<u>INDEX</u>

Program

INTRODUCTION –

```
print('Welcome to Club-70')
import mysql.connector as sq
con=sq.connect(host="localhost",user="root",passwd="1234",database="pro12")
cur=con.cursor()
print('Enter 1 for viewing record of a player')
print('Enter 2 for entering a fresh player')
print("Enter 3 for updating a player's record")
print('Enter 4 for removing a player')
f=int(input('Your choice?  '))
```

The program starts here. The user is asked to enter one of the given numbers in order to edit the databases of the club.

Entering '1' will allow the user to view the record of any player.

Entering '2' will allow the user to enter a new player.

Entering '3' will allow the user to update the record of an existing player.

Entering '4' will allow the user to delete any player's data.

All the data is stored in and accessed from MySQL.

In the following pages, we will see how each of the four functions of this program works.

f==1 –

```
if f==1:
    import pandas as pd
    print('Enter 11 for viewing a batsman')
    print('Enter 12 for viewing a bowler')
    q=int(input('Your choice? '))
```

The user will be further asked to enter numbers.

q==11 –

```python
if q==11:
    j=input("Enter id of the player you want to view. Enter 'x' if you don't know the id ")
    if j=='x':
        n=input("Enter name of the player ")
        qry1="select * from bat where name='%s'" %(n)
        cur.execute(qry1)
        rows=cur.fetchall()
        w=list(rows[0])
        a=pd.Series(w,index=['ID','NAME','NATIONALITY','RUNS','MATCHES','AVERAGE'])
        print(a)
    else:
        qry="select * from bat where id="+j
        cur.execute(qry)
        rows=cur.fetchall()
        w=list(rows[0])
        a=pd.Series(w,index=['ID','NAME','NATIONALITY','RUNS','MATCHES','AVERAGE'])
        print(a)
```

The user will be asked to enter the id of the batsman.

The user can enter 'x' if he/she doesn't know the player's id, provided the player's name is known.

Entering 'x' will ask the user to enter the name and entering the correct name will display all the details of the batsman which are id, name, nationality, runs scored, matches played, average runs per match.

Entering the correct id of the player will also generate all the above details of a player.

Entering wrong id or name will give an error.

The details would be generated in the form of a series.

q==12 –

```python
elif q==12:
    j=input("Enter id of the player you want to view. Enter 'x' if you don't know the id ")
    if j=='x':
        n=input("Enter name of the player ")
        qry1="select * from bowl where name='%s'" %(n)
        cur.execute(qry1)
        rows=cur.fetchall()
        w=list(rows[0])
        a=pd.Series(w,index=['ID','NAME','NATIONALITY','WICKETS','MATCHES','WICKETS PER MATCH'])
        print(a)
    else:
        qry2="select * from bowl where id="+j
        cur.execute(qry2)
        rows=cur.fetchall()
        w=list(rows[0])
        a=pd.Series(w,index=['ID','NAME','NATIONALITY','WICKETS','MATCHES','WICKETS PER MATCH'])
        print(a)
```

The user will be asked to enter the id of the bowler.

The user can enter 'x' if he/she doesn't know the player's id, provided the player's name is known.

Entering 'x' will ask the user to enter the name and entering the correct name will display all the details of the bowler which are id, name, nationality, wickets taken, matches played, wickets per match.

Entering the correct id of the player will also generate all the above details of a player.

Entering wrong id or name will give an error.

The details would be generated in the form of a series.

f==2 –

```python
if f==2:
    h=input("Is he an allrounder( Enter 'y' for yes or 'n' for no )? ")
    if h=='y':
        qr="select max(id) from bat"
        cur.execute(qr)
        r=cur.fetchall()
        r=r[0]
        r=list(r)
        i=r[0]
        i+=1
        n=input("Enter player's name ")
        c=input("Enter his nationality ")
        qry1="insert into bat values(%s,'%s','%s',0,0,NULL)" %(i,n,c)
        cur.execute(qry1)
        qry2="insert into bowl values(%s,'%s','%s',0,0,NULL)" %(i,n,c)
        cur.execute(qry2)
        print('record added')
```

The user is asked whether the new player is an all-rounder.

