Building a database of baseball historical games

Emmanuel Messori

09/09/2021

> We will work with data from Major League Baseball games compiled by Retrosheet, a non-profit organization that's gathered game statistics going back to the 1800s to today. The main file we will work from is game_log.csv, which has been compiled and pre-cleaned from 127 separate CSV files from Retrosheet. This file has hundreds of data points on each game. The goal of this to convert and normalize this data into several separate tables using SQL and create a robust database of game-level statistics.
>
> Dataquest

Reading the data

```
head(game_log)
```

```
## # A tibble: 6 x 161
##       date number_of_game day_of_week v_name v_league v_game_number h_name
##      <dbl>          <dbl> <chr>       <chr>  <chr>            <dbl> <chr>
## 1 18710504              0 Thu         CL1    <NA>                 1 FW1
## 2 18710505              0 Fri         BS1    <NA>                 1 WS3
## 3 18710506              0 Sat         CL1    <NA>                 2 RC1
## 4 18710508              0 Mon         CL1    <NA>                 3 CH1
## 5 18710509              0 Tue         BS1    <NA>                 2 TRO
## 6 18710511              0 Thu         CH1    <NA>                 2 CL1
## # ... with 154 more variables: h_league <chr>, h_game_number <dbl>,
## #   v_score <dbl>, h_score <dbl>, length_outs <dbl>, day_night <chr>,
## #   completion <lgl>, forfeit <lgl>, protest <chr>, park_id <chr>,
## #   attendance <dbl>, length_minutes <dbl>, v_line_score <chr>,
## #   h_line_score <chr>, v_at_bats <dbl>, v_hits <dbl>, v_doubles <dbl>,
## #   v_triples <dbl>, v_homeruns <dbl>, v_rbi <dbl>, v_sacrifice_hits <dbl>,
## #   v_sacrifice_flies <dbl>, v_hit_by_pitch <dbl>, v_walks <dbl>, ...
```

- `game_log` is the main file of our database. It contains information about 171907 baseball matche encoded in 161 variables : date, place, game statistics, players information.
- `park_codes` adds info pertaining the baseball grounds.
- `person_code` contains info about the players. It joins with the `game_log` filt through the `id` variable.
- `team_codes` contains information about the baseball teams.

All the fields in `game_log` are explained in the `game_log_fields.txt` file.

In the game log, each player has a defensive position listed in these columns:

- 106-132 Visiting starting players ID, name and defensive position,listed in the order (1-9) they appeared in the batting order.
- 133-159 Home starting players ID, name and defensive position listed in the order (1-9) they appeared in the batting order.

This [article] (http://probaseballinsider.com/baseball-instruction/baseball-basics/baseball-basics-positions/) gives us a list of names for each numbered position:

- Pitcher
- Catcher
- 1st Base
- 2nd Base
- 3rd Base
- Shortstop
- Left Field
- Center Field
- Right Field

This information is repeated in columns 72-77 for the home teams.

Wikipedia tells us there are currently two leagues - the American (AL) and National (NL).The fields 5 and 8 contain information about the historical leagues which existed at the time:

```
unique(game_log$h_league)
```

```
## [1] NA   "NL" "AA" "UA" "PL" "AL" "FL"
```

> All of the (candidate) major leagues in baseball have standardized two-letter abbreviations such as NA — namely, NA, NL, AA, UA, PL, AL, FL — whose crucial value is in this encyclopedic context.
>
> fandom

- **NL** National League
- **AL** American League
- **UA** Union Association
- **AA** American Association
- **PL** Player's League
- **FL** Federal League

Tables

```
library(DBI)
library(RSQLite)

con <- dbConnect(RSQLite::SQLite(), "mlb.db")


dbWriteTable(conn = con, name = "game_log", value = game_log,
             row.names = FALSE, header = TRUE)
```

```
dbWriteTable(conn = con, name = "park_codes", value = park_codes,
             row.names = FALSE, header = TRUE)

dbWriteTable(conn = con, name = "person_codes", value = person_codes,
             row.names = FALSE, header = TRUE)

dbWriteTable(conn = con, name = "team_codes", value = team_codes,
             row.names = FALSE, header = TRUE)
```

Since we do not have it yet, we will create a compound primary key for the game_log table using the `h_name`, `date` and `number` of game field.

```
new_c <- 'ALTER TABLE game_log
ADD COLUMN game_id TEXT;'

dbExecute(con, new_c)
```

```
## [1] 0
```

```
data<- 'UPDATE game_log SET game_id = h_name || date || number_of_game
WHERE game_id IS NULL'

dbExecute(con, data)
```

```
## [1] 171907
```

```
dbGetQuery(con, "SELECT COUNT(DISTINCT(game_id)) FROM game_log")
```

```
##   COUNT(DISTINCT(game_id))
## 1                   171907
```

