

# Client Report - [Project 3: Finding relationships in baseball. ]

---

**Course CSE 250** James Lule

Elevator pitch

*paste your elevator pitch here*

GRAND QUESTION 1

Write an SQL query to create a new dataframe  
about baseball players who attended BYU-Idaho.

---

The new table should contain

---

five columns: playerID, schoolID, salary, and the yearID/teamID associated with each salary.

---

Order the table by salary (highest to lowest) and  
print out the table in your report.

---

## TECHNICAL DETAILS

#paste chart code in this snippet box

```
#paste your table code in this snippet box
df = pd.read_sql_query("""
SELECT c.playerID, c.schoolID, s.salary, s.teamID FROM Salaries AS s
JOIN CollegePlaying AS c
ON c.playerID = s.playerID
WHERE schoolID = "idbyuid"
ORDER BY salary DESC
LIMIT 5
""", con)
```

df

replace the table below with your table

	playerID	schoolID	salary	teamID
0	lindsma01	idbyuid	4000000.0	CHA
1	lindsma01	idbyuid	4000000.0	CHA
2	lindsma01	idbyuid	3600000.0	BAL
3	lindsma01	idbyuid	3600000.0	BAL
4	lindsma01	idbyuid	2800000.0	COL

GRAND QUESTION 2

COPY PASTE GRAND QUESTION 2 FROM THE PROJECT HERE

Write an SQL query that provides playerID, yearID, and batting average for players with at least one at bat. Sort the table from highest batting average to lowest, and show the top 5 results in your report. *type your results and analysis here*

TECHNICAL DETAILS

```
#paste chart code in this snippet box
df1 = pd.read_sql_query("""SELECT playerID, yearID, H, AB, (H / AB ) as
Batting_Average
FROM Batting
WHERE AB >= 1
ORDER BY Batting_Average DESC
LIMIT 5;
""", con)

df1
```

#paste your table code in this snippet box

replace the table below with your table

	playerID	yearID	H	AB	Batting_Average
0	snowch01	1874	1	1	1
1	baldwki01	1884	1	1	1

	playerID	yearID	H	AB	Batting_Average
2	mccafsp01	1889	1	1	1
3	gumbebi01	1893	1	1	1
4	oconnfr01	1893	2	2	1

### GRAND QUESTION 3

**# Write an SQL query that provides playerID, yearID, and batting average for players with at least one at bat. Sort the table from highest batting average to lowest, and show the top 5 results in your report.**

Use the same query as above, but only include players with more than 10 "at bats" that year. Print the top 5 results.

---

Now calculate the batting average for players over their entire careers (all years combined). Only include players with more than 100 at bats, and print the top 5 results.

---

Pick any two baseball teams and compare them using a metric of your choice (average salary, home runs, number of wins, etc.). Write an SQL query to get the data you need. Use Python if additional data wrangling is needed, then make a graph in Altair to visualize the comparison. Provide the visualization and its description.

---

*type your results and analysis here*

#### TECHNICAL DETAILS

#paste chart code in this snippet box

*insert your chart png here*

```
#paste your table code in this snippet box
df2 = pd.read_sql_query("""SELECT playerID, H, AB, round(SUM(H + 0.0) / SUM(AB),
3) as Batting_Average
FROM Batting
GROUP BY playerID
HAVING AB >= 11
ORDER BY Batting_Average DESC
LIMIT 5;
""", con)

df2
```

replace the table below with your table

	playerID	H	AB	Batting_Average
0	silvelu01	6	11	0.545
1	durhado01	7	14	0.500
2	lucefr01	6	12	0.500
3	dappecl01	8	17	0.471
4	mattetr01	7	15	0.467

## GRAND QUESTION 4

# Now calculate the batting average for players over their

entire careers (all years combined). Only include players

with more than 100 at bats, and print the top 5 results.

type your results and analysis here

### TECHNICAL DETAILS

```
#paste your table code in this snippet box
df3 = pd.read_sql_query("""SELECT playerID, H, AB, round(SUM(H + 0.0) / SUM(AB),
3) as Batting_Average
FROM Batting
GROUP BY playerID
HAVING AB > 100
ORDER BY Batting_Average DESC
LIMIT 5;
""", con)
```

```
df3
```

	playerID	H	AB	Batting_Average
0	cobbty01	36	151	0.366
1	barnero01	63	157	0.360
2	lucefr01	64	130	0.356
3	kingst01	57	144	0.353
4	delahed01	66	290	0.346

