

IF2240 - Basis Data

Pembahasan Praktikum 2

Operasi SQL Lanjut

14 Maret 2024

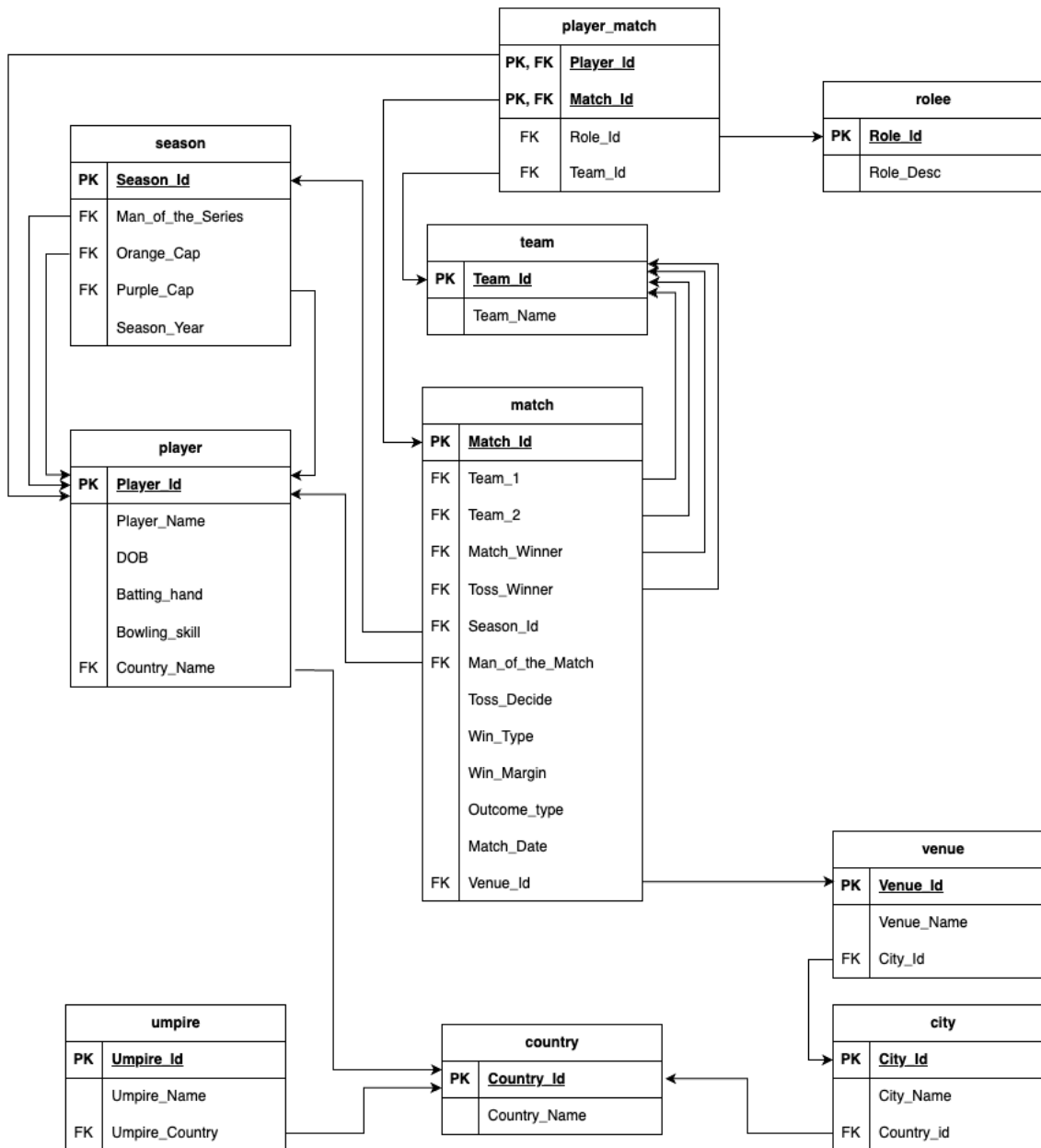


Nama Praktikan

NIM Praktikan

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
2024

Indian Premier League



Legenda

No	Nama Tabel	Keterangan
1	player_match	Keterlibatan seorang pemain pada sebuah pertandingan
2	rolee	Deskripsi peran pemain
3	season	Musim pertandingan
4	team	Tim
5	player	Pemain
6	match	Pertandingan
7	venue	Tempat diadakannya pertandingan
8	city	Kota
9	country	Negara
10	umpire	Wasit

