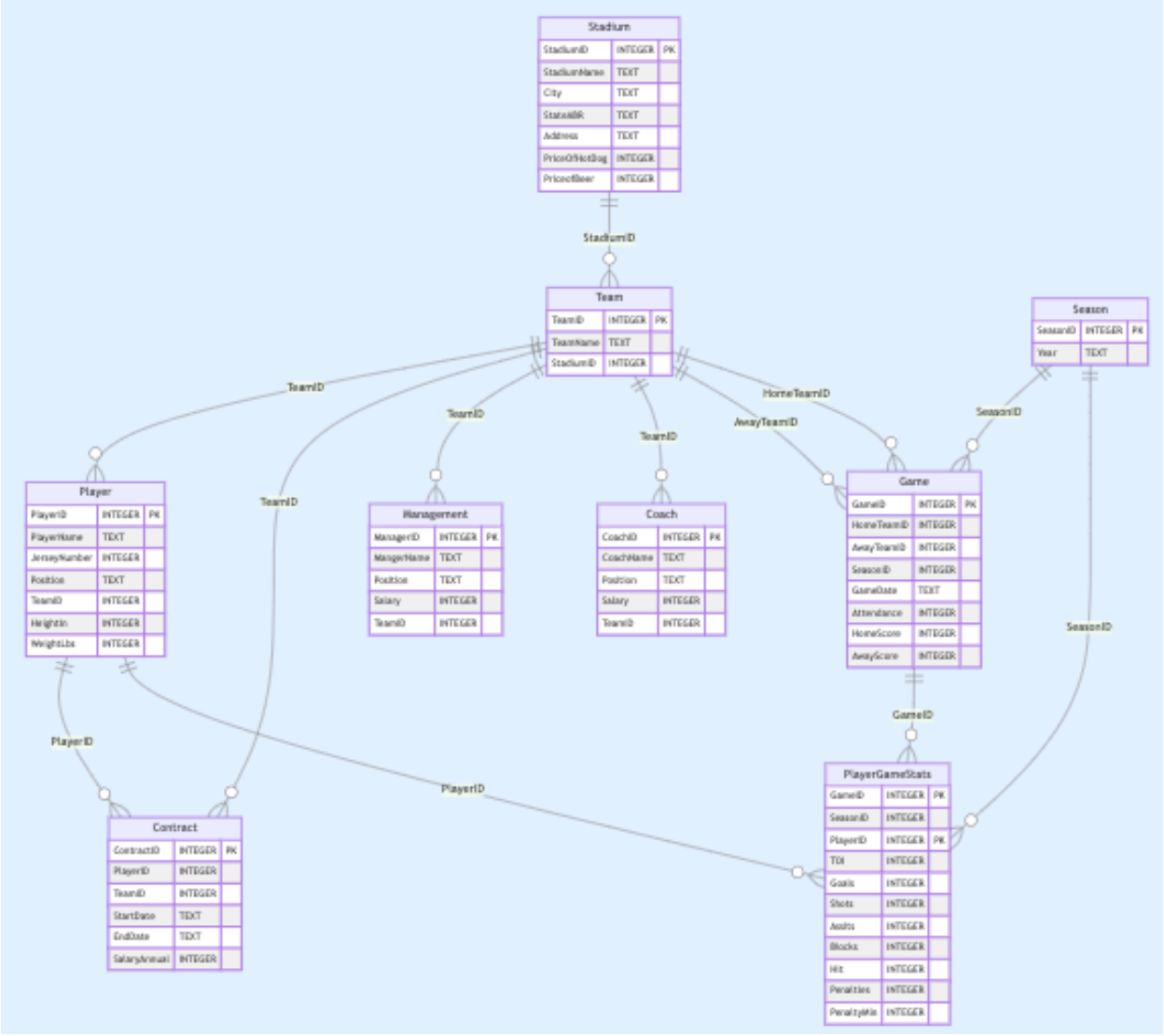
Operational Database

Overview and Diagram

I first developed my Relational Database for a hockey league in dbdiagram.io then exported it to a “MySQL” file. Where I then used an LLM to convert the MySQL code into SQLlite which could be read with https://orpinots.github.io/sql-sandbox/ to visualize the diagram and construct queries. The Dataset contains 9 entities; Stadium, Team, Player, Coach, Management, Contract, Game, Season, and PlayerGameStats. The schema is normalized to a level of 3NF, but is essentially in BCNF. In my schema every determinant is either a primary key or a foreign key pointing to a primary key, with only slight exceptions. In Season, ‘Year’ could theoretically be unique but it’s not declared unique. In Stadium, ‘StadiumName’ could be unique, but it’s not declared as such. While BCNF does not require those to be declared unique; it requires no functional dependency violations, which my schema does not have.



