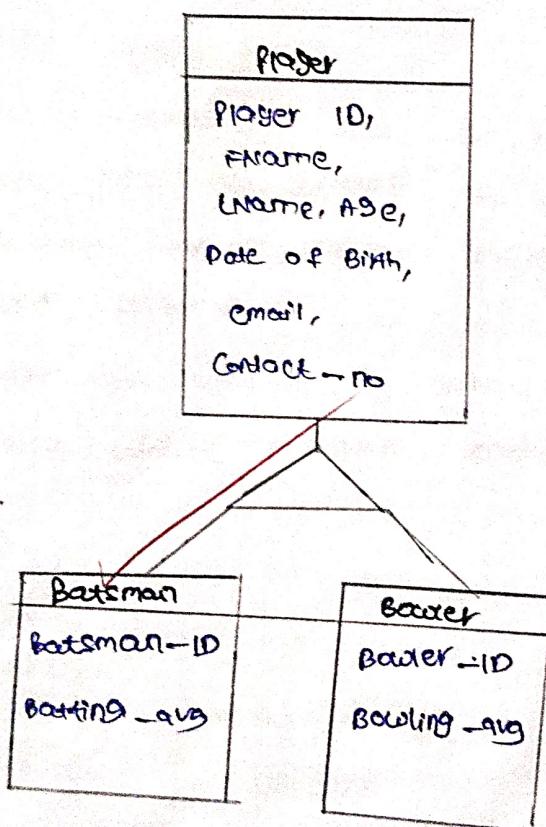
By using generalization we can reduce data redundancy, improve data integrity, and simplify the structure of the ER diagram. This approach also allows for easier maintenance and updates, as changes made to the attributes shared by all "Person" entities will be automatically reflected in the subclasses.

Specialization:

In the context of Entity-Relationship (ER) diagrams, specialization refers to the process of defining subtypes with an entity type. It allows two different 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 specific roles that players can have in cricket.

Here's the modified ER diagram with the specialization:



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

Attribute	Domain	Check constraint Example
Age	integer	CHECK (Age >= 18)
Contact - No	VARCHAR (0-15)	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 - Con CHECK (age >= 18);
Table altered.

Q.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 altered.

Name	Null?	Type
UNI		VARCHAR 2(10)
UMPIRE_ID		VARCHAR 2(30)
FNAME		VARCHAR 2(30)
LNAME		NUMBER (5,2)
AGE		DATE
DATE OF BIRTH		VARCHAR 2(30)
COUNTRY		VARCHAR 2(10)
EMAIL		NUMBER
PHONE_NO		

a.e Perform SQL Relations using DDL, DCL Commands.

DCL stands for "data control language," which is a subset of SQL (structure 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 (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> grant create session to Raj;

Grant succeeded.

SQL> Conn

Enter user-name: Raj

Enter pass word:

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 - CSE	
EX NO.	2
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
PIVA VOCE (5)	1
RECORD (5)	5
TOTAL (29)	12
DATE WITH DATE	2

Result: Thus the Hierarchical model and network model has been successfully created.