GRAND QUESTION 5

Pick any two baseball teams and compare them using a metric of your

choice (average salary, home runs, number of wins, etc.). Write an SQL query to get the data you need. Use Python if additional data wrangling is needed, then make a graph in Altair to visualize the comparison. Provide the visualization and its description. *type your results and analysis here*

TECHNICAL DETAILS

```
#paste chart code in this snippet box
```

*insert your chart png here*

```
#paste your table code in this snippet box
```

*replace the table below with your table*

teams	avg
NYN	2.463020e+06
NYA	3.968910e+06

APPENDIX A (PYTHON CODE)

```
#paste all your code from your python file (.py) here

#%%

import pandas as pd
import numpy as np
```

```

import json
import sqlite3
import altair as alt
# %%
con = sqlite3.connect('lahmansbaseballdb.sqlite')

# %%

# %%
# df = pd.read_sql_query("""SELECT sum(H) FROM CollegePlaying WHERE yearID""",
con)

# Write an SQL query to create a new dataframe
# about baseball players who attended BYU-Idaho.
# The new table should contain
# five columns: playerID, schoolID, salary, and the yearID/teamID associated with
each salary.
# Order the table by salary (highest to lowest) and
# print out the table in your report.

# df = pd.read_sql_query("""SELECT sum(H)
# FROM batting WHERE yearID
# """, con)
# print(df)
df = pd.read_sql_query("""
SELECT c.playerID, c.schoolID, s.salary, s.teamID FROM Salaries AS s
JOIN CollegePlaying AS c
ON c.playerID = s.playerID
WHERE schoolID = "idbyuid"
ORDER BY salary DESC
LIMIT 5
""", con)

df

# %%
# This three-part question requires you to calculate
# batting average (number of hits divided by the number of at-bats)

df1 = pd.read_sql_query("""SELECT playerID, yearID, H, AB, (H / AB ) as
Batting_Average
FROM Batting
WHERE AB >= 1
ORDER BY Batting_Average DESC
LIMIT 5;
""", con)

```

df1

#%

# Write an SQL query that provides playerID, yearID, and batting average for players with at least one at bat. Sort the table from highest batting average to lowest, and show the top 5 results in your report.

# Use the same query as above, but only include players with more than 10 “at bats” that year. Print the top 5 results.

# Now calculate the batting average for players over their entire careers (all years combined). Only include players with more than 100 at bats, and print the top 5 results.

# Pick any two baseball teams and compare them using a metric of your choice (average salary, home runs, number of wins, etc.). Write an SQL query to get the data you need. Use Python if additional data wrangling is needed, then make a graph in Altair to visualize the comparison. Provide the visualization and its description.

```
df2 = pd.read_sql_query("""SELECT playerID, H, AB, round(SUM(H + 0.0) / SUM(AB),
3) as Batting_Average
FROM Batting
GROUP BY playerID
HAVING AB >= 11
ORDER BY Batting_Average DESC
LIMIT 5;
""", con)
```

df2

# %%

# Now calculate the batting average for players over their  
# entire careers (all years combined). Only include players  
# with more than 100 at bats, and print the top 5 results.

```
df3 = pd.read_sql_query("""SELECT playerID, H, AB, round(SUM(H + 0.0) / SUM(AB),
3) as Batting_Average
FROM Batting
GROUP BY playerID
HAVING AB > 100
ORDER BY Batting_Average DESC
LIMIT 5;
""", con)
```

df3

#%

# Pick any two baseball teams and compare them using a metric of your  
# choice (average salary, home runs, number of wins, etc.).  
# Write an SQL query to get the data you need.  
# Use Python if additional data wrangling is needed,

```
# then make a graph in Altair to visualize the comparison.  
# Provide the visualization and its description.
```

```
q3 =pd.read_sql_query (''  
SELECT teamid AS teams, avg(cast(salary AS float)) AS avg  
FROM Salaries  
WHERE teamid = 'NYA' OR teamid = 'NYN'  
GROUP BY teamid  
ORDER BY teamid DESC;
```

```
''', con)
```

```
q3
```

```
chart = (alt.Chart(q3,  
title = 'Average Salary NYC and NYN'  
).  
encode(  
    x = alt.X('teams', title = 'Teams'),  
    y = alt.Y('avg', title = 'Average')  
  
).  
mark_bar()  
)  
chart
```

```
# %%
```