Querries

Query 1, CASE

Identify a player's BMI and resulting classification; "Heavy", "Mid", "Light"

```
SELECT
   PlayerID,
   PlayerName,
   HeightIn,
   WeightLbs,
   ROUND(WeightLbs * 703.0 / (HeightIn * HeightIn), 2) AS BMI,
   CASE
      WHEN WeightLbs * 703.0 / (HeightIn * HeightIn) >= 30 THEN 'Heavy'
      WHEN WeightLbs * 703.0 / (HeightIn * HeightIn) >= 25 THEN 'Mid'
      ELSE 'Lean'
   END AS BMI_Class
FROM Player;
```

PlayerID	PlayerName	HeightIn	WeightLbs	BMI	BMI_Class
1	Alex Carter	72	185	25.09	Mid
2	Max Dalton	73	190	25.06	Mid
3	Tim Johnson	75	205	25.62	Mid
4	Evan Lee	74	195	25.03	Mid
5	Jake Reilly	71	180	25.1	Mid
6	Chris Stone	74	200	25.68	Mid
7	Owen Phillips	76	210	25.56	Mid
8	Ryan Murphy	73	198	26.12	Mid

Query 2 GROUP BY / HAVING,

Calculate the offensive efficiency of players having at least 1 goal

```
Select
        P.playerid,
        sum(pgs.toi) as Tot_TOI,
        sum(pgs.goals) as Tot_Goals,
        round((sum(pgs.shots) / sum(pgs.TOI)),2) as GPM
from player as p
join playergamestats as pgs on p.playerid = pgs.playerid
group by p.playerid
        having Tot_Goals > 0
order by Tot_Goals desc
```

PlayerID	Tot_TOI	Tot_Goals	GPM
2	20	2	0
6	19	1	0
5	17	1	0
1	18	1	0

Query 3 Subquery

Retrieve the p;ayers where the total TOI exceeded the average TOI for all games.

```
SELECT
   p.PlayerID,
   p.PlayerName,
   SUM(s.TOI) AS Tot_TOI
FROM Player p
JOIN PlayerGameStats s
   ON p.PlayerID = s.PlayerID
GROUP BY p.PlayerID
HAVING SUM(s.TOI) > (
   SELECT AVG(PlayerTotalTOI)
   FROM (
      SELECT SUM(TOI) AS PlayerTotalTOI
      FROM PlayerGameStats
      GROUP BY PlayerID)
)
```

PlayerID	PlayerName	TotalTOI
4	Evan Lee	60
8	Ryan Murphy	60

Query 4 OUTER JOIN

Return all teams and their stadium info

```
SELECT
      t.TeamID,
      t.TeamName,
      s.StadiumName,
      s.City,
      s.StateABR
FROM Team t
LEFT JOIN Stadium s
      ON t.StadiumID = s.StadiumID;
```

TeamID	TeamName	StadiumName	City	StateABR
1	Eugene Yetis	Cascade Arena	Eugene	OR
2	Boulder Blizzards	Rocky Rink	Boulder	CO

Query 5 PARTITION BY

Rank players **within each team** by salary (highest to lowest)

```
SELECT
        p.TeamID,
        t.TeamName,
        p.PlayerID,
        p.PlayerName,
        c.SalaryAnnual,
        RANK() OVER (PARTITION BY p.TeamID ORDER BY c.SalaryAnnual DESC)
    AS SalaryRank
    FROM Player p
    JOIN Contract c
        ON p.PlayerID = c.PlayerID
    JOIN Team t
        ON p.TeamID = t.TeamID
    ORDER BY p.TeamID, SalaryAnnual DESC
```

TeamID	TeamName	PlayerID	PlayerName	SalaryAnnual	SalaryRank
1	Eugene Yetis	3	Tim Johnson	3100000	1
1	Eugene Yetis	1	Alex Carter	2500000	2
1	Eugene Yetis	4	Evan Lee	2200000	3
1	Eugene Yetis	2	Max Dalton	1800000	4
2	Boulder Blizzards	7	Owen Phillips	3300000	1
2	Boulder Blizzards	5	Jake Reilly	2400000	2
2	Boulder Blizzards	8	Ryan Murphy	2100000	3
2	Boulder Blizzards	6	Chris Stone	2000000	4

