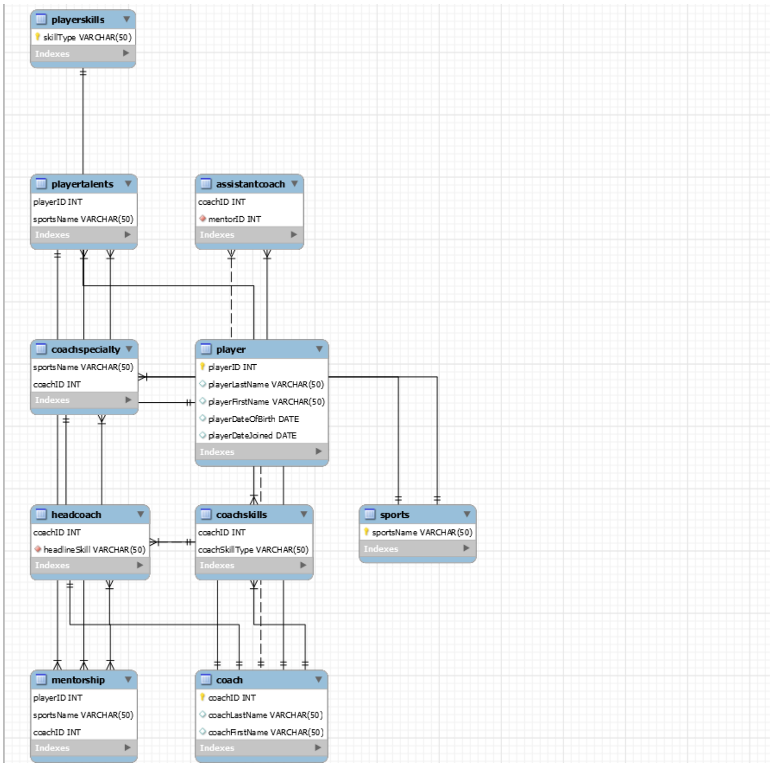
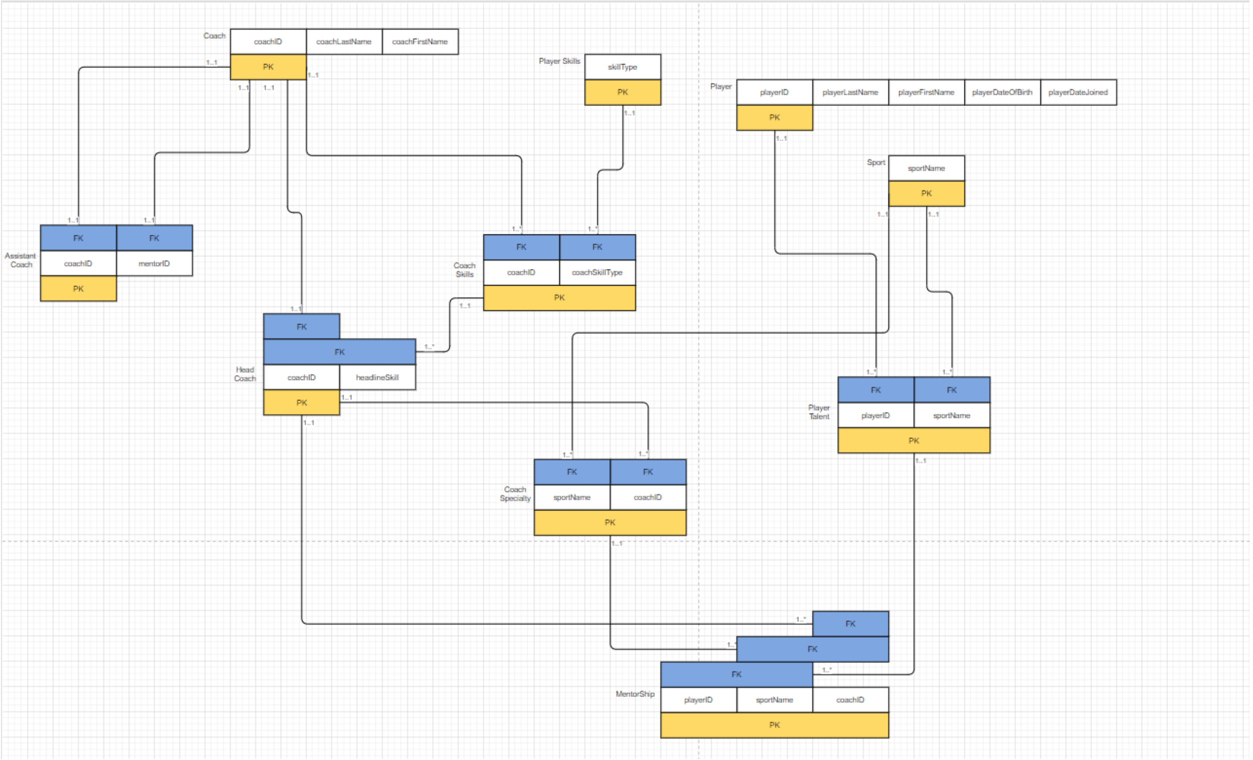
Group 14

