Write 20 SQL queries to extract insights from the data.

1. Top 10 batsman by total runs in Test matches.

import sqlite3

# Connect to the database

conn = sqlite3.connect("cricsheet\_database.db")

cursor = conn.cursor()

# SQL query to find top 10 batsmen by total runs

query = """

SELECT

    Batter AS Batsman,

    SUM("Batter\_Runs") AS Total\_Runs

FROM

    test\_matches

GROUP BY

    Batter

ORDER BY

    Total\_Runs DESC

LIMIT 10;

"""

# Execute the query

cursor.execute(query)

results = cursor.fetchall()

# Display the results

print("Top 10 Batsmen by Total Runs in Test Matches:")

for row in results:

    print(f"{row[0]}: {row[1]} runs")

Output:



1. Top 10 batsman by total runs in ODI matches.

import sqlite3

# Connect to the database

conn = sqlite3.connect("cricsheet\_database.db")

cursor = conn.cursor()  # Create a cursor object

# Define the query

query\_odi = """

SELECT

    Batter AS Batsman,

    SUM(Batter\_Runs) AS Total\_Runs

FROM odi\_matches

GROUP BY Batsman

ORDER BY Total\_Runs DESC

LIMIT 10;

"""

try:

    # Execute the query

    cursor.execute(query\_odi)

    results = cursor.fetchall()

    # Display the results

    print("Top 10 Batsmen by Total Runs in ODIs:")

    for row in results:

        print(f"Batsman: {row[0]}, Total Runs: {row[1]}")

except sqlite3.Error as e:

    print("An error occurred:", e)

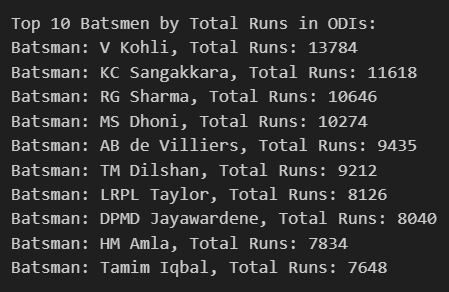
finally:

    # Close the cursor and connection

    cursor.close()

    conn.close()

Output:



1. Top 10 batsman by total runs in T20 matches.

import sqlite3

# Connect to the database

conn = sqlite3.connect("cricsheet\_database.db")

cursor = conn.cursor()  # Create a cursor object

# Define the query

query\_odi = """

SELECT

    Batter AS Batsman,

    SUM(Batter\_Runs) AS Total\_Runs

FROM t20\_matches

GROUP BY Batsman

ORDER BY Total\_Runs DESC

LIMIT 10;

"""

try:

    # Execute the query

    cursor.execute(query\_odi)

    results = cursor.fetchall()

    # Display the results

    print("Top 10 Batsmen by Total Runs in T20s:")

    for row in results:

        print(f"Batsman: {row[0]}, Total Runs: {row[1]}")

except sqlite3.Error as e:

    print("An error occurred:", e)

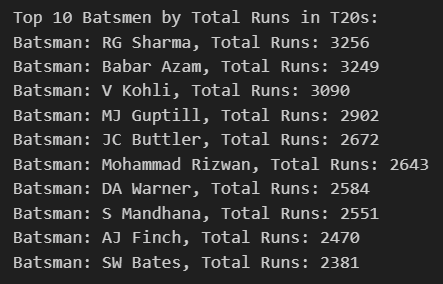
finally:

    # Close the cursor and connection

    cursor.close()

    conn.close()

Output:



1. Find the **top 10 bowlers** by the number of wickets in Test matches.

import sqlite3

# Connect to the database

connection = sqlite3.connect('cricsheet\_database.db')

cursor = connection.cursor()

# SQL query to find the highest wicket-taker

query = """

SELECT

    Bowler AS Highest\_Wicket\_Taker,

    COUNT(Player\_Out) AS Total\_Wickets

FROM

    test\_matches

WHERE

    Bowler IS NOT NULL

    AND Bowler != ''

    AND Player\_Out IS NOT NULL

    AND Player\_Out != '0'

GROUP BY

    Bowler

ORDER BY

    Total\_Wickets DESC

LIMIT 1;

"""

try:

    # Execute the query

    cursor.execute(query)

    result = cursor.fetchone()

    if result:

        print(f"Highest Wicket Taker: {result[0]}, Total Wickets: {result[1]}")

    else:

        print("No data found for highest wicket-taker.")

except sqlite3.Error as e:

    print(f"An error occurred: {e}")

finally:

    # Close the database connection

    connection.close()

Output:



1. Find the **top 10 bowlers** by the number of wickets in ODI matches.

import sqlite3

# Connect to the database

connection = sqlite3.connect('cricsheet\_database.db')

cursor = connection.cursor()

# SQL query to find the highest wicket-taker

query = """

SELECT

    Bowler AS Highest\_Wicket\_Taker,

    COUNT(Player\_Out) AS Total\_Wickets

FROM

    odi\_matches

WHERE

    Bowler IS NOT NULL

    AND Bowler != ''

    AND Player\_Out IS NOT NULL

    AND Player\_Out != '0'

GROUP BY

    Bowler

ORDER BY

    Total\_Wickets DESC

LIMIT 1;

"""

try:

    # Execute the query

    cursor.execute(query)

    result = cursor.fetchone()

    if result:

        print(f"Highest Wicket Taker: {result[0]}, Total Wickets: {result[1]}")

    else:

        print("No data found for highest wicket-taker.")