SQL Table and Insert Functions

```
PRAGMA foreign_keys = ON;

CREATE TABLE Player (
  PlayerID INTEGER PRIMARY KEY,
  PlayerName TEXT,
  JerseyNumber INTEGER,
  Position TEXT,
  TeamID INTEGER,
  HeightIn INTEGER,
  WeightLbs INTEGER,
  FOREIGN KEY (TeamID) REFERENCES Team(TeamID)
);

CREATE TABLE Contract (
  ContractID INTEGER PRIMARY KEY,
  PlayerID INTEGER,
  TeamID INTEGER,
  StartDate TEXT,
  EndDate TEXT,
  SalaryAnnual INTEGER,
  FOREIGN KEY (PlayerID) REFERENCES Player(PlayerID),
  FOREIGN KEY (TeamID) REFERENCES Team(TeamID)
);

CREATE TABLE Team (
  TeamID INTEGER PRIMARY KEY,
  TeamName TEXT,
  StadiumID INTEGER,
  FOREIGN KEY (StadiumID) REFERENCES Stadium(StadiumID)
);

CREATE TABLE Management (
  ManagerID INTEGER PRIMARY KEY,
  MangerName TEXT,
  Position TEXT,
  Salary INTEGER,
  TeamID INTEGER,
  FOREIGN KEY (TeamID) REFERENCES Team(TeamID)
);

CREATE TABLE Coach (
  CoachID INTEGER PRIMARY KEY,
  CoachName TEXT,
  Position TEXT,
  Salary INTEGER,
  TeamID INTEGER,
  FOREIGN KEY (TeamID) REFERENCES Team(TeamID)
);

CREATE TABLE Stadium (
  StadiumID INTEGER PRIMARY KEY,
  StadiumName TEXT,
```

