

January 19, 2024

Relacyjno obiektowe bazy danych

Baza ligi sportowej

Adam Bijacik 240634

Joram Mumb Mulaj Kambaj 240358

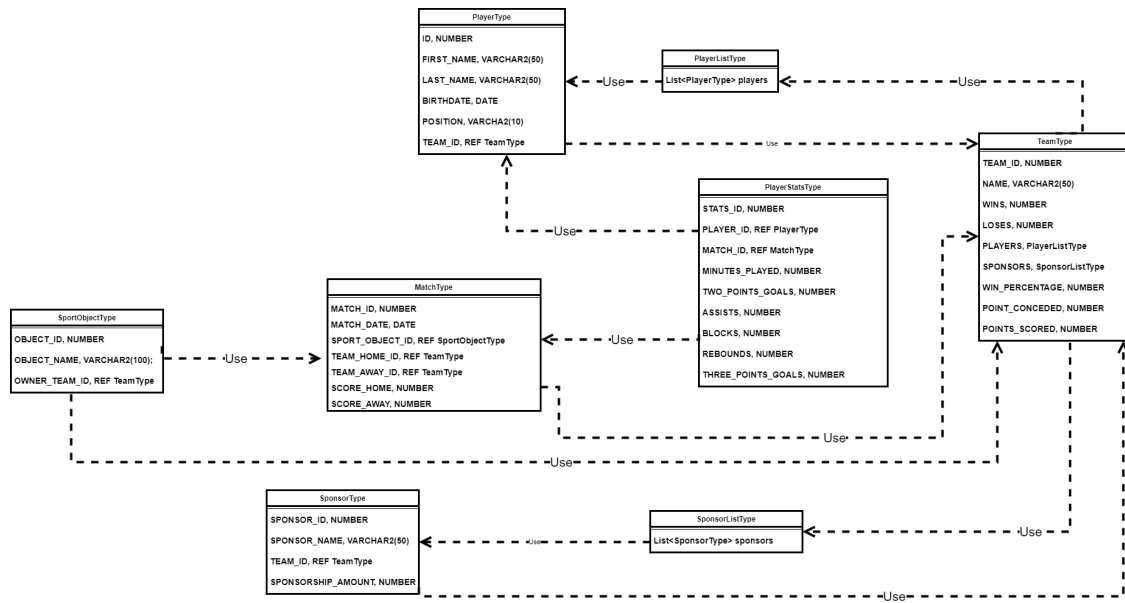
1 Opis projektu

Projekt zaliczeniowy zakłada stworzenie kompleksowej bazy danych dla ligi koszykówki, obejmującej rejestrowanie wyników, przebiegu sezonu oraz zbieranie statystyk indywidualnych graczy. System umożliwi dostęp do szczegółowych informacji o meczach, w tym statystyk poszczególnych zawodników, zarówno z konkretnego meczu, jak i całego sezonu. Dodatkowo, użytkownicy będą mogli sprawdzić aktualny ranking ligi, monitorując hierarchię drużyn w trakcie sezonu.

