

# Sprawozdanie Projekt 1 cz. 2

## Baza danych turniejów League of Legends

Maxymilian Kowalski

### Procedury

```
--procedura do ustalania zwyciezcy turnieju, zakladamy ze wygrywa ten kto wygrał najwięcej gier do tej pory
DROP PROCEDURE IF EXISTS SetTournamentWinner;

DELIMITER //

CREATE PROCEDURE SetTournamentWinner(IN p_tournament_name VARCHAR(40))
BEGIN
    DECLARE v_tournament_id INT;
    DECLARE v_winner_team_id INT;

    -- Get the tournament ID
    SELECT tournament_id INTO v_tournament_id FROM Tournaments WHERE tournament_name = p_tournament_name;
    IF v_tournament_id IS NULL THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Tournament not found';
    END IF;

    -- Find the team with the most wins
    SELECT winner_id INTO v_winner_team_id
    FROM Matches
    WHERE tournament_id = v_tournament_id
    GROUP BY winner_id
    ORDER BY COUNT(*) DESC
    LIMIT 1;

    -- Update the tournament's winner
    UPDATE Tournaments
    SET winner_id = v_winner_team_id
    WHERE tournament_id = v_tournament_id;
END //

DELIMITER ;
```

Jest to pierwsza z dwóch procedur i jest ona wywoływana przez wyzwalacz zaprezentowany w późniejszej części sprawozdania. Jest ona też przyczyną wielu problemów na drugim szbd - Postgres, ale o tym również później.

Druga procedura jest bardziej złożona od pierwszej, ponieważ zawiera wiele wewnętrznych walidacji, transakcje i obsługę błędów. Aby podejrzeć jej kod proszę sprawdzić plik `create_and_populate.sql` w katalogu projektu, dziękuję. W kwestii dokumentacji działania procedury można spojrzeć na jej wywołania i przykładowe komunikaty:

```
-- CALL AddMatchToTournament(
--     'Summer Cup 2023',
--     '2023-07-15',
--     'Winner Team Name',
--     'Team Alpha',
--     'Team Beta',
--     '00:35:00',
--     '1:1:Mid:12:6/2/4:18500:12500:15000:1:2:3:4:5:6,2:7:Top:13:5/4/7:17500:13000:
-- );
-- ERROR 1644 (45000) at line 493: Winner team is not a playing team

-- CALL AddMatchToTournament(
--     'Summer Cup 2023',
--     '2023-07-15',
--     'Team Alpha',
--     'Team Alpha',
--     'Team Beta',
--     '00:35:00',
--     '1:1:Mid:12:6/2/4:18500:12500:15000:1:2:3:4:5:6,2:7:Top:13:5/4/7:17500:13000:
-- );
-- ERROR 1644 (45000) at line 505: Player is not part of the playing teams
```



```
-- select * from Tournaments;
-- tournament_id  tournament_name  winner_id  prize_pool
-- 1      Worlds 2022      1      2000000
-- 2      MSI 2022      1      400000
-- 3      LCS Summer Split 2022      2      50000
-- 4      LCK Spring Split 2022      3      100000
-- 5      Summer Cup 2023 NULL      NULL

-- select * from Matches;
-- match_id      date_played      winner_id      duration_minutes      tournament_id
-- 1      2023-01-15      1      25:00:00      1      1      2
-- 2      2023-01-16      2      30:32:00      2      3      4
-- 3      2023-01-16      1      28:12:00      3      2      4
-- 4      2023-01-17      3      22:30:00      4      1      3
-- 5      2023-07-15      1      00:35:00      5      1      2

-- select * from Picks where match_id=5;
-- pick_id  match_id      player_id      champion_id      position      level      stats
-- 11      5      1      12      6/2/4      18500      12500      15000      6
-- 12      5      2      7      Top      13      5/4/7      17500      13000      14800      12
-- 13      5      3      10      Jungle      11      3/5/12      16000      14000      13500      18
-- 14      5      4      15      Bot      11      7/3/9      19000      12700      15500      14
-- 15      5      5      21      Support      10      1/4/15      13000      13500      12500      3
-- 16      5      6      2      Mid      14      8/2/3      20000      12000      16000      12
-- 17      5      7      8      Top      15      4/4/6      19500      12800      15700      14
-- 18      5      8      13      Jungle      12      3/6/10      16800      14300      14500      6
-- 19      5      10      18      Support      11      2/5/14      13500      13900      12600      11
-- 20      5      9      20      Bot      13      6/2/8      18400      12600      15400      18
```

Jak widać na załączonych zrzutach ekranu, procedura nie tylko sprawdziła poprawność przekazanych danych ale również dodała turniej którego nie było wcześniej w tabeli, dodała mecz i picki wszystkich graczy do odpowiednich tabel sprawdzając przy okazji czy nie ma przypadkiem duplikatów postaci w żadnej drużynie.

## Wyzwalacze (triggery)

```
--wyzwalacze

DELIMITER //

CREATE TRIGGER UpdateTournamentWinner AFTER INSERT ON Matches
FOR EACH ROW
BEGIN
    DECLARE v_tournament_name VARCHAR(30);
    SELECT tournament_name into v_tournament_name from Tournaments where tournament_id = NEW.tournament_id;
    CALL SetTournamentWinner(v_tournament_name);
END//

DELIMITER ;
```

Udokumentowanie działania tego wyzwalacza zostało przeprowadzone w poprzednim punkcie - zmieniające się id zwycięzcy turnieju po wprowadzaniu nowych meczy.

## Funkcje

```
--funkcje
DROP FUNCTION IF EXISTS CalculateTeamWinRate;
DELIMITER //

CREATE FUNCTION CalculateTeamWinRate(p_team_name VARCHAR(30)) RETURNS DECIMAL(5,2)
BEGIN
    DECLARE total_matches INT;
    DECLARE total_wins INT;
    DECLARE v_team_id INT;
    DECLARE win_rate DECIMAL(5,2);

    SELECT team_id into v_team_id FROM Teams WHERE team_name = p_team_name;

    SELECT COUNT(*) INTO total_matches
    FROM Matches
    WHERE team1_id = v_team_id OR team2_id = v_team_id;

    SELECT COUNT(*) INTO total_wins
    FROM Matches
    WHERE winner_id = v_team_id;

    IF total_matches > 0 THEN
        SET win_rate = (total_wins / total_matches) * 100;
    ELSE
        SET win_rate = 0;
    END IF;

    RETURN win_rate;
END //

DELIMITER ;

-- SELECT CalculateTeamWinRate('DWG KIA');
-- SELECT CalculateTeamWinRate('Team Liquid');
-- CalculateTeamWinRate('DWG KIA')
-- 60.00
-- CalculateTeamWinRate('Team Liquid')
-- 60.00
```

## Kopia zapasowa bazy danych MariaDB

Aby wykonać kopię zapasową wywołujemy komendę:

```
mariadb-dump -u root -p league > db_backup.sql
```

Niestety ta komenda skopiuje jedynie tabele, klucze i indeksy, a pominie procedury, funkcje i wyzwalacze. Aby wykonać pełną kopię zapasową, uwzględniającą te elementy, należy wywołać komendę:

```
mariadb-dump -u root -p --routines --triggers league > db_backup.sql
```

Następnie dropujemy całą bazę danych i przywracamy ją z backupu:

```
> mariadb -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 365
Server version: 11.1.3-MariaDB Arch Linux

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> drop database league;
Query OK, 11 rows affected (0.094 sec)

MariaDB [(none)]> create database league;
Query OK, 1 row affected (0.001 sec)

MariaDB [(none)]> Bye
```