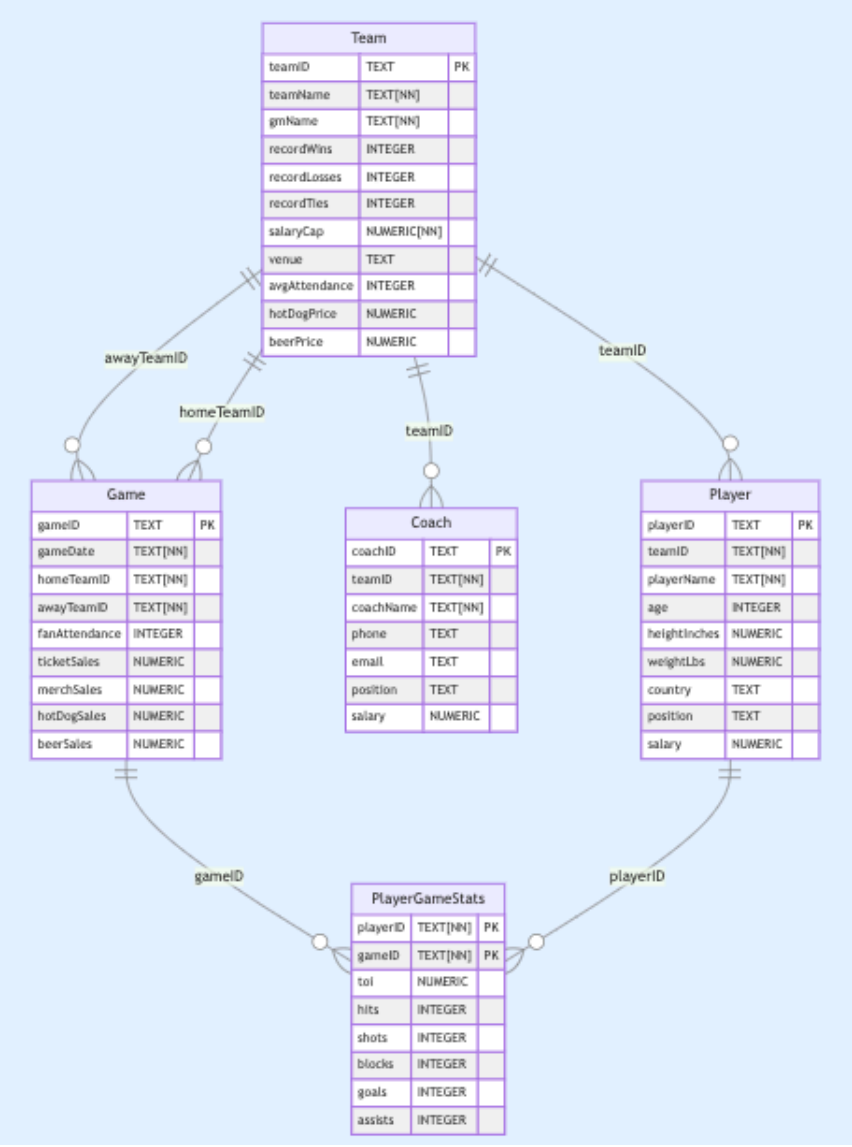
```
  City TEXT,
  StateABR TEXT,
  Address TEXT,
  PriceOfHotDog INTEGER,
  PriceofBeer INTEGER
);

CREATE TABLE Season (
  SeasonID INTEGER PRIMARY KEY,
  Year TEXT
);

CREATE TABLE Game (
  GameID INTEGER PRIMARY KEY,
  HomeTeamID INTEGER,
  AwayTeamID INTEGER,
  SeasonID INTEGER,
  GameDate TEXT,
  Attendance INTEGER,
  HomeScore INTEGER,
  AwayScore INTEGER,
  FOREIGN KEY (HomeTeamID) REFERENCES Team(TeamID),
  FOREIGN KEY (AwayTeamID) REFERENCES Team(TeamID),
  FOREIGN KEY (SeasonID) REFERENCES Season(SeasonID)
);

CREATE TABLE PlayerGameStats (
  GameID INTEGER,
  SeasonID INTEGER,
  PlayerID INTEGER,
  TOI INTEGER,
  Goals INTEGER,
  Shots INTEGER,
  Assits INTEGER,
  Blocks INTEGER,
  Hit INTEGER,
  Penalties INTEGER,
  PenaltyMin INTEGER,
  PRIMARY KEY (GameID, PlayerID),
  FOREIGN KEY (GameID) REFERENCES Game(GameID),
  FOREIGN KEY (SeasonID) REFERENCES Season(SeasonID),
  FOREIGN KEY (PlayerID) REFERENCES Player(PlayerID)
);

-- Insert Functions
INSERT INTO Stadium (StadiumID, StadiumName, City, StateABR, Address, PriceOfHotDog, PriceofBeer)
VALUES
(1, 'Cascade Arena', 'Eugene', 'OR', '100 Ice Ave', 7, 12),
(2, 'Rocky Rink', 'Boulder', 'CO', '42 Mountain Blvd', 6, 10);

INSERT INTO Team (TeamID, TeamName, StadiumID) VALUES
(1, 'Eugene Yetis', 1),
(2, 'Boulder Blizzards', 2);
```

```
INSERT INTO Season (SeasonID, Year) VALUES
(1, '2025-2026');

INSERT INTO Player (PlayerID, PlayerName, JerseyNumber, Position, TeamID, HeightIn, WeightLbs)
VALUES
(1, 'Alex Carter',     9,  'LW', 1, 72, 185),
(2, 'Max Dalton',      13,  'RW', 1, 73, 190),
(3, 'Tim Johnson',     5,  'D',  1, 75, 205),
(4, 'Evan Lee',        30,  'G',  1, 74, 195),
(5, 'Jake Reilly',     22,  'LW', 2, 71, 180),
(6, 'Chris Stone',     7,  'RW', 2, 74, 200),
(7, 'Owen Phillips',  55,  'D',  2, 76, 210),
(8, 'Ryan Murphy',     29,  'G',  2, 73, 198);

INSERT INTO Contract (ContractID, PlayerID, TeamID, StartDate, EndDate, SalaryAnnual) VALUES
(1,  1, 1, '2025-07-01', '2027-06-30', 2500000),
(2,  2, 1, '2025-07-01', '2026-06-30', 1800000),
(3,  3, 1, '2025-07-01', '2028-06-30', 3100000),
(4,  4, 1, '2025-07-01', '2026-06-30', 2200000),
(5,  5, 2, '2025-07-01', '2027-06-30', 2400000),
(6,  6, 2, '2025-07-01', '2026-06-30', 2000000),
(7,  7, 2, '2025-07-01', '2028-06-30', 3300000),
(8,  8, 2, '2025-07-01', '2026-06-30', 2100000);

INSERT INTO Game (GameID, HomeTeamID, AwayTeamID, SeasonID, GameDate, Attendance,
HomeScore, AwayScore) VALUES
(1, 1, 2, 1, '2025-10-22', 12800, 4, 3);

INSERT INTO PlayerGameStats (GameID, SeasonID, PlayerID, TOI, Goals, Shots, Assits, Blocks, Hit,
Penalties, PenaltyMin) VALUES
(1, 1, 1, 18, 1, 4, 1, 0, 2, 0, 0),
(1, 1, 2, 20, 2, 5, 0, 1, 3, 1, 2),
(1, 1, 3, 24, 0, 2, 1, 4, 1, 0, 0),
(1, 1, 4, 60, 0, 0, 0, 0, 0, 0, 0),
(1, 1, 5, 17, 1, 3, 0, 0, 1, 0, 0),
(1, 1, 6, 19, 1, 4, 1, 1, 2, 0, 0),
(1, 1, 7, 26, 0, 1, 1, 5, 2, 1, 2),
(1, 1, 8, 60, 0, 0, 0, 0, 0, 0, 0);

INSERT INTO Management (ManagerID, MangerName, Position, Salary, TeamID) VALUES
(1, 'Sarah Kim', 'General Manager', 1500000, 1),
(2, 'Leo Grant', 'General Manager', 1450000, 2);

INSERT INTO Coach (CoachID, CoachName, Position, Salary, TeamID) VALUES
(1, 'Mike Turner', 'Head Coach', 1200000, 1),
(2, 'Tom Alvarez', 'Head Coach', 1150000, 2);
```

STAR Schema Database

Overview and Diagram

I first developed my Relational Database for a hockey league in dbdiagram.io then exported it to a “MySQL” file. Where i then used an LLM to convert the MySQL code into SQLite which could be read with https://orpinots.github.io/sql-sandbox/ to visualize the diagram and construct queries. This database models a hockey league with five well-structured entities and clear relationships. The Team table stores organizational and financial information; both Player and Coach reference Team, creating one-to-many relationships. The Game table links two teams (home and away) with attendance and revenue fields. The PlayerGameStats table forms a many-to-many bridge between Player and Game, storing individual performance metrics for each matchup. The schema is normalized to 3NF, tables use valid primary keys, attributes depend only on those keys, and no transitive dependencies remain.



