1. Query

```
SELECT
   batting.yearID AS Year,
   COUNT(batting.masterID) AS 'Number of Batters'
FROM
   batting
      JOIN
   teams
WHERE
   batting.teamID = teams.teamID
      AND teams.name = 'New York Yankees'
      AND teams.yearID = batting.yearID
GROUP BY batting.yearID
ORDER BY batting.yearID
```

Comments : joining two tables batting and teams because we do not have team name in batting table and counting number of batters grouping by yearID.

2. Query

```
SELECT
   t1.name 'Team Name (in first year of streak)',
   t1.yearid 'Year 1',
   t5.yearid 'Year 5',
   t1.w 'Wins in Year 1',
   t2.w 'Wins in Year 2',
   t3.w 'Wins in Year 3',
   t4.w 'Wins in Year 4',
   t5.w 'Wins in Year 5'
FROM
   teams t1
      JOIN
   teams t2 ON t1.teamid = t2.teamid
      JOIN
   teams t3 ON t1.teamid = t3.teamid
      JOIN
   teams t4 ON t1.teamid = t4.teamid
      JOIN
   teams t5 ON t1.teamid = t5.teamid
WHERE
   t1.yearid + 1 = t2.yearid
      AND t2.yearid + 1 = t3.yearid
      AND t3.yearid + 1 = t4.yearid
      AND t4.yearid + 1 = t5.yearid
      AND t1.w < t2.w
```

```
    AND t2.w < t3.w
    AND t3.w < t4.w
    AND t4.w < t5.w;
```

3. Query

4. Query

```
Select  * from
    (select distinct nameFirst, nameLast,dense_rank() over(order by 3B desc) as "Rank", 3B
as "Triples"
    from batting join
    master where master.masterID = batting.masterID ) d where d.Rank < 11
```

Comments: In above query using dense_rank function over column 3B so that we will get rank. Joining batting and master table because we need nameFirst and nameLast column.

5. Query

```
SELECT
   master.nameFirst,
   master.nameLast,
   batting.yearID AS yearID,
   batting.HR
FROM
   master
      JOIN
   batting
WHERE
   batting.masterID = master.masterID
      AND (batting.masterID , batting.yearID) IN (SELECT
         batting.masterID, batting.yearID
      FROM
         batting
            JOIN
         teams
      WHERE
         batting.yearID = teams.yearID
            AND batting.teamID = teams.teamID
            AND teams.name = 'New York Yankees'
            AND (batting.HR , batting.yearID) IN (SELECT
               MAX(batting.HR), batting.yearID
            FROM
               batting
            GROUP BY batting.yearID))
```

```
ORDER BY yearID ASC
```

Comments: Above query gives the result of new york yankees batters if they scored highest home runs than any other player in respective year. So I am using aggregate function max and using group by on column yearID, Joining with master table to get nameFirst and nameLast

6. Query

```
select master.nameFirst,master.nameLast,batting.yearID,batting.HR
from master join batting where batting.masterID=master.masterID and
(batting.HR,batting.yearID) IN
((select HR, yearID from
(SELECT HR, yearID, dense_rank() over(partition by yearID order by HR desc) as "rank" FROM
batting) a where a.rank = 3))
order by yearID
```

Comments: above query we need to find the 3rd max highest home runs by any batter in the respective year, to find the rank of batter I am using dense_rank function. Joining master and batter table to get first and last name of batter.

7. Query

```
SELECT DISTINCT
    namefirst, namelast
FROM
    (SELECT DISTINCT
        a2.teamid, a2.yearid, a2.lgid
    FROM
        (SELECT DISTINCT
        a.teamid, a.yearid, a.masterid
    FROM
        (SELECT DISTINCT
        teamid, yearid, lgid
    FROM
        appearances
    NATURAL JOIN master
    WHERE
        namefirst = 'Yogi'
            AND namelast = 'Berra') AS yogiMatches
    JOIN appearances AS a ON a.teamid = yogiMatches.teamid
        AND a.yearid = yogiMatches.yearid
        AND a.lgid = yogiMatches.lgid) AS MappingTable
    JOIN appearances AS a2 ON a2.masterid = MappingTable.masterid) AS DegreeOne
        JOIN
    appearances AS a3 ON a3.teamid = DegreeOne.teamid
```

```
    AND a3.yearid = DegreeOne.yearid
    AND a3.lgid = DegreeOne.lgid
    NATURAL JOIN
  master
ORDER BY namelast
```

Comments : finding the second degree players to yogi berra, first find the players(First Degree) that played against yogi berra then , find the teams played by first degree player, then find the players played by those teams.

8. Query

```
SELECT DISTINCT
  name
FROM
  teams
WHERE
  name NOT IN (SELECT DISTINCT
      name
    FROM
      teams
    WHERE
      teamID IN (SELECT DISTINCT
          teamID
        FROM
          master
            JOIN
          appearances
        WHERE
          master.masterID = appearances.masterID
            AND master.nameFirst = 'Rickey'
            AND master.nameLast = 'Henderson'))
    AND teams.yearID > (SELECT
      EXTRACT(YEAR FROM debut) AS startyear
    FROM
      master
    WHERE
      nameFirst = 'Rickey'
        AND nameLast = 'Henderson')
    AND teams.yearID < (SELECT
      EXTRACT(YEAR FROM finalgame) AS endyear
    FROM
      master
    WHERE
```

```
        nameFirst = 'Rickey'
            AND nameLast = 'Henderson')
ORDER BY name
```

Comments : above query gives the result of teams that ricky henderson did not play in his career. First we need to find the teams that ricky henderson played in his career. Then exclude those teams from all the teams that are there in his career.

9. Query

```
SELECT name, rn as "rank" FROM (SELECT name, W, Rank() OVER(order by W) AS rn
FROM teams WHERE yearID = 1970 AND lgID = "NL") a WHERE rn =(SELECT
CEILING(MAX(rn)/2)
FROM (SELECT name, W, Rank() OVER(order by W) AS rn FROM teams WHERE
yearID=1970 AND lgID = "NL") b)
```

Comments : we need to find the teams that has median number of total wins in national league. In the year 1970 - 1980.