```sql
-- AGGREGATED DENORMALIZED TABLES FOR PLAYERS RETIREMENT STATISTICS IN TABLEAU
CREATE OR REPLACE TEMPORARY TABLE TEMP_PLA_GAME_YEAR AS
-- join of the select below with the TOURNAMENT table for collums YEAR OF TOURNAMENT and GAME_ID
SELECT
       TOURNAMENT_ID::int AS TOURNAMENT_ID,
       PLAYER_ID::int AS PLAYER_ID,
       SUM(SUM_PRIZE::float) AS SUM_PRIZE,
       YEAR(T_END_DATE)::int as YEAR_OF_TOURNAMENT,
       T_GAME_ID::int AS GAME_ID
FROM
    (
    -- union of TOURNAMENT_RESULTS_INDIVIDUAL and TOURNAMENT_RESULTS_PLAYER_IN_TEAM tables
    SELECT
        TRI_PLAYER_ID::int AS PLAYER_ID,
        SUM(TRI_PRIZE_USD)::float AS SUM_PRIZE,
        TRI_TOURNAMENT_ID::int AS TOURNAMENT_ID,
        NULL AS TOURNAMENT_TEAM_ID
    FROM TOURNAMENT_RESULTS_INDIVIDUAL
    WHERE TRI_PLAYER_ID < 900000 -- fake IDs are bigger than 900 000
    GROUP BY TRI_PLAYER_ID,TRI_TOURNAMENT_ID, TOURNAMENT_TEAM_ID

    UNION ALL

    SELECT
        TRP_PLAYER_ID::int AS PLAYER_ID,
        SUM(TRP_PRIZE_USD_FOR_PLAYER)::float AS SUM_PRIZE,
        TRP_TOURNAMENT_ID::int AS TOURNAMENT_ID,
        TRP_TOURNAMENT_TEAM_ID AS TOURNAMENT_TEAM_ID
    FROM TOURNAMENT_RESULTS_PLAYER_IN_TEAM
    GROUP BY PLAYER_ID, TOURNAMENT_ID, TOURNAMENT_TEAM_ID
    ) AS AAA
JOIN TOURNAMENT ON AAA.TOURNAMENT_ID=T_TOURNAMENT_ID
GROUP BY TOURNAMENT_ID, PLAYER_ID, YEAR_OF_TOURNAMENT, GAME_ID;
```

```sql
-- WHAT IS THE AVERAGE, MEDIAN AND QUANTILE VALUE FOR A GAME?
CREATE OR REPLACE TEMPORARY TABLE TEMP_AVG_MED_KVA_PERGAME AS
WITH cte AS (
       SELECT PLAYER_ID, GAME_ID, SUM(SUM_PRIZE) as SUM2_PRIZE
       FROM TEMP_PLA_GAME_YEAR
       GROUP BY PLAYER_ID, GAME_ID)

SELECT
       GAME_ID,
       AVG(SUM2_PRIZE) as AVG_PRIZE_FOR_GAME,
       MEDIAN(SUM2_PRIZE) as MED_PRIZE_FOR_GAME,
       PERCENTILE_CONT( 0.9 ) WITHIN GROUP (ORDER BY SUM2_PRIZE) as KVA09_PRIZE_FOR_GAME
FROM cte
GROUP BY GAME_ID
ORDER BY GAME_ID;


-- Merge TABLE with TEMP_PLA_GAME_YEAR and TEMP_AVG_MED_KVA_PERGAME
-- add columns if PRIZE >= AVG, MED, KVA PER GAME

CREATE OR REPLACE TEMPORARY TABLE TEMP_PLAYER_GAME_AVG_MED_KVA AS
WITH CTE_UNIK AS (
       SELECT pp.GAME_ID, pp.PLAYER_ID, SUM(pp.SUM_PRIZE) AS SUM2_PRIZE
       FROM TEMP_PLA_GAME_YEAR pp
       GROUP BY pp.GAME_ID, pp.PLAYER_ID)

SELECT c.GAME_ID, PLAYER_ID, SUM2_PRIZE,
    CASE WHEN SUM2_PRIZE >= KVA09_PRIZE_FOR_GAME THEN 1
        ELSE 0
        END as ISMORETHAN_KVA,
    CASE WHEN SUM2_PRIZE >= AVG_PRIZE_FOR_GAME THEN 1
        ELSE 0
        END as ISMORETHAN_AVG,
    CASE WHEN SUM2_PRIZE >= MED_PRIZE_FOR_GAME THEN 1
        ELSE 0
```

```
        END as ISMORETHAN_MED
FROM CTE_UNIK c
JOIN TEMP_AVG_MED_KVA_PERGAME a ON a.GAME_ID = c.GAME_ID
WHERE SUM2_PRIZE is not null
ORDER BY PLAYER_ID, c.GAME_ID, SUM2_PRIZE DESC;

-- the TEMP_PLAYER_GAME_AVG_MED_KVA table shows that MEDIAN is not suitable for our purposes because it divides the players ca into
two halves (48.6% and 51.4%).