This is done because, if the player is an all-rounder, then his entry will be created in both the tables (bat & bowl).

This will facilitate the user in performing further operations.

h=='y' –

This means that the player to be added is an all-rounder.

Further, the user would be asked to enter the player's name and nationality.

Once the record gets added, the user will be shown a message informing the same.

h=='n' –

```
elif h=='n':
    print('Enter 21 for entering fresh batsman')
    print('Enter 22 for entering fresh bowler')
    q=int(input('Your choice? '))
    if q==21:
        qr="select max(id) from bat where id not like '___'"
        cur.execute(qr)
        r=cur.fetchall()
        r=r[0]
        r=list(r)
        i=r[0]
        i+=1
        n=input("Enter player's name ")
        c=input("Enter his nationality ")
        qry="insert into bat values(%s,'%s','%s',0,0,NULL)" %(i,n,c)
        cur.execute(qry)
        print('record added')
    elif q==22:
        qr="select max(id) from bowl where id not like '___'"
        cur.execute(qr)
        r=cur.fetchall()
        r=r[0]
        r=list(r)
        i=r[0]
        i+=1
        n=input("Enter player's name ")
        c=input("Enter his nationality ")
        qry="insert into bowl values(%s,'%s','%s',0,0,NULL)" %(i,n,c)
        cur.execute(qry)
        print('record added')
    else:
        print("You didn't follow instructions properly")
else:
    print("You didn't follow instructions properly")
```

Now, the user has to choose whether the new player is a batsman or a bowler.

q==21 –

This means that the player to be added is a batsman.

Further, the user would be asked to enter the player's name and nationality.

Once the record gets added, the user will be shown a message informing the same.

q==22 –

This means that the player to be added is a bowler.

Further, the user would be asked to enter the player's name and nationality.

Once the record gets added, the user will be shown a message informing the same.

f==3 –

```
if f==3:
    h=input("Is he an allrounder( Enter 'y' for yes or 'n' for no )? ")
    if h=='y':
        i=input("Enter id. Enter 'x' if you don't know the id ")
        if i=='x':
            n=input('Enter name of the player ')
            qry1="select * from bat where name='%s'" %(n)
            cur.execute(qry1)
            rows=cur.fetchall()
            w=list(rows[0])
            i=w[0]
            m=w[4]
            m=m+1
            r=int(input('Enter runs scored in that match '))
            R=w[3]
            r=r+R
            av=r/m
            qry2="update bat set runs=%s, matches=%s, average=%s where id=%s" %(r,m,av,i)
            cur.execute(qry2)
            qry3="select * from bowl where name='%s'" %(n)
            cur.execute(qry3)
            rows=cur.fetchall()
            w=list(rows[0])
            m=w[4]
            m=m+1
            r=int(input('Enter wickets taken in that match '))
            R=w[3]
            r=r+R
            av=r/m
            qry="update bowl set wickets=%s, matches=%s, wickets_per_match=%s where id=%s" %(r,m,av,i)
            cur.execute(qry)
            print('record updated')

        else:
            qry1="select * from bat where id="+i
            cur.execute(qry1)
            rows=cur.fetchall()
            w=list(rows[0])
            m=w[4]
            m=m+1
            r=int(input('Enter runs scored in that match '))
            R=w[3]
            r=r+R
            av=r/m
            qry2="update bat set runs=%s, matches=%s, average=%s where id=%s" %(r,m,av,i)
            cur.execute(qry2)
            qry="select * from bowl where id="+i
            cur.execute(qry)
            rows=cur.fetchall()
            w=list(rows[0])
            m=w[4]
            m=m+1
            r=int(input('Enter wickets taken in that match '))
            R=w[3]
            r=r+R
            av=r/m
            qry="update bowl set wickets=%s, matches=%s, wickets_per_match=%s where id=%s" %(r,m,av,i)
            cur.execute(qry)
            print('record updated')
```

The user is further asked if the record to be updated is of an all-rounder.

h=='y' –

This means that the record to be updated is of an all-rounder.

The user will be asked to enter the id of the player.