Querries

Query 1, CASE

Identify if a team's fanbase favors either Beer or Hot Dog for home games

```
Select
        p.TeamID,
        round(avg(p.Age),2) as TeamAge,
        round(avg(p.heightInches),2) as TeamHeight,
        round(avg(p.weightLbs),2) as TeamWeight,
        CASE when g.beerSales > g.hotDogSales then 'Beer Fans' else 'Hot Dog Fans'
        end as 'FanFavorite'
from player as p
join game as g on g.homeTeamID = p.TeamID
group by p.TeamID
order by TeamAge desc
```

teamID	TeamAge	TeamHeight	TeamWeight	FanFavorite
T004	27.29	73.79	208.57	Beer Fans
T003	27	73.29	204.43	Beer Fans
T002	26.43	73.14	202.14	Beer Fans
T001	26.29	72.71	196.86	Beer Fans

Query 2 GROUP BY / HAVING,

Calculate the offensive efficiency of players having at least 10 shots

```
Select
        pg.playerID,
        P.salary,
        round(sum(pg.TOI),1) as Tot_TOI,
        sum(pg.shots) as Tot_Shots,
        round((sum(pg.shots) / sum(pg.TOI)),2) as SPM
from PlayerGameStats as pg
join Player as p on p.playerID = pg.playerID
group by pg.playerID
        having sum(pg.shots) > 10
order by SPM desc
```

playerID	salary	Tot_TOI	Tot_Shots	GPM
P003	5100000	88.7	19	0.21
P008	5200000	89.6	15	0.17
P010	5300000	84.2	14	0.17
P017	5400000	90	15	0.17
P022	4800000	88	15	0.17
P001	5800000	91.4	15	0.16
P015	5100000	91	15	0.16
P024	5200000	81.9	11	0.13

Query 3 Subquery

Retrieve the games where the total ticket sales exceeded the average ticket sales for all games.

```
Select
        G.gameid,
        G.hometeamid,
        G.awayteamid,
        g.ticketsales
from game as g
where g.ticketsales > (select avg(ticketsales) from game group by gameid)
```

gameID	homeTeamID	awayTeamID	ticketSales
G002	T003	T004	1240000

Query 4 OUTER JOIN

List every team whether or not they have games recorded, showing total ticket revenue for home games.

```
select
        t.teamID,
        t.teamName,
        SUM(g.ticketSales) AS totalHomeTicketRevenue
from team as t
full outer JOIN Game as g on t.teamid = g.hometeamid
group by t.teamID, t.teamName
order by totalHomeTicketRevenue desc
```

teamID	teamName	totalHomeTicketRevenue
T001	Portland Penguins	2225000
T003	Salem Blades	1240000
T002	Eugene Elk	910000
T004	Bend Yetis	770000

Query 5 PARTITION BY

Find the top 3 tallest skaters within each team

```
select *
from (
        Select p.playerName, t.teamName, p.heightinches,
                rank() over(partition by p.teamid order by p.heightinches desc) as
        heightRank
        from player as p
        join team as t on t.teamid = p.teamid
        order by p.teamid, heightRank )
where heightRank <= 3
```

playerName	teamName	heightinches	heightRank
Niko Talbot	Portland Penguins	75	1
Zach Olesen	Portland Penguins	74	2
Dylan Brooks	Portland Penguins	73	3
Henrik Nystrom	Eugene Elk	76	1
Toby Muller	Eugene Elk	75	2
Ryan Stewart	Eugene Elk	74.5	3
Jack Sorensen	Salem Blades	76.5	1
Josef Kravchuk	Salem Blades	75	2
Brodie Quinn	Salem Blades	74	3
Martin Ilves	Bend Yetis	77	1
Trevor Kostas	Bend Yetis	75	2
Max Benson	Bend Yetis	74.5	3