except sqlite3.Error as e:

    print(f"An error occurred: {e}")

finally:

    # Close the database connection

    connection.close()

Output:



As well as in T20 cricket:



1. Top 5 Teams with the Highest Win Percentage in Test Cricket:

import sqlite3

# Connect to the database

connection = sqlite3.connect('cricsheet\_database.db')

cursor = connection.cursor()

# SQL query to find the top 5 teams with the highest win percentage

query = """

SELECT

    Team AS Team\_Name,

    COUNT(CASE WHEN Match\_Winner = Team THEN 1 END) \* 100.0 / COUNT(\*) AS Win\_Percentage

FROM

    test\_matches

GROUP BY

    Team

ORDER BY

    Win\_Percentage DESC

LIMIT 5;

"""

try:

    # Execute the query

    cursor.execute(query)

    results = cursor.fetchall()

    # Display the results

    print("Top 5 Teams with the Highest Win Percentage:")

    for rank, row in enumerate(results, start=1):

        print(f"{rank}. {row[0]} - {row[1]:.2f}%")

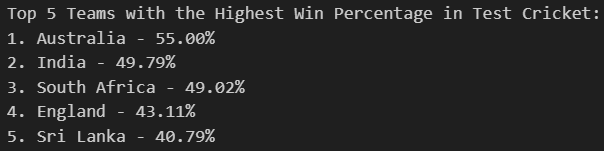
except sqlite3.Error as e:

    print(f"An error occurred: {e}")

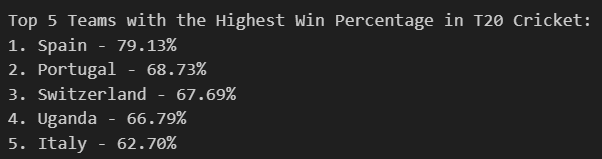
finally:

    # Close the connection

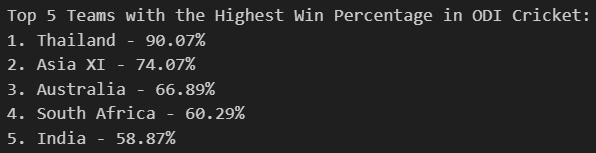
    connection.close()



As well as for T20 Cricket:



For ODI Cricket:



1. How does winning the toss influence the match outcome?

import sqlite3

# Connect to the database

connection = sqlite3.connect('cricsheet\_database.db')

cursor = connection.cursor()

# SQL query to find toss match outcome percentages that add up to 100%

query = """

SELECT

    CASE

        WHEN Toss\_Winner = Match\_Winner THEN 'Toss Winner Wins Match'

        ELSE 'Toss Winner Loses Match'

    END AS Toss\_Match\_Outcome,

    COUNT(\*) AS Match\_Count

FROM

    t20\_matches

GROUP BY

    Toss\_Match\_Outcome;

"""

# Execute the query

cursor.execute(query)

# Fetch and print the results

results = cursor.fetchall()

total\_matches = sum(row[1] for row in results)  # Calculate the total number of matches

# Adjust percentages to ensure they sum to 100%

adjusted\_results = [(row[0], row[1], round((row[1] / total\_matches) \* 100, 2)) for row in results]

# Display the results

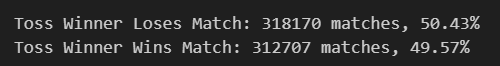
for outcome, count, percentage in adjusted\_results:

    print(f"{outcome}: {count} matches, {percentage}%")

# Close the connection

connection.close()

Output:



1. which bowler gives more extra runs in test cricket?

import sqlite3

# Connect to the database

connection = sqlite3.connect('cricsheet\_database.db')

cursor = connection.cursor()

# SQL query to find the bowler who gave the most extra runs

query = """

SELECT

    Bowler AS Bowler\_Name,

    SUM([Extras\_Runs]) AS Total\_Extras

FROM

    test\_matches

WHERE

    Bowler IS NOT NULL AND Bowler != ''

GROUP BY

    Bowler

ORDER BY

    Total\_Extras DESC

LIMIT 1;

"""

# Execute the query

cursor.execute(query)

# Fetch and display the result

result = cursor.fetchone()

if result:

    print(f"The bowler who gave the most extra runs is {result[0]} with {result[1]} extra runs.")

# Close the connection

connection.close()

Output:



1. Top 10 bowler who gives most extra runs in test cricket.

import sqlite3

# Connect to the database

connection = sqlite3.connect('cricsheet\_database.db')

cursor = connection.cursor()

# SQL query to find the top 10 bowlers who gave the most extra runs

query = """

SELECT

    Bowler AS Bowler\_Name,

    SUM([Extras\_Runs]) AS Total\_Extras

FROM

    test\_matches

WHERE

    Bowler IS NOT NULL AND Bowler != ''

GROUP BY

    Bowler

ORDER BY

    Total\_Extras DESC

LIMIT 10;

"""

# Execute the query

cursor.execute(query)

# Fetch and display the results

results = cursor.fetchall()

print("Top 10 Bowlers with Most Extra Runs:")

for rank, (bowler, total\_extras) in enumerate(results, start=1):

    print(f"{rank}. {bowler} - {total\_extras} extra runs")

# Close the connection

connection.close()

Output:

