

Problem 1: Consider the relations `PLAYERS` and `PLAYS` given by the schemas below.

- `PLAYERS` (`playerID`, `firstName`, `lastName`, `gender`, `DOB`, `height`, `weight`, `drafted`)
- `PLAYS` (`playerID`, `teamID`, `teamName`, `number`, `position`, `startYear`)

`PLAYERS` provides information on all basketball players in the league, giving the `playerID`, first name and last name of the player, the gender, the date of birth (`DOB`), the player's height and weight, and the year they were drafted into the league.

`PLAYS` provides information about which players play on which teams. A player with `playerID` plays on a team with a `teamID` and team name. The player has a number, the position they play on the team, and the year they started playing with this team.

For example, `playerID` 5 plays with `teamID` 1, the Toronto Raptors, with the number 4, in the point guard position, since 2021. Given these schemas, answer the following questions:

1.a (9 marks)

Identify three candidate keys. For each candidate key, describe the key, and briefly state the assumptions or conditions under which each candidate key would be valid

Candidate keys:

1. `playerID` in `PLAYERS` relation:

- description: `playerID` contains a sole attribute, so it is minimal superkey. Given that each player will have unique `playerID`
- assumption: each players has unique `playerID`

2. `{playerID, teamID, number}` in `PLAYS` relation:

- description: `{playerID, teamID, number}` is minimal superkey given assumption.
- assumption: A player uses the same number for their duration at a given team.

3. {playerID, teamID, startYear} in PLAYS relation:

- description: {playerID, teamID, startYear} identifies the assumption, making it a minimal superkey.
- assumption: A player can only be associated with a team at a given period in time.

🔍 1.b (6 marks)

List three integrity constraints that should hold over these relations. For each constraint, describe in one sentence why your constraint is necessary.

1. playerID in PLAYS references playerID in PLAYERS:

- reason: foreign key constraint is necessary to ensure referential integrity, in other word, every player in PLAY must exist in PLAYERS

2. drafted in PLAYERS must be less than or equal to startYear in PLAYS:

- reason: temporal integrity constraint, i.e., a player cannot start playing for a team before they were drafted into the league

3. {teamID, number} in PLAYS table must be unique per playerID

- reason: uniqueness constraint, i.e., no two players on the same team have the same number at any point in time