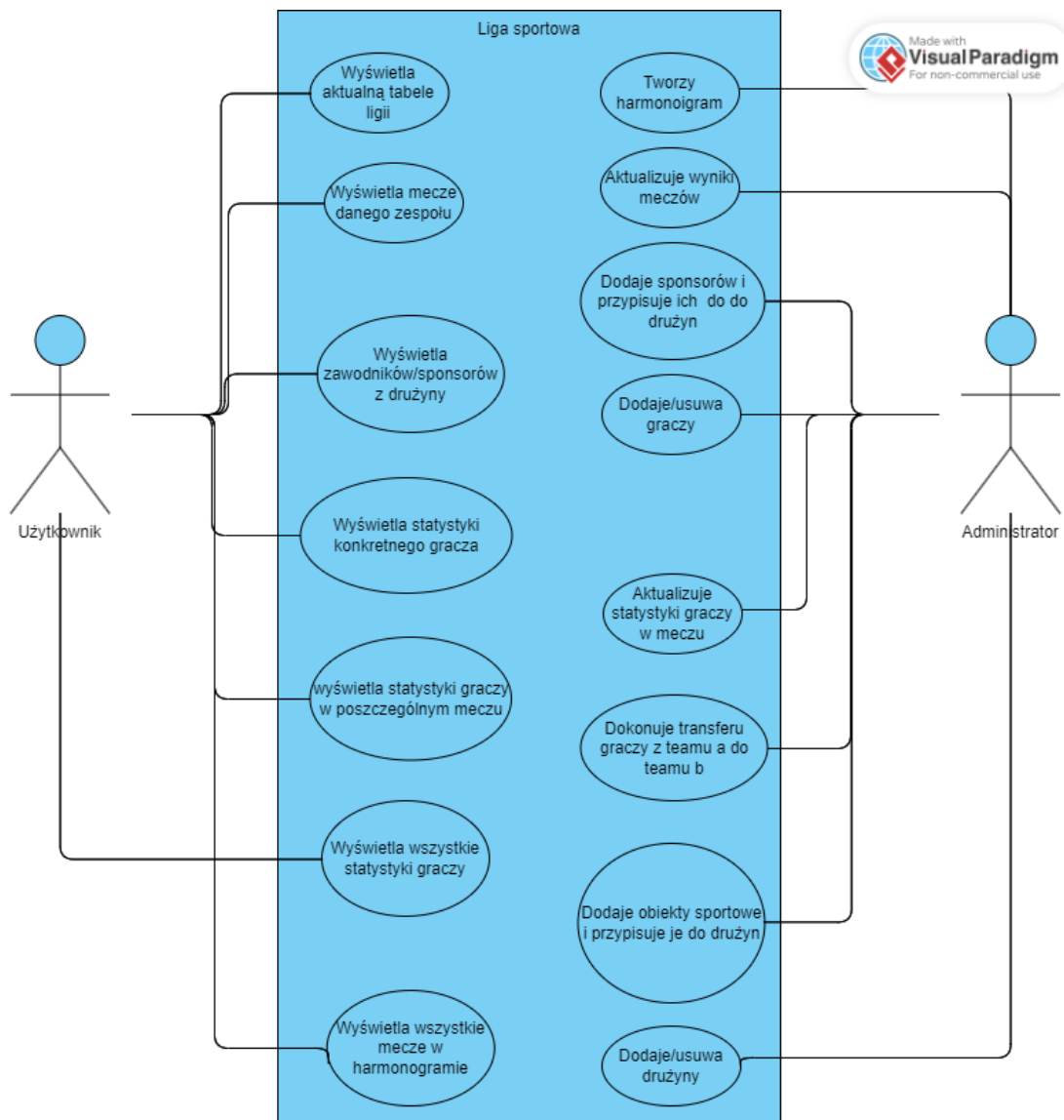
2 Założenia

- Baza obsługuje tylko jedną ligę sportową.
- Baza obsługuje tylko jeden sezon ligi.
- W lidze może być maksymalnie 15 drużyn.
- Drużyna nie może rozgrywać więcej niż jednego spotkania równocześnie.
- Drużyna może zagrać jeden mecz na 24 godziny.
- Mecze będą odbywały się w określonym przedziale czasowym między godziną 12 a 22, przy czym rozpoczęcie każdego meczu musi następować o pełnej godzinie.
- Każda drużyna może mieć maksymalnie 15 graczy.
- Zawodnik grający w lidze musi mieć ukończone 18 lat.
- Zawodnik może grać tylko w jednej drużynie równocześnie.
- Drużyna musi mieć unikalną nazwę.
- Drużyna może zagrać maksymalnie 2 mecze z tą samą drużyną.
- Drużyna może mieć wiele sponsorów, ale sponsor może mieć tylko jedną drużynę.
- Harmonogram jest losowany na początku sezonu, nie ma opcji dodania nowych meczów ani ich przełożenia.
- Każda drużyna musi mieć własny obiekt sportowy.

