1 SQLEquiQuest ground truth SQL queries

- Question 1: Show top 10 batsmen with highest total Caught stealing count in their entire lifetime.
  - Output format: playerid, firstname, lastname, total caught stealing
  - Order by: 1. total caught stealing (descending order), 2. firstname (ascending order), 3. lastname (ascending order), 4. playerid (ascending order)

```
-- Table: "Batting"
-- Columns: "playerID", "yearID", "CS"

-- Table: "People"
-- Columns: "playerID", "nameFirst", "nameLast"

with result as (
  select playerid, sum(coalesce(cs,0)) as
      total_caught_stealing
  from batting
  GROUP BY batting.playerid
  order by total_caught_stealing desc )

select result.playerid, coalesce(people.namefirst,
    '') as firstname, coalesce(people.namelast,'')
     as lastname, result.total_caught_stealing
  from result
  join people on result.playerid = people.playerid
  order by total_caught_stealing desc, namefirst
      asc, namelast asc, playerid asc
  limit 10;
```

- Question 2: Show details of top 10 batsmen with highest runscore calculated for their entire career (runscore for player's entire career is the sum of runscores across all games he played).
  - Output format: playerid, firstname, runscore.
  - Order by: 1. runscore (descending order), 2. firstname (descending order), 3. playerid (ascending order)

```
-- Table: "Batting"
-- Columns: "playerID", "h2b", "h3b", "hr"

-- Table: "People"
-- Columns: "playerID", "nameFirst", "nameLast"

with result as (
  select sum(2*coalesce(h2b,0) + 3*coalesce(h3b,0)
       + 4*coalesce(hr,0)) as runscore, playerid
  from batting
  GROUP BY playerid
  ORDER BY runscore desc )

select result.playerid, coalesce(people.namefirst,
    '') as firstname, result.runscore
  from result
  join people on result.playerid = people.playerid
  order by result.runscore desc, firstname desc,
      playerid asc
  limit 10;
```

- Question 3: For each player show the name of the player and total points received by them from the year 2000 and later.
  - Output format: playerid, playername, total points.
  - Order by: 1. total points (descending order), 2. playerid (ascending order)

```
-- Table: "AwardsShareManagers"
-- Coulumns: "playerID", "pointsWon" "yearID"

-- Table: "People"
-- Columns: "playerID", "nameFirst", "nameLast"

with result as (
  select playerid, sum(coalesce(pointsWon,0)) as
      total_points
  from awardsshareplayers
  where yearid>=2000 group by playerid ),
player_name_table as (
  select playerid, namefirst, namelast, coalesce(
      namefirst, '') || CASE WHEN (namefirst ||
      namelast) is not NULL THEN ' ' ELSE '' END
      || coalesce(namelast, '') as playername
  from people )

select result.playerid, PN.playername, result.
    total_points
  from result
  join player_name_table PN on result.playerid =
      PN.playerid
  order by total_points desc, playerid asc;
```

- Question 5: Output players with the number of seasons they have played in decreasing order. Note that a player might have played multiple roles out of batter, pitcher, and fielder in the same season; it still counts as one season.
  - Output format: playerid, firstname, lastname, date of birth, num seasons.
  - Order by: 1. num seasons (descending order) 2. playerid (ascending order) 3. firstname (ascending order) 4. lastname (ascending order) 5. date of birth (ascending order)

```
-- Table: "Batting"
-- Columns: "playerID", "yearID", "CS"

-- Table: "Fielding"
-- Columns: "playerID", "yearID"

-- Table: "Pitching"
-- Columns: "playerID", "yearID"

-- Table: "People"
-- Columns: "playerID", "nameFirst", "nameLast"

with all_tables as (
  select playerid, yearid
  from batting
  group by playerid, yearid
  union
  select playerid, yearid
  from fielding
  group by playerid, yearid
  union
  select playerid, yearid
  from pitching
  group by playerid, yearid ),
result as (
  select playerid, count(distinct(yearid)) as
      num_seasons
  from all_tables
  group by playerid )

select r.playerid, coalesce(p.namefirst,'') as
    firstname, coalesce(p.namelast,'') as lastname
    , coalesce(birthyear || '-' || lpad(birthmonth
    ::text,2,'0') || '-' || lpad(birthday::text,2,
    '0'),'') as date_of_birth, r.num_seasons
  from result r
  join people p on r.playerid = p.playerid
  order by r.num_seasons desc, r.playerid asc;
```

- Question 24:

  Graph 1: The concept of graphical analysis can be applied to this dataset. Consider the tables pitching and allstarfull. Now, we can make a graph such that all the players who appear in these tables are nodes of this graph. The edges are defined such that there exists an edge between two nodes (players) if they have played in the same team in the same season; the weight of the edge is the number of seasons played together in the same team. The graph thus formed is undirected and weighted. If players A and B played 5 seasons together in team X and 2 seasons together in team Y, the graph will have one edge between A and B with weight 7. For allstarfull table, consider only the tuples where the player actually played in the game i.e., GP = 1.

  Query: Using Graph 1, find whether there exists a path of length three or more between webbbr01 and clemero02. Output a boolean value: True for yes and False for no.
  – Output format: pathexists

```
-- Table: "AllStarFull"
-- Columns: "playerID", "YearID", "teamID", "GP"

-- Table: "Pitching"
-- Columns: "playerID", "yearID", "teamID"

create or replace view graph1 as
  select p1, p2, count(*) as w
  from (
    select table1.playerid as p1, table2.playerid
        as p2, table1.teamid, table1.yearid
    from (
      select * from allstarfull where GP = 1
    ) as table1, (
      select * from allstarfull where GP = 1
    ) as table2
    where table1.teamid = table2.teamid and table1
        .yearid = table2.yearid and not table1.
        playerid = table2.playerid
    union
    select table1.playerid as p1, table2.playerid
        as p2, table1.teamid, table1.yearid
    from pitching as table1, pitching as table2
    where table1.teamid = table2.teamid and table1
        .yearid = table2.yearid and not table1.
        playerid = table2.playerid
    union
    select table1.playerid as p1, table2.playerid
        as p2, table1.teamid, table1.yearid
    from pitching as table1, (
      select * from allstarfull where GP = 1
    ) as table2
    where table1.teamid = table2.teamid and table1
        .yearid = table2.yearid and not table1.
        playerid = table2.playerid
    union
    select table1.playerid as p1, table2.playerid
        as p2, table1.teamid, table1.yearid
    from (
      select * from allstarfull where GP = 1
    ) as table1, pitching as table2
    where table1.teamid = table2.teamid and table1
        .yearid = table2.yearid and not table1.
        playerid = table2.playerid
  ) as temp
  group by p1, p2;

with recursive sub as (
  select array[p1::text, p2::text] as path, p2, w
      as dist
  from graph1
  where p1 = 'webbbr01'
  union all
  select recur.path || graph1.p2::text, graph1.p2,
       (dist + w) as dist
  from sub as recur, graph1
  where graph1.p1 = recur.p2 and not graph1.p2 =
      any (recur.path) and not recur.p2 = '
      clemero02' )

select case when count(*) > 0 then True else False
     end as pathexists
  from sub
  where dist >= 3 and p2 = 'clemero02';
```