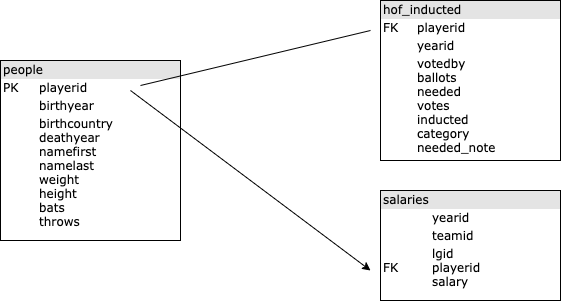
1. Draw a schema of the relationship between the people, salaries, and hof\_inducted tables.
   1. Label the primary and foreign keys.



* 1. What are the parent and child tables? Are these one-to-one, one-to-many, or many-to-many relationships?

The people table is the parent and the hof\_inducted and salaries are child tables. The people to hof\_inducted table is a one-to-one relationship while the people to salaries tables is a one-to-many relationship.

1. Write a query that returns the namefirst and namelast fields of the people table, along with the inducted field from the hof\_inducted table. All rows from the people table should be returned, and NULL values for the fields from hof\_inducted should be returned when there is no match found.

SELECT namefirst, namelast, inducted

FROM people LEFT OUTER JOIN hof\_inducted

ON people.playerid = hof\_inducted.playerid;

1. In 2006, a special Baseball Hall of Fame induction was conducted for players from the negro baseball leagues of the 20th century. In that induction, 17 players were posthumously inducted into the Baseball Hall of Fame. Write a query that returns the first and last names, birth and death dates, and birth countries for these players. Note that the year of induction was 2006, and the value for votedby will be “Negro League.”

SELECT namefirst, namelast, birthyear, deathyear, birthcountry

FROM people INNER JOIN hof\_inducted

ON people.playerid = hof\_inducted.playerid

WHERE hof\_inducted.yearid = 2006 AND

hof\_inducted.votedby = 'Negro League';

1. Write a query that returns the yearid, playerid, teamid, and salary fields from the salaries table, along with the category field from the hof\_inducted table. Keep only the records that are in both salaries and hof\_inducted. Hint: While a field named yearid is found in both tables, don’t JOIN by it. You must, however, explicitly name which field to include.

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

FROM salaries INNER JOIN hof\_inducted

ON salaries.playerid = hof\_inducted.playerid;

1. Write a query that returns the playerid, yearid, teamid, lgid, and salary fields from the salaries table and the inducted field from the hof\_inducted table. Keep all records from both tables.

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

FROM salaries FULL OUTER JOIN hof\_inducted

ON salaries.playerid = hof\_inducted.playerid;

1. There are 2 tables, hof\_inducted and hof\_not\_inducted, indicating successful and unsuccessful inductions into the Baseball Hall of Fame, respectively.
   1. Combine these 2 tables by all fields. Keep all records.

SELECT \*

FROM hof\_inducted UNION ALL

SELECT \*

FROM hof\_not\_inducted;

* 1. Get a distinct list of all player IDs for players who have been put up for HOF induction.

SELECT playerid

FROM hof\_inducted UNION

SELECT playerid

FROM hof\_not\_inducted;

1. Write a query that returns the last name, first name (see people table), and total recorded salaries for all players found in the salaries table.

SELECT namefirst, namelast, SUM(salary) AS total\_salary

FROM people RIGHT OUTER JOIN salaries

ON people.playerid = salaries.playerid

GROUP BY namefirst, namelast;

1. Write a query that 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 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

1. Return a table including all records from both hof\_inducted and hof\_not\_inducted, and 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 field, A to Z.

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 ASC, inducted DESC, namefull ASC

1. Write a query that returns the highest annual salary for each teamid, ranked from high to low, along with the corresponding playerid. Bonus! Return namelast and namefirst in the resulting table. (You can find these in the people table.)

I couldn’t figure this out and the example answer isn’t correct. It returns multiple high salaries for each team.

1. Select birthyear, deathyear, namefirst, and namelast of all the players born since the birth year of Babe Ruth (playerid = 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;

1. Using the people table, write a query that returns namefirst, namelast, and a field called usaborn where. The usaborn field should show the following: if the player's birthcountry is the USA, then the record is 'USA.' Otherwise, it's 'non-USA.' Order the results by 'non-USA' records first.

SELECT namefirst, namelast,

CASE

WHEN birthcountry = 'USA' THEN 'USA'

ELSE 'Non-USA'

END as usa\_born

FROM people

ORDER BY 3;

1. 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

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

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

FROM people;

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

WITH cte AS

(

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

FROM salaries

WHERE yearid > 2010

GROUP BY teamid, yearid

)

SELECT teamid, AVG(max\_salary) AS avg\_max\_salary

FROM cte

GROUP BY teamid

ORDER BY avg\_max\_salary DESC;