I wywołujemy testy sprawdzające poprawność (kod dostępny w archiwum projektu):

```
> mariadb -u root -p league < project1/db_backup.sql
Enter password:

> mariadb -u root -p league < project1/only_function_calls.sql
Enter password:
winner_id
1
winner_id
2
winner_id
2
tournament_id  tournament_name winner_id  prize_pool
1 Worlds 2022 1 2000000
2 MSI 2022 1 400000
3 LCS Summer Split 2022 2 50000
4 LCK Spring Split 2022 3 100000
5 Summer Cup 2023 2 NULL
match_id date_played winner_id duration_minutes tournament_id team1_id team2_id
1 2023-01-15 1 25:00:00 1 1 2
2 2023-01-16 2 30:32:00 2 3 4
3 2023-01-16 1 28:12:00 3 2 4
4 2023-01-17 3 22:30:00 4 1 3
5 2023-07-15 1 00:35:00 5 1 2
6 2023-07-16 2 00:35:00 5 1 2
7 2023-07-17 2 00:35:00 5 1 2
8 2023-07-15 1 00:35:00 5 1 2
9 2023-07-16 2 00:35:00 5 1 2
10 2023-07-17 2 00:35:00 5 1 2
pick_id match_id player_id champion_id position level statistics dealt_damage received_damage earned_gold
item2_id item3_id item4_id item5_id item6_id item6_id
11 5 1 1 Mid 12 6/2/4 18500 12500 15000 6 6 6 6 6 6
12 5 2 7 Top 13 5/4/7 17500 13000 14800 12 12 12 12 12 12
13 5 3 10 Jungle 11 3/5/12 16000 14000 13500 18 18 18 18 18 18
14 5 4 15 Bot 11 7/3/9 19000 12700 15500 14 14 14 14 14 14
15 5 5 21 Support 10 1/4/15 13000 13500 12500 3 3 3 3 3 3
16 5 6 2 Mid 14 8/2/3 20000 12000 16000 12 12 12 12 12 12
17 5 7 8 Top 15 4/4/6 19500 12800 15700 14 14 14 14 14 14
18 5 8 13 Jungle 12 3/6/10 16800 14300 14500 6 6 6 6 6 6
19 5 10 18 Support 11 2/5/14 13500 13900 12600 11 11 11 11 11 11
20 5 9 20 Bot 13 6/2/8 18400 12600 15400 18 18 18 18 18 18
CalculateTeamWinRate('DWG KIA')
50.00
CalculateTeamWinRate('Team Liquid')
62.50
```

## Drugi system zarządzania bazą danych

Drugim systemem, na który przenieśliśmy moją bazę został Postgres, ponieważ jest on otwartoźródłowy. Instalacja i konfiguracja Postgresa na systemie Arch Linux jest banalnie prosta. Wymaga ona od nas wykonania kilku komend w terminalu, które zostały dobrze opisane w poradniku na stronie geeksforgeeks:

<https://www.geeksforgeeks.org/how-to-install-postgresql-on-arch-based-linux-distributions-manjaro/>

# Migracja MariaDB do Postgres

Migracja bazy danych z MariaDB do PostgreSQL wymaga przekształcenia schematów, danych, procedur składowanych i innych funkcji specyficznych dla bazy danych. Istnieje kilka narzędzi, które mogą pomóc w tym procesie, a oto niektóre z najpopularniejszych:

1. **pgloader**: pgloader to popularne narzędzie open-source, które umożliwia ładowanie danych do PostgreSQL z różnych źródeł, w tym z MariaDB. Może przekształcać schemat i dane podczas procesu migracji, co czyni go potężną opcją dla skomplikowanych migracji. Pgloader umożliwia szybkie i efektywne przeniesienie danych, automatycznie zarządzając różnicami w typach danych i strukturach bazy danych.
2. **Pentaho Data Integration (Kettle)**: Pentaho oferuje narzędzie ETL (Extract, Transform, Load) typu open-source, które może być używane do skomplikowanych migracji danych. Posiada graficzny interfejs, który może być łatwiejszy w użyciu dla osób nie znających się na programowaniu. Umożliwia przekształcanie danych podczas ich przenoszenia i może służyć do bardziej złożonych migracji wymagających transformacji danych.
3. **AWS Database Migration Service (DMS)**: Choć głównie używany w ekosystemie AWS, DMS może być wykorzystany do migracji bazy danych z MariaDB do PostgreSQL. DMS obsługuje migracje schematów i danych, a także zapewnia możliwość ciągłej replikacji danych z minimalnym wpływem na wydajność źródłowej bazy danych. Jest to szczególnie przydatne w scenariuszach, gdzie wymagana jest minimalna przerwa w dostępności bazy danych.