Normalization

> Within a table, all of the columns should be related, or be an attribute, to the primary key. Any column that is not an attribute of the primary key is better placed in her own table. The primary key of our game log is `game_id`, and the players' names are not attributes of a game, but of the player ID. If the only data we had was the `game_log`, we would remove this column and create a new table that had the names of each player. As it happens, our `person_codes` table already has a list of our player IDs and names, so we can remove these without the need for creating a new table first.
>
> We want to also eliminate any redundant data that is available elsewhere. This second example can be found in the `park_codes` table.The start and end columns indicate the dates for the first and last games played at the park. This information can also be derived by looking at the park information for each game, so we might want to remove these columns from this table. The same observation can be applied to the `team_codes` table.
>
> Dataquest

- Some info contained in the `person_codes` and `team_codes` table can be probably deduced from `game_log`: the player's and career start and the eventual roles that he covered afterwards, the teams debut and final matches .

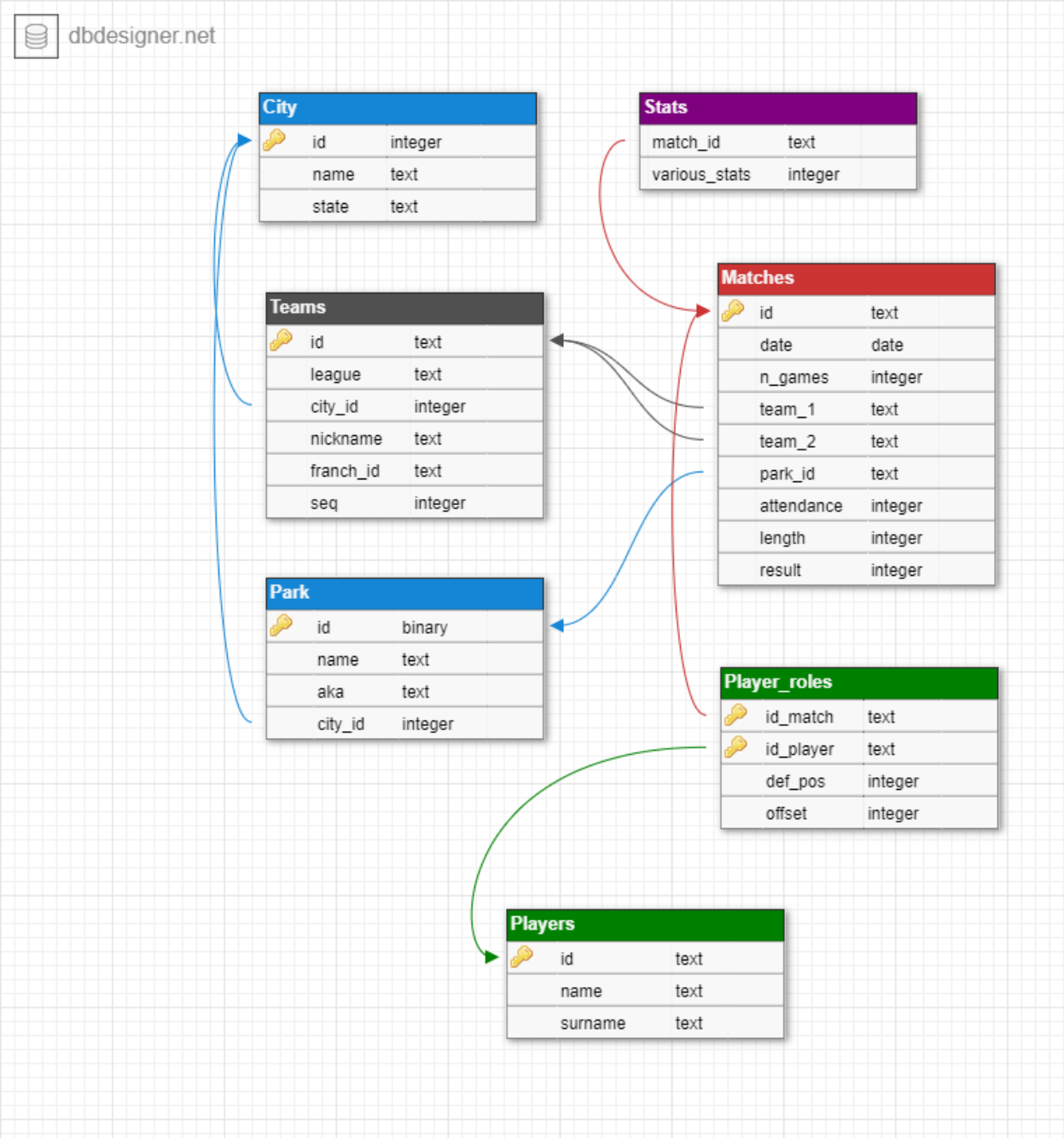

Figure 1: starting schema

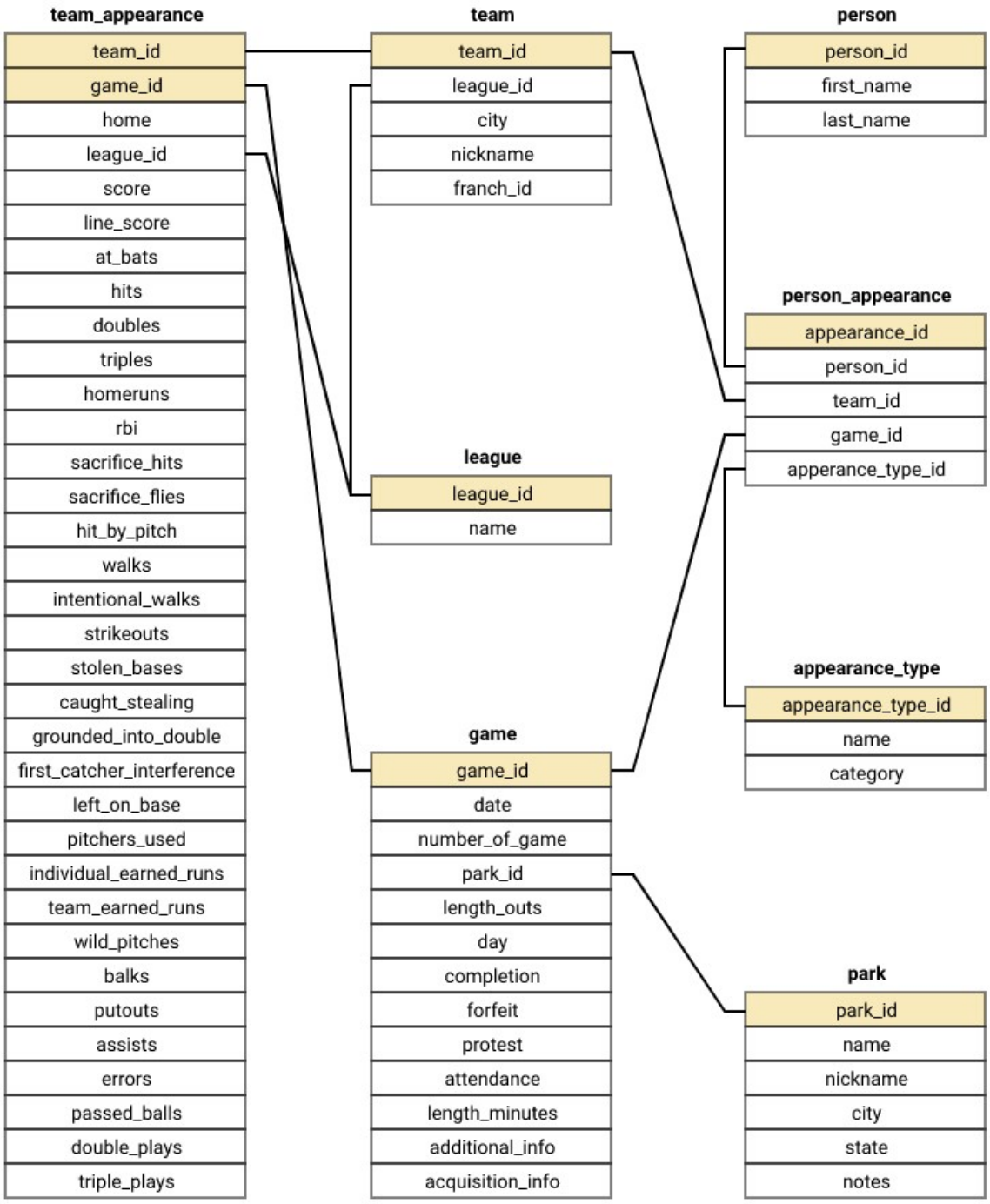

Figure 2: final schema

Database Creation

```
# person table

person <- "CREATE TABLE person (
           person_id TEXT PRIMARY KEY,
           first_name TEXT,
           last_name TEXT);"

insert_person <- "INSERT INTO person
                  SELECT id, first, last
                  FROM person_codes"

dbExecute(con, person)
```

```
## [1] 0
```

```
dbExecute(con, insert_person)
```

```
## [1] 20494
```

```
dbGetQuery(con,"SELECT * FROM person
           LIMIT 10")
```

```
##    person_id first_name   last_name
## 1   aardd001      David     Aardsma
## 2   aaroh101       Hank       Aaron
## 3   aarot101     Tommie       Aaron
## 4   aased001        Don        Aase
## 5   abada001       Andy        Abad
## 6   abadf001   Fernando        Abad
## 7   abadj101       John      Abadie
## 8   abbae101         Ed Abbaticchio
## 9   abbeb101       Bert       Abbey
## 10  abbec101    Charlie       Abbey
```

```
#park table

dbExecute(con,'DROP TABLE IF EXISTS park')
```

```
## [1] 0
```

```
park <- 'CREATE TABLE park(
         park_id TEXT PRIMARY KEY,
         name TEXT,
         nickname TEXT,
         city TEXT,
         state TEXT,
         notes TEXT
);'
```

```
insert_park <- 'INSERT INTO park