SQL Table and Insert Functions

```
PRAGMA foreign_keys = ON;

-- TEAMS
CREATE TABLE Team (
    teamID          TEXT PRIMARY KEY,      -- T001
    teamName        TEXT NOT NULL,
    gmName          TEXT NOT NULL,
    recordWins      INTEGER DEFAULT 0,
    recordLosses    INTEGER DEFAULT 0,
    recordTies      INTEGER DEFAULT 0,
    salaryCap       NUMERIC NOT NULL,
    venue           TEXT,
    avgAttendance   INTEGER,
    hotDogPrice     NUMERIC,
    beerPrice       NUMERIC
);

-- PLAYERS
CREATE TABLE Player (
    playerID        TEXT PRIMARY KEY,      -- P001
    teamID          TEXT NOT NULL,
    playerName      TEXT NOT NULL,
    age             INTEGER,
    heightInches    NUMERIC,
    weightLbs       NUMERIC,
    country         TEXT,
    position        TEXT,             -- C, LW, RW, D, G
    salary          NUMERIC,
    FOREIGN KEY (teamID) REFERENCES Team(teamID)
);

-- COACHES
CREATE TABLE Coach (
    coachID         TEXT PRIMARY KEY,      -- C001
    teamID          TEXT NOT NULL,
    coachName       TEXT NOT NULL,
    phone           TEXT,
    email           TEXT,
    position        TEXT,                -- HC or AC
    salary          NUMERIC,
    FOREIGN KEY (teamID) REFERENCES Team(teamID)
);

-- GAMES
CREATE TABLE Game (
    gameID          TEXT PRIMARY KEY,      -- G001
    gameDate        TEXT NOT NULL,         -- 'YYYY-MM-DD'
    homeTeamID      TEXT NOT NULL,
    awayTeamID      TEXT NOT NULL,
    fanAttendance   INTEGER,
    ticketSales     NUMERIC,
    merchSales      NUMERIC,
```

```
    hotDogSales     NUMERIC,
    beerSales       NUMERIC,
    FOREIGN KEY (homeTeamID) REFERENCES Team(teamID),
    FOREIGN KEY (awayTeamID) REFERENCES Team(teamID)
);

-- PLAYER & GAME STATS
CREATE TABLE PlayerGameStats (
    playerID        TEXT NOT NULL,
    gameID          TEXT NOT NULL,
    toi             NUMERIC,            -- time on ice (mins)
    hits            INTEGER,
    shots           INTEGER,
    blocks          INTEGER,
    goals           INTEGER,
    assists         INTEGER,
    PRIMARY KEY(playerID, gameID),
    FOREIGN KEY (playerID) REFERENCES Player(playerID),
    FOREIGN KEY (gameID) REFERENCES Game(gameID)
);

INSERT INTO Team (teamID, teamName, gmName, recordWins, recordLosses, recordTies, salaryCap,
venue, avgAttendance, hotDogPrice, beerPrice) VALUES
('T001','Portland Penguins','Steve Miller',14,6,0,600000000,'Rose Garden',15000,6.50,9.00),
('T002','Eugene Elk','Mark Thompson',11,9,0,550000000,'Civic Arena',12000,5.75,8.50),
('T003','Salem Blades','Rachel Carter',16,4,0,700000000,'Salem Coliseum',17000,6.00,9.25),
('T004','Bend Yetis','Kyle Jackson',9,11,1,450000000,'Mountain Arena',10000,6.25,9.75);

INSERT INTO Player (playerID, teamID, playerName, age, heightInches, weightLbs, country, position,
salary) VALUES
-- Portland T001
('P001','T001','Adam Wexler',26,72.0,190,'USA','C',5800000),
('P002','T001','Jonas Makarov',24,71.5,185,'RUS','LW',5400000),
('P003','T001','Dylan Brooks',27,73.0,200,'CAN','RW',5100000),
('P004','T001','Zach Olesen',25,74.0,205,'SWE','D',5200000),
('P005','T001','Carlos Diaz',29,72.5,198,'MEX','D',4800000),
('P006','T001','Mason Lee',23,71.0,182,'USA','RW',4500000),
('P007','T001','Niko Talbot',30,75.0,218,'FIN','G',6500000),
('P008','T002','Noah Richards',24,71.5,186,'USA','C',5200000),
('P009','T002','Luca Romano',26,72.0,193,'ITA','LW',5000000),
('P010','T002','Evan Baker',25,73.0,201,'USA','RW',5300000),
('P011','T002','Ryan Stewart',29,74.5,210,'CAN','D',5100000),
('P012','T002','Toby Muller',27,75.0,216,'GER','D',4700000),
('P013','T002','Declan Forrest',23,70.0,179,'USA','LW',4400000),
('P014','T002','Henrik Nystrom',31,76.0,230,'SWE','G',6300000),
('P015','T003','Carter Holland',23,71.0,184,'USA','C',5100000),
('P016','T003','Leo Vasiliev',25,72.5,195,'RUS','LW',4900000),
('P017','T003','Tyler Shan',27,73.5,205,'USA','RW',5400000),
('P018','T003','Brodie Quinn',28,74.0,212,'CAN','D',5500000),
('P019','T003','Josef Kravchuk',30,75.0,220,'CZE','D',5100000),
('P020','T003','Eli Ramirez',22,70.5,180,'USA','RW',4300000),
('P021','T003','Jack Sorensen',34,76.5,235,'DEN','G',6400000),
('P022','T004','Connor Beck',25,72.0,192,'USA','C',4800000),
('P023','T004','Roman Chen',26,73.0,200,'CHN','LW',4700000),
```

