**Final Relational Schema**

```
CREATE TABLE Team(
        teamID CHAR(5),
        record CHAR(7) ,
        goals INT,
        shots INT,
        shotsOnGoal INT,
        saves INT,
        headCoach TEXT,
        assistantCoach TEXT,
        PRIMARY KEY (teamID)
);

CREATE TABLE Player (
        email TEXT NOT NULL,
        name TEXT NOT NULL,
        position CHAR(3),
        class INT,
        gamesPlayed INT,
        gamesStarted INT,
        goals INT,
        assists INT,
        shots INT,
        shotsOnGoal INT,
        goalsAllowed INT,
        saves INT,
        PRIMARY KEY (email)
);

CREATE TABLE Session(
        date CHAR(5) NOT NULL,
        type TEXT NOT NULL,
        PRIMARY KEY (date)
);


CREATE TABLE Device(
        deviceID CHAR(3) NOT NULL,
        PRIMARY KEY (deviceID)
);
```

**Relationship Sets to Tables**

```
CREATE TABLE Holds (
        teamID CHAR(5) NOT NULL,
        date CHAR(5) NOT NULL,
        PRIMARY KEY (date, teamID),
        FOREIGN KEY (teamID) REFERENCES Team,
        FOREIGN KEY (date) REFERENCES Session
);


CREATE TABLE ParticipatesIn (
        email CHAR(20) NOT NULL,
        date CHAR(5) NOT NULL,
        teamID CHAR(5) NOT NULL,
        PRIMARY KEY (email, date, teamID)
        FOREIGN KEY (email) references Player,
        FOREIGN KEY (date, teamID) references Holds
);

CREATE TABLE RecordsStatsOn (
        deviceID CHAR(3) NOT NULL,
        email CHAR(20) NOT NULL,
        date CHAR(5) NOT NULL,
        teamID CHAR(5) NOT NULL,
        distance FLOAT,
        sprintDistance FLOAT,
        energy FLOAT,
        playerLoad FLOAT,
        topSpeed FLOAT,
        distancePerMin FLOAT,
        PRIMARY KEY (deviceID, email, date, teamID)
        FOREIGN KEY (email, date, teamID) references ParticipatesIn,
        FOREIGN KEY (devideID) references Device
);


CREATE TABLE Tracks (
        deviceID CHAR(3) NOT NULL,
        email TEXT NOT NULL,
        season CHAR(6) NOT NULL,
```

PRIMARY KEY (season, email, deviceID)
FOREIGN KEY (email) references Player,
FOREIGN KEY (deviceID) references Device
)

**App Example Queries**

The backend server has endpoints that support basic, straightforward SELECT, INSERT, and UPDATE queries.

SELECT format: `` `SELECT ${field} FROM ${table} WHERE ${condition};` ``
INSERT format: `` `INSERT INTO ${table} ${field} VALUES ${values} WHERE ${condition};` ``
UPDATE format: `` `UPDATE ${table} SET ${field}=${values} WHERE ${condition};` ``
Subject to change given various request parameters as indicated in string literal above

It also has an endpoint that supports more complex queries; this endpoint is called with a full custom query given as a request parameter.

**Session Specific Queries**

SELECT date, sessionid, type FROM session;

SELECT date, sessionid, type FROM session WHERE date ILIKE '%_Month_Day_Year%';

SELECT AVG(distance) as distance, AVG(sprintdistance) as sprintdistance, AVG(topspeed) as topspeed, AVG(energy) as energy, AVG(playerload) as playerload FROM recordsstatson WHERE sessionid = `_sessionid`;

**Player Specific Queries**

SELECT name, email FROM player;

SELECT name, email FROM player WHERE name ILIKE '%_name%';

SELECT distance, sprintdistance, energy, topspeed, playerload FROM recordsstatson WHERE email = 'nameClass@amherst.edu' AND sessionid = 'xxxxx';

The player and session specific queries are used throughout the app to compile comprehensive player data on a per-session and per-season basis; that data is then further filtered (according to user input, if any) using queries that retrieve the relevant players/sessions.

**Leaderboard Queries**

For a given stat:

```
SELECT R.email, R.stat FROM recordsstatson R
WHERE R.stat = (SELECT MAX(stat) FROM recordsstatson WHERE email = R.email)
ORDER BY R.stat DESC;
```

Queried for each relevant stat (distance, sprintdistance, energy, playerload, topspeed, distanceperminute)

**Update/Insert Queries (next steps!!)**

- We want to try to add an accessible means for our strength and conditioning coach to mass-populate/update data