                SELECT park_id, name, aka, city, state, notes
                FROM park_codes;'

dbExecute(con, park)
```

```
## [1] 0
```

```
dbExecute(con, insert_park)
```

```
## [1] 252
```

```
dbGetQuery(con, 'SELECT * FROM park LIMIT 10')
```

```
##    park_id                           name
## 1    ALB01                 Riverside Park
## 2    ALT01                  Columbia Park
## 3    ANA01       Angel Stadium of Anaheim
## 4    ARL01              Arlington Stadium
## 5    ARL02 Rangers Ballpark in Arlington
## 6    ATL01 Atlanta-Fulton County Stadium
## 7    ATL02                   Turner Field
## 8    ATL03                  Suntrust Park
## 9    BAL01         Madison Avenue Grounds
## 10   BAL02                 Newington Park
##                                     nickname      city state
## 1                                       <NA>    Albany    NY
## 2                                       <NA>   Altoona    PA
## 3              Edison Field; Anaheim Stadium   Anaheim    CA
## 4                                       <NA> Arlington    TX
## 5  The Ballpark in Arlington; Ameriquest Fl Arlington    TX
## 6                                       <NA>   Atlanta    GA
## 7                                       <NA>   Atlanta    GA
## 8                                       <NA>   Atlanta    GA
## 9                                       <NA> Baltimore    MD
## 10                                      <NA> Baltimore    MD
##                                            notes
## 1  TRN:9/11/80;6/15&9/10/1881;5/16-5/18&5/30/1882
## 2                                           <NA>
## 3                                           <NA>
## 4                                           <NA>
## 5                                           <NA>
## 6                                           <NA>
## 7                                           <NA>
## 8                                           <NA>
## 9                                            WS3
## 10               BL1:1872-74; BL4:1873; BL2: 1882
```

```
#league table

league <- 'CREATE TABLE league (
```

```
          league_id TEXT PRIMARY KEY,
          name TEXT);'

insert_league <- 'INSERT INTO league
                  SELECT DISTINCT(league) as league_id,
                         CASE league
                         WHEN "NL" THEN "National League"
                         WHEN "AL" THEN "American League"
                         WHEN "UA" THEN "Union Association"
                         WHEN "AA" THEN "American Association"
                         WHEN "PL" THEN  "Player\'s League"
                         WHEN "FL" THEN "Federal League"
                         ELSE NULL