The user can enter 'x' if he/she doesn't know the player's id, provided the player's name is known.

After entering the correct name/correct id, the user will be further asked to enter the runs scored & wickets taken in the match of which the record is being added.

The record will be updated in both the tables.

Once the record gets updated, the user will be shown a message informing the same.

h=='n' –

```
elif h=='n':
    print('Enter 31 for updating batting record')
    print('Enter 32 for updating bowling record')
    q=int(input('Your choice? '))
    if q==31:
        i=input("Enter id. Enter 'x' if you don't know the id ")
        if i=='x':
            n=input('Enter name of the player ')
            qry1="select * from bat where name='%s'" %(n)
            cur.execute(qry1)
            rows=cur.fetchall()
            w=list(rows[0])
            i=w[0]
            m=w[4]
            m=m+1
            r=int(input('Enter runs scored in that match '))
            R=w[3]
            r=r+R
            av=r/m
            qry2="update bat set runs=%s, matches=%s, average=%s where id=%s" %(r,m,av,i)
            cur.execute(qry2)
            print('record updated')
        else:
            qry="select * from bat where id="+i
            cur.execute(qry)
            rows=cur.fetchall()
            w=list(rows[0])
            m=w[4]
            m=m+1
            r=int(input('Enter runs scored in that match '))
            R=w[3]
            r=r+R
            av=r/m
            qry="update bat set runs=%s, matches=%s, average=%s where id=%s" %(r,m,av,i)
            cur.execute(qry)
            print('record updated')
```

Now, the user has to choose whether to update a batsman's record or a bowler's.

q==31 –

This means that user is updating a batsman's record.

The user will be asked to enter the id of the player.

The user can enter 'x' if he/she doesn't know the player's id, provided the player's name is known.

After entering the correct name/correct id, the user will be further asked to enter the runs scored in the match of which the record is being added.

Once the record gets updated, the user will be shown a message informing the same.

q==32 –

```python
        elif q==32:
            i=input("Enter id. Enter 'x' if you don't know the id ")
            if i=='x':
                n=input('Enter name of the player ')
                qry3="select * from bowl where name='%s'" %(n)
                cur.execute(qry3)
                rows=cur.fetchall()
                w=list(rows[0])
                i=w[0]
                m=w[4]
                m=m+1
                r=int(input('Enter wickets taken in that match '))
                R=w[3]
                r=r+R
                av=r/m
                qry="update bowl set wickets=%s, matches=%s, wickets_per_match=%s where id=%s" %(r,m,av,i)
                cur.execute(qry)
                print('record updated')
            else:
                qry="select * from bowl where id="+i
                cur.execute(qry)
                rows=cur.fetchall()
                w=list(rows[0])
                m=w[4]
                m=m+1
                r=int(input('Enter wickets taken in that match '))
                R=w[3]
                r=r+R
                av=r/m
                qry="update bowl set wickets=%s, matches=%s, wickets_per_match=%s where id=%s" %(r,m,av,i)
                cur.execute(qry)
                print('record updated')
        else:
            print("You didn't follow instructions properly")
    else:
        print("You didn't follow instructions properly")
```

This means that user is updating a bowler's record.

The user will be asked to enter the id of the player.

The user can enter 'x' if he/she doesn't know the player's id, provided the player's name is known.

After entering the correct name/correct id, the user will be further asked to enter the wickets taken in the match of which the record is being added.

Once the record gets updated, the user will be shown a message informing the same.

It must be noted that entering any character which is not specified, will give an error.

Entering any id/name which doesn't exist, will also give an error.

f==4 –

```
if f==4:
    l=input("Is he an allrounder( Enter 'y' for yes or 'n' for no )? ")
    if l=='y':
        i=input("Enter id. Enter 'x' if you don't know the id ")
        if i=='x':
            n=input('Enter name of the player ')
            q1="delete from bat where name='%s'" %(n)
            cur.execute(q1)
            q2="delete from bowl where name='%s'" %(n)
            cur.execute(q2)
            print('record deleted')
        else:
            q1="delete from bat where id=%s" %(i)
            cur.execute(q1)
            q2="delete from bowl where id=%s" %(i)
            cur.execute(q2)
            print('record deleted')
```

The user is further asked if the record to be deleted is of an all-rounder.

l=='y' –

This means the user is deleting an all-rounder's record.

The user will be asked to enter the id of the player.

The user can enter 'x' if he/she doesn't know the player's id, provided the player's name is known.

After entering the correct name/correct id, the record of that player will be deleted.

This record will be deleted from both the tables.

l=='n' –

```
    elif l=='n':
        print("Enter 41 for deleting a batsman's record")
        print("Enter 42 for deleting a bowler's record")
        q=int(input('Your choice? '))
        if q==41:
            i=input("Enter id. Enter 'x' if you don't know the id ")
            if i=='x':
                n=input('Enter name of the player ')
                q1="delete from bat where name='%s'" %(n)
                cur.execute(q1)
                print('record deleted')
            else:
                q1="delete from bat where id=%s" %(i)
                cur.execute(q1)
                print('record deleted')
        elif q==42:
            i=input("Enter id. Enter 'x' if you don't know the id ")
            if i=='x':
                n=input('Enter name of the player ')
                q1="delete from bowl where name='%s'" %(n)
                cur.execute(q1)
                print('record deleted')
            else:
                q2="delete from bowl where id=%s" %(i)
                cur.execute(q2)
                print('record deleted')
        else:
            print("You didn't follow instructions properly")
    else:
        print("You didn't follow instructions properly")
con.commit()
con.close()
```

Now, the user has to choose whether to delete a batsman's record or a bowler's.

q==41 –

This means that user is deleting a batsman's record.

The user will be asked to enter the id of the player.

The user can enter 'x' if he/she doesn't know the player's id, provided the player's name is known.

After entering the correct name/correct id, the record of that player will be deleted.

q==42 –

This means that user is deleting a bowler's record.

The user will be asked to enter the id of the player.

The user can enter 'x' if he/she doesn't know the player's id, provided the player's name is known.

After entering the correct name/correct id, the record of that player will be deleted.

Once the record gets deleted, the user will be shown a message informing the same.

Output

```
Welcome to Club-70
Enter 1 for viewing record of a player
Enter 2 for entering a fresh player
Enter 3 for updating a player's record
Enter 4 for removing a player
```

f==1 –

```
Your choice?  1
Enter 11 for viewing a batsman
Enter 12 for viewing a bowler
Your choice? 11
Enter id of the player you want to view. Enter 'x' if you don't know the id x
Enter name of the player John
ID                    101
NAME                 John
NATIONALITY    New Zealand
RUNS                  418
MATCHES                12
AVERAGE          34.8333
dtype: object
```

It generated the batsman's record taking his name as the input.

```
Your choice?  1
Enter 11 for viewing a batsman
Enter 12 for viewing a bowler
Your choice? 12
Enter id of the player you want to view. Enter 'x' if you don't know the id 2
ID                        2
NAME                 Kailash
NATIONALITY          India
WICKETS                  5
MATCHES                  2
WICKETS PER MATCH      2.5
dtype: object
```

It generated the bowler's record taking his id as the input.

f==2 –

```
Your choice?  2
Is he an allrounder( Enter 'y' for yes or 'n' for no )? n
Enter 21 for entering fresh batsman
Enter 22 for entering fresh bowler
Your choice? 21
Enter player's name Yash
Enter his nationality India
record added
```

Record gets added. Since the record is just added, his runs and matches are 0. We can further update his runs using the update function of this program.

```
Your choice?  2
Is he an allrounder( Enter 'y' for yes or 'n' for no )? n
Enter 21 for entering fresh batsman
Enter 22 for entering fresh bowler
Your choice? 22
Enter player's name Paul
Enter his nationality West Indies
record added
```

The same case follows here. His wickets and matches are 0, which can be further updated.

```
Enter 4 for removing a player
Your choice?  2
Is he an allrounder( Enter 'y' for yes or 'n' for no )? y
Enter player's name Alex
Enter his nationality England
record added
>>>
```

This player is an all-rounder. So, his record gets added in both the tables. He will have his runs, wickets & his matches 0, which can be further updated.