UML:



RSD:



Class Definitions and Descriptions of the associations

Player Class represents a person who is playing sports and registered with the clinic, looking to be coached. Its attributes are:

- playerID: Represents a unique identification number used to identify each player.
- playerLastName: Represents the surname of the player.
- playerFirstName: Represents the given name of the player.
- playerDateOfBirth: Represents the year, month, and day the player was born in.
- playerDateJoined: Represents the year, month, and day the player joined the coaching cl.

Player Skills Class represents qualities or abilities required for a player to excel in a sport. Its attributes are:

- skillType: Represents the specific name/type of the player skill.

Coach Class represents a person who teaches and trains the players, makes decisions about the player schedules, how they play during games, etc. Its attributes are:

- coachID: Represents a unique identification number used to identify each coach.
- coachLastName: Represents the surname of the coach.
- coachFirstName: Represents the given name of the coach.

Assistant Coach Class represents the coach who supports the head coach. Its attributes are:

- mentorID: Represents the identification number of the coach who mentors the assistant coach.

Coach Skills Class represents the skills that the coach can teach the players. Its attributes are:

- coachSkillType: Represents the specific name of the skill the coach teaches the player.

Sport Class represents the athletic activity that the coaches teach to the players. Its attributes are:

- sportsName: Represents the name of the athletic activity the player is learning.

Player Talent Class represents the name of the sports that the players can play.

Coach Specialty Class represents the name of the sports the individual coaches specialize in.

Mentorship Class represents which sports the coaches teach the players, along with their identification numbers.

Head Coach Class represents the coaches who oversee the assistant coaches. Its attributes are:

- headlineSkill: Represents the name of the skill the coach is the best at.

DDL:

```
DROP TABLE player;                              -- drops the table called player
/* Creating the Player table with the player's information using the
  playerID as the primary key. */
CREATE TABLE player
(
        playerID                    INT NOT NULL,       -- column for the Player ID
   playerLastName           VARCHAR (50),        -- column for the Last Name
   playerFirstName          VARCHAR (50),        -- column for the First Name
   playerDateOfBirth   DATE,                 -- column for the Date of Birth
   playerDateJoined    DATE,                 -- column for the Date Joined

   PRIMARY KEY (playerID)                       -- the column playerID is the primary key of
the player table
);

DROP TABLE playerSkills;                        -- drops the table called playerSkills
-- Creating the PlayerSkills using the skillType as the primary key
CREATE TABLE playerSkills
(
        skillType       VARCHAR (50),                   -- column for the Skill Type

   PRIMARY KEY (skillType)                      -- the column skillType is the primary key of
the playerSkills table
);

DROP TABLE coach;                                       -- drops the table called coach
-- Creating the Coach table with the coach's information using coachID as primary key
CREATE TABLE coach
(
        coachID                     INT NOT NULL,               -- column for the Coach ID
   coachLastName    VARCHAR (50),              -- column for the Coach Last Name
   coachFirstName   VARCHAR (50),              -- column for the Coach First Name
   PRIMARY KEY (coachID)                            -- column coachID is the primary key of the
coach table
);

DROP TABLE assistantCoach;
/* Creating the assistantCoach table using coach ID as the primary key
  and also referencing coach ID from the coach table. */
CREATE TABLE assistantCoach
(
        coachID              INT NOT NULL,                        -- column for the Coach ID
from the coach table
   mentorID    INT NOT NULL,                        -- column for the Mentor ID

   PRIMARY KEY (coachID),                           -- column coachID is the primary key of the
assisstantCoach table
   CONSTRAINT assistantCoachID_fk     FOREIGN KEY (coachID) -- the column coachID is a
foreign from the coach table
   REFERENCES coach(coachID),
   CONSTRAINT assistantCoachMentor_fk FOREIGN KEY (mentorID) -- the column mentor ID
is a foreign key referencing
   REFERENCES coach(coachID)                                                          --
coach ID from the coach table
);

DROP TABLE coachSkills; -- drops the table called coachSkills
/* Creating coachSkills table using the coachID and coachSkilltype as primary key
  and also referencing coach ID from coach table and skillType from playerSkills. */
CREATE TABLE coachSkills
(
        coachID                        INT NOT NULL,              -- column for the coach ID
from coach table
   coachSkillType      VARCHAR (50),                -- column for the Coach Skill Type

   PRIMARY KEY (coachID,coachSkillType),           -- the columns coachID and coachSkillType
are primary keys for coachSkills table
   CONSTRAINT coachID_fk                  FOREIGN KEY (coachID) -- the column coachID is
a foreign key from the coach table
   REFERENCES coach(coachID),
   CONSTRAINT coachSkillType_fk FOREIGN KEY (coachSkillType) -- the column
coachSkillType is a foreign key referencing
   REFERENCES playerSkills(skillType)                                      -- skillType
from the playerSkills table
);
```

```
DROP TABLE headCoach; -- drops the table called headCoach
/* Creating table headCoach which uses the coachID as the primary key while also referencing
   coachID from coach table and both coachID and coachSkillType from coachSkills table. */
CREATE TABLE headCoach
(
        coachID                         INT NOT NULL,                           -- column for the
Coach ID from coach table
   headlineSkill         VARCHAR (50) NOT NULL,   -- column for the Headline Skill

   PRIMARY KEY (coachID),            -- the column coachID is the primary key for the
headCoach table
   CONSTRAINT headCoach_fk           FOREIGN KEY (coachID) -- the column coachID is a
foreign key from the coach table
   REFERENCES coach(coachID),
   CONSTRAINT headlineSkill_fk FOREIGN KEY (coachID, headlineSkill) -- the columns
coachID and headlineSkill are foreign keys
   REFERENCES coachSkills(coachID, coachSkillType)                              --
referencing coachID and coachSkillType from coachSkills table
);

DROP TABLE sports;           -- drops the table called sports
-- Creating sports table using sportsName as the primary key
CREATE TABLE sports
(
        sportsName    VARCHAR (50) NOT NULL,   -- column for the Sports Name

   PRIMARY KEY (sportsName)       -- the column sportsName is the primary key for the sports
table
);

DROP TABLE playerTalents; -- drops the table called playerTalents
/* Creating playerTalents table using playerID and sportsName as primary key while referencing
   playerID  from the player table and sportsName from sports table. */
CREATE TABLE playerTalents
(
        playerID         INT NOT NULL,                          -- column for the Player ID from
player table
   sportsNameVARCHAR (50) NOT NULL,   -- column for the Sports Name from sports table

   PRIMARY KEY (playerID, sportsName),                          -- the columns playerID and
sportsName is the primary key
   CONSTRAINT playerID_fk  FOREIGN KEY (playerID)  -- the column playerID is a foreign key
from player table
   REFERENCES player(playerID),
   CONSTRAINT sportsName_fk FOREIGN KEY (sportsName) -- the column sportsname is a
foreign key from sports table
   REFERENCES sports(sportsName)
);

DROP TABLE coachSpecialty;          -- drops the table called coachSpecialty
/* Creating CoachSpecialty table using both the sportsName and coachID as the primary key
 while also referencing sportsName from sports table and coachID from coach table. */
CREATE TABLE coachSpecialty
(
         sportsName    VARCHAR (50) NOT NULL,   -- column for the Sports Name from sports
table
   coachID                INT NOT NULL,                          -- column for the Coach ID from
coach table

   PRIMARY KEY (sportsName, coachID),    -- the columns sportsName and coachID is the
primary key
   CONSTRAINT coachSportName_fk   FOREIGN KEY (sportsName)       -- the column
sportsName is a foreign key from sports table
   REFERENCES sports(sportsName),
   CONSTRAINT coachSpecialtyID_fk FOREIGN KEY (coachID)              -- the column coachID
is a foreign key from coach table
   REFERENCES coach(coachID)
);

DROP TABLE mentorship;     -- drops the table called mentorship
/* Creating mentorship table using the playerID, sportsName and coachID as the primary key
while
  also referencing playerID and sportsName from playerTalents table, coachID from coach table
and
  sportsName, coachID from coachspecialty table */
CREATE TABLE mentorship
(
         playerID INT NOT NULL,                               -- column for the Player ID from the
player table
   sportsName VARCHAR (50) NOT NULL,  -- column for the Sports Name from the sports table
   coachID INT NOT NULL,                              -- column for the Coach ID from the coach
table

   PRIMARY KEY (playerID, sportsName, coachID),          -- the columns playerID,
sportsName, coachID, is the primary keys for the mentorship table
   CONSTRAINT mentorID_fk                   FOREIGN KEY (playerID, sportsName)
-- the columns playerID and sportsName are foreign keys from
```

```
   REFERENCES playerTalents(playerID, sportsName),
         -- player and sports table
   CONSTRAINT mentorCoach_fk     FOREIGN KEY (coachID)
-- the column coachID is a foreign key from the coach table
   REFERENCES headCoach(coachID),

   CONSTRAINT coachSpecSportName_fk FOREIGN KEY (sportsName, coachID)
-- the columns sportsName and coachID are foreign keys from
   REFERENCES coachSpecialty(sportsName, coachID)
         -- the sports and coach table
);
```

```
#TRIGGERS
-- Prevents a player from being coached in more than three sports
DROP TRIGGER maxCoachInsert;
DELIMITER $$
CREATE TRIGGER maxCoachesInsert
BEFORE INSERT ON mentorship FOR EACH ROW
BEGIN
         DECLARE maxCoach INT;
   SELECT count(*) INTO maxCoach
   FROM mentorship
   WHERE playerID = NEW.playerID;
   IF (maxCoach = 3) THEN
                  SIGNAL SQLSTATE '45000'
      SET MESSAGE_TEXT = 'Cannot be coach in more than three sports';
         END IF;
END$$
DELIMITER ;
```

```
-- Prevent a player from being coached no more than three sports
DELIMITER $$
CREATE TRIGGER maxCoachesUpdate
BEFORE UPDATE ON mentorship FOR EACH ROW
BEGIN
         DECLARE maxCoach INT;
   SELECT COUNT(*) INTO maxCoach
   FROM mentorship
   WHERE playerID = NEW.playerID;
   IF (maxCoach > 3) THEN
                  SIGNAL SQLSTATE '45000'
```

```
    SET MESSAGE_TEXT = 'Player connot be a coach in more than three sports';
      END IF;
END$$
DELIMITER ;


-- Trigger to prevent deletion of coach skills when coach only has two skills
DELIMITER $$
CREATE TRIGGER minCoachSports
BEFORE DELETE ON coachskills FOR EACH ROW
BEGIN
      DECLARE minSports INT;
  SELECT COUNT(*) INTO minSports
  FROM coachSkills
  WHERE coachID = OLD.coachID;
  IF (minSports = 2) THEN
            SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Coaches must have at least two skills';
      END IF;
END$$
DELIMITER ;



-- Prevents players under the age of 10 from joining the Clinic
DELIMITER $$
CREATE TRIGGER minPlayerAge
BEFORE INSERT ON player FOR EACH ROW
BEGIN
      DECLARE minAge INT;
  SELECT TIMESTAMPDIFF(year, NEW.playerDateOfBirth, NOW()) INTO minAge;
      IF (minAge < 10) THEN
            SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Players must be at least ten years old to join this Clinic';
      END IF;
END$$
DELIMITER ;



-- Trigger to enforce constraint, not inherently in structure,
-- to prevent coaches from mentoring themselves
DELIMITER $$
CREATE TRIGGER asstDuplicateCoach
```

```
BEFORE INSERT ON assistantcoach FOR EACH ROW
BEGIN
        IF (NEW.coachID = NEW.mentorID) THEN
                SIGNAL SQLSTATE '45000'
      SET MESSAGE_TEXT = 'Coaches cannot mentor themselves';
        END IF;
END$$
DELIMITER ;
```

-- Trigger to update player column

```
DELIMITER $$
CREATE TRIGGER sportsCoachInsert
AFTER INSERT ON mentorship FOR EACH ROW
BEGIN
        DECLARE sports VARCHAR(100);
   DECLARE coachings VARCHAR(100);
   DECLARE sportsCoach VARCHAR(100);
   DECLARE coachName VARCHAR(100);
        SET sports = NEW.sportsName;
   SET coachName = (SELECT CONCAT(coachLastName, ' ', coachFirstName)
                                        FROM coach
               WHERE coachID = NEW.coachID);
        SET sportsCoach = CONCAT(sports, ': ', coachName);
   UPDATE player
   SET coachings = CONCAT_WS(', ', coachings, sportsCoach)
   WHERE playerID = NEW.playerID;
END$$
DELIMITER ;
```

-- Three different cases of sports --> coach deletions

```
DROP TRIGGER IF EXISTS sportsCoachDelete;
DELIMITER $$
CREATE TRIGGER sportsCoachDelete
AFTER DELETE ON mentorship FOR EACH ROW
BEGIN
        DECLARE sports VARCHAR(100);
   DECLARE coachName VARCHAR(100);
   DECLARE sportsCoach VARCHAR(100);
   DECLARE coachings VARCHAR(100);
   DECLARE currCoachings VARCHAR(100);
   DECLARE fullCoachMatch VARCHAR(100);
```

```
        SET sports = OLD.sportsName;
   SET coachName = (SELECT CONCAT(coachLastName, ' ', coachFirstName)
                                        FROM coach
              WHERE coachID = OLD.coachID);
        SET sportsCoach = CONCAT(sports, ': ', coachName);
   SELECT TRIM(coachings) INTO currCoachings
   FROM player WHERE playerID = OLD.playerID;
   SELECT STRCMP(currCoachings, sportsCoach) INTO fullCoachMatch;
   UPDATE player
   SET coachings = CASE
                                              WHEN fullCoachMatch = 0 THEN
                                                     REPLACE(coachings, sportsCoach, NULL)
                                              WHEN fullCoachMatch != 0 THEN
                                                     REPLACE(coachings, sportsCoach, '')
                                              ELSE coachings
                                       END
        WHERE playerID = OLD.playerID;

   -- Easier to just TRIM the leading and double comma manually
   UPDATE player
   SET coachings = TRIM(BOTH ', ' FROM coachings),
                coachings = REPLACE(coachings, ', ,', ',')
        WHERE playerID = OLD.playerID;
END$$
DELIMITER ;
```

DML:

```
-- Inserting values to the player table which contains the attributes in order:
-- playerID, playerLastName, playerFirstName, playerDateOfBirth and playerDateJoined
INSERT INTO player
VALUES          (1, 'Kent', 'Clark', '1938-02-28','1952-04-18'),
                (2, 'Wayne', 'Bruce', '1915-04-17', '1935-04-08'),
     (3, 'Prince', 'Diana', '1976-03-22', '1988-04-06'),
     (4, 'Jordan', 'Hal', '1976-02-20', '1988-04-07'),
     (5, 'Allen', 'Barry', '1989-03-14', '2008-04-17'),
     (6, 'Curry', 'Arthur', '1989-01-29', '2008-02-16'),
     (7, 'Maximoff', 'Wanda', '1976-02-10', '1998-03-11'),
     (8, 'Kane', 'Kate', '1990-01-26', '2012-05-17'),
     (9, 'Walters', 'Jennifer', '1980-11-09', '1998-07-09'),
     (10, 'Rogers', 'Steve', '1918-07-04', '1932-04-18'),
     (11, 'Odinson', 'Thor', '1964-08-11', '1987-02-24'),
     (12, 'Grey', 'Jean', '1967-09-12', '1989-08-25'),
        (13, 'Danvers', 'Carol', '1977-08-17', '1999-10-08'),
        (14, 'MURDOCK', 'MATTHEW', '1974-08-04', '1989-09-23');

-- Inserting values to the coach table which contains the attributes in order:
-- coachID, coachLastName and coachFirstName
INSERT INTO coach
VALUES          (1, 'Bunny', 'Lola'), (2, 'Martian', 'Marvin'), (3, 'Duck', 'Daffy'),
        (4, 'Pig', 'Porky'), (5, 'Bunny', 'Bugs'), (6, 'Fudd', 'Elmer'),
        (7, 'Devil', 'Tasmanian'), (8, 'Yosemite', 'Sam'), (9, 'Le Pew', 'Pepe'),
        (10, 'Gonzales', 'Speedy'), (11, 'Bulldog', 'Hector'), (12, 'Hawk', 'Henery'),
        (13, 'Dog', 'Charlie'), (14, 'Runner', 'Road'), (15, 'Coyote', 'Wile'),
        (16, 'Voorhees', 'Jason'), (17, 'Pennyworth', 'Alfred'), (18, 'Waller', 'Amanda'),
        (19, 'Constatine', 'John'), (20, 'Stewart', 'John'), (21, 'Apokolips', 'Darkseid');

-- Inserting values to the playerSkills table which contains the attribute skillType
INSERT INTO playerSkills
VALUES  ('Speed'), ('Defense'), ('Offense'), ('High Jumping'), ('Agility'),
        ('Endurance'), ('Flexability'), ('Strength'), ('Quick Foot Work'),
        ('Highten Sense'), ('Long Jumping'), ('Advanved Handeling'),
        ('Advanced Throwing'), ('Advanced Catching'), ('Cunning'),
        ('Speed Passing'), ('Under Water Basket Weaving'), ('Dunking'),
        ('Trick Plays'), ('Stopping Power'), ('Power Sliding'),
        ('Warm up Movments'), ('Energy Conservation'), ('Arm Strengthing'),
        ('Puzzle Solving'), ('Deep Diving'), ('Core Strengthing'),
        ('Leg Strengthing'), ('Team Unity'), ('Top Leadership'),
        ('Pressure Handling'), ('Slam Dunking'), ('Assult Swimming'),
        ('Motivation'), ('Intellegence'), ('Speed Swimming'), ('Strength Endurance');
 -- Inserting more values to playerskills table
 INSERT INTO playerskills
 VALUES ('Power Lifting'), ('Speed Running'), ('Speed Endurance'), ('Strategy Planning'),
        ('Sprinting'), ('Power Jumping'), ('Speed Jumping');
 INSERT INTO playerskills
 VALUES ('Power Running');

-- Inserting values to the sports table which contains the attribute sportsName
INSERT INTO sports
VALUES  ('Basketball'), ('Football'), ('Baseball'),
        ('Ping Pong'), ('Tennis'), ('Soccer'),
        ('Volleyball'), ('Pool'), ('Bowling'),
        ('Golf'), ('Swimming');

-- Inserting values to the assistantcoach table which contains the attributes in order:
-- coachID and mentorID
insert into assistantcoach
```

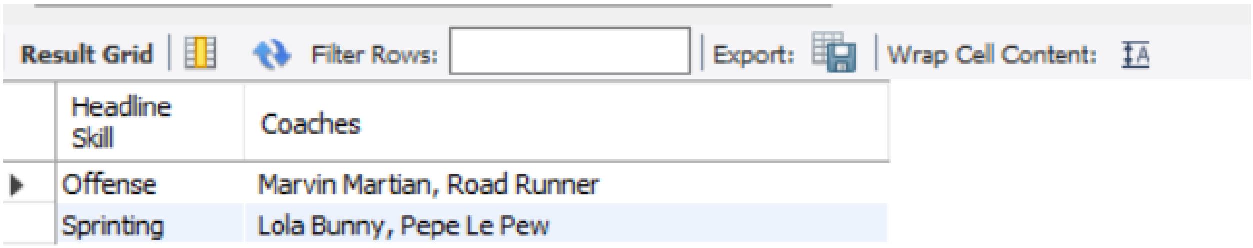
```
values (1,9), (2, 15), (3, 20), (4,12), (5,19), (6,10), (7,14), (8,18), (11, 13), (16, 17);

-- Inserting values to the playerTalents table which contains the attributes in order:
-- PlayerID and sportsName
INSERT INTO playerTalents
VALUES (1, 'Football'), (1, 'Swimming'),
       (2, 'Tennis'), (2, 'Volleyball'),
       (3, 'Bowling'),
       (4, 'Pool'), (4, 'Golf'),
       (5, 'Football'), (5, 'Swimming'),
       (6, 'Bowling'),
       (7, 'Basketball'),
       (8, 'Baseball'),
       (9, 'Swimming'),
       (10, 'Ping Pong'), (10, 'Golf'),
       (11, 'Volleyball'),
       (12, 'Bowling'),
       (13, 'Baseball'), (13, 'Basketball'),
       (14, 'Pool'), (14, 'Volleyball');

-- Inserting values to the coachSkills table which contains the attributes in order:
-- coachID and coachSkillTyoe. Inserting was done in individual to check if constraints for key values are met
INSERT INTO coachSkills
VALUES  (1, 'Speed'), (2, 'Defense'), (2, 'Offense');
INSERT INTO coachSkills
VALUES (2, 'High Jumping'), (3, 'Agility'), (3, 'Endurance');
INSERT INTO coachSkills
VALUES  (4, 'Power Lifting'), (4, 'Speed Running'), (5, 'Advanved Handeling');
INSERT INTO coachSkills
VALUES  (5, 'Speed Swimming'), (6, 'Strength Endurance'), (6, 'Speed Endurance');
INSERT INTO coachSkills
VALUES (7, 'Strategy Planning'), (7, 'Pressure Handling'), (8, 'Slam Dunking');
INSERT INTO coachSkills
VALUES   (8, 'Dunking'), (9, 'Sprinting'), (9, 'Assult Swimming'), (10, 'Power Jumping');
INSERT INTO coachSkills
VALUES     (10, 'Puzzle Solving'), (11, 'Trick Plays'), (11, 'Speed Passing');
INSERT INTO coachSkills
VALUES     (12, 'Agility'), (12, 'Top Leadership'), (13, 'Advanced Throwing');
INSERT INTO coachSkills
VALUES    (13, 'High Jumping'), (14, 'Offense'), (14, 'Speed');
INSERT INTO coachSkills
VALUES     (15, 'Advanced Catching'), (15, 'Cunning'), (16, 'Under Water Basket Weaving');
INSERT INTO coachSkills
```

```
VALUES   (16, 'Stopping Power'), (17, 'Deep Diving'), (17, 'Warm up Movments');
INSERT INTO coachSkills
VALUES     (18, 'Arm Strengthing'), (18, 'Core Strengthing'), (19, 'Leg Strengthing');
INSERT INTO coachSkills
VALUES     (19, 'Team Unity'), (20, 'Speed Jumping'), (20, 'Power Running');
INSERT INTO coachSkills
VALUES     (21, 'Power Sliding'), (21, 'Energy Conservation');

-- Inserting values to the headcoach table which contains the attributes in order:
-- coachID and headlineSkill
INSERT INTO headcoach
VALUES  (21, 'Power Sliding'), (18, 'Arm Strengthing'),
        (17, 'Deep Diving'), (14, 'Offense'), (15, 'Advanced Catching'),
        (2, 'Offense'), (3, 'Endurance'), (4, 'Power Lifting'), (5, 'Speed Swimming'), (6, 'Strength
Endurance'),
        (7, 'Pressure Handling'),(8, 'Dunking'),(9, 'Sprinting'),(10, 'Puzzle Solving'),(11, 'Trick
Plays'),
        (12, 'Top Leadership'),(13, 'Advanced Throwing'), (16, 'Stopping Power'), (19, 'Team Unity'),
        (20, 'Speed Jumping');
INSERT INTO headcoach
VALUES (1, 'Sprinting');


-- Inserting values to the coachspecialty table which contains the attributes in order:
-- sportsName and coachID
insert into coachspecialty
values  ('Football', 1), ('Ping Pong', 2), ('Soccer', 3),
        ('Basketball', 4), ('Baseball', 5), ('Swimming', 6),
        ('Golf', 7), ('Bowling', 8), ('Pool', 9), ('Tennis', 20), ('Volleyball', 15);

-- Inserting values to the mentorship table which contains the attributes in order:
-- PlayerID, sportsName, coachID.
insert into mentorship
values  (1, 'Swimming', 6), (2, 'Tennis', 20), (3, 'Bowling', 8),
        (4, 'Golf', 7), (5, 'Swimming', 6), (7, 'Basketball', 4),
        (8, 'Baseball', 5), (9, 'Swimming', 6), (10, 'Ping Pong', 2),
        (11, 'Volleyball', 15), (12, 'Bowling', 8), (13, 'Baseball', 5),
        (14, 'Pool', 9);

-- Inserting values to coachSkills to have cases for query #2
insert into coachskills
values (7, 'Offense'), (7, 'Speed'), (16, 'Deep Diving'), (1, 'Sprinting'), (10, 'Speed Endurance');
insert into coachskills
values (6, 'Puzzle Solving'), (4,'Agility'), (11, 'High Jumping'), (13, 'Trick Plays'), (15, 'Offense');
```

#QUERY 1

List all of the headline skills that are shared by two or more coaches, and list the names of the coaches who have that skill. One row of output for each skill. List the headline skills in alphabetical order.

```
SELECT headlineSkill AS "Headline Skill",
GROUP_CONCAT(CONCAT(' ', coachFirstName, ' ', coachLastName)) AS "Coaches"
FROM headcoach
NATURAL JOIN coach
GROUP BY headlineSkill
HAVING COUNT(headlineSkill) >= 2;
```



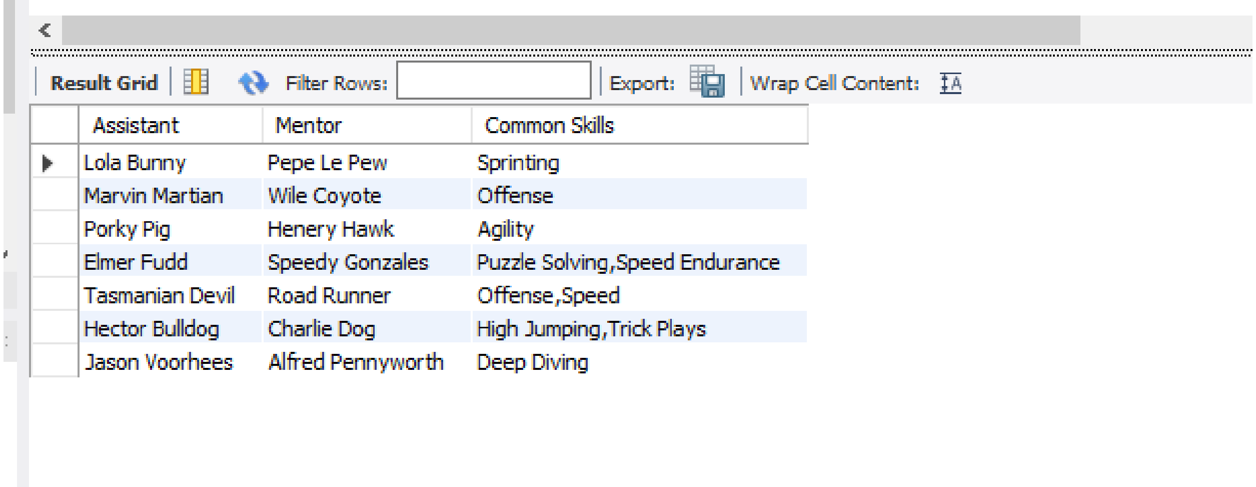
Headline Skill	Coaches
Offense	Marvin Martian, Road Runner
Sprinting	Lola Bunny, Pepe Le Pew

#QUERY 2

List each assistant coach's first name, last name, their mentor's first name, last name, and any skills that the two coaches have in common. One row of output for each assistant coach. List the assistant coaches sorted by last name, then their first name.

```
select concat(assistant.coachFirstName, ' ', assistant.coachLastName) 'Assistant',
concat(mentor.coachFirstName, ' ',  mentor.coachLastName) 'Mentor',
group_concat(cs1.coachSkillType)'Common Skills'
from assistantcoach inner join coach assistant on assistantcoach.coachID = assistant.coachID
inner join coach mentor on assistantcoach.mentorID = mentor.coachID
inner join coachskills cs1 on cs1.coachID = assistant.coachID
inner join coachskills cs2 on cs2.coachID = mentor.coachID
where cs1.coachSkillType = cs2.coachSkillType
```

```
group by cs1.coachID;
```

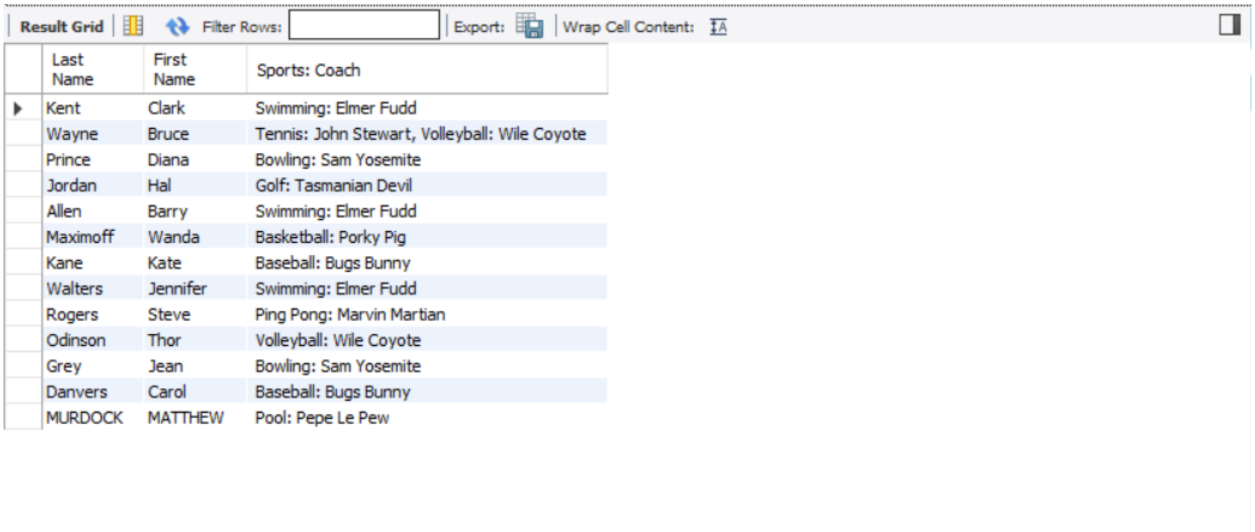


Assistant	Mentor	Common Skills
Lola Bunny	Pepe Le Pew	Sprinting
Marvin Martian	Wile Coyote	Offense
Porky Pig	Henery Hawk	Agility
Elmer Fudd	Speedy Gonzales	Puzzle Solving,Speed Endurance
Tasmanian Devil	Road Runner	Offense,Speed
Hector Bulldog	Charlie Dog	High Jumping,Trick Plays
Jason Voorhees	Alfred Pennyworth	Deep Diving

#QUERY 3

For each player, list their first name and last name, each sport that we coach them in, and the coach’s first name and last name. One row of output for each player. List them by their last name, then their first name.

```
select playerLastName 'Last Name', playerFirstName 'First Name', group_concat(' ',
concat(sportsName),  ': ',  concat(coachFirstName,  ' ', coachLastName)) as 'Sports: Coach'
from mentorship
natural join player natural join coach
group by playerID;
```



Last Name	First Name	Sports: Coach
Kent	Clark	Swimming: Elmer Fudd
Wayne	Bruce	Tennis: John Stewart, Volleyball: Wile Coyote
Prince	Diana	Bowling: Sam Yosemite
Jordan	Hal	Golf: Tasmanian Devil
Allen	Barry	Swimming: Elmer Fudd
Maximoff	Wanda	Basketball: Porky Pig
Kane	Kate	Baseball: Bugs Bunny
Walters	Jennifer	Swimming: Elmer Fudd
Rogers	Steve	Ping Pong: Marvin Martian
Odinson	Thor	Volleyball: Wile Coyote
Grey	Jean	Bowling: Sam Yosemite
Danvers	Carol	Baseball: Bugs Bunny
MURDOCK	MATTHEW	Pool: Pepe Le Pew