-- The mean and quantile are relatively similar, the quantile is more in line with our idea, so we only work with the quantile below
-- The average divides the players into 83.6% and 16.7%
-- The quantile divides the players into 89.6% and 10.4%
-- Percentages should be viewed with the understanding that this is a rough preview that does not take into account fact,
-- that the players are always in the source table broken down by a game.

-- QUANTILE
-- SELECT COUNT(DISTINCT PLAYER_ID)
-- FROM TEMP_PLAYER_GAME_AVG_MED_KVA
-- WHERE ISMORETHAN_KVA = 1; -- 7 149 = 10,4 %
-- SELECT COUNT(DISTINCT PLAYER_ID)
-- FROM TEMP_PLAYER_GAME_AVG_MED_KVA
-- WHERE ISMORETHAN_KVA = 0; -- 61 849 = 89,6 %

-- AVERAGE
-- SELECT COUNT(DISTINCT PLAYER_ID)
-- FROM TEMP_PLAYER_GAME_AVG_MED_KVA
-- WHERE ISMORETHAN_AVG = 1; -- 11 396 = 16,7 %
-- SELECT COUNT(DISTINCT PLAYER_ID)
-- FROM TEMP_PLAYER_GAME_AVG_MED_KVA
-- WHERE ISMORETHAN_AVG = 0; -- 58 220 = 83,6 %

-- MEDIAN
-- SELECT COUNT(DISTINCT PLAYER_ID)
-- FROM TEMP_PLAYER_GAME_AVG_MED_KVA
-- WHERE ISMORETHAN_MED = 1; -- 34 057 = 48,6 %
-- SELECT COUNT(DISTINCT PLAYER_ID)
```

```sql
-- FROM TEMP_PLAYER_GAME_AVG_MED_KVA
-- WHERE ISMORETHAN_MED = 0; -- 36 029 = 51,4 %
-- creating final tables

CREATE OR REPLACE TEMPORARY TABLE TEMP_WHOSEARLYERRETIRED_KVA AS

-- counting Players and Average count of years they played and summing Prizes for each game, adding Game names
SELECT
       x.GAME_ID::int AS W_GAME_ID,
       G_GAME_NAME AS W_GAME_NAME,
       x.ISMORETHAN_KVA::int AS W_ISMORETHAN_KVA,
       COUNT(DISTINCT x.PLAYER_ID)::int as W_CTN_PLAYERS,
       AVG(x.CNT_YEARS)::float as W_AVG_CNT_YEARS,
       SUM(x.SUM3_PRIZE)::float as W_SUM_PRIZE_USD
FROM
    -- adding Prize and Quantile by joining table TEMP_PLAYER_GAME_AVG_MED_KVA
    (
    SELECT
         tityp.GAME_ID,
         tityp.PLAYER_ID,
         tityp.CNT_YEARS,
         SUM(tg.SUM2_PRIZE) AS SUM3_PRIZE,
         ISMORETHAN_KVA
    FROM
        -- counting years of tournaments and grouping other collums
        (
        SELECT
            GAME_ID,
            PLAYER_ID,
            COUNT(DISTINCT YEAR_OF_TOURNAMENT) AS CNT_YEARS
        FROM
            -- adding year of tournament from Tournament table
            (
            SELECT GAME_ID, PLAYER_ID, YEAR(T_END_DATE) AS YEAR_OF_TOURNAMENT
            FROM
```

```
            -- core data: game, player, tournament ID is obtained by merging two tables
            (
            SELECT
                GAME_ID,
                PLAYER_ID,
                TRI_TOURNAMENT_ID AS TOURNAMENT_ID
            FROM TEMP_PLAYER_GAME_AVG_MED_KVA
            LEFT JOIN TOURNAMENT_RESULTS_INDIVIDUAL ti ON PLAYER_ID=TRI_PLAYER_ID
            GROUP BY GAME_ID, PLAYER_ID, TRI_TOURNAMENT_ID
            UNION ALL
            SELECT
                GAME_ID,
                PLAYER_ID,
                TRP_TOURNAMENT_ID AS TOURNAMENT_ID
            FROM TEMP_PLAYER_GAME_AVG_MED_KVA