3 Diagram relacji pomiędzy obiektami



4 Diagram użycia przypadków dla ról



5 Wyjaśnienia typów

TeamType - typ meczu

- team_id NUMBER - identyfikator drużyny,
- name VARCHAR2(50) - nazwa drużyny,
- wins NUMBER - ilość zwycięstw,
- loses NUMBER - ilość porażek,
- players PlayerListType - zawodnicy,
- sponsors SponsorListType - sponsorze

PlayerType - typ zawodnika

- player_id NUMBER - identyfikator zawodnika,
- first_name VARCHAR2(50) - imie,
- last_name VARCHAR2(50) - nazwisko,
- birthdate DATE - data urodzenia,
- position VARCHAR2(50) - pozycja,
- team_id REF TeamType - drużyna

PlayerStatsType - typ statystyk zawodnika w danym meczu

- stats_id NUMBER - identyfikator statystyk,
- player_id REF PlayerType - zawodnik którego dot. statystyki,
- match_id REF MatchType - mecz którego dot. statystyki,
- minutes_played NUMBER - ilość zagranych minut,
- two_points_goals NUMBER - ilość trafionych dwójek w meczu,
- assists NUMBER - ilość asyst w meczu,
- blocks NUMBER - ilość bloków w meczu,
- rebounds NUMBER - ilość zbiórek w meczu,
- three_points_goal NUMBER - ilość trafionych trójek

MatchType - typ meczu

- match_id NUMBER - identyfikator meczu,
- match_date DATE - data w której mecz się odbył,
- sport_object_id REF SportObjectType - obiekt na którym mecz został rozegrany,
- team_home_id REF TeamType - identyfikator gospodarzy,
- team_away_id REF TeamType - identyfikator gości,
- score_home NUMBER - wynik gospodarzy,
- score_away NUMBER - wynik gości

SponsorType - typ sponsora

- sponsor_id NUMBER - identyfikator sponsora,
- sponsor_name VARCHAR2(50) - nazwa sponsora,
- team_id REF TeamType - identyfikator drużyny która sponsoruje,
- sponsorship_amount NUMBER - kwota która drużyna otrzymuje

SportObjectType - typ obiektu sportowego

- object_id NUMBER - identyfikator obiektu,
- object_name VARCHAR2(50) - nazwa obiektu,
- owner_team_id REF TeamType - identyfikator drużyny która gra na tym obiekcie

PlayerListType - tablica referencji do typu PlayerType

SponsorListType - tablica referencji do typu SponsorType

6 Pakiety

Pakiet SCHEDULE_PACKAGE jest odpowiedzialny za generowanie całego harmonogramu rozgrywek.

- GENERATE_SCHEDULE - procedura generująca harmonogram rozgrywek dla aktualnych drużyn.
- CHECK_IF_READY_FOR_NEXT_MATCH - funkcja sprawdzająca, czy dana drużyna może rozgrywać mecz w danym okresie.
- CHECK_IF_DATA_IS_OCCUPIED - funkcja sprawdzająca, czy dany termin jest wolny.
- ADD_MATCH - procedura dodająca spotkanie do tabeli.
- FIND_DATA_FOR_MATCH - funkcja zwracająca prawidłową datę dla meczu.
- PRINT_SCHEDULE - Wyświetla cały harmonogram rozgrywek.
- PRINT_MATCHES_FOR_TEAM - Wyświetla wszystkie mecze danej drużyny.

Pakiet PLAYER_PACKAGE

Pakiet PLAYER_PACKAGE umożliwia operacje na zawodnikach.

- TRANSFER_PLAYER - procedura przenosząca gracza z drużyny A do drużyny B.
- CHECK_IF_PLAYER_EXISTS - funkcja sprawdzająca, czy gracz znajduje się w bazie.
- ADD_PLAYER - funkcja dodająca nowego zawodnika do bazy.
- REMOVE_PLAYER_FROM_TEAM - funkcja usuwająca gracza z drużyny.

Pakiet TEAM_PACKAGE

Pakiet TEAM_PACKAGE jest odpowiedzialny za wszystkie operacje związane z drużynami.