Jako narzędzie do migracji wybrałem **pgloader**, użycie w moim przypadku wyglądało następująco:

```
> pgloader mysql://root:admin@0.0.0.0:3306/league pgsql://root@127.0.0.1:5432/league
2023-11-19T22:17:06.006667+01:00 LOG pgloader version "3.6.ef5d846"
2023-11-19T22:17:06.010000+01:00 LOG Data errors in '/tmp/pgloader/'
2023-11-19T22:17:06.033335+01:00 LOG Migrating from #<MYSQL-CONNECTION mysql://root@0.0.0.0:3306/league {1005B9BCA3}>
2023-11-19T22:17:06.033335+01:00 LOG Migrating into #<PGSQL-CONNECTION pgsql://root@127.0.0.1:5432/league {1005D0ECC3}>
2023-11-19T22:17:06.360014+01:00 LOG report summary reset
```

	table name	errors	rows	bytes	
total time					
-----	-----	-----	-----	-----	-----
	fetch meta data	0	54		0.030s
	Create Schemas	0	0		0.000s
	Create SQL Types	0	5		0.003s
	Create tables	0	16		0.020s
	Set Table OIDs	0	8		0.000s
-----	-----	-----	-----	-----	-----
	league.champions	0	25	11.1 kB	0.063s
	league.items	0	22	4.3 kB	0.017s
	league.players	0	20	1.6 kB	0.020s
	league.achievements	0	6	0.2 kB	0.030s
	league.teams	0	4	0.0 kB	0.037s
	league.picks	0	10	0.6 kB	0.000s
	league.matches	0	4	0.1 kB	0.003s
	league.tournaments	0	4	0.1 kB	0.003s
-----	-----	-----	-----	-----	-----
COPY Threads Completion		0	4		0.063s
Create Indexes		0	29		0.163s
Index Build Completion		0	29		0.087s
Reset Sequences		0	8		0.023s
Primary Keys		0	8		0.003s
Create Foreign Keys		0	17		0.007s
Create Triggers		0	0		0.000s
Set Search Path		0	1		0.000s
Install Comments		0	0		0.000s
-----	-----	-----	-----	-----	-----
Total import time	✓		95	18.0 kB	0.347s

## Różnice w dialektach

### 1. Aktualizacja Procedur i Funkcji:

- W MariaDB, aby zaktualizować procedurę lub funkcję, najpierw trzeba ją usunąć, a następnie ponownie utworzyć.
- W PostgreSQL można użyć `CREATE OR REPLACE FUNCTION`, co pozwala na łatwą aktualizację istniejących funkcji.

### 2. Konkatenacja Łańcuchów:

- MariaDB: `SELECT CONCAT(column1, column2) FROM tabela.`
- PostgreSQL: `SELECT column1 || column2 FROM tabela.`

### 3. Obsługa Czasu i Dat:

- Funkcje dotyczące czasu i daty mogą mieć różne nazwy i składnię. Na przykład, `NOW()` w MariaDB odpowiada `CURRENT_TIMESTAMP` w PostgreSQL.

### 4. Zmienne Sesji:

- MariaDB: Ustawienie zmiennej sesji `SET @x = 5.`
- PostgreSQL: Używa `SET` do ustawienia parametrów konfiguracyjnych, ale nie obsługuje zmiennych użytkownika w ten sam sposób.

### 5. Definiowanie Procedur:

- W MariaDB używa się `DELIMITER` do zmiany standardowego zakończenia instrukcji, aby umożliwić definicje procedur z wieloma instrukcjami. W PostgreSQL nie jest to potrzebne, gdyż używa się bloków `$$` do grupowania instrukcji wewnątrz funkcji lub procedur.
- Składnia tworzenia procedur również się różni. W MariaDB procedury są tworzone za pomocą `CREATE PROCEDURE`, podobnie jak w PostgreSQL, ale składnia argumentów i opcji jest różna.

