**--1. all namefirst and namelast from people**

**--along with inducted field from hof\_inducted**

SELECT namefirst, namelast, inducted

FROM people LEFT OUTER JOIN hof\_inducted

ON people.playerid = hof\_inducted.playerid;

**--2. hof\_inducted and salaries INNER JOIN**

SELECT salaries.yearid, salaries.playerid, teamid, salary, category

FROM salaries INNER JOIN hof\_inducted

ON salaries.playerid = hof\_inducted.playerid;

**--3. salaries and hof\_inducted FULL OUTER JOIN**

SELECT salaries.playerid, salaries.yearid, teamid, lgid, salary, inducted

FROM hof\_inducted FULL OUTER JOIN salaries

ON hof\_inducted.playerid = salaries.playerid;

**--4. hof\_inducted and hof\_inducted UNION**

SELECT \* FROM hof\_inducted

UNION ALL

SELECT \* FROM hof\_not\_inducted;

SELECT playerid FROM hof\_inducted

UNION

SELECT playerid FROM hof\_not\_inducted;

**--5. SUM of salaries by name**

SELECT

namelast,

namefirst,

SUM(salary) AS total\_salary

FROM salaries AS s

INNER JOIN people AS p

ON s.playerid = p.playerid

GROUP BY p.playerid, namelast, namefirst;

**--6. namefirst and last with all hof records**

SELECT hof\_inducted.playerid, yearid, namefirst, namelast

FROM hof\_inducted LEFT OUTER JOIN people

ON hof\_inducted.playerid = people.playerid

UNION ALL

SELECT hof\_not\_inducted.playerid, yearid, namefirst, namelast

FROM hof\_not\_inducted LEFT OUTER JOIN people

ON hof\_not\_inducted.playerid = people.playerid;

**--7. Like 6. but Filtered since 1980 and**

**--sorted by year and a field "lastname, firstname"**

SELECT concat(namelast,', ', namefirst) AS namefull, yearid, inducted

FROM hof\_inducted LEFT OUTER JOIN people

ON hof\_inducted.playerid = people.playerid

WHERE yearid >= 1980

UNION ALL

SELECT concat(namelast,', ', namefirst) AS namefull, yearid, inducted

FROM hof\_not\_inducted LEFT OUTER JOIN people

ON hof\_not\_inducted.playerid = people.playerid

WHERE yearid >= 1980

ORDER BY yearid, inducted DESC, namefull;

**--8. Return a table containing the highest annual salary**

**-- for each teamid, ranked high to low along with the**

**-- matching playerid.**

**-- BONUS! In addition to playerid, return namelast**

**-- and namefirst in this table (These are in the people table.).**

WITH max AS

(SELECT MAX(salary) as max\_salary, teamid, yearid

FROM salaries

GROUP BY teamid, yearid)

SELECT salaries.yearid, salaries.teamid, playerid, max.max\_salary

FROM max LEFT OUTER JOIN salaries

ON salaries.teamid = max.teamid AND salaries.yearid = max.yearid AND salaries.salary = max.max\_salary

ORDER BY max.max\_salary DESC;

**--9. Bonus!**

WITH max AS

(SELECT MAX(salary) as max\_salary, teamid, yearid

FROM salaries

GROUP BY teamid, yearid)

SELECT salaries.yearid, salaries.teamid, salaries.playerid, namelast, namefirst, max.max\_salary

FROM salaries LEFT OUTER JOIN people

ON salaries.playerid = people.playerid

RIGHT OUTER JOIN max

ON salaries.teamid = max.teamid AND salaries.yearid = max.yearid AND salaries.salary = max.max\_salary

ORDER BY max.max\_salary DESC;

**--10. Select birthyear, deathyear, namefirst, namelast**

**-- of all players born since birthyear of**

**-- Babe Ruth (ID = ruthba01). Sort low to high.**

SELECT birthyear, deathyear, namefirst, namelast

FROM people

WHERE birthyear > ANY

(SELECT birthyear

FROM people

WHERE playerid = 'ruthba01')

ORDER BY birthyear;

**-- 11. Using the people table,**

**-- create a table containing**

**-- namefirst, namelast, and a**

**-- field called usaborn where**

**-- if the player's birthcountry is USA**

**-- then 'USA', else 'non-USA'.**

**-- order by 'non-USA' records first.**

SELECT namefirst, namelast,

CASE

WHEN birthcountry = 'USA' THEN 'USA'

ELSE 'non-USA'

END AS usaborn

FROM people

ORDER BY 3;

**-- 12. Calculate the average height**

**-- for players throwing with their**

**-- right versus left hand.**

**-- Name these fields right\_height**

**-- and left\_height**

SELECT

AVG(CASE WHEN throws = 'R' THEN height END) AS right\_height,

AVG(CASE WHEN throws = 'L' THEN height END) AS left\_height

FROM people;

**-- 13. Get average of team**

**-- maximum salaries since 2010.**

WITH max\_sal\_by\_team\_by\_year AS

(

SELECT teamid, yearid, MAX(salary) AS max\_sal

FROM salaries

GROUP BY teamid, yearid

)

SELECT teamid, AVG(max\_sal) AS avg\_max\_sal\_since\_2010

FROM max\_sal\_by\_team\_by\_year

WHERE yearid >= 2010

GROUP BY teamid;