Petunjuk

1. Praktikum ini wajib dikerjakan di dalam Laboratorium Teknik Informatika ITB.
2. Tidak ada restriksi penggunaan keyword pada praktikum ini. Anda dapat menggunakan keyword apapun untuk menjawab soal praktikum.
3. Aturan No 2 tidak berlaku untuk *join* relasi. Anda harus menggunakan *keyword join* yang bersesuaian, tidak boleh melakukan *join* secara manual (seperti pada praktikum 1).

1. Pemain Sepuh

Tampilkan ID pemain yang pernah menjadi *man of the match* pada pertandingan yang diadakan pada bulan Maret sekaligus *man of the series* pada musim pertandingan yang sama. Berikan judul kolom 'Pemain Sepuh' pada hasil pencarian. Pastikan jawaban tidak duplikat.

[HINT] Gunakan backtick (`) untuk nama tabel yang sama dengan salah satu reserved keyword SQL. Gunakan fungsi dasar SQL untuk mengekstrak bagian dari sebuah tanggal seperti MONTH(), YEAR(), WEEK().

Query	<pre>SELECT DISTINCT Man_of_the_series AS `Pemain Sepuh` FROM `match` NATURAL JOIN season WHERE man_of_the_series = man_of_the_match AND MONTH(match_date) = 3;</pre>
SS Query dan Hasil	
<pre>MariaDB [ip1]> SELECT DISTINCT -> Man_of_the_series AS `Pemain Sepuh` -> FROM -> `match` -> NATURAL JOIN season -> WHERE -> man_of_the_series = man_of_the_match AND -> MONTH(match_date) = 3; +-----+ Pemain Sepuh +-----+ 133 +-----+ 1 row in set (0.002 sec)</pre>	

2. Riwayat Jumlah Pemain

Pak Jay ingin melihat jumlah pemain yang pernah bertanding dalam tim. Tampilkan **nama tim** serta **jumlah pemainnya** (nama kolom dibebaskan) lalu urutkan berdasarkan jumlah pemain secara menurun. Pastikan pemain tidak duplikat.

Query	<pre>SELECT team_name, COUNT(DISTINCT player_id) AS player_count FROM player_match NATURAL JOIN team GROUP BY team_id ORDER BY player_count DESC;</pre>
SS Query dan Hasil	
<pre>+-----+-----+ team_name player_count +-----+-----+ Royal Challengers Bangalore 100 Mumbai Indians 89 Delhi Daredevils 88 Kings XI Punjab 87 Rajasthan Royals 86 Kolkata Knight Riders 83 Deccan Chargers 62 Chennai Super Kings 58 Pune Warriors 46 Sunrisers Hyderabad 43 Rising Pune Supergiants 23 Kochi Tuskers Kerala 20 Gujarat Lions 19 +-----+-----+ 13 rows in set (0.011 sec)</pre>	

3. Pemain India OP

Tampilkan 5 baris nama pemain dan tanggal pertandingan terurut membesar untuk pemain yang menjadi *man_of_the_match* dan memiliki *bowling_skill* "Right-arm fast-medium" yang berasal dari India. (Note: Pemain bisa menjadi *man_of_the_match* lebih dari sekali pada *match* yang berbeda)

[HINT] Gunakan backtick (`) untuk nama tabel yang sama dengan salah satu reserved keyword SQL. Gunakan LIMIT untuk membatasi jumlah rows yang ditampilkan

Query	<pre>SELECT Player_Name, Match_Date FROM `match` m INNER JOIN player p ON p.Player_id = m.Man_of_the_Match INNER JOIN country c ON c.Country_id = p.Country_Name WHERE Bowling_skill = "Right-arm fast-medium" AND c.Country_Name = "India" ORDER BY m.Match_Date ASC LIMIT 5;</pre>
SS Query dan Hasil	

```

+-----+-----+
| Player_Name | Match_Date |
+-----+-----+
| MF MaharooF | 2008-04-19 00:00:00 |
| S Sreesanth | 2008-05-05 00:00:00 |
| I Sharma    | 2011-04-27 00:00:00 |
| UT Yadav    | 2012-05-15 00:00:00 |
| UT Yadav    | 2012-05-19 00:00:00 |
+-----+-----+
5 rows in set (0.000 sec)

