



Mateusz Molik

Nazwa projektu:

SYSTEM BAZODANOWY KLUBY PIŁKARSKIE



Projekt wykonany na zaliczenie przedmiotu Bazy Danych

Dokumentacja techniczna

- Wprowadzenie
- Funkcjonalności
 - 1. Klub piłkarski
 - 2. Właściciel
 - 3. Piłkarz
 - 4. Mecz
 - 5. Statystyki
- Wykorzystanie PL/SQL
 - 1. Wyzwalacze
 - 2. Procedury
 - 3. Funkcje
 - 4. Cursor
 - 5. Pakiety
- Użyte technologie
- Diagram ERD (Entity-Relationship Diagram)

Wprowadzenie

Aplikacja Kluby Piłkarskie to webowa wersja systemu bazodanowego , nieco podobnego do portalu informacyjnego „Łączy nas Piłka”. Aplikacja pozwala przechowywać informacje na temat klubów piłkarskich, piłkarzy, właścicieli oraz rozegranych meczy pomiędzy klubami. Aplikacja została wyposażona również w moduł statystyk. Głównymi atutami aplikacji są praktyczność oraz intuicyjność. Aplikacja w całości prowadzona jest w języku polskim. Na interfejs aplikacji składają się następujące elementy:

1. Strona startowa
2. Moduł statystyk
3. Moduł Klub, prezentuje informacje na temat klubów piłkarskich
4. Moduł Właściciel, prezentuje informacje na temat właścicieli klubów piłkarskich
5. Moduł Właściciel, prezentuje informacje na temat właścicieli klubów piłkarskich
6. Moduł Piłkarze , prezentuje informacje na temat piłkarzy klubów piłkarskich
7. Moduł Mecz , prezentuje informacje na temat meczów rozegranych pomiędzy klubami

Funkcjonalności

1. Klub piłkarski

W module klub piłkarski mamy dostępne funkcjonalności:

- Tworzenie klubu piłkarskiego
- Edycja klubu piłkarskiego
- Wyszukiwanie klubu piłkarskiego po nazwie
- Usunięcie klubu piłkarskiego
- Dodanie właściciela do klubu
- Dodanie właścicieli do klubu
- Usunięcie właściciela/ właścicieli do klubu
- Wyświetlanie szczegółowych informacji na temat klubu

2. Właściciel

W module właściciel mamy dostępne funkcjonalności:

- Tworzenie właściciela
- Edycja właściciela
- Wyszukiwanie właściciela po nazwisko/imie
- Usunięcie właściciela
- Wyświetlanie szczegółowych informacji na właściciela

3. Piłkarz

W module piłkarz mamy dostępne funkcjonalności:

- Tworzenie piłkarza
- Edycja piłkarza
- Wyszukiwanie piłkarza po nazwisko/imie
- Usunięcie piłkarza
- Dodanie osiągnięcia do piłkarza (np. Zawodnik roku/dnia/miesiąca)
- Dodanie przerwy do piłkarza (np. Reprezentacja/Kontuzja/Wolne)
- Wyświetlanie szczegółowych informacji na piłkarza

4. Mecz

W module mecz mamy dostępne funkcjonalności:

- Tworzenie meczu
- Dodanie strzelców bramek w meczu
- Usunięcie meczu
- Dodanie żółtych kartek do piłkarzy
- Dodanie czerwonych kartek do piłkarzy

5. Statystyki

W module statystyki mamy dostępne funkcjonalności:

- Wyświetlanie najbogatszego klub ligi

- Wyświetlanie klubu z klubu z największą liczbą bramek na wyjeździe/u siebie
- Wyświetlanie bramkostrzelnego zawodnika(najwięcej strzelonych bramek)
- Wyświetlanie niebezpiecznego zawodnik (najwięcej czerwonych kartek)
- Wyświetlanie najbardziej opłacanego zawodnika
- Wyświetlanie najmłodszego piłkarza ligi
- Wyświetlanie najpopularniejszego zawodnik
- Wyświetlanie TOP3 właścicieli (osoby poniżej 22 roku życia którzy mają przynajmniej 100000zł gotówki)

Wykorzystanie PL/SQL

1. Wyzwalacze

Klub

TR_DODANIE_WLASCICIELA_DO_KLUBU-Jeżeli dodamy właściciela do klubu ,budżet klubu się zwiększy o gotówkę jaką dysponuje właściciel, analogicznie jeżeli usuniemy to budżet się zmniejszy

```

create or replace TRIGGER TR_DODANIE_WLASCICIELA_DO_KLUBU
BEFORE INSERT ON "KlubWlasciels"

FOR EACH ROW DECLARE
dodatek int;
BEGIN

    select "Gotowka" into dodatek from "Wlasciels" where "WlascicielId"= :new."WlascicielId";

    UPDATE "Klubs"
    SET