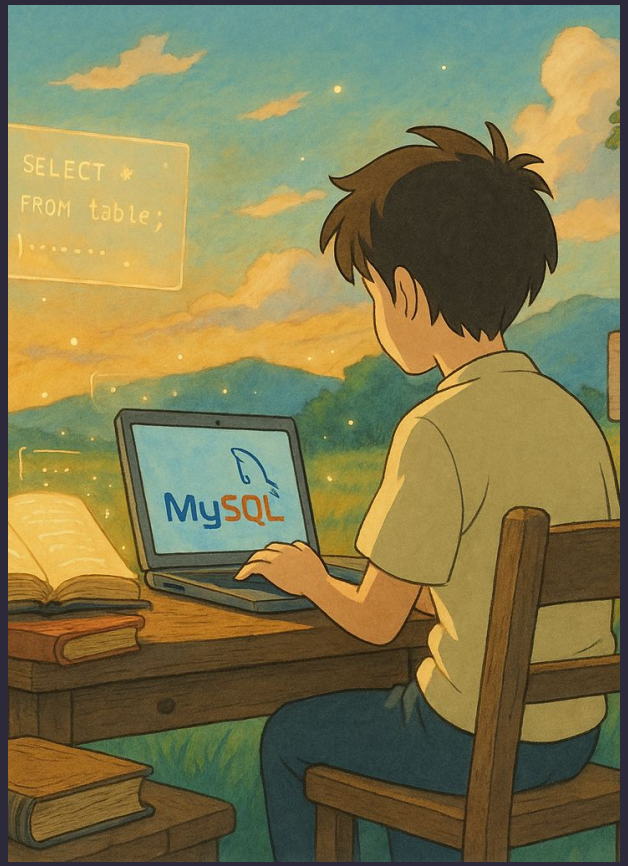
Major League Baseball (MLB) Analysis

MySQL Project

The Introduction

Leverage everything I have learned in "SQL for Data Analysis: Advanced SQL Querying Techniques" to track how Major League Baseball (MLB) player statistics have changed over time and across different teams in the league.



Knowing **The Project**

The **Situation**

You've just been hired as a **Data Analyst** Intern for Major League Baseball (MLB), who has recently gotten access to a large of historical player data.

The **Assignment**

You have access to **decades worth of data** including player statistics like schools attended, salaries, teams played for, height and weight, and more. Your task is to **use advanced SQL querying** techniques to **track how player statistics** have changed over time and across different teams in the league.

The Documentation

You can access the Documentation in my Github Repository Click This Link <MySQL Project>

Key Objectives

01 School Analysis

What schools do MLB players attend?

02 Salary Analysis

How much do teams spend on player salaries?

03 Player Career Analysis

What does each player's career look like?

04 Player Comparison Analysis

How do player attributes compare?

01 { ..

School Analysis

< What schools do MLB players attend? >

}

a) In each decade, how many schools were there that produced MLB players?

< Query >

```sql
SELECT
    FLOOR(yearID/10)*10 AS decade
    ,COUNT(DISTINCT schoolID) AS total_school
FROM
    schools
GROUP BY
    decade
ORDER BY
    decade DESC;
```

Result Grid | Filter Rows:

decade	total_school
2010	57
2000	372
1990	494
1980	473
1970	427
1960	301
1950	176
1940	142
1930	162
1920	196
1910	178

< Result >

b) What are the names of the top 5 schools that produced the most players?

```sql
WITH sch_d AS (
    SELECT
        pl.playerID
        ,pl.nameFirst
        ,pl.nameLast
        ,pl.debut
        ,sch.schoolID
        ,sch.yearID
        ,sd.name_full AS school_name
    FROM
        players AS pl
    LEFT JOIN
        schools AS sch
        ON pl.playerID = sch.playerID
    LEFT JOIN
        school_details AS sd
        ON sch.schoolID = sd.schoolID
    )

SELECT
    school_name
    ,COUNT(playerID) AS total_player
FROM
    sch_d
WHERE
    school_name IS NOT NULL
GROUP BY
    school_name
ORDER BY
    COUNT(playerID) DESC
LIMIT 5;
```

