

Challenge July 2022

Evaluate Team Performance

A solution with DT5GL by Jack Jansonius – 4 July 2022

Problem Statement (from the web site):

Team	Player	Game Date	Efficiency
Mustungs	Brown	4/1/2022	good
	Brown	4/2/2022	better
	Brown	4/3/2022	best
	Robinson	4/1/2022	worst
	Robinson	4/2/2022	better
	Robinson	4/3/2022	best
	Smith	4/1/2022	bad
	Smith	4/2/2022	good
	Smith	4/3/2022	bad
Eagles	Black	4/1/2022	good
	Black	4/2/2022	better
	Black	4/3/2022	best
	White	4/1/2022	worst
	White	4/2/2022	better
	White	4/3/2022	best
	Green	4/1/2022	bad
	Green	4/2/2022	good
	Green	4/3/2022	worst

Your decision model should evaluate performance of different teams based on efficiency of their players. On the left you can see how different players performed during different games. Each player receives 5 points for “best” efficiency, 3 points for “better” efficiency, 2 for “good” efficiency. You need to subtract 2 points for “bad” efficiency and 5 points for “worst” efficiency. Which team got the most points?

Please [submit](#) your solutions using your favorite BR/DM tools.

The table above taken over in the SQLite database tables:

Game:

	Teamid	Playerid	Date	Efficiency
1	1	1	04-01-2022	good
2	1	1	04-02-2022	better
3	1	1	04-03-2022	best
4	1	2	04-01-2022	worst
5	1	2	04-02-2022	better
6	1	2	04-03-2022	best
7	1	3	04-01-2022	bad
8	1	3	04-02-2022	good
9	1	3	04-03-2022	bad
10	2	4	04-01-2022	good
11	2	4	04-02-2022	better
12	2	4	04-03-2022	best
13	2	5	04-01-2022	worst
14	2	5	04-02-2022	better
15	2	5	04-03-2022	best
16	2	6	04-01-2022	bad
17	2	6	04-02-2022	good
18	2	6	04-03-2022	worst

Player:

	Id	Name
1	1	Brown
2	2	Robinson
3	3	Smith
4	4	Black
5	5	White
6	6	Green

Team:

	Id	Name
1	1	Mustungs
2	2	Eagles

and for solution one 1 additional table Summation (can also as a temp table):

	TeamId	TotalPoints
1	1	0
2	2	0

Implementation of the decision model in DT5GL, solution 1:

SQLite_database: "Database/TeamPerformance.sqlite3"

Table 0:

```
If:                                     | 0| 1|
'Next Game to process?'                 | Y| N|
Then:
Action is Process_game                   | X|  |
Action is Games_processed                 |  | X|
# .....
```

Proposition: 'Next Game to process?'
Obtain_instance_from_database_view: Game

Attribute: Efficiency Type: Text
Obtain_value_from_database_view: Game.Efficiency

Attribute: Points Type: Integer
Equals: 5 if Efficiency == "best" \\
 else 3 if Efficiency == "better" \\
 else 2 if Efficiency == "good" \\
 else -2 if Efficiency == "bad" \\
 else -5 if Efficiency == "worst" \\
 else 99999

Database_view: Game
With_attributes:
Teamid, Efficiency
Query:
SELECT Teamid, Efficiency
FROM Game
LIMIT 1 OFFSET %s
With_arguments: Game.auto_index

Database_view: TopScore
With_attributes:
Teamid, Teamname, TotalPoints
Query:
SELECT a.Teamid, b.Name, a.TotalPoints
FROM Summation AS a
INNER JOIN
Team AS b on (a.TeamId = b.Id)
ORDER BY TotalPoints DESC
LIMIT 1
End_Query

GoalAttribute: Action
Repeat_until: Games_processed

Case: Games_processed
Print: "Winner is team %s with %s points!" TopScore.Teamname TopScore.TotalPoints

Case: Process_game
Print: "Efficiency for player is %s so add %s points for team with id: %s "
Game.Efficiency Points Game.Teamid
>SQL: "UPDATE Summation "
-SQL: " SET TotalPoints = TotalPoints + %s " Points
<SQL: " WHERE Teamid = %s " Game.Teamid

Initial_database_setup: make_start_summation
Query:
UPDATE Summation SET TotalPoints = 0
End_Query

Testrun solution 1:

Efficiency for player is good so add 2 points for team with id: 1
Efficiency for player is better so add 3 points for team with id: 1
Efficiency for player is best so add 5 points for team with id: 1
Efficiency for player is worst so add -5 points for team with id: 1
Efficiency for player is better so add 3 points for team with id: 1
Efficiency for player is best so add 5 points for team with id: 1
Efficiency for player is bad so add -2 points for team with id: 1
Efficiency for player is good so add 2 points for team with id: 1
Efficiency for player is bad so add -2 points for team with id: 1
Efficiency for player is good so add 2 points for team with id: 2
Efficiency for player is better so add 3 points for team with id: 2
Efficiency for player is best so add 5 points for team with id: 2
Efficiency for player is worst so add -5 points for team with id: 2
Efficiency for player is better so add 3 points for team with id: 2
Efficiency for player is best so add 5 points for team with id: 2
Efficiency for player is bad so add -2 points for team with id: 2
Efficiency for player is good so add 2 points for team with id: 2
Efficiency for player is worst so add -5 points for team with id: 2
Winner is team Mustangs with 11 points!
Time elapsed: 0:00:01.114105

Implementation of the decision model in DT5GL, solution 2:

SQLite_database: "Database/TeamPerformance.sqlite3"

```
rTable 0:
If:                | 0|
'Get_TopScore'    | Y|
Then:
Action is Display_TopScore | X|
# .....
```

Proposition: 'Get_TopScore'
Obtain_instance_from_database_view: TopScore

```
Database_view: TopScore
With_attributes:
Teamid, Teamname, TotalPoints
Query:
SELECT a.Teamid, b.Name,
       Sum(CASE WHEN a.Efficiency = 'best' THEN 5
              WHEN a.Efficiency = 'better' THEN 3
              WHEN a.Efficiency = 'good' THEN 2
              WHEN a.Efficiency = 'bad' THEN -2
              WHEN a.Efficiency = 'worst' THEN -5
              END) AS TotalPoints
FROM Game AS a
JOIN
Team AS b on (a.TeamId = b.Id)
GROUP BY Teamid
ORDER BY TotalPoints DESC
LIMIT 1
End_Query
```

GoalAttribute: Action

Case: Display_TopScore
Print: "Winner is team %s with %s points!" TopScore.Teamname TopScore.TotalPoints

Testrun solution 2:

Winner is team Mustangs with 11 points!

Time elapsed: 0:00:01.136424