---------Schema----------

USE APDA\_Tournament;

CREATE TABLE Tournament (

tournament\_id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100) NOT NULL,

date DATE NOT NULL

);

CREATE TABLE Round (

round\_id INT AUTO\_INCREMENT PRIMARY KEY,

tournament\_id INT NOT NULL,

round\_number INT NOT NULL,

FOREIGN KEY (tournament\_id) REFERENCES Tournament(tournament\_id)

ON DELETE CASCADE

);

CREATE TABLE Matchup (

matchup\_id INT AUTO\_INCREMENT PRIMARY KEY,

round\_id INT NOT NULL,

FOREIGN KEY (round\_id) REFERENCES Round(round\_id)

ON DELETE CASCADE

);

CREATE TABLE Team (

team\_id INT AUTO\_INCREMENT PRIMARY KEY,

team\_name VARCHAR(100) NOT NULL

);

CREATE TABLE Debater (

debater\_id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100) NOT NULL,

experience\_level ENUM('Novice', 'Varsity') NOT NULL,

team\_id INT,

FOREIGN KEY (team\_id) REFERENCES Team(team\_id)

ON DELETE SET NULL

);

CREATE TABLE Role (

role\_id INT AUTO\_INCREMENT PRIMARY KEY,

role\_name ENUM('Prime Minister', 'Member of Government', 'Leader of Opposition', 'Member of Opposition') NOT NULL

);

CREATE TABLE Judge (

judge\_id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100) NOT NULL,

experience\_level ENUM('Novice', 'Experienced') NOT NULL

);

CREATE TABLE Performance (

performance\_id INT AUTO\_INCREMENT PRIMARY KEY,

debater\_id INT NOT NULL,

round\_id INT NOT NULL,

speaks DECIMAL(4,2) NOT NULL,

feedback TEXT,

FOREIGN KEY (debater\_id) REFERENCES Debater(debater\_id)

ON DELETE CASCADE,

FOREIGN KEY (round\_id) REFERENCES Round(round\_id)

ON DELETE CASCADE

);

CREATE TABLE User (

user\_id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100) NOT NULL,

role ENUM('Organizer', 'Administrator', 'Judge') NOT NULL

);

CREATE TABLE Tournament\_User (

tournament\_id INT NOT NULL,

user\_id INT NOT NULL,

PRIMARY KEY (tournament\_id, user\_id),

FOREIGN KEY (tournament\_id) REFERENCES Tournament(tournament\_id)

ON DELETE CASCADE,

FOREIGN KEY (user\_id) REFERENCES User(user\_id)

ON DELETE CASCADE

);

————VALUES————

INSERT INTO Tournament (name, date)

VALUES

('Northeastern Invitational', '2024-12-15'),

('Boston Debate Open', '2025-01-20');

INSERT INTO Round (tournament\_id, round\_number)

VALUES

(1, 1),

(1, 2),

(2, 1);

INSERT INTO Matchup (round\_id)

VALUES

(1),

(2),

(3);

INSERT INTO Team (team\_name)

VALUES

('Team Alpha'),

('Team Beta'),

('Team Gamma');

INSERT INTO Debater (name, experience\_level, team\_id)

VALUES

('Alice', 'Novice', 1),

('Bob', 'Varsity', 1),

('Charlie', 'Novice', 2),

('Dana', 'Varsity', 3);

INSERT INTO Role (role\_name)

VALUES

('Prime Minister'),

('Member of Government'),

('Leader of Opposition'),

('Member of Opposition');

INSERT INTO Judge (name, experience\_level)

VALUES

('Eve', 'Experienced'),

('Frank', 'Novice');

INSERT INTO Performance (debater\_id, round\_id, speaks, feedback)

VALUES

(1, 1, 78.5, 'Good structure but needs stronger arguments'),

(2, 1, 80.0, 'Excellent rebuttals'),

(3, 2, 75.0, 'Good effort, needs to work on delivery'),

(4, 3, 82.0, 'Outstanding performance');

INSERT INTO User (name, role)

VALUES

('Grace', 'Organizer'),

('Hank', 'Administrator'),

('Eve', 'Judge');

INSERT INTO User (name, role)

VALUES

('Grace', 'Organizer'),

('Hank', 'Administrator'),

('Eve', 'Judge');

———— Queries —————

USE APDA\_Tournament;

# List all rounds in the "Northeastern Invitational" tournament

SELECT r.round\_id, r.round\_number

FROM Round r

JOIN Tournament t ON r.tournament\_id = t.tournament\_id

WHERE t.name = 'Northeastern Invitational';

# Get debater names and their teams

SELECT d.name AS Debater, t.team\_name AS Team

FROM Debater d

JOIN Team t ON d.team\_id = t.team\_id;

