--1. return the namefirst and namelast fields of the people table, along with the inducted field from the hof\_inducted table.--

SELECT namefirst, namelast, inducted

FROM people

LEFT OUTER JOIN hof\_inducted

ON people.playerid=hof\_inducted.playerid;

--2. return the yearid, playerid, teamid, and salary fields from the salaries table, along with the category field from the hof\_inducted table.--

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

FROM salaries

INNER JOIN hof\_inducted

ON salaries.playerid=hof\_inducted.playerid;

--3. return the playerid, yearid, teamid, lgid, and salary fields from the salaries table and the inducted field from the hof\_inducted table.--

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

FROM salaries

FULL OUTER JOIN hof\_inducted

ON salaries.playerid=hof\_inducted.playerid;

--4. Combine hof\_inducted and hof\_not\_inducted by all fields. Then, Get a distinct list of all player IDs for players who have been put up for HOF induction.--

SELECT \*

FROM hof\_inducted

UNION

SELECT \*

FROM hof\_not\_inducted;

SELECT playerid

FROM hof\_inducted

UNION

SELECT playerid

FROM hof\_not\_inducted;

--5. return the last name, first name (see the people table), and total recorded salaries for all players found in the salaries table.--

SELECT namelast,namefirst, salary

FROM people

RIGHT OUTER JOIN salaries

ON people.playerid=salaries.playerid;

--6. returns all records from the hof\_inducted and hof\_not\_inducted tables that include playerid, yearid, namefirst, and namelast. Hint: Each FROM statement will include a LEFT OUTER JOIN.--

SELECT \*

FROM hof\_inducted

LEFT OUTER JOIN hof\_not\_inducted

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

LEFT OUTER JOIN people

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

--7. Return a table including all records from both hof\_inducted and hof\_not\_inducted. Include a new field, namefull, which is formatted as namelast , namefirst (in other words, the last name, followed by a comma, then a space, then the first name). The query should also return the yearid and inducted fields. Include only records since 1980 from both tables. Sort the resulting table by yearid, then inducted so that Y comes before N. Finally, sort by the namefull a-z..--

SELECT CONCAT (namelast , ', ' , namefirst) AS namefull, HOF\_combined.yearid, HOF\_combined.inducted

FROM people

LEFT OUTER JOIN

(SELECT \* FROM hof\_inducted

UNION

SELECT \* FROM hof\_not\_inducted) HOF\_combined

ON people.playerid=HOF\_combined.playerid

WHERE HOF\_combined.yearid>'1980'

ORDER BY yearid, inducted DESC, namefull;

--9. return each year's highest annual salary for each team ID, ranked from high to low, along with the corresponding player ID. Bonus: Return namelast and namefirst in the resulting table. --

SELECT MAX(salary), yearid, teamid, salaries.playerid, namelast, namefirst

FROM salaries

INNER JOIN people

ON salaries.playerid=people.playerid

GROUP BY yearid, teamid, salaries.playerid, namelast, namefirst

ORDER BY yearid DESC, max DESC;

--10. Select birthyear, deathyear, namefirst, and namelast of all the players born since the birth year of Babe Ruth (whose playerid is ruthba01). Sort the results by birth year from low to high.--

SELECT birthyear, deathyear, namefirst, namelast

FROM people

WHERE birthyear >=

(SELECT birthyear

FROM people

WHERE playerid ='ruthba01')

ORDER BY birthyear;

--11. Using the people table, write a query that returns namefirst, namelast, and a field called usaborn.--

SELECT namefirst,namelast,

CASE

WHEN birthcountry LIKE 'USA' THEN 'USA'

ELSE 'non-USA'

END AS usaborn

FROM people

ORDER BY usaborn;

--12. Calculate the average height for players throwing with their right hand versus their left hand. Name these fields right\_height and left\_height, respectively.--

SELECT

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

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

FROM people

WHERE height IS NOT NULL

AND throws IS NOT NULL;

--13. Get the average of each team's maximum player salary since 2010. Hint: WHERE will go outside of your CTE.--

WITH max\_salary\_per\_team\_per\_year AS

(

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

FROM salaries

GROUP BY teamid, yearid

)

SELECT teamid, AVG(max\_salary) AS max\_salary\_since\_2010

FROM max\_salary\_per\_team\_per\_year

WHERE yearid>'2010'

GROUP BY teamid

ORDER BY teamid;