

**Introduction to Database System (Lab)**

**Spring 2023**

# **Final Term Lab**

# **VERSION I**

**Time Allowed: 90 min**

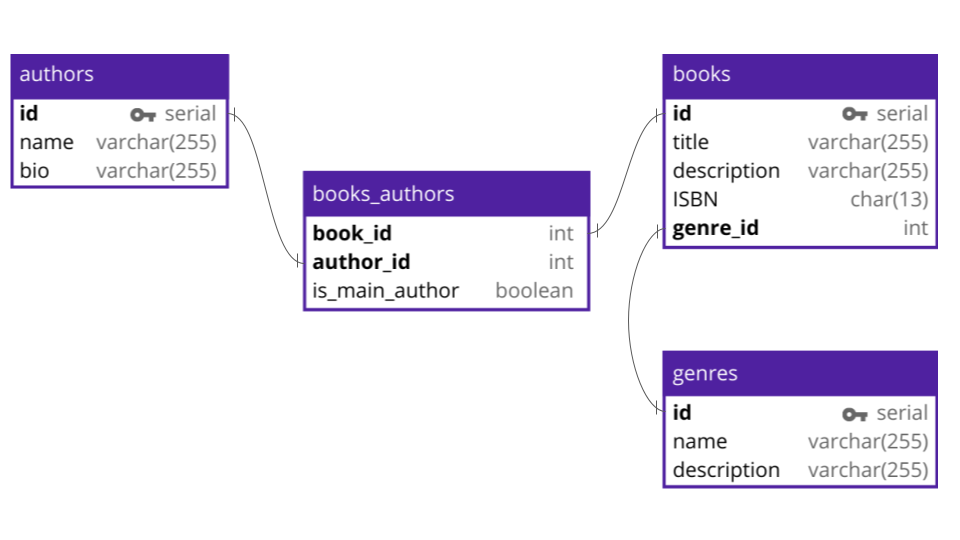
|  |  |
| --- | --- |
| Name: |  |
| Registration No |  |
| Serial Number (of attendance sheet) |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | DDL | DML | Total |
| Total Marks | 20 | 30 | 50 |
| Ob. Marks |  |  |  |

**Instructions:**

1. **This is a closed book, closed notes paper.**
2. **Understanding questions is part of the paper**. Therefore, no queries will be entertained during examination.
3. **chess.sql** file is uploaded on teams/portal. Use those files to attempt the DML section of paper.
4. Use proper indentation/formatting while writing queries. Not properly indenting will deduct 5% marks.
5. **You need to make an MS Word with your name and registration should be mentioned on each page.**
6. **You need to write only queries with output in MS Word file, you need to write query (text form) + its output table (picture) if any.**
7. **You will solve the DDL + DML section only in word docx.**

# **Part 1 DDL** **[20 Marks]**



**Q#1:** Implement all the table with its attributes. Make primary keys in all tables (Without foreign keys).

CREATE TABLE Books (

BookID INT PRIMARY KEY,

Title VARCHAR(255),

ISBN CHAR(13),

PublicationYear INT,

Publisher VARCHAR(255)

);

CREATE TABLE Authors (

AuthorID INT PRIMARY KEY,

AuthorName VARCHAR(255)

);

CREATE TABLE Books\_Authors (

BookID INT,

AuthorID INT,

PRIMARY KEY (BookID, AuthorID)

);

CREATE TABLE Genres (

GenreID INT PRIMARY KEY,

GenreName VARCHAR(255),

ABC CHAR(255)

);

**Q#2:** Alter **books** and **books\_authors** tables and add foreign keys by altering these tables.

ALTER TABLE Books

ADD FOREIGN KEY (GenreID) REFERENCES Genres (GenreID);

ALTER TABLE Books\_Authors

ADD FOREIGN KEY (BookID) REFERENCES Books (BookID),

ADD FOREIGN KEY (AuthorID) REFERENCES Authors (AuthorID);