< Query >

< Result >

Result Grid | Filter Rows: | Exp

school_name	total_player
University of Texas at Austin	265
University of Southern California	250
Stanford University	248
Arizona State University	236
University of Michigan	191

c) For each decade, what were the names of the top 3 schools that produced the most players?

```sql
WITH sch_d AS (
    SELECT
        pl.playerID
        ,pl.nameFirst
        ,pl.nameLast
        ,pl.debut
        ,sch.schoolID
        ,sch.yearID
        ,sd.name_full AS school_name
    FROM
        players AS pl
    LEFT JOIN
        schools AS sch
        ON pl.playerID = sch.playerID
    LEFT JOIN
        school_details AS sd
        ON sch.schoolID = sd.schoolID
    )
    ,rank_total_player AS (
    SELECT
        DISTINCT school_name
        ,FLOOR(yearID/10)*10 AS decade
        ,COUNT(DISTINCT playerID) AS total_player
        ,ROW_NUMBER() OVER(PARTITION BY FLOOR(yearID/10)*10 ORDER BY COUNT(playerID) DESC) AS ranking
    FROM
        sch_d
    WHERE
        school_name IS NOT NULL
    GROUP BY
        school_name
        ,decade
    ORDER BY
        decade DESC, total_player DESC
    )
SELECT
    school_name
    ,decade
    ,total_player
FROM
    rank_total_player
WHERE
    ranking BETWEEN 1 AND 3;
```

< Query >

< Result >

Result Grid | Filter Rows: | Export: | Wra

school_name	decade	total_player
University of Florida	2010	5
Georgia Institute of Technology	2010	3
University of South Carolina	2010	3
California State University Long Beach	2000	23
Arizona State University	2000	23
Stanford University	2000	22
Stanford University	1990	25
University of Southern California	1990	23
Florida State University	1990	21
University of Arizona	1980	24
University of California, Los Angeles	1980	22

Result 3

02 { ..

Salary Analysis

< How much do teams spend on player salaries? >

}

a) Return the top 20% of teams in terms of average annual spending

```sql
WITH total_spend AS (
    SELECT
        teamID
        ,yearID
        ,SUM(salary) AS total_spend
    FROM
        salaries
    GROUP BY
        teamID
        ,yearID
    ORDER BY
        teamID
        ,yearID
    )

    ,avg_spend AS (
    SELECT
        teamID
        ,AVG(total_spend) AS avg_spending
        ,NTILE(5) OVER (ORDER BY AVG(total_spend) DESC) AS spend_percentg
    FROM
        total_spend
    GROUP BY
        teamID
    )

SELECT
    teamID
    ,ROUND(avg_spending/1000000, 2) AS avg_spending_in_mil
FROM
    avg_spend
WHERE
    spend_percentg = 1;
```

< Query >

< Result >

Result Grid | Filter Rows:

teamID	avg_spending_in_mil
SFG	143.51
LAA	118.47
NYA	109.44
BOS	81.09
LAN	74.59
WAS	71.54
ARI	71.18
PHI	66.08

b) For each team, show the cummulative sum of spending over the years

```sql
WITH team_spending AS (
    SELECT
        yearID
        ,teamID
        ,SUM(salary) AS total_spend
    FROM
        salaries
    GROUP BY
        yearID
        ,teamID
    )
SELECT
    yearID
    ,teamID
    ,ROUND(total_spend/1000000, 2) AS total_spend_in_mil
    ,ROUND(SUM(total_spend) OVER(PARTITION BY teamID ORDER BY yearID)/1000000, 2) AS cum_spend_in_mil
FROM
    team_spending;
```

< Result >

Result Grid | Filter Rows: | Export:

yearID	teamID	total_spend_in_mil	cum_spend_in_mil
1997	ANA	31.14	31.14
1998	ANA	41.28	72.42
1999	ANA	55.39	127.80
2000	ANA	51.46	179.27
2001	ANA	47.54	226.80
2002	ANA	61.72	288.53
2003	ANA	79.03	367.56
2004	ANA	100.53	468.09
1998	ARI	32.35	32.35
1999	ARI	68.70	101.05
2000	ARI	81.03	182.08

< Query >

c) Return the first year that each team's cumulative spending surpassed 1 billion

```sql
WITH team_spending AS (
    SELECT
        yearID
        ,teamID
        ,SUM(salary) AS spending
    FROM
        salaries
    GROUP BY
        yearID
        ,teamID
    )
    ,cumulative_spending AS (
    SELECT
        yearID
        ,teamID
        ,SUM(spending) OVER(PARTITION BY teamID ORDER BY yearID) AS cum_spend
    FROM
        team_spending
    )
SELECT
    teamID
    ,MIN(yearID) AS year_spend
FROM
    cumulative_spending
WHERE
    cum_spend > 1000000000
GROUP BY
    teamID;
```

< Query >

< Result >

Result Grid | Filter Rows:

teamID	year_spend
ARI	2012
ATL	2005
BAL	2007
BOS	2004
CHA	2008
CHN	2007
CIN	2010
CLE	2009
COL	2011
DET	2009
HOU	2008

03 { ..

Player Career Analysis

< What does each player's career look like? >

}

a) For each player, calculate their age at their first (debut) game, their last game, and their career length (all in years). Sort from longest career to shortest career?

```sql
WITH datePlayers AS (
    SELECT
        playerID
        ,CONCAT(nameFirst, ' ', nameLast) AS playerName
        ,CAST(CONCAT(birthYear, '-', birthMonth, '-',birthDay) AS DATE) AS dateBirth
        ,debut
        ,finalGame
    FROM
        players
    )
SELECT
    playerName
    ,ROUND(DATEDIFF(debut, dateBirth)/364) AS ageDebut
    ,ROUND(DATEDIFF(finalGame, dateBirth)/364) AS ageRetire
    ,ROUND(DATEDIFF(finalGame, debut)/364) AS lenghtCareerInYear
FROM
    datePlayers
ORDER BY
    lenghtCareerInYear DESC;
```

< Result >

Result Grid | Filter Rows: | Export: | Wrap

playerName	ageDebut	ageRetire	lenghtCareerInYear
Nick Altrock	22	57	35
Jim O'Rourke	22	54	33
Minnie Minoso	23	55	32
Charley O'Leary	29	59	31
Arlie Latham	20	50	29
Deacon McGuire	21	49	28
Nolan Ryan	20	47	27
Hughie Jennings	22	50	27
Gabby Street	22	49	27
Cap Anson	19	46	27
Johnny Evers	21	48	27

Result 1

< Query >

b) What team did each player play on for their starting and ending years?

```sql
SELECT
    pl.playerID
    ,CONCAT(pl.nameFirst, ' ', pl.nameLast) AS playerName
    ,sl1.yearID AS debutYear
    ,sl1.teamID AS debutTeam
    ,sl2.yearID AS retireYear
    ,sl2.teamID AS retireTeam
FROM
    players AS pl
INNER JOIN
    salaries AS sl1
    ON pl.playerID = sl1.playerID
    AND YEAR(pl.debut) = sl1.yearID
INNER JOIN
    salaries AS sl2
    ON pl.playerID = sl2.playerID
    AND YEAR(finalGame) = sl2.yearID
ORDER BY
    YEAR(finalGame) DESC;
```

< Result >

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

playerID	playerName	debutYear	debutTeam	retireYear	retireTeam
wrighja01	Jamey Wright	1996	COL	2014	LAN
colonba01	Bartolo Colon	1997	CLE	2014	NYM
gonzaal02	Alex Gonzalez	1998	FLO	2014	DET
beltrad01	Adrian Beltre	1998	LAN	2014	TEX
ramirar01	Aramis Ramirez	1998	PIT	2014	MIL
molinjo01	Jose Molina	1999	CHN	2014	TBA
burneaj01	A. J. Burnett	1999	FLO	2014	PHI
hudsoti01	Tim Hudson	1999	OAK	2014	SFG
nathajo01	Joe Nathan	1999	SFN	2014	DET
beimejo01	Joe Beimel	2001	PIT	2014	SEA
suzukic01	Ichiro Suzuki	2001	SEA	2014	NYA

< Query >

c) How many players started and ended on the same team and also played for over a decade?

```sql
SELECT
    pl.playerID
    ,CONCAT(pl.nameFirst, ' ', pl.nameLast) AS playerName
    ,sl1.yearID AS debutYear
    ,sl1.teamID AS debutTeam
    ,sl2.yearID AS retireYear
    ,sl2.teamID AS retireTeam
    ,sl2.yearID - sl1.yearID AS lengthCareer
    ,ROW_NUMBER() OVER() AS counting
FROM
    players AS pl
INNER JOIN
    salaries AS sl1
    ON pl.playerID = sl1.playerID
    AND YEAR(pl.debut) = sl1.yearID
INNER JOIN
    salaries AS sl2
    ON pl.playerID = sl2.playerID
    AND YEAR(finalGame) = sl2.yearID
WHERE
    sl1.teamID = sl2.teamID AND
    sl2.yearID - sl1.yearID > 10
ORDER BY
    counting DESC;
```

< Result >

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

playerID	playerName	debutYear	debutTeam	retireYear	retireTeam	lengthCareer	counting
utleych01	Chase Utley	2003	PHI	2014	PHI	11	19
woodke02	Kerry Wood	1998	CHN	2012	CHN	14	18
heltoto01	Todd Helton	1997	COL	2013	COL	16	17
aurilri01	Rich Aurilia	1995	SFN	2009	SFN	14	16
riverma01	Mariano Rivera	1995	NYA	2013	NYA	18	15
pettian01	Andy Pettitte	1995	NYA	2013	NYA	18	14
radkebr01	Brad Radke	1995	MIN	2006	MIN	11	13
jonesch06	Chipper Jones	1993	ATL	2012	ATL	19	12
hentgpa01	Pat Hentgen	1991	TOR	2004	TOR	13	11
willibe02	Bernie Williams	1991	NYA	2006	NYA	15	10
lankfra01	Ray Lankford	1990	SLN	2004	SLN	14	9

< Query >

04 { ..

Player Comparison Analysis

< How do player attributes compare? >

}

a) Which player have the same birthday?

```sql
WITH birth AS (
    SELECT
        playerID
        ,CONCAT(nameFirst, ' ', nameLast) AS namePlayer
        ,CAST(CONCAT(birthYear, '-', birthMonth, '-', birthDay) AS DATE) AS birthDate
    FROM
        players
    )
SELECT
    birthDate
    ,GROUP_CONCAT(namePlayer SEPARATOR'; ') AS players
FROM
    birth
WHERE
    birthDATE IS NOT NULL
GROUP BY
    birthDate
HAVING
    COUNT(namePlayer) >1;
```

< Result >

Result Grid | Filter Rows: | Export: | Wr

birthDate	players
1845-01-31	Freeman Brown; Bob Ferguson
1854-05-04	Flip Lafferty; Jim Shanley
1854-10-06	Pop Snyder; Frank McCarton
1855-01-01	Bill Sharsig; Tom Mansell; Bill McGunnigle
1855-02-14	Lou Sylvester; Joe Gerhardt
1855-08-20	George Fisher; Dave Pierson
1855-10-02	Jack Allen; Bob Blakiston
1856-09-05	Tug Thompson; Jimmy Knowles
1857-03-09	George Daisy; Sam Moffet
1857-10-24	Dick Pierson; Ned Williamson
1858-03-03	Monk Cline; Harry Wheeler

< Query >

b) Create a summary table that shows for each team, what percent of players bat right, left, and both

```sql
SELECT
    playerID, bats
FROM
    players;

SELECT
    sl.teamID
    ,COUNT(sl.playerID) AS total_players
    ,ROUND(SUM(CASE WHEN bats = 'R' THEN 1 ELSE 0 END) / COUNT(sl.playerID) *100, 2) AS bats_right_inPercent
    ,ROUND(SUM(CASE WHEN bats = 'L' THEN 1 ELSE 0 END) / COUNT(sl.playerID) *100, 2) AS bats_left_inPercent
    ,ROUND(SUM(CASE WHEN bats = 'B' THEN 1 ELSE 0 END) / COUNT(sl.playerID) *100, 2) AS bats_both_inPercent
FROM
    salaries AS sl
LEFT JOIN
    players AS pl
    ON sl.playerID = pl.playerID
GROUP BY
    sl.teamID
ORDER BY
    COUNT(sl.playerID) DESC;
```

< Result >

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

teamID	total_players	bats_right_inPercent	bats_left_inPercent	bats_both_inPercent
PHI	893	58.45	31.35	10.19
CLE	893	59.57	29.68	10.75
LAN	893	62.93	27.77	9.18
SLN	886	61.85	26.52	11.63
BAL	883	61.83	29.56	8.61
BOS	883	61.95	29.33	8.61
OAK	881	62.66	27.47	9.88
NYA	879	58.82	30.72	10.47
PIT	879	64.39	27.42	8.19
CIN	874	62.59	29.41	8.01
KCA	874	64.30	27.23	8.47

< Query >

c) How have average height and weight at debut game changed over the years, and what's the decade-over-decade difference?

```sql
WITH wh AS (
SELECT
    FLOOR(YEAR(debut)/10)*10 AS decade
    ,ROUND(AVG(weight), 2) AS avg_weight
    ,ROUND(AVG(height), 2) AS avg_height
FROM
    players
GROUP BY
    decade
    )

SELECT
    decade
    ,avg_weight - LAG(avg_weight) OVER(ORDER BY decade) weight_diff
    ,avg_height - LAG(avg_height) OVER(ORDER BY decade) height_diff
FROM
    wh
WHERE
    decade IS NOT NULL;
```

< Result >

Result Grid | Filter Rows:

decade	weight_diff	height_diff
1870	NULL	NULL
1880	5.87	0.74
1890	1.32	0.41
1900	3.75	0.54
1910	-2.21	0.25
1920	1.23	0.13
1930	5.71	0.73
1940	3.54	0.41
1950	2.06	0.42
1960	1.46	0.41
1970	0.18	0.19

< Query >

udemy

Certificate no: UC-45c3e16e-84b0-43d0-a8c2-8bf4d61d92b7
Certificate url: ude.my/UC-45c3e16e-84b0-43d0-a8c2-8bf4d61d92b7
Reference Number: 0004

CERTIFICATE OF COMPLETION

SQL for Data Analysis: Advanced SQL Querying Techniques

Instructors **Maven Analytics • 1,500,000 Learners, Alice Zhao**

Andrian Wijaya

Date **Aug. 16, 2025**
Length **8.5 total hours**

Link
Certification

Profile;

Andrian Wijaya

enhance your skills AND
gain you experience;

Access this project in this link <github>

Thank You