                         END as name
                         FROM team_codes'

dbExecute(con, league)
```

```
## [1] 0
```

```
dbExecute(con, insert_league)
```

```
## [1] 7
```

```
dbGetQuery(con, 'SELECT * FROM league')
```

```
##   league_id                 name
## 1        UA    Union Association
## 2        NL      National League
## 3        PL      Player's League
## 4      <NA>                 <NA>
## 5        AA American Association
## 6        AL      American League
## 7        FL       Federal League
```

```
#appearance type

dbWriteTable(con, name = "appearance_type", value = "baseball_data/appearance_type.csv",
             row.names = FALSE, header = TRUE )

dbGetQuery(con, 'SELECT * FROM appearance_type')
```

```
##    appearance_type_id              name category
## 1                  01          Batter 1  offense
## 2                  02          Batter 2  offense
## 3                  03          Batter 3  offense
## 4                  04          Batter 4  offense
## 5                  05          Batter 5  offense
## 6                  06          Batter 6  offense
## 7                  07          Batter 7  offense
## 8                  08          Batter 8  offense
```

```
## 9                  O9           Batter 9  offense
## 10                 D1            Pitcher  defense
## 11                 D2            Catcher  defense
## 12                 D3           1st Base  defense
## 13                 D4           2nd Base  defense
## 14                 D5           3rd Base  defense
## 15                 D6          Shortstop  defense
## 16                 D7         Left Field  defense
## 17                 D8       Center Field  defense
## 18                 D9        Right Field  defense
## 19                D10   Unknown Position  defense
## 20                UHP         Home Plate   umpire
## 21                U1B         First Base   umpire
## 22                U2B        Second Base   umpire
## 23                U3B         Third Base   umpire
## 24                ULF         Left Field   umpire
## 25                URF        Right Field   umpire
## 26                 MM            Manager  manager
## 27                AWP    Winning Pitcher    award
## 28                ALP     Losing Pitcher    award
## 29                ASP     Saving Pitcher    award
## 30                AWB Winning RBI Batter    award
## 31                PSP   Starting Pitcher  pitcher
```

Game and team tables

Here are some notes on the normalization choices made with each of these tables:

Team

The start, end, and sequence columns can be derived from the game level data.