**Q#3:** Add a column ABC (data type must be char (255)) in the **genres’s** table.

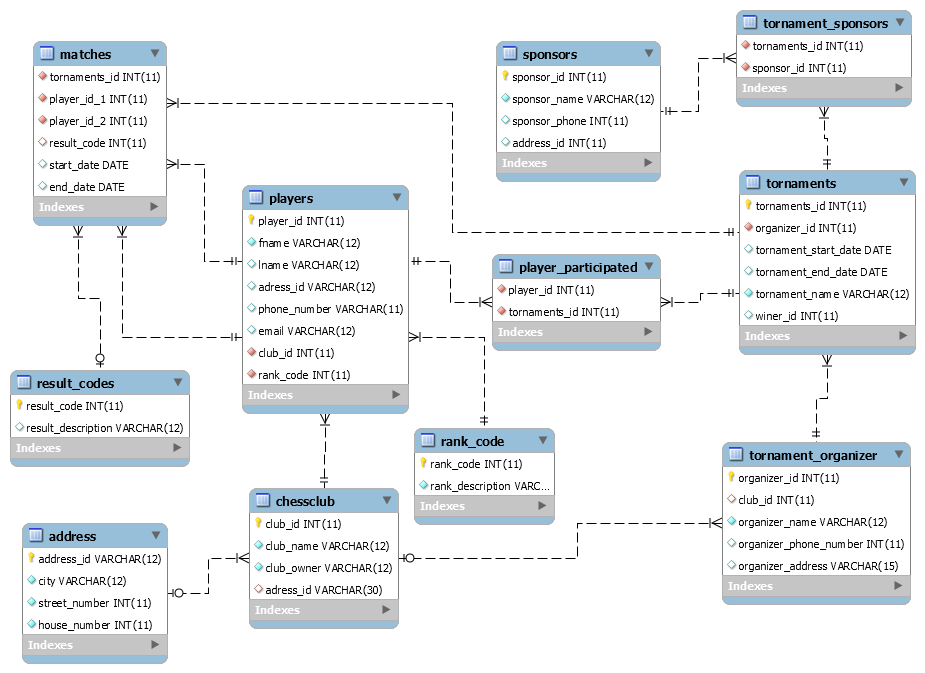
Alter table genres ADD column ABC char(255);

**Q#4:** Change the data type of ISBN from char to varchar (255).

Alter table books MODIFY column ISBN varchar(255);

**Part 2 DML**  **[30 Marks]**

**Import chess.sql schema.**



**Solve the following questions from the above schema:**

**Q#1:**

Write a query to display the count of tournaments sponsored by Mr. Azhar.

**Note: [use Join]**

**Select count(\*) AS tournament\_count**

**From tournament t**

**Join sponsor s on**

**Q#2:**

**Write a query to display the ID, first name, and last name of players who play for the club "Ned Chess" and have the club organizer named "Hafiz Asif"**

Note: **[use correlated query]**

**SELECT p.player\_id, p.first\_name, p.last\_name**

**FROM players p**

**WHERE p.club\_id = (**

**SELECT c.club\_id**

**FROM clubs c**

**WHERE c.club\_name = 'Ned Chess'**

**AND c.organizer\_name = 'Hafiz Asif'**

**);**

**Q#3:**

Create a view that displays the names of players who have played the maximum number of matches.

CREATE VIEW max\_matches\_players AS

SELECT p.player\_name

FROM players p

JOIN player\_match pm ON p.player\_id = pm.player\_id

GROUP BY p.player\_id, p.player\_name

HAVING COUNT(\*) = (

SELECT MAX(match\_count)

FROM (

SELECT COUNT(\*) AS match\_count

FROM player\_match

GROUP BY player\_id

) AS matches\_count

);

**Q#4:**

Write a query to display the details of players who have participated in more than one tournament, including the tournament name, tournament sponsor name, and the name of the organizer.

SELECT p.player\_name, t.tournament\_name, s.sponsor\_name, c.organizer\_name

FROM players p

JOIN player\_tournament pt ON p.player\_id = pt.player\_id

JOIN tournaments t ON pt.tournament\_id = t.tournament\_id

JOIN sponsors s ON t.sponsor\_id = s.sponsor\_id

JOIN clubs c ON t.club\_id = c.club\_id

GROUP BY p.player\_id, p.player\_name, t.tournament\_name, s.sponsor\_name, c.organizer\_name

HAVING COUNT(DISTINCT t.tournament\_id) > 1;

**Q#5:**

Write a query to print the count of players who participated in the tournament named 'Grand Master'

**Note: [use nested query]**

**SELECT COUNT(\*) AS player\_count**

**FROM (**

**SELECT DISTINCT player\_id**

**FROM player\_tournament**

**WHERE tournament\_id = (**

**SELECT tournament\_id**

**FROM tournaments**

**WHERE tournament\_name = 'Grand Master'**

**)**

**) AS subquery;**

**Q#6**

Create a stored procedure that displays the names of players as participants along with their rank description for all players who participated in a tournament with an end date in the year 2002. Also, call the procedure.

CREATE PROCEDURE get\_participant\_ranks

AS

BEGIN

SET NOCOUNT ON;

SELECT p.player\_name, r.rank\_description

FROM players p

JOIN player\_tournament pt ON p.player\_id = pt.player\_id

JOIN tournaments t ON pt.tournament\_id = t.tournament\_id

JOIN ranks r ON p.rank\_id = r.rank\_id

WHERE YEAR(t.end\_date) = 2002;

END;