            LEFT JOIN TOURNAMENT_RESULTS_PLAYER_IN_TEAM tt ON PLAYER_ID=TRP_PLAYER_ID
            GROUP BY GAME_ID, PLAYER_ID, TRP_TOURNAMENT_ID
            ORDER BY GAME_ID, PLAYER_ID
           ) tit
       LEFT JOIN TOURNAMENT t ON TOURNAMENT_ID=T_TOURNAMENT_ID
        ) tity
    GROUP BY GAME_ID, PLAYER_ID
    ) tityp
  JOIN TEMP_PLAYER_GAME_AVG_MED_KVA tg ON tg.GAME_ID=tityp.GAME_ID AND tg.PLAYER_ID=tityp.PLAYER_ID
  GROUP BY tityp.GAME_ID, tityp.PLAYER_ID, tityp.CNT_YEARS, ISMORETHAN_KVA
  -- ORDER BY tityp.PLAYER_ID,tityp.GAME_ID
) x
JOIN GAME g ON G_GAME_ID=x.GAME_ID
GROUP BY x.GAME_ID, G_GAME_NAME, x.ISMORETHAN_KVA
ORDER BY x.GAME_ID, G_GAME_NAME, x.ISMORETHAN_KVA DESC;
```

```sql
-- creating final table, where sample of players for a game is more than 100 players (Because smaller sample is not statisticaly
representative for us)
CREATE OR REPLACE TABLE WHOSEARLYERRETIRED_KVA AS
SELECT
       W_GAME_ID,
       W_GAME_NAME,
       W_ISMORETHAN_KVA,
       W_CTN_PLAYERS,
       W_AVG_CNT_YEARS,
       W_SUM_PRIZE_USD
FROM TEMP_WHOSEARLYERRETIRED_KVA
WHERE W_GAME_ID IN
        (SELECT DISTINCT W_GAME_ID FROM
            (
            SELECT W_GAME_ID, SUM(W_CTN_PLAYERS) AS SUM_PLAYERS
            FROM TEMP_WHOSEARLYERRETIRED_KVA
            GROUP BY W_GAME_ID
            HAVING SUM_PLAYERS >=100
            )
        );

-- it is necessary (for the desired visualizations in Tableau) to transform the table so that each column is split into two, depending
on whether W_ISMORETHAN_KVA = 1 or 0.
CREATE OR REPLACE TABLE WHOSEARLYERRETIRED_KVA_ROWS AS
SELECT
    W_GAME_ID,
    W_GAME_NAME,
    SUM(IFNULL(CTN_PLAYERS_0,0)) AS CTN_PLAYERS_0,
    SUM(IFNULL(CTN_PLAYERS_1,0)) AS CTN_PLAYERS_1,
    SUM(IFNULL(SUM_PRIZE_USD_1,0)) AS SUM_PRIZE_USD_1,
    SUM(IFNULL(W_AVG_CNT_YEARS_0,0)) AS W_AVG_CNT_YEARS_0,
    SUM(IFNULL(W_AVG_CNT_YEARS_1,0)) AS W_AVG_CNT_YEARS_1
FROM
    (SELECT
        W_GAME_ID,
```

```
        W_GAME_NAME,
        0 AS CTN_PLAYERS_0,
        W_CTN_PLAYERS AS CTN_PLAYERS_1,
        0 AS SUM_PRIZE_USD_0,
        W_SUM_PRIZE_USD AS SUM_PRIZE_USD_1,
        0 AS W_AVG_CNT_YEARS_0,
        W_AVG_CNT_YEARS AS W_AVG_CNT_YEARS_1
    FROM TEMP_WHOSEARLYERRETIRED_KVA
    WHERE W_ISMORETHAN_KVA = 1
    UNION
    SELECT
        W_GAME_ID,
        W_GAME_NAME,
        W_CTN_PLAYERS AS CTN_PLAYERS_0,
        0 AS CTN_PLAYERS_1,
        W_SUM_PRIZE_USD AS SUM_PRIZE_USD_0,
        0 AS SUM_PRIZE_USD_1,
        W_AVG_CNT_YEARS AS W_AVG_CNT_YEARS_0,
        0 AS W_AVG_CNT_YEARS_1
    FROM TEMP_WHOSEARLYERRETIRED_KVA
    WHERE W_ISMORETHAN_KVA = 0)
GROUP BY W_GAME_ID, W_GAME_NAME
ORDER BY W_GAME_ID;


-- table for maximum age of players

CREATE OR REPLACE TABLE WHOSEARLYERRETIRED_MAXAGE AS
WITH CTE_PDAK AS (
   SELECT
       pp.TOURNAMENT_ID,
       pp.YEAR_OF_TOURNAMENT,
       pp.GAME_ID, pp.PLAYER_ID,
       P_BIRTH_YEAR AS YEAR_OF_BIRTH,
       pp.YEAR_OF_TOURNAMENT - P_BIRTH_YEAR AS AGE_ON_TOURNAMENT_YEAR,
```

```
       pp.SUM_PRIZE,
       CASE WHEN pp.SUM_PRIZE >= KVA09_PRIZE_FOR_GAME THEN 1