Game

> We have chosen to include all columns for the game log that don't refer to one specific team or player, instead putting those in two appearance tables. We have removed the column with the day of the week, as this can be derived from the date. We have changed the day_night column to day, with the intention of making this a boolean column. Even though SQLite doesn't support the BOOLEAN type, we can use this when creating our table and SQLite will manage the underlying types behind the scenes (for more on how this works refer to the SQLite documentation. This means that anyone quering the schema of our database in the future understands how that column is intended to be used.
>
> Dataquest

```
#team table
dbExecute(con, 'DROP TABLE IF EXISTS team')
```

```
## [1] 0
```

```
team <- 'CREATE TABLE team (
        team_id TEXT PRIMARY KEY,
        league_id TEXT,
        city TEXT,
        nickname TEXT,
        franch_id TEXT,
        FOREIGN KEY(league_id) REFERENCES league(league_id));'

insert_team <- 'INSERT INTO team
                SELECT team_id,
                      league,
                      city,
                      nickname,
                      franch_id
                      FROM team_codes'

dbExecute(con, team)
```

```
## [1] 0
```

```
dbExecute(con, insert_team)
```

```
## [1] 150
```

```
dbGetQuery(con, 'SELECT * FROM team LIMIT 10')
```

```
##    team_id league_id      city        nickname franch_id
## 1      ALT        UA   Altoona Mountain Cities       ALT
## 2      ARI        NL   Arizona    Diamondbacks       ARI
## 3      BFN        NL   Buffalo          Bisons       BFN
## 4      BFP        PL   Buffalo          Bisons       BFP
## 5      BL1      <NA> Baltimore        Canaries       BL1
## 6      BL2        AA Baltimore         Orioles       BL2
## 7      BLN        NL Baltimore         Orioles       BL2
## 8      BL4      <NA> Baltimore       Marylands       BL4
## 9      BLA        AL Baltimore         Orioles       BLA
## 10     NYA        AL  New York         Yankees       BLA
```

```
# Game table
dbExecute(con, 'DROP TABLE IF EXISTS game')
```

```
## [1] 0
```

```
game <- 'CREATE TABLE game(
          game_id TEXT PRIMARY KEY,
          date TEXT,
          number_of_game INTEGER,
          park_id TEXT,
          length_outs INTEGER,
          day BOOLEAN,
```

```
          completion TEXT,
          forfeit TEXT,
          attendance INTEGER,
          length_minutes INTEGER,
          additional_info TEXT,
          acquisition_info TEXT,
          FOREIGN KEY(park_id) REFERENCES park(park_id)

)'

insert_game <- 'INSERT INTO GAME
                SELECT game_id,
                date,
                number_of_game,
                park_id,
                length_outs,
                day_night,
                completion,
                forfeit,
                attendance,
                length_minutes,
                additional_info,
                acquisition_info
                FROM game_log
;'

dbExecute(con, game)
```

```
## [1] 0
```

```
dbExecute(con, insert_game)
```

```
## [1] 171907
```

```
dbGetQuery(con, 'SELECT * FROM game LIMIT 10')
```

```
##             game_id       date number_of_game park_id length_outs day
## 1  FW118710504.00.0 18710504.0              0   FOR01          54   D
## 2  WS318710505.00.0 18710505.0              0   WAS01          54   D
## 3  RC118710506.00.0 18710506.0              0   RCK01          54   D
## 4  CH118710508.00.0 18710508.0              0   CHI01          54   D
## 5  TRO18710509.00.0 18710509.0              0   TRO01          54   D
## 6  CL118710511.00.0 18710511.0              0   CLE01          48   D
## 7  CL118710513.00.0 18710513.0              0   CIN01          54   D
## 8  FW118710513.00.0 18710513.0              0   FOR01          54   D
## 9  FW118710515.00.0 18710515.0              0   FOR01          54   D
## 10 BS118710516.00.0 18710516.0              0   BOS01          54   D
##    completion forfeit attendance length_minutes additional_info
## 1        <NA>    <NA>        200            120            <NA>
## 2        <NA>    <NA>       5000            145            HTBF
## 3        <NA>    <NA>       1000            140            <NA>
## 4        <NA>    <NA>       5000            150            <NA>
```

```
## 5        <NA>    <NA>       3250            145            HTBF
## 6        <NA>    <NA>       2500            120            <NA>
## 7        <NA>    <NA>       1200            150            <NA>
## 8        <NA>    <NA>       1500            105            <NA>
## 9        <NA>    <NA>         NA            140            <NA>
## 10       <NA>    <NA>       2500             NA            HTBF
##    acquisition_info
## 1                 Y
## 2                 Y
## 3                 Y
## 4                 Y
## 5                 Y
## 6                 Y
## 7                 Y
## 8                 Y
## 9                 Y
## 10                Y
```