- CHECK_IF_TEAM_EXISTS - funkcja sprawdzająca, czy dana drużyna istnieje w bazie.
- CHECK_IF_SPONSOR_EXISTS - funkcja sprawdzająca, czy dany sponsor istnieje w bazie.
- CHECK_IF_TEAM_HAS_ALREADY_THIS_SPONSOR - funkcja sprawdzająca, czy drużyna nie ma już takiego sponsora.
- ADD_TEAM - procedura dodająca nową drużynę do bazy.
- DELETE_TEAM - procedura usuwająca drużynę z bazy.
- CREATE_SPONSOR - procedura dodająca nowego sponsora do bazy.
- ADD_SPONSOR_TO_THE_TEAM - procedura przypisująca danego sponsora do konkretnej drużyny.
- PRINT_PLAYERS_FROM_TEAM - procedura wyświetlająca wszystkich zawodników z danej drużyny.
- PRINT_SPONSORS_FROM_TEAM - procedura wyświetlająca wszystkich sponsorów z danej drużyny.
- ADD_SPORT_OBJECT - procedura dodająca obiekt drużyny, na którym rozgrywane są mecze.
- CHECK_IF_TEAM_IS_FULL - funkcja sprawdzająca, czy drużyna ma już maksymalną liczbę zawodników.

Pakiet PLAYER_STATS_PACKAGE

Pakiet PLAYER_STATS_PACKAGE umożliwia operacje na statystykach zawodników.

- UPDATE_PLAYER_STATS_PLAYER - aktualizuje statystyki gracza.
- ADD_PLAYER_STATS - dodaje statystyki dla gracza w danym meczu.
- REMOVE_PLAYER_STATS - usuwa statystyki gracza z danego meczu.
- DISPLAY_PLAYER_STATS - wyświetla statystyki danego gracza.
- DISPLAY_ALL_PLAYER_STATS - wyświetla statystyki wszystkich graczy
- DISPLAY_ALL_PLAYER_STATS_FROM_MATCH - wyświetla statystyki wszystkich graczy z danego meczu

Pakiet MATCH_PACKAGE

Pakiet MATCH_PACKAGE jest odpowiedzialny za wszystkie metody związane z meczem.

- UPDATE_TEAM_RANKING - aktualizuje ranking drużyny.
- DISPLAY_TEAM_RANKING - wyświetla aktualny ranking ligii.
- GET_MATCH_ID - pobiera ID meczu dla danej daty.
- UPDATE_SCORE_WITH_MATCH_ID - aktualizuje wynik meczu przyjmując ID meczu jako argument.
- UPDATE_SCORE_WITH_TEAMS - aktualizuje wynik meczu przyjmując nazwę drużyny jako argument.

7 Wyzwalacze

- PLAYER_AGE_CHECK - sprawdza czy dodawany zawodnik jest pełnoletni.
- SPONSORSHIP_AMOUNT_CHECK - sprawdza czy stawka sponsora nie jest mniejsza bądź równa zero.
- TEAM_NAME_CHECK - sprawdza czy nazwa drużyny nie jest pusta.

8 Role

- Użytkownik - Posiada możliwość wyświetlania aktualnego rankingu ligii, statystyk indywidualnych graczy, listy członków danego zespołu oraz informacji dotyczących sponsorów, umożliwiając pełniejsze zrozumienie struktury ligi koszykówki.

```
[ ]: BEGIN
    MATCH_PACKAGE.update_score_with_match_id(0, 115, 110);
    TEAM_PACKAGE.CREATE_SPONSOR( 'GOOGLE', 420000);
    TEAM_PACKAGE.ADD_SPONSOR_TO_THE_TEAM(1,0);
    TEAM_PACKAGE.PRINT_PLAYERS_FROM_TEAM(0);
    PLAYER_PACKAGE.transfer_player(0, 2);
END;
```

- Administrator - Posiada pełne uprawnienia do zarządzania ligą, w tym dodawania, transferowania oraz usuwania graczy, tworzenia nowych drużyn, a także zawierania umów sponsorskich i innych kluczowych działań administracyjnych.

```
[ ]: BEGIN
    MATCH_PACKAGE.update_score_with_match_id(0, 115, 110);
    TEAM_PACKAGE.CREATE_SPONSOR( 'GOOGLE', 420000);
    TEAM_PACKAGE.ADD_SPONSOR_TO_THE_TEAM(1,0);
    TEAM_PACKAGE.PRINT_PLAYERS_FROM_TEAM(0);
    PLAYER_PACKAGE.transfer_player(0, 2);
END;
```