### 6. Autoinkrementacja:

- MariaDB: `AUTO_INCREMENT.`
- PostgreSQL: Używa sekwencji i `SERIAL.`

### 7. Zwracane Typy:

- PostgreSQL pozwala funkcjom zwracać różnorodne typy, w tym typy złożone i zestawy wierszy. W MariaDB funkcje są bardziej ograniczone pod względem zwracanych typów.

## Implementacja w Postgresie

Migracja danych do Postgresa odbyła się za pomocą narzędzia pgloader i została opisana już wcześniej, dlatego teraz zaprezentuję jak zostały zaimplementowane procedury, wyzwalacze i funkcje oraz pokażę wyniki ich działania.



## Procedury

```
CREATE OR REPLACE PROCEDURE SetTournamentWinner(p_tournament_name VARCHAR(40))
LANGUAGE plpgsql AS $$
DECLARE
    v_tournament_id INT;
    v_winner_team_id INT;
BEGIN
    -- Get the tournament ID
    SELECT tournament_id INTO v_tournament_id FROM Tournaments WHERE tournament_name = p_tournament_name;
    IF v_tournament_id IS NULL THEN
        RAISE EXCEPTION 'Tournament not found';
    END IF;

    -- Find the team with the most wins
    SELECT winner_id INTO v_winner_team_id
    FROM Matches
    WHERE tournament_id = v_tournament_id
    GROUP BY winner_id
    ORDER BY COUNT(*) DESC
    LIMIT 1;

    -- Update the tournament's winner
    UPDATE Tournaments
    SET winner_id = v_winner_team_id
    WHERE tournament_id = v_tournament_id;
END;
$$;
```

```
--procedury
CREATE OR REPLACE PROCEDURE _AddMatchToTournament(
    p_tournament_name VARCHAR(40),
    p_date_played DATE,
    p_winner_team_name VARCHAR(30),
    p_team1_name VARCHAR(30),
    p_team2_name VARCHAR(30),
    p_duration_minutes INTERVAL,
    p_player_data TEXT
)
LANGUAGE plpgsql AS $$
DECLARE
    v_tournament_id INT;
    v_winner_team_id INT;
    v_team1_id INT;
    v_team2_id INT;
    v_match_id INT;
    v_player_id INT;
    v_champion_id INT;
    v_position TEXT;
    v_level INT;
    v_statistics VARCHAR(20);
    v_dealt_damage INT;
    v_received_damage INT;
    v_earned_gold INT;
    v_item_ids TEXT;
    v_index INT := 1;
    v_player_data_entry TEXT;
    v_end_index INT;
```

Druga procedura jest znacznie dłuższa więc pokazałem tylko jej początek, resztę można obejrzeć w pliku `translated_procedures.sql`. Procedura zmieniła nazwę, ponieważ powodowała ona problemy z wyzwalaczem:



```

~ /studia_mgr/zbd on master 12 ?2
> psql -U root -d league -f project1/translated_procedures.sql
CREATE PROCEDURE
CREATE PROCEDURE
CREATE PROCEDURE
psql:project1/translated_procedures.sql:253: NOTICE: Value2: 5
psql:project1/translated_procedures.sql:253: ERROR: Tournament not found
CONTEXT: PL/pgSQL function settournamentwinner(character varying) line 9 at RAISE
SQL statement "CALL SetTournamentWinner(NEW.tournament_id::VARCHAR)"

```

Problem ten wynikał z faktu, że turniej był dodawany w trakcie działania procedury, a następnie dodawany mecz sprawdzał czy taki turniej istnieje. W ciele procedury dany mecz istniał, ale zmiany nie zostały zcommitowane więc wyzwalacz myślał, że taki turniej nie istnieje, stąd błąd.