From here onwards I follow the solution file.

```
#team_appearance table

ta <- 'CREATE TABLE IF NOT EXISTS team_appearance (
      team_id TEXT,
      game_id TEXT,
      home BOOLEAN,
      league_id TEXT,
      score INTEGER,
      line_score TEXT,
      at_bats INTEGER,
      hits INTEGER,
      doubles INTEGER,
      triples INTEGER,
      homeruns INTEGER,
      rbi INTEGER,
      sacrifice_hits INTEGER,
      sacrifice_flies INTEGER,
      hit_by_pitch INTEGER,
      walks INTEGER,
      intentional_walks INTEGER,
      strikeouts INTEGER,
      stolen_bases INTEGER,
      caught_stealing INTEGER,
      grounded_into_double INTEGER,
      first_catcher_interference INTEGER,
      left_on_base INTEGER,
      pitchers_used INTEGER,
      individual_earned_runs INTEGER,
      team_earned_runs INTEGER,
      wild_pitches INTEGER,
      balks INTEGER,
      putouts INTEGER,
      assists INTEGER,
      errors INTEGER,
```

```
    passed_balls INTEGER,
    double_plays INTEGER,
    triple_plays INTEGER,
    PRIMARY KEY (team_id, game_id),
    FOREIGN KEY (team_id) REFERENCES team(team_id),
    FOREIGN KEY (game_id) REFERENCES game(game_id),
    FOREIGN KEY (league_id) REFERENCES league(league_id)
  );'

dbExecute(con, ta)
```

```
## [1] 0
```

```
insert_to_team_appearance <- "
  INSERT OR IGNORE INTO team_appearance
      SELECT
          h_name,
          game_id,
          1 AS home,
          h_league,
          h_score,
          h_line_score,
          h_at_bats,
          h_hits,
          h_doubles,
          h_triples,
          h_homeruns,
          h_rbi,
          h_sacrifice_hits,
          h_sacrifice_flies,
          h_hit_by_pitch,
          h_walks,
          h_intentional_walks,
          h_strikeouts,
          h_stolen_bases,
          h_caught_stealing,
          h_grounded_into_double,
          h_first_catcher_interference,
          h_left_on_base,
          h_pitchers_used,
          h_individual_earned_runs,
          h_team_earned_runs,
          h_wild_pitches,
          h_balks,
          h_putouts,
          h_assists,
          h_errors,
          h_passed_balls,
          h_double_plays,
          h_triple_plays
      FROM game_log

  UNION
```

```
      SELECT
          v_name,
          game_id,
          0 AS home,
          v_league,
          v_score,
          v_line_score,
          v_at_bats,
          v_hits,
          v_doubles,
          v_triples,
          v_homeruns,
          v_rbi,
          v_sacrifice_hits,
          v_sacrifice_flies,
          v_hit_by_pitch,
          v_walks,
          v_intentional_walks,
          v_strikeouts,
          v_stolen_bases,
          v_caught_stealing,
          v_grounded_into_double,
          v_first_catcher_interference,
          v_left_on_base,
          v_pitchers_used,
          v_individual_earned_runs,
          v_team_earned_runs,
          v_wild_pitches,
          v_balks,
          v_putouts,
          v_assists,
          v_errors,
          v_passed_balls,
          v_double_plays,
          v_triple_plays
      from game_log;
"

dbExecute(con, insert_to_team_appearance)
```

```
## [1] 343814
```

```
check <- "
  SELECT * FROM team_appearance
 LIMIT 10;
"

dbGetQuery(con, check)
```

```
##    team_id          game_id home league_id score line_score at_bats hits
## 1      ALT ALT18840430.00.0    1        UA     2       <NA>      NA   NA
## 2      ALT ALT18840502.00.0    1        UA     3       <NA>      NA   NA
```

```