('P024','T004','Gabriel Ortiz',28,74.0,210,'MEX','RW',5200000),
('P025','T004','Trevor Kostas',27,75.0,218,'CAN','D',5300000),
('P026','T004','Max Benson',29,74.5,215,'USA','D',4900000),
('P027','T004','Liam Barrett',24,71.0,185,'USA','LW',4600000),
('P028','T004','Martin Ilves',32,77.0,240,'EST','G',6100000);

INSERT INTO Coach (coachID, teamID, coachName, phone, email, position, salary) VALUES
('C001','T001','Dan Coleman','503-555-9101','dan@penguins.com','Head Coach',8000000),
('C002','T001','Erik Sand','503-555-9102','erik@penguins.com','Assistant Coach',3500000),
('C003','T002','Tom Reid','541-555-8731','treid@elk.com','Head Coach',7200000),
('C004','T002','Jake Foster','541-555-8732','jfoster@elk.com','Assistant Coach',3300000),
('C005','T003','Chad Lowry','503-555-1175','lowry@blades.com','Head Coach',8400000),
('C006','T003','Owen Cyr','503-555-1176','ocy@blades.com','Assistant Coach',3200000),
('C007','T004','Jim Keller','541-555-2401','keller@yeti.com','Head Coach',6900000),
('C008','T004','Felix Moran','541-555-2402','fmoran@yeti.com','Assistant Coach',3100000);

INSERT INTO Game (gameID, gameDate, homeTeamID, awayTeamID, fanAttendance, ticketSales, merchSales) VALUES
('G001','2025-10-01','T001','T002',16000,1120000,320000),
('G002','2025-10-07','T003','T004',17000,1240000,345000),
('G003','2025-10-14','T002','T003',13000,910000,250000),
('G004','2025-10-21','T004','T001',10500,770000,190000),
('G005','2025-10-29','T001','T003',15800,1105000,310000);

INSERT INTO PlayerGameStats (playerID, gameID, toi, hits, shots, blocks, goals, assists) VALUES
('P001','G001',18.5,2,3,1,1,0),
('P002','G001',16.2,1,2,0,0,1),
('P003','G001',17.8,3,4,1,1,1),
('P004','G001',20.1,4,1,3,0,0),
('P005','G001',19.5,3,1,4,0,1),
('P006','G001',15.8,1,2,0,0,1),
('P007','G001',59.0,0,0,0,0,0),
('P008','G001',18.2,2,3,1,0,1),
('P009','G001',15.9,1,2,0,1,0),
('P010','G001',16.8,2,3,1,0,1),
('P011','G001',21.1,4,1,5,0,0),
('P012','G001',18.5,3,1,3,0,0),
('P013','G001',14.8,1,1,0,0,0),
('P014','G001',59.0,0,0,0,0,0),
('P015','G001',17.5,2,3,1,1,0),
('P016','G001',15.8,1,2,0,1,1),
('P017','G001',18.0,2,3,1,0,1),
('P018','G001',21.5,4,1,4,0,0),
('P019','G001',20.0,3,1,3,0,1),
('P020','G001',13.2,1,1,0,0,0),
('P021','G001',59.0,0,0,0,0,0),
('P022','G001',17.0,2,3,1,1,0),
('P023','G001',15.6,1,1,0,0,1),
('P024','G001',16.2,2,2,1,0,1),
('P025','G001',20.2,3,1,4,0,0),
('P026','G001',19.0,3,1,3,0,0),
('P027','G001',14.4,1,1,0,0,0),
('P028','G001',59.0,0,0,0,0,0),
('P015','G002',19.1,2,3,1,1,1),

```
('P016','G002',16.0,1,2,0,0,1),
('P017','G002',18.3,2,3,1,1,0),
('P018','G002',22.0,4,1,5,0,0),
('P019','G002',20.3,3,1,3,0,1),
('P020','G002',14.0,1,1,0,0,0),
('P021','G002',60.0,0,0,0,0,0),
('P022','G002',17.8,2,3,1,1,0),
('P023','G002',15.7,1,2,0,0,1),
('P024','G002',16.4,2,2,1,0,1),
('P025','G002',20.0,3,1,4,0,0),
('P026','G002',18.8,3,1,3,0,0),
('P027','G002',14.2,1,1,0,0,0),
('P028','G002',60.0,0,0,0,0,0),
('P001','G002',18.2,2,3,1,1,0),
('P002','G002',16.1,1,2,0,0,1),
('P003','G002',17.7,3,4,1,1,1),
('P004','G002',21.0,4,1,3,0,0),
('P005','G002',19.4,3,1,4,0,1),
('P006','G002',15.7,1,2,0,0,1),
('P007','G002',60.0,0,0,0,0,0),
('P008','G002',17.9,2,3,1,0,1),
('P009','G002',15.8,1,2,0,1,0),
('P010','G002',16.6,2,3,1,0,1),
('P011','G002',20.8,4,1,5,0,0),
('P012','G002',18.6,3,1,3,0,0),
('P013','G002',14.7,1,1,0,0,0),
('P014','G002',60.0,0,0,0,0,0),
('P008','G003',18.0,2,3,1,1,0),
('P009','G003',16.2,1,2,0,0,1),
('P010','G003',17.4,2,3,1,1,1),
('P011','G003',21.5,4,1,5,0,0),
('P012','G003',19.2,3,1,3,0,1),
('P013','G003',14.4,1,1,0,0,0),
('P014','G003',60.0,0,0,0,0,0),
('P015','G003',18.3,2,3,1,1,1),
('P016','G003',15.9,1,2,0,1,0),
('P017','G003',17.9,2,3,1,0,1),
('P018','G003',21.8,4,1,4,0,0),
('P019','G003',20.1,3,1,3,0,1),
('P020','G003',13.8,1,1,0,0,0),
('P021','G003',60.0,0,0,0,0,0),
('P001','G003',17.6,2,3,1,1,0),
('P002','G003',15.7,1,2,0,0,1),
('P003','G003',17.5,3,4,1,0,1),
('P004','G003',20.2,4,1,3,0,0),
('P005','G003',19.5,3,1,4,0,0),
('P006','G003',15.2,1,2,0,0,1),
('P007','G003',60.0,0,0,0,0,0),
('P022','G003',17.3,2,3,1,1,0),
('P023','G003',15.4,1,2,0,0,1),
('P024','G003',16.1,2,2,1,0,1),
('P025','G003',19.9,3,1,4,0,0),
('P026','G003',18.7,3,1,3,0,0),
('P027','G003',14.1,1,1,0,0,0),
```