Zamiast tego wyzwalacza została stworzona funkcja opakowująca te dwie funkcje aby aktualizacja zwycięzcy turnieju następowała zawsze po dodaniu meczu, a składnia niczym nie różniła się między naszymi dwoma systemami baz danych:

```

CREATE OR REPLACE PROCEDURE AddMatchToTournament(
    p_tournament_name VARCHAR(40),
    p_date_played DATE,
    p_winner_team_name VARCHAR(30),
    p_team1_name VARCHAR(30),
    p_team2_name VARCHAR(30),
    p_duration_minutes INTERVAL,
    p_player_data TEXT
)
LANGUAGE plpgsql AS $$
BEGIN
    CALL _AddMatchToTournament(
        p_tournament_name,
        p_date_played,
        p_winner_team_name,
        p_team1_name,
        p_team2_name,
        p_duration_minutes,
        p_player_data
    );
    CALL SetTournamentWinner(p_tournament_name);
END;
$$;

```

Walidacja w procedurze w Postgresie:

```
-- CALL AddMatchToTournament(  
--     'Summer Cup 2023',  
--     '2023-07-15',  
--     'Winner Team Name',  
--     'Team Alpha',  
--     'Team Beta',  
--     '00:35:00',  
--     '1:1:Mid:12:6/2/4:18500:12500:15000:1:2:3:4:5:6,2:7:Top:13:5/4/7:17500:13000:14800:7:8:9:10',  
-- );  
-- psql:project1/translated_procedures.sql:227: ERROR: Winner team is not a playing team  
  
-- CALL AddMatchToTournament(  
--     'Summer Cup 2023',  
--     '2023-07-15',  
--     'Team Alpha',  
--     'Team Alpha',  
--     'Team Beta',  
--     '00:35:00',  
--     '1:1:Mid:12:6/2/4:18500:12500:15000:1:2:3:4:5:6,2:7:Top:13:5/4/7:17500:13000:14800:7:8:9:10',  
-- );  
-- psql:project1/translated_procedures.sql:239: ERROR: Player is not part of the playing teams
```

Wyniki działania procedury (takie same jak w MariaDB):

```
CALL AddMatchToTournament(  
    'Summer Cup 2023',  
    '2023-07-15',  
    'DWG KIA',  
    'DWG KIA',  
    'Team Liquid',  
    '00:35:00',  
    '1:1:Mid:12:6/2/4:18500:12500:15000:1:2:3:4:5:6,2:7:Top:13:5/4/7:17500:13000:14800:7:8:9:10',  
);  
  
-- SELECT winner_id FROM Tournaments WHERE tournament_name='Summer Cup 2023';  
-- winner_id  
-- -----  
--          1  
  
CALL AddMatchToTournament(  
    'Summer Cup 2023',  
    '2023-07-16',  
    'Team Liquid',  
    'DWG KIA',  
    'Team Liquid',  
    '00:35:00',  
    '1:1:Mid:12:6/2/4:18500:12500:15000:1:2:3:4:5:6,2:7:Top:13:5/4/7:17500:13000:14800:7:8:9:10',  
);  
  
-- SELECT winner_id FROM Tournaments WHERE tournament_name='Summer Cup 2023';  
-- winner_id  
-- -----  
--          1  
  
CALL AddMatchToTournament(  
    'Summer Cup 2023',  
    '2023-07-17',  
    'Team Liquid',  
    'DWG KIA',  
    'Team Liquid',  
    '00:35:00',  
    '1:1:Mid:12:6/2/4:18500:12500:15000:1:2:3:4:5:6,2:7:Top:13:5/4/7:17500:13000:14800:7:8:9:10',  
);  
  
-- SELECT winner_id FROM Tournaments WHERE tournament_name='Summer Cup 2023';  
-- winner_id  
-- -----  
--          2
```

```
-- SELECT winner_id FROM Tournaments WHERE tournament_name='Summer Cup 2023';
-- winner_id
-- -----
--      2

-- select * from Tournaments;
-- tournament_id | tournament_name | winner_id | prize_pool
-- -----
--      1 | Worlds 2022 | 1 | 2000000
--      2 | MSI 2022 | 1 | 400000
--      3 | LCS Summer Split 2022 | 2 | 50000
--      4 | LCK Spring Split 2022 | 3 | 100000
--      5 | Summer Cup 2023 | 2 |

-- select * from Matches;
-- match_id | date_played | winner_id | duration_minutes | tournament_id | team1_id | team2_id
-- -----
--      1 | 2023-01-15 | 1 | 00:25:00 | 1 | 1 | 2
--      2 | 2023-01-16 | 2 | 00:30:32 | 2 | 3 | 4
--      3 | 2023-01-16 | 1 | 00:28:12 | 3 | 2 | 4
--      4 | 2023-01-17 | 3 | 00:22:30 | 4 | 1 | 3
--      5 | 2023-07-15 | 1 | 00:35:00 | 5 | 1 | 2
--      6 | 2023-07-16 | 2 | 00:35:00 | 5 | 1 | 2
--      7 | 2023-07-17 | 2 | 00:35:00 | 5 | 1 | 2
-- (7 rows)

-- select * from Picks where match_id=5;
-- pick_id | match_id | player_id | champion_id | position | level | statistics | dealt_damage | received_damage | earned
-- -----
--      11 | 5 | 1 | 1 | Mid | 12 | 6/2/4 | 18500 | 12500 |
--      12 | 5 | 2 | 7 | Top | 13 | 5/4/7 | 17500 | 13000 |
--      13 | 5 | 3 | 10 | Jungle | 11 | 3/5/12 | 16000 | 14000 |
--      14 | 5 | 4 | 15 | Bot | 11 | 7/3/9 | 19000 | 12700 |
--      15 | 5 | 5 | 21 | Support | 10 | 1/4/15 | 13000 | 13500 |
--      16 | 5 | 6 | 2 | Mid | 14 | 8/2/3 | 20000 | 12000 |
--      17 | 5 | 7 | 8 | Top | 15 | 4/4/6 | 19500 | 12800 |
--      18 | 5 | 8 | 13 | Jungle | 12 | 3/6/10 | 16800 | 14300 |
--      19 | 5 | 10 | 18 | Support | 11 | 2/5/14 | 13500 | 13900 |
--      20 | 5 | 9 | 20 | Bot | 13 | 6/2/8 | 18400 | 12600 |
-- (10 rows)
```

## Funkcje

```
-- Create the function
CREATE or replace FUNCTION CalculateTeamWinRate(p_team_name VARCHAR(30)) RETURNS DECIMAL(5,2) AS $$
DECLARE
    total_matches INT;
    total_wins INT;
    v_team_id INT;
    win_rate DECIMAL(5,2);
BEGIN
    SELECT team_id INTO v_team_id FROM Teams WHERE team_name = p_team_name;

    SELECT COUNT(*) INTO total_matches
    FROM Matches
    WHERE team1_id = v_team_id OR team2_id = v_team_id;

    SELECT COUNT(*) INTO total_wins
    FROM Matches
    WHERE winner_id = v_team_id;

    IF total_matches > 0 THEN
        win_rate := (total_wins::DECIMAL / total_matches) * 100;
    ELSE
        win_rate := 0;
    END IF;

    RETURN win_rate;
END;
$$ LANGUAGE plpgsql;

-- SELECT CalculateTeamWinRate('DwG KIA');
-- SELECT CalculateTeamWinRate('Team Liquid');
-- calculateteamwinrate
-- -----
--                60.00
-- (1 row)

-- calculateteamwinrate
-- -----
--                60.00
```

## Wyzwalacz

Jak wspominałem wcześniej wyzwalacz został usunięty ze względu na napotkane problemy, ale przed usunięciem prezentował się tak:

```
-- -- Create the trigger

CREATE OR REPLACE FUNCTION trigger_function_UpdateTournamentWinner()
RETURNS TRIGGER AS $$
BEGIN
    CALL SetTournamentWinner(NEW.tournament_id::VARCHAR);
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE OR REPLACE TRIGGER UpdateTournamentWinner
AFTER INSERT ON Matches
FOR EACH ROW
EXECUTE FUNCTION trigger_function_UpdateTournamentWinner();
```

Wyzwalacze w Postgresie nie mogą wywoływać procedur tylko funkcje, w tym celu stworzyłem funkcję wywołującą naszą procedurę i przekazałem ją do wywołania naszemu wyzwalaczowi. Możliwe, że właśnie to jest powodem problemów z transakcjami w procedurze i wyzwalaczu.