## 3      ALT ALT18840503.00.0    1        UA     5       <NA>      NA   NA
## 4      ALT ALT18840505.00.0    1        UA     2       <NA>      NA   NA
## 5      ALT ALT18840510.00.0    1        UA     9       <NA>      NA   NA
## 6      ALT ALT18840512.00.0    1        UA     3       <NA>      NA   NA
## 7      ALT ALT18840514.00.0    1        UA     2       <NA>      NA   NA
## 8      ALT ALT18840515.00.0    1        UA     7       <NA>      NA   NA
## 9      ALT ALT18840516.00.0    1        UA     6       <NA>      NA   NA
## 10     ALT ALT18840517.00.0    1        UA     8       <NA>      NA   NA
##    doubles triples homeruns rbi sacrifice_hits sacrifice_flies hit_by_pitch
## 1       NA      NA       NA  NA             NA              NA           NA
## 2       NA      NA       NA  NA             NA              NA           NA
## 3       NA      NA       NA  NA             NA              NA           NA
## 4       NA      NA       NA  NA             NA              NA           NA
## 5       NA      NA       NA  NA             NA              NA           NA
## 6       NA      NA       NA  NA             NA              NA           NA
## 7       NA      NA       NA  NA             NA              NA           NA
## 8       NA      NA       NA  NA             NA              NA           NA
## 9       NA      NA       NA  NA             NA              NA           NA
## 10      NA      NA       NA  NA             NA              NA           NA
##    walks intentional_walks strikeouts stolen_bases caught_stealing
## 1     NA                NA         NA           NA              NA
## 2     NA                NA         NA           NA              NA
## 3     NA                NA         NA           NA              NA
## 4     NA                NA         NA           NA              NA
## 5     NA                NA         NA           NA              NA
## 6     NA                NA         NA           NA              NA
## 7     NA                NA         NA           NA              NA
## 8     NA                NA         NA           NA              NA
## 9     NA                NA         NA           NA              NA
## 10    NA                NA         NA           NA              NA
##    grounded_into_double first_catcher_interference left_on_base pitchers_used
## 1                    NA                          NA           NA            NA
## 2                    NA                          NA           NA            NA
## 3                    NA                          NA           NA            NA
## 4                    NA                          NA           NA            NA
## 5                    NA                          NA           NA            NA
## 6                    NA                          NA           NA            NA
## 7                    NA                          NA           NA            NA
## 8                    NA                          NA           NA            NA
## 9                    NA                          NA           NA            NA
## 10                   NA                          NA           NA            NA
##    individual_earned_runs team_earned_runs wild_pitches balks putouts assists
## 1                      NA               NA           NA    NA      NA      NA
## 2                      NA               NA           NA    NA      NA      NA
## 3                      NA               NA           NA    NA      NA      NA
## 4                      NA               NA           NA    NA      NA      NA
## 5                      NA               NA           NA    NA      NA      NA
## 6                      NA               NA           NA    NA      NA      NA
## 7                      NA               NA           NA    NA      NA      NA
## 8                      NA               NA           NA    NA      NA      NA
## 9                      NA               NA           NA    NA      NA      NA
## 10                     NA               NA           NA    NA      NA      NA
##    errors passed_balls double_plays triple_plays
## 1      NA           NA           NA           NA
```

```
## 2      NA           NA           NA           NA
## 3      NA           NA           NA           NA
## 4      NA           NA           NA           NA
## 5      NA           NA           NA           NA
## 6      NA           NA           NA           NA
## 7      NA           NA           NA           NA
## 8      NA           NA           NA           NA
## 9      NA           NA           NA           NA
## 10     NA           NA           NA           NA
```