('P028','G003',60.0,0,0,0,0,0),
('P022','G004',18.5,2,3,1,2,0),
('P023','G004',15.6,1,2,0,0,1),
('P024','G004',16.9,2,3,1,0,1),
('P025','G004',21.3,4,1,4,0,0),
('P026','G004',19.3,3,1,3,0,1),
('P027','G004',14.3,1,1,0,0,0),
('P028','G004',60.0,0,0,0,0,0),
('P001','G004',18.7,2,3,1,1,0),
('P002','G004',16.3,1,2,0,1,1),
('P003','G004',17.8,2,3,1,0,1),
('P004','G004',21.0,4,1,4,0,0),
('P005','G004',19.4,3,1,3,0,0),
('P006','G004',15.9,1,1,0,0,0),
('P007','G004',60.0,0,0,0,0,0),
('P015','G004',18.1,2,3,1,1,0),
('P016','G004',15.8,1,2,0,0,1),
('P017','G004',18.0,2,3,1,0,1),
('P018','G004',21.6,4,1,5,0,0),
('P019','G004',20.2,3,1,4,0,1),
('P020','G004',13.9,1,1,0,0,0),
('P021','G004',60.0,0,0,0,0,0),
('P008','G004',17.8,2,3,1,1,0),
('P009','G004',15.7,1,2,0,0,1),
('P010','G004',16.8,2,2,1,0,1),
('P011','G004',21.2,4,1,4,0,0),
('P012','G004',19.1,3,1,3,0,0),
('P013','G004',14.2,1,1,0,0,0),
('P014','G004',60.0,0,0,0,0,0),
('P001','G005',18.4,2,3,1,1,0),
('P002','G005',16.2,1,2,0,0,1),
('P003','G005',17.9,2,4,2,1,1),
('P004','G005',20.9,4,1,5,0,0),
('P005','G005',19.6,3,1,4,0,1),
('P006','G005',15.7,1,2,0,0,1),
('P007','G005',60.0,0,0,0,0,0),
('P015','G005',18.0,2,3,1,1,1),
('P016','G005',16.0,1,2,0,1,0),
('P017','G005',17.8,2,3,1,0,1),
('P018','G005',21.7,4,1,4,0,0),
('P019','G005',20.1,3,1,3,0,0),
('P020','G005',14.1,1,1,0,0,0),
('P021','G005',60.0,0,0,0,0,0),
('P008','G005',17.7,2,3,1,0,1),
('P009','G005',15.9,1,2,0,1,0),
('P010','G005',16.6,2,3,1,0,1),
('P011','G005',20.7,4,1,4,0,0),
('P012','G005',18.8,3,1,3,0,0),
('P013','G005',14.5,1,1,0,0,0),
('P014','G005',60.0,0,0,0,0,0),
('P022','G005',17.4,2,3,1,1,0),
('P023','G005',15.5,1,2,0,0,1),
('P024','G005',16.3,2,2,1,0,1),
('P025','G005',19.8,3,1,4,0,0),

```
('P026','G005',18.9,3,1,3,0,0),
('P027','G005',14.2,1,1,0,0,0),
('P028','G005',60.0,0,0,0,0,0);
```