He will have the same id in both the tables.

f==3 –

```
Your choice?  3
Is he an allrounder( Enter 'y' for yes or 'n' for no )? n
Enter 31 for updating batting record
Enter 32 for updating bowling record
Your choice? 31
Enter id. Enter 'x' if you don't know the id 3
Enter runs scored in that match 27
record updated
```

The program took his id as input to let the user update his record.

The user only gave the runs scored in that match and the runs, matches and runs per match data got updated accordingly.

```
Your choice?  3
Is he an allrounder( Enter 'y' for yes or 'n' for no )? n
Enter 31 for updating batting record
Enter 32 for updating bowling record
Your choice? 32
Enter id. Enter 'x' if you don't know the id x
Enter name of the player Phyl
Enter wickets taken in that match 2
record updated
```

The program took his name as input to let the user update his record.

The user only gave the wickets taken in that match and the wickets, matches and wickets per match data got updated accordingly.

```
Enter 4 for removing a player
Your choice?  3
Is he an allrounder( Enter 'y' for yes or 'n' for no )? y
Enter id. Enter 'x' if you don't know the id 101
Enter runs scored in that match 18
Enter wickets taken in that match 1
record updated
```

The program took his id as input to let the user update his record.

This player is all-rounder. Thus, the user was asked to provide information about runs scored and wickets taken in that match.

The program then updated his runs, runs per match, wickets, matches and wickets per match accordingly.

All-rounders have same id in both the tables.

f==4 –

```
Your choice?  4
Is he an allrounder( Enter 'y' for yes or 'n' for no )? n
Enter 41 for deleting a batsman's record
Enter 42 for deleting a bowler's record
Your choice? 41
Enter id. Enter 'x' if you don't know the id x
Enter name of the player Bob
record deleted
```

The program took name as input to let the user delete the record.

```
Your choice?  4
Is he an allrounder( Enter 'y' for yes or 'n' for no )? n
Enter 41 for deleting a batsman's record
Enter 42 for deleting a bowler's record
Your choice? 42
Enter id. Enter 'x' if you don't know the id 2
record deleted
```

The program took id as input to let the user delete the record.

```
Enter 4 for removing a player
Your choice?  4
Is he an allrounder( Enter 'y' for yes or 'n' for no )? y
Enter id. Enter 'x' if you don't know the id x
Enter name of the player Alex
record deleted
```

The program took name as input to let the user delete the record.

This was an all-rounder's record and hence it was deleted from both the tables.

Tables used

Two tables from MYSQL were used in this program.

  1) Bat

```
+------+--------+-------------+------+---------+---------+
| id   | name   | nationality | runs | matches | average |
+------+--------+-------------+------+---------+---------+
|    1 | Upkaar | India       |  519 |      14 | 37.0714 |
|  101 | John   | New Zealand |  436 |      13 | 33.5385 |
|  102 | Puneet | India       |   34 |       2 |      17 |
|    2 | Bob    | England     |   56 |       2 |      28 |
|    3 | Yash   | India       |   27 |       1 |      27 |
|  103 | Alex   | England     |    0 |       0 |    NULL |
+------+--------+-------------+------+---------+---------+
```

  2) Bowl

```
+------+---------+-------------+---------+---------+-------------------+
| id   | name    | nationality | wickets | matches | wickets_per_match |
+------+---------+-------------+---------+---------+-------------------+
|    1 | Phyl    | South Africa|      18 |      14 |           1.28571 |
|  101 | John    | New Zealand |      11 |      13 |          0.846154 |
|  102 | Puneet  | India       |       1 |       2 |               0.5 |
|    2 | Kailash | India       |       5 |       2 |               2.5 |
|    3 | Paul    | West Indies |       0 |       0 |              NULL |
|  103 | Alex    | England     |       0 |       0 |              NULL |
+------+---------+-------------+---------+---------+-------------------+
```

It must be noted that all-rounders have the same id in both the tables.

Bibliography

- Informatics practices by Sumita Arora – Class XI
- Informatics practices by Sumita Arora – Class XII