```
USE BUDT703_Project_0507_08

-- Created for Tableau output
DROP VIEW IF EXISTS LongestWinStreaks;
DROP VIEW IF EXISTS LongestLossStreaks;
DROP VIEW IF EXISTS ConsecutiveWinGroups;
DROP VIEW IF EXISTS ConsecutiveLossGroups;
DROP VIEW IF EXISTS RankedGames;
DROP VIEW IF EXISTS Moneyball_Game_View;


--Essential
DROP VIEW IF EXISTS LocationWiseScore;
DROP VIEW IF EXISTS TournamentWinRate;
DROP VIEW IF EXISTS OpponentWinRate;
DROP TABLE IF EXISTS [Moneyball.Hold];
DROP TABLE IF EXISTS [Moneyball.Game];
DROP TABLE IF EXISTS [Moneyball.Tournament];
DROP TABLE IF EXISTS [Moneyball.Location];
DROP TABLE IF EXISTS [Moneyball.Opponent];



CREATE TABLE [Moneyball.Opponent] (
    oppId VARCHAR (20) NOT NULL,
    oppName VARCHAR (40),
    CONSTRAINT pk_Opponent_oppId PRIMARY KEY (oppId)
);

CREATE TABLE [Moneyball.Location] (
    locId VARCHAR (20) NOT NULL,
    locName VARCHAR (100),
    CONSTRAINT pk_Location_locId PRIMARY KEY (locId)
);


CREATE TABLE [Moneyball.Tournament] (
    trnName VARCHAR(50) NOT NULL,
    CONSTRAINT pk_Tournament_trnName PRIMARY KEY (trnName)
);

CREATE TABLE [Moneyball.Game] (
    gmeId VARCHAR(20) NOT NULL,
    gmeDate DATE,
    gmeTime TIME,
    gmeTimeOfDay AS (
        CASE
            WHEN gmeTime >= '06:00:00' AND gmeTime < '12:00:00' THEN 'Morning'
            WHEN gmeTime >= '12:00:00' AND gmeTime < '17:00:00' THEN 'Afternoon'
            WHEN gmeTime >= '17:00:00' AND gmeTime < '20:00:00' THEN 'Evening'
            ELSE 'Night'
        END
    ),
    gmeScore VARCHAR(20),
    gmeAt VARCHAR(15),
    trnName VARCHAR(50),
    CONSTRAINT pk_Game_gmeId PRIMARY KEY (gmeId),
    CONSTRAINT fk_Game_trnName FOREIGN KEY (trnName)
        REFERENCES [Moneyball.Tournament] (trnName)
```

```
        ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE [Moneyball.Hold] (
    gmeId VARCHAR (20) NOT NULL,
    oppId VARCHAR (20) NOT NULL,
    locId VARCHAR (20) NOT NULL,
    CONSTRAINT pk_Hold_gmeId_oppId_locId PRIMARY KEY (gmeId, oppId, locId),
    CONSTRAINT fk_Hold_gmeId FOREIGN KEY (gmeId)
        REFERENCES [Moneyball.Game] (gmeId)
        ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT fk_Hold_oppId FOREIGN KEY (oppId)
        REFERENCES [Moneyball.Opponent] (oppId)
        ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT fk_Hold_locId FOREIGN KEY (locId)
        REFERENCES [Moneyball.Location] (locId)
        ON DELETE CASCADE ON UPDATE CASCADE
);

GO
CREATE VIEW OpponentWinRate AS (
      SELECT
            opp.oppId AS 'Opponent ID',
            opp.oppName AS 'Opponent Team',
            COUNT(hold.gmeId) AS 'Games Played Against',
            SUM(CASE WHEN CAST(SUBSTRING(gme.gmeScore, 1, CHARINDEX('-',
gme.gmeScore) - 1) AS INT) >
                                    CAST(SUBSTRING(gme.gmeScore, CHARINDEX('-',
gme.gmeScore) + 1, LEN(gme.gmeScore)) AS INT) THEN 1 ELSE 0 END) AS 'Wins',
            SUM(CASE WHEN CAST(SUBSTRING(gme.gmeScore, 1, CHARINDEX('-',
gme.gmeScore) - 1) AS INT) <
                                    CAST(SUBSTRING(gme.gmeScore, CHARINDEX('-',
gme.gmeScore) + 1, LEN(gme.gmeScore)) AS INT) THEN 1 ELSE 0 END) AS 'Losses',
            SUM(CASE WHEN CAST(SUBSTRING(gme.gmeScore, 1, CHARINDEX('-',
gme.gmeScore) - 1) AS INT) =
                                    CAST(SUBSTRING(gme.gmeScore, CHARINDEX('-',
gme.gmeScore) + 1, LEN(gme.gmeScore)) AS INT) THEN 1 ELSE 0 END) AS 'Ties',
            (SUM(CASE WHEN CAST(SUBSTRING(gme.gmeScore, 1, CHARINDEX('-',
gme.gmeScore) - 1) AS INT) >
                                    CAST(SUBSTRING(gme.gmeScore, CHARINDEX('-',
gme.gmeScore) + 1, LEN(gme.gmeScore)) AS INT) THEN 1 ELSE 0 END) * 1.0 /
COUNT(hold.gmeId)) AS 'Win Rate Against Opponent'
      FROM
            [Moneyball.Opponent] opp
      JOIN
            [Moneyball.Hold] hold
            ON opp.oppId = hold.oppId
      JOIN
            [Moneyball.Game] gme
            ON hold.gmeId = gme.gmeId
      GROUP BY
            opp.oppId, opp.oppName
);
GO
CREATE VIEW TournamentWinRate AS (
      SELECT
            trn.trnName AS 'Tournament',
            COUNT(g.gmeId) AS 'Games Played Against',
            SUM(CASE WHEN CAST(SUBSTRING(g.gmeScore, 1, CHARINDEX('-', g.gmeScore)
```

```
- 1) AS INT) >
                                    CAST(SUBSTRING(g.gmeScore, CHARINDEX('-',
g.gmeScore) + 1, LEN(g.gmeScore)) AS INT) THEN 1 ELSE 0 END) AS 'Wins',
            SUM(CASE WHEN CAST(SUBSTRING(g.gmeScore, 1, CHARINDEX('-', g.gmeScore)
- 1) AS INT) <
                                    CAST(SUBSTRING(g.gmeScore, CHARINDEX('-',
g.gmeScore) + 1, LEN(g.gmeScore)) AS INT) THEN 1 ELSE 0 END) AS 'Losses',
            SUM(CASE WHEN CAST(SUBSTRING(g.gmeScore, 1, CHARINDEX('-', g.gmeScore)
- 1) AS INT) =
                                    CAST(SUBSTRING(g.gmeScore, CHARINDEX('-',
g.gmeScore) + 1, LEN(g.gmeScore)) AS INT) THEN 1 ELSE 0 END) AS 'Ties',
            (SUM(CASE WHEN CAST(SUBSTRING(g.gmeScore, 1, CHARINDEX('-', g.gmeScore)
- 1) AS INT) >
                                    CAST(SUBSTRING(g.gmeScore, CHARINDEX('-',
g.gmeScore) + 1, LEN(g.gmeScore)) AS INT) THEN 1 ELSE 0 END) * 1.0 /
COUNT(g.gmeId)) AS 'Tournament Win Rate'
      FROM
            [Moneyball.Tournament] trn
      LEFT JOIN
            [Moneyball.Game] g ON trn.trnName = g.trnName
      GROUP BY
            trn.trnName
);
GO
CREATE VIEW LocationWiseScore AS (
      SELECT
            gme.gmeAt AS 'Location Type',
            COUNT(g.gmeId) AS 'Games Played',
            SUM(CASE WHEN CAST(SUBSTRING(g.gmeScore, 1, CHARINDEX('-', g.gmeScore)
- 1) AS INT) >
                  CAST(SUBSTRING(g.gmeScore, CHARINDEX('-', g.gmeScore) + 1,
LEN(g.gmeScore)) AS INT) THEN 1 ELSE 0 END) AS 'Wins',
            SUM(CASE WHEN CAST(SUBSTRING(g.gmeScore, 1, CHARINDEX('-', g.gmeScore)
- 1) AS INT) <
                  CAST(SUBSTRING(g.gmeScore, CHARINDEX('-', g.gmeScore) + 1,
LEN(g.gmeScore)) AS INT) THEN 1 ELSE 0 END) AS 'Losses',
            SUM(CASE WHEN CAST(SUBSTRING(g.gmeScore, 1, CHARINDEX('-', g.gmeScore)
- 1) AS INT) =
                  CAST(SUBSTRING(g.gmeScore, CHARINDEX('-', g.gmeScore) + 1,
LEN(g.gmeScore)) AS INT) THEN 1 ELSE 0 END) AS 'Ties',
            (SUM(CASE WHEN CAST(SUBSTRING(g.gmeScore, 1, CHARINDEX('-', g.gmeScore)
- 1) AS INT) >
                  CAST(SUBSTRING(g.gmeScore, CHARINDEX('-', g.gmeScore) + 1,
LEN(g.gmeScore)) AS INT) THEN 1 ELSE 0 END) * 1.0 / COUNT(g.gmeId)) AS 'Win Rate'
            FROM
                  [Moneyball.Game] gme
            LEFT JOIN
                  [Moneyball.Hold] hold ON gme.gmeId = hold.gmeId
            LEFT JOIN
                  [Moneyball.Game] g ON hold.gmeId = g.gmeId
            WHERE
                  gme.gmeAt IS NOT NULL
            GROUP BY
                  gme.gmeAt
);

GO
CREATE VIEW Moneyball_Game_View AS
SELECT
```

```
    gmeTimeOfDay,
    CASE
        WHEN CAST(SUBSTRING(gmeScore, 1, CHARINDEX('-', gmeScore) - 1) AS INT) >
             CAST(SUBSTRING(gmeScore, CHARINDEX('-', gmeScore) + 1, LEN(gmeScore))
AS INT) THEN 1
        ELSE 0
    END AS WinFlag
FROM
    [Moneyball.Game]
WHERE
    gmeTimeOfDay IS NOT NULL;

GO
CREATE VIEW RankedGames AS (
      SELECT
        gmeId, gmeDate, gmeScore, trnName,
        ROW_NUMBER() OVER (ORDER BY g.gmeDate DESC) -
        ROW_NUMBER() OVER
            (PARTITION BY YEAR(g.gmeDate), CASE WHEN SUBSTRING(g.gmeScore, 2, 1) >
SUBSTRING(g.gmeScore, 4, 1) THEN 'W' ELSE 'L' END
            ORDER BY g.gmeDate DESC) AS GroupNum
    FROM
        [Moneyball.Game] g
)
GO
CREATE VIEW ConsecutiveLossGroups AS (
    SELECT
        YEAR(r.gmeDate) AS GameYear,
        r.GroupNum,
        COUNT(*) AS ConsecutiveGamesLost
    FROM
        RankedGames r
    WHERE
        CASE WHEN SUBSTRING(r.gmeScore, 2, 1) < SUBSTRING(r.gmeScore, 4, 1) THEN
'L' ELSE 'W' END = 'L'
    GROUP BY
        YEAR(r.gmeDate),
        r.GroupNum
)
GO
CREATE VIEW ConsecutiveWinGroups AS (
    SELECT
        YEAR(r.gmeDate) AS GameYear,
        r.GroupNum,
        COUNT(*) AS ConsecutiveGamesWon
    FROM
        RankedGames r
    WHERE
        CASE WHEN SUBSTRING(r.gmeScore, 2, 1) < SUBSTRING(r.gmeScore, 4, 1) THEN
'L' ELSE 'W' END = 'W'
    GROUP BY
        YEAR(r.gmeDate),
        r.GroupNum
)
GO
CREATE VIEW LongestLossStreaks AS (
    SELECT
        GameYear,
        MAX(ConsecutiveGamesLost) AS LongestLossStreak
```

```
    FROM
        ConsecutiveLossGroups
    GROUP BY
        GameYear
)
GO
CREATE VIEW LongestWinStreaks AS (
    SELECT
        GameYear,
        MAX(ConsecutiveGamesWon) AS LongestWinStreak
    FROM
        ConsecutiveWinGroups
    GROUP BY
        GameYear
)
```