```

4. Terhebat pada Musimnya

Untuk setiap season, terdapat pemain-pemain hebat yang meraih penghargaan seperti *Man of the Series*, *Orange Cap*, maupun *Purple Cap*. Tampilkan *player_id* dan *player_name* untuk seluruh pemain yang pernah mendapatkan penghargaan *Man of the Series* tetapi tidak pernah mendapatkan penghargaan *Orange Cap*! Terlebih, tampilkan juga *season_id* dimana pemain mendapatkan penghargaan *Man of the Series* tersebut! Anda harus menggunakan OUTER JOIN pada soal ini.

Query	<pre> SELECT p.player_id, p.player_name, s1.season_id FROM player p JOIN season s1 ON p.player_id = s1.man_of_the_series LEFT OUTER JOIN season s2 ON p.player_id = s2.orange_cap WHERE s2.season_id IS NULL; </pre>
-------	--

SS Query dan Hasil

```
MariaDB [prak_basdat_2]> SELECT p.player_id, p.player_name, s1.season_id FROM player p JOIN
season s1 ON p.player_id = s1.man_of_the_series LEFT OUTER JOIN season s2 ON p.player_id =
s2.orange_cap WHERE s2.season_id IS NULL;
```

player_id	player_name	season_id
32	SR Watson	1
32	SR Watson	6
53	AC Gilchrist	2
305	GJ Maxwell	7
315	SP Narine	5
334	AD Russell	8

6 rows in set (0.001 sec)

5. Pemain Beban

Tampilkan 5 nama pemain yang tidak memiliki *bowling skill* (ditandai dengan *bowling skill* bernilai NULL) dengan jumlah pertandingan kalah terbanyak. Berikan pula jumlah pertandingan kalah tersebut

[HINT] Gunakan backtick (`) untuk nama tabel yang sama dengan salah satu reserved keyword SQL. Gunakan keyword LIMIT untuk membatasi jumlah baris keluaran.

Query	<pre> SELECT Player_name, COUNT(*) FROM player_match NATURAL JOIN player NATURAL JOIN `match` WHERE Bowling_skill IS NULL AND ((team_Id = Team_1 AND team_1 <> Match_Winner) OR (team_Id = Team_2 AND team_2 <> Match_Winner)) GROUP BY Player_id, player_name ORDER BY </pre>
-------	--

	COUNT(*) DESC LIMIT 5;
SS Query dan Hasil	
<pre> +-----+-----+ Player_name COUNT(*) +-----+-----+ KD Karthik 62 PA Patel 52 NV Ojha 48 WP Saha 47 SS Tiwary 35 +-----+-----+ 5 rows in set (0.005 sec) </pre>	

6. Venue Terpopuler

Anak pak Jay ingin mengetahui **5 venue** terpopuler di Indian Premier League, yaitu yang paling sering menyelenggarakan pertandingan. Tampilkan **nama venue**, banyaknya pertandingan yang diselenggara (kolom diberi nama **total_match**), dan **nama negara** tempat *venue*. Urutkan berdasarkan *total_match*.

[HINT] Gunakan backtick (`) untuk nama tabel yang sama dengan salah satu reserved keyword SQL. Gunakan keyword LIMIT untuk membatasi jumlah baris keluaran.

Query	<pre> SELECT v.venue_name, COUNT(m.venue_id) AS total_match, co.country_name FROM `match` m JOIN venue v ON m.venue_id = v.venue_id JOIN city c ON c.city_id = v.city_id JOIN country co ON co.country_id = c.country_id LIMIT 5; </pre>
-------	--