## Dokumentacja działania

```
10 ~/studio1_mq/zbd on 0.1 master 12 22 took 65 s
> psql -U root -d league -f project1/translated_procedures.sql
CREATE PROCEDURE
CREATE PROCEDURE
CREATE PROCEDURE
CALL
CALL
CALL
CALL
match_id | date_played | winner_id | duration_minutes | tournament_id | team1_id | team2_id
-----
1 | 2023-01-15 | 1 | 00:25:00 | 1 | 1 | 2
2 | 2023-01-16 | 2 | 00:30:32 | 2 | 3 | 4
3 | 2023-01-16 | 1 | 00:28:12 | 3 | 2 | 4
4 | 2023-01-17 | 3 | 00:22:30 | 4 | 1 | 3
5 | 2023-07-15 | 1 | 00:35:00 | 5 | 1 | 2
6 | 2023-07-16 | 2 | 00:35:00 | 5 | 1 | 2
7 | 2023-07-17 | 2 | 00:35:00 | 5 | 1 | 2
(7 rows)

pick_id | match_id | player_id | champion_id | position | level | statistics | dealt_damage | received_damage | earned_gold | item1_id | item2_id | item3_id | item4_id | item5_id | item6_id
-----
11 | 5 | 1 | 1 | Mid | 12 | 6/2/4 | 18500 | 12500 | 15000 | 1 |  |  |  |  |  |  |
12 | 5 | 2 | 7 | Top | 13 | 5/4/7 | 17500 | 13000 | 14800 | 7 |  |  |  |  |  |  |
13 | 5 | 3 | 10 | Jungle | 11 | 3/5/12 | 16000 | 14000 | 13500 | 13 |  |  |  |  |  |  |
14 | 5 | 4 | 15 | Bot | 11 | 7/3/9 | 19000 | 12700 | 15500 | 19 |  |  |  |  |  |  |
15 | 5 | 5 | 21 | Support | 10 | 1/4/15 | 13000 | 13500 | 12500 | 15 |  |  |  |  |  |  |
16 | 5 | 6 | 2 | Mid | 14 | 8/2/3 | 20000 | 12000 | 16000 | 2 |  |  |  |  |  |  |
17 | 5 | 7 | 8 | Top | 15 | 4/4/6 | 19500 | 12800 | 15700 | 14 |  |  |  |  |  |  |
18 | 5 | 8 | 13 | Jungle | 12 | 3/6/10 | 16800 | 14300 | 14500 | 22 |  |  |  |  |  |  |
19 | 5 | 10 | 18 | Support | 11 | 2/5/14 | 13500 | 13900 | 12600 | 1 |  |  |  |  |  |  |
20 | 5 | 9 | 20 | Bot | 13 | 6/2/8 | 18400 | 12600 | 15400 | 8 |  |  |  |  |  |  |
(10 rows)

CREATE FUNCTION
calculateteamwinrate
-----
60.00
(1 row)

calculateteamwinrate
-----
60.00
(1 row)
```

## Wnioski

Wnioski do poszczególnych punktów zostały rozsiiane po sprawozdaniu więc nie będę ich przepisywał kolejny raz na końcu. Myślę, że głównym wnioskiem do wyciągnięcia z tego projektu jest to, że pomimo znajomości jednego szbd nie możemy zakładać, że znamy je wszystkie oraz, że przejście z jednego na drugi będzie szybkie i przyjemne. Ze względu na różnice w podejściach do transakcji i elementów programowalnych migracja bazy danych może okazać się czasochłonnym zadaniem jeśli naszym celem jest zachować pełną funkcjonalność.