       ELSE 0
       END as ISMORETHAN_KVA
   FROM TEMP_PLA_GAME_YEAR pp
   JOIN PLAYER ON pp.PLAYER_ID = P_PLAYER_ID
   JOIN TEMP_AVG_MED_KVA_PERGAME a ON a.GAME_ID = pp.GAME_ID
   WHERE P_BIRTH_YEAR is not null
   )

SELECT
    x.GAME_ID::int AS W_GAME_ID,
    G_GAME_NAME AS W_GAME_NAME,
    x.ISMORETHAN_KVA::int AS W_ISMORETHAN_KVA,
    COUNT(DISTINCT x.PLAYER_ID)::int as W_CTN_PLAYERS,
    AVG(CNT_AGE)::float as W_AVG_CNT_YEARS,
    MAX(MAX_AGE)::int as W_MAX_AGE
FROM
    (
    SELECT
        GAME_ID,
        PLAYER_ID,
        COUNT(DISTINCT AGE_ON_TOURNAMENT_YEAR) as CNT_AGE,
        MAX(AGE_ON_TOURNAMENT_YEAR) as MAX_AGE,
        SUM(SUM_PRIZE) as SUM2_PRIZE,
        ISMORETHAN_KVA
    FROM CTE_PDAK
    GROUP BY GAME_ID, PLAYER_ID, ISMORETHAN_KVA
    ) x
JOIN GAME g ON G_GAME_ID=x.GAME_ID
GROUP BY x.GAME_ID, G_GAME_NAME, x.ISMORETHAN_KVA
ORDER BY x.GAME_ID, G_GAME_NAME, x.ISMORETHAN_KVA DESC;
```

```sql
-- just a control table for a distribution of players in their years played
CREATE OR REPLACE TABLE WHOSEARLIERRETIRED_PLAYERSBYYERS AS
SELECT
    x.ISMORETHAN_KVA::int AS ISMORETHAN_KVA,
    CNT_YEARS,
    COUNT(DISTINCT x.PLAYER_ID)::int as CTN_PLAYERS
FROM
    (
    SELECT
        tityp.GAME_ID,
        tityp.PLAYER_ID,
        tityp.CNT_YEARS,
        SUM(SUM2_PRIZE) AS SUM3_PRIZE,
        ISMORETHAN_KVA,
        tityp.CNT_TOURNAMENTS
    FROM
        (
        SELECT
            GAME_ID,
            PLAYER_ID,
            COUNT(DISTINCT YEAR_OF_TOURNAMENT) AS CNT_YEARS,
            COUNT(DISTINCT TOURNAMENT_ID) AS CNT_TOURNAMENTS
        FROM
            (
            SELECT
                GAME_ID,
                PLAYER_ID,
                TOURNAMENT_ID,
                YEAR(T_END_DATE) AS YEAR_OF_TOURNAMENT
            FROM
                (
                SELECT
                    GAME_ID, PLAYER_ID,
                    TRI_TOURNAMENT_ID AS TOURNAMENT_ID
                FROM TEMP_PLAYER_GAME_AVG_MED_KVA
```

```
                LEFT JOIN TOURNAMENT_RESULTS_INDIVIDUAL ti ON PLAYER_ID=TRI_PLAYER_ID
                GROUP BY GAME_ID, PLAYER_ID, TRI_TOURNAMENT_ID
                UNION ALL
                SELECT
                    GAME_ID,
                    PLAYER_ID,
                    TRP_TOURNAMENT_ID AS TOURNAMENT_ID
                FROM TEMP_PLAYER_GAME_AVG_MED_KVA
                LEFT JOIN TOURNAMENT_RESULTS_PLAYER_IN_TEAM tt ON PLAYER_ID=TRP_PLAYER_ID
                GROUP BY GAME_ID, PLAYER_ID, TRP_TOURNAMENT_ID
                ORDER BY GAME_ID, PLAYER_ID
               ) tit
            LEFT JOIN TOURNAMENT t ON TOURNAMENT_ID=T_TOURNAMENT_ID
             ) tity
        GROUP BY GAME_ID, PLAYER_ID
        ) tityp
    JOIN TEMP_PLAYER_GAME_AVG_MED_KVA tg ON tg.GAME_ID=tityp.GAME_ID AND tg.PLAYER_ID=tityp.PLAYER_ID
    GROUP BY tityp.GAME_ID, tityp.PLAYER_ID, tityp.CNT_YEARS, tityp.CNT_TOURNAMENTS, ISMORETHAN_KVA
  ) x
GROUP BY ISMORETHAN_KVA, CNT_YEARS
ORDER BY ISMORETHAN_KVA, CNT_YEARS DESC;
```