	<pre>GROUP BY m.venue_id ORDER BY total_match DESC LIMIT 5;</pre>																		
SS Query dan Hasil																			
<pre>MariaDB [prak2]> SELECT v.venue_name, COUNT(m.venue_id) AS total_match, co.country_name -> FROM `match` m JOIN venue v ON m.venue_id = v.venue_id -> JOIN city c ON c.city_id = v.city_id -> JOIN country co ON co.country_id = c.country_id -> GROUP BY m.venue_id -> ORDER BY total_match DESC -> LIMIT 5;</pre> <table><tr><th>venue_name</th><th>total_match</th><th>country_name</th></tr><tr><td>M Chinnaswamy Stadium</td><td>58</td><td>India</td></tr><tr><td>Eden Gardens</td><td>54</td><td>India</td></tr><tr><td>Feroz Shah Kotla</td><td>53</td><td>India</td></tr><tr><td>Wankhede Stadium</td><td>49</td><td>India</td></tr><tr><td>MA Chidambaram Stadium, Chepauk</td><td>48</td><td>India</td></tr></table> <pre>5 rows in set (0.002 sec)</pre>		venue_name	total_match	country_name	M Chinnaswamy Stadium	58	India	Eden Gardens	54	India	Feroz Shah Kotla	53	India	Wankhede Stadium	49	India	MA Chidambaram Stadium, Chepauk	48	India
venue_name	total_match	country_name																	
M Chinnaswamy Stadium	58	India																	
Eden Gardens	54	India																	
Feroz Shah Kotla	53	India																	
Wankhede Stadium	49	India																	
MA Chidambaram Stadium, Chepauk	48	India																	

7. MVP Royal Challengers Bangalore

Tampilkan nama pemain dan *batting_hand* pemain untuk pemain yang pernah meraih *man_of_the_match* lebih dari 5 kali untuk tim “Royal Challengers Bangalore”

[HINT] Gunakan CONCAT() untuk melakukan konkatenasi string.

Query	<pre> SELECT p.Player_Name, p.Batting_hand FROM `match` m INNER JOIN player p ON p.Player_Id = m.Man_of_the_Match INNER JOIN team t ON t.Team_Id = m.Match_Winner NATURAL JOIN player_match WHERE WHERE t.Team_Name = "Royal Challengers Bangalore" </pre>
-------	--

	GROUP BY p.Player_Id HAVING COUNT(m.Man_of_the_Match) > 5;
SS Query dan Hasil	
<pre> +-----+-----+ Player_Name Batting_hand +-----+-----+ V Kohli Right-hand bat AB de Villiers Right-hand bat CH Gayle Left-hand bat +-----+-----+ 3 rows in set (0.001 sec) </pre>	

8. Pemain Terbaik dari yang Terburuk

Seorang *database administrator*, Julala, menganggap bahwa tim dengan *skill* yang buruk adalah tim yang memiliki akumulasi *win_margin* kurang dari 100. Maka dari itu, ia beranggapan bahwa jika tim yang termasuk ke dalam golongan tersebut memenangi pertandingan, maka hal tersebut akan disebut *anomali*. Tampilkan atribut pemain yakni ***player_id***, ***player_name***, dan jumlah pemain tersebut menjadi ***man_of_the_match*** (tampilkan sebagai ***count_motm***) untuk pemain yang menjadi *man_of_the_match* dari *match* yang dianggap *anomali*. Urutkan hasil berdasarkan *player_id* secara *ascending*. Anda diwajibkan menerapkan *subquery* untuk persoalan ini. *Keyword subquery* yang diperbolehkan adalah IN (tidak boleh menggunakan *keyword subquery* lain).

[HINT] Gunakan AS untuk melakukan penamaan ulang atribut yang ditampilkan.

Query	<pre> SELECT p.player_id, p.player_name, COUNT(man_of_the_match) AS count_motm FROM player p, `match` m WHERE p.player_id = m.man_of_the_match AND m.match_winner IN (SELECT team_id FROM team t JOIN `match` m ON t.team_id = m.match_winner GROUP BY t.team_id HAVING SUM(win_margin) < 100) GROUP BY p.player_id ORDER BY p.player_id ASC; </pre>
-------	--