The person_appearance table

The final table we have to create is the person_appearance table. It will be used to store information on appearances in games by managers, players, and umpires as detailed in the appearance_type table.

```
# Adding the Person Appearance Table

drop_person_appearance_precaution <- "DROP TABLE IF EXISTS person_appearance"
dbExecute(con, drop_person_appearance_precaution)
```

```
## [1] 0
```

```
create_person_appearance_command <- "
  CREATE TABLE person_appearance (
      appearance_id INTEGER PRIMARY KEY,
      person_id TEXT,
      team_id TEXT,
      game_id TEXT,
      appearance_type_id,
      FOREIGN KEY (person_id) REFERENCES person(person_id),
      FOREIGN KEY (team_id) REFERENCES team(team_id),
      FOREIGN KEY (game_id) REFERENCES game(game_id),
      FOREIGN KEY (appearance_type_id) REFERENCES appearance_type(appearance_type_id)
  );
"
dbExecute(con, create_person_appearance_command)
```

```
## [1] 0
```

```
insert_to_person_appearance <- '
  INSERT OR IGNORE INTO person_appearance (
      game_id,
      team_id,
      person_id,
      appearance_type_id
  )
      SELECT
          game_id,
          NULL,
          hp_umpire_id,
          "UHP"
      FROM game_log
```

```
    WHERE hp_umpire_id IS NOT NULL

UNION

    SELECT
        game_id,
        NULL,
        [1b_umpire_id],
        "U1B"
    FROM game_log
    WHERE "1b_umpire_id" IS NOT NULL

UNION

    SELECT
        game_id,
        NULL,
        [2b_umpire_id],
        "U2B"
    FROM game_log
    WHERE [2b_umpire_id] IS NOT NULL

UNION

    SELECT
        game_id,
        NULL,
        [3b_umpire_id],
        "U3B"
    FROM game_log
    WHERE [3b_umpire_id] IS NOT NULL

UNION

    SELECT
        game_id,
        NULL,
        lf_umpire_id,
        "ULF"
    FROM game_log
    WHERE lf_umpire_id IS NOT NULL

UNION

    SELECT
        game_id,
        NULL,
        rf_umpire_id,
        "URF"
    FROM game_log
    WHERE rf_umpire_id IS NOT NULL

UNION
```

```
    SELECT
        game_id,
        v_name,
        v_manager_id,
        "MM"
    FROM game_log
    WHERE v_manager_id IS NOT NULL

UNION

    SELECT
        game_id,
        h_name,
        h_manager_id,
        "MM"
    FROM game_log
    WHERE h_manager_id IS NOT NULL

UNION

    SELECT
        game_id,
        CASE
            WHEN h_score > v_score THEN h_name
            ELSE v_name
            END,
        winning_pitcher_id,
        "AWP"
    FROM game_log
    WHERE winning_pitcher_id IS NOT NULL

UNION

    SELECT
        game_id,
        CASE
            WHEN h_score < v_score THEN h_name
            ELSE v_name
            END,
        losing_pitcher_id,
        "ALP"
    FROM game_log
    WHERE losing_pitcher_id IS NOT NULL

UNION

    SELECT
        game_id,
        CASE
            WHEN h_score > v_score THEN h_name
            ELSE v_name
            END,
        saving_pitcher_id,
```

```
        "ASP"
    FROM game_log
    WHERE saving_pitcher_id IS NOT NULL

  UNION

    SELECT
        game_id,
        CASE
            WHEN h_score > v_score THEN h_name
            ELSE v_name
            END,
        winning_rbi_batter_id,
        "AWB"
    FROM game_log
    WHERE winning_rbi_batter_id IS NOT NULL

  UNION

    SELECT
        game_id,
        v_name,
        v_starting_pitcher_id,
        "PSP"
    FROM game_log
    WHERE v_starting_pitcher_id IS NOT NULL

  UNION

    SELECT
        game_id,
        h_name,
        h_starting_pitcher_id,
        "PSP"
    FROM game_log
    WHERE h_starting_pitcher_id IS NOT NULL;
'
dbExecute(con, insert_to_person_appearance)
```

```
## [1] 1646109
```

```
for (letter in c("h", "v")) {
  for (num in 1:9) {
    template <- '
      INSERT INTO person_appearance (
          game_id,
          team_id,
          person_id,
          appearance_type_id
      )
          SELECT
              game_id,
              %s_name,
```

```
            %s_player_%f_id,
            "O%f"
        FROM game_log
        WHERE %s_player_%f_id IS NOT NULL

    UNION

        SELECT
            game_id,
            %s_name,
            %s_player_%f_id,
            "D" || CAST(%s_player_%f_def_pos AS INT)
        FROM game_log
        WHERE %s_player_%f_id IS NOT NULL;
    '
    # replace all of the %s and %f with the correct letter number
    template <- gsub("%s", letter, template, fixed = TRUE)
    template <- gsub("%f", num, template, fixed = TRUE)

    dbExecute(con, template)
  }
}
```

```
dbListTables(con)
```

```
##  [1] "appearance_type"   "game"              "game_log"
##  [4] "league"            "park"              "park_codes"
##  [7] "person"            "person_appearance" "person_codes"
## [10] "team"              "team_appearance"   "team_codes"
```

```
dbGetQuery(con, 'SELECT * FROM person_appearance LIMIT 10')
```

```
##    appearance_id person_id team_id          game_id appearance_type_id
## 1              1  maplb901    <NA> ALT18840430.00.0                UHP
## 2              2  curte801     ALT ALT18840430.00.0                 MM
## 3              3  murpj104     ALT ALT18840430.00.0                PSP
## 4              4  hodnc101     SLU ALT18840430.00.0                PSP
## 5              5  sullt101     SLU ALT18840430.00.0                 MM
## 6              6  hoopm101    <NA> ALT18840502.00.0                UHP
## 7              7  curte801     ALT ALT18840502.00.0                 MM
## 8              8  learj102     ALT ALT18840502.00.0                PSP
## 9              9  sullt101     SLU ALT18840502.00.0                 MM
## 10            10  taylb103     SLU ALT18840502.00.0                PSP
```

Dropping the starting tables

```
drops <- c('DROP TABLE game_log','DROP TABLE person_codes','DROP TABLE park_codes', 'DROP TABLE team_co

walk(drops, dbExecute, conn=con)

dbListTables(con)
```

```
## [1] "appearance_type"   "game"              "league"
## [4] "park"              "person"            "person_appearance"
## [7] "team"              "team_appearance"
```