# Show performance feedback for a given round:

SELECT d.name AS Debater, p.speaks, p.feedback

FROM Performance p

JOIN Debater d ON p.debater\_id = d.debater\_id

WHERE p.round\_id = 2;

# Find Rounds Without Judges Assigned

SELECT r.round\_id, r.round\_number, t.name AS tournament\_name

FROM Round r

LEFT JOIN Judge j ON r.judge\_id = j.judge\_id

JOIN Tournament t ON r.tournament\_id = t.tournament\_id

WHERE j.judge\_id IS NULL;

#View All Novice Debaters in a Tournament

SELECT d.debater\_id, d.name

FROM Debater d

JOIN Team t ON d.team\_id = t.team\_id

JOIN Matchup m ON m.matchup\_id = t.team\_id

JOIN Round r ON m.round\_id = r.round\_id

JOIN Tournament tmt ON r.tournament\_id = tmt.tournament\_id

WHERE d.experience\_level = 'Novice' AND tmt.name = 'Northeastern Invitational';

#Calculate Average speaks for a debater

SELECT d.name AS Debater, AVG(p.speaks) AS Average\_Speaks

FROM Debater d

JOIN Performance p ON d.debater\_id = p.debater\_id

GROUP BY d.debater\_id;

——-Stored procedures ———

USE APDA\_Tournament

#Procedure to Add a New Tournament

DELIMITER $$

CREATE PROCEDURE AddTournament(IN tournamentName VARCHAR(100), IN tournamentDate DATE)

BEGIN

INSERT INTO Tournament (name, date)

VALUES (tournamentName, tournamentDate);

END$$

DELIMITER ;

# Procedure to Assign Judges to a Tournament

DELIMITER $$

CREATE PROCEDURE AssignJudgeToTournament(IN judgeId INT, IN tournamentId INT)

BEGIN

INSERT INTO Tournament\_User (tournament\_id, user\_id)

VALUES (tournamentId, judgeId);

END$$

DELIMITER ;

# Procedure to Record Debater Performance

DELIMITER $$

CREATE PROCEDURE RecordPerformance(

IN debaterId INT,

IN roundId INT,

IN speaks DECIMAL(4,2),

IN feedbackText TEXT

)

BEGIN

INSERT INTO Performance (debater\_id, round\_id, speaks, feedback)

VALUES (debaterId, roundId, speaks, feedbackText);

END$$

DELIMITER ;

CALL AddTournament('BU Nov 2025', '2025-1-15');

SELECT \* FROM Tournament;

CALL AssignJudgeToTournament(1, 1); -- Assigns Judge ID 1 to Tournament ID 2 (no duplicates allowed)

CALL RecordPerformance(1,2, 75.50, 'Excellent rebuttals and structured arguments.');

———Triggers ————

USE APDA\_Tournament;

#Prevent Duplicate Entries in Tournament\_User

DELIMITER $$

CREATE TRIGGER prevent\_duplicate\_tournament\_user

BEFORE INSERT ON Tournament\_User

FOR EACH ROW

BEGIN

IF EXISTS (

SELECT 1

FROM Tournament\_User

WHERE tournament\_id = NEW.tournament\_id AND user\_id = NEW.user\_id

) THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Duplicate entry for Tournament\_User.';

END IF;

END$$

DELIMITER ;

INSERT INTO Tournament (name, date) VALUES ('Debate Fest 2024', '2024-12-20');

INSERT INTO User (name, role) VALUES ('Alice', 'Organizer');

INSERT INTO Tournament\_User (tournament\_id, user\_id) VALUES (1, 1);

INSERT INTO Tournament\_User (tournament\_id, user\_id) VALUES (1, 1);

——— Views —————

USE APDA\_Tournament;

# Provides tournament details along with the number of rounds

CREATE VIEW TournamentDetails AS

SELECT t.tournament\_id, t.name AS Tournament, t.date AS Date, COUNT(r.round\_id) AS NumberOfRounds

FROM Tournament t

LEFT JOIN Round r ON t.tournament\_id = r.tournament\_id

GROUP BY t.tournament\_id;

SELECT \* FROM TournamentDetails;

# Summarizes debater performance across tournaments

CREATE VIEW DebaterPerformanceOverview AS

SELECT d.name AS Debater, t.name AS Tournament, AVG(p.speaks) AS AvgSpeaks

FROM Debater d

JOIN Performance p ON d.debater\_id = p.debater\_id

JOIN Round r ON p.round\_id = r.round\_id

JOIN Tournament t ON r.tournament\_id = t.tournament\_id

GROUP BY d.debater\_id, t.tournament\_id;

SELECT \* FROM DebaterPerformanceOverview;