SS Query dan Hasil

```

MariaDB [prak_basdat_2]> SELECT p.player_id, p.player_name, count(man_of_the_match) as count_motm FROM player p,
`match` m WHERE p.player_id = m.man_of_the_match AND m.match_winner IN (SELECT team_id FROM team t JOIN `match`
m ON t.team_id = m.match_winner GROUP BY t.team_id HAVING SUM(win_margin) < 100) GROUP BY p.player_id ORDER BY
p.player_id ASC;
+-----+-----+-----+
| player_id | player_name | count_motm |
+-----+-----+-----+
| 2 | BB McCullum | 2 |
| 8 | V Kohli | 1 |
| 14 | P Kumar | 1 |
| 20 | MS Dhoni | 1 |
| 21 | SK Raina | 1 |
| 64 | DPMD Jayawardene | 1 |
| 85 | AM Rahane | 2 |
| 104 | BJ Hodge | 2 |
| 106 | AB Dinda | 2 |
| 147 | DR Smith | 2 |
| 254 | AJ Finch | 3 |
| 294 | P Parameswaran | 1 |
| 350 | CH Morris | 1 |
+-----+-----+-----+
13 rows in set (0.001 sec)

```

9. MyHand Menyerang Tim

Setelah puas melakukan phising pada pemilik waralaba, MyHand ingin menemukan nama dari tim yang pernah menang dalam pertandingan pada tahun 2009 dan menang *toss* pada tahun 2016 untuk dikirimkan sebuah paket. Ia juga

ingin memastikan bahwa tim yang terpilih, pada tahun 2009, *man of the match*-nya harus berasal dari negara yang diawali huruf A. Bantulah MyHand menemukan nama-nama tim yang sesuai dengan keperluannya tanpa duplikat! Anda diwajibkan menerapkan *subquery* untuk persoalan ini. *Keyword subquery* yang diperbolehkan adalah EXISTS (tidak boleh menggunakan *keyword subquery* lain)

HINT: Ada beberapa nama tabel yang merupakan reserved keywords SQL. Gunakan backtick (`) untuk membungkus nama tabel tersebut jika digunakan. Gunakan fungsi bawaan SQL untuk mendapatkan bagian dari sebuah data dengan tipe data timestamp.

Query	<pre> SELECT DISTINCT t.Team_Name FROM team t INNER JOIN `match` m ON t.Team_Id = m.Match_Winner INNER JOIN player p ON m.Man_of_the_match = p.Player_Id INNER JOIN country c ON c.Country_Id = p.Country_Name WHERE YEAR(m.Match_Date) = 2009 AND c.Country_Name LIKE 'A%' AND EXISTS (SELECT * FROM `match` WHERE YEAR(Match_Date) = 2016 AND Toss_Winner = t.Team_Id); </pre>
SS Query dan Hasil	

```

+-----+
| Team_Name |
+-----+
| Kolkata Knight Riders |
| Kings XI Punjab |
| Delhi Daredevils |
+-----+
3 rows in set (0.001 sec)

```

10. Pemain Baru

Coach Didot ingin mencari pemain baru untuk timnya. Pemain-pemain yang ingin ia cari pernah bermain untuk suatu tim yang paling sering bertanding sebagai team_1 di Sharjah Cricket Stadium. Bantulah Coach Didot untuk menemukan nama-nama pemain tersebut dengan catatan pemain-pemain tersebut harus memiliki bowling_skill = Legbreak dan batting_hand = Left-hand bat.

Anda diwajibkan menerapkan *subquery* untuk persoalan ini. *Keyword subquery* yang diperbolehkan adalah WITH (tidak boleh menggunakan *keyword* lain)

[HINT]: *Gunakan backtick (`) untuk nama tabel yang sama dengan salah satu reserved keyword SQL.*

Query	<pre> WITH teams AS (SELECT team_1 as team_id, COUNT(*) AS num_of_appearance FROM `match` NATURAL INNER JOIN venue WHERE venue_name = 'Sharjah Cricket Stadium' GROUP BY team_id ORDER BY num_of_appearance DESC </pre>
-------	--

```
        LIMIT 1
    )

SELECT DISTINCT player_name
FROM
    teams NATURAL INNER JOIN
    player_match NATURAL INNER JOIN
    player
WHERE
    batting_hand = 'Left-hand bat' AND
    bowling_skill = 'Legbreak';
```

SS Query dan Hasil

```
+-----+
| player_name |
+-----+
| G Gambhir   |
| DA Warner   |
| CA Ingram   |
| BMAJ Mendis |
+-----+
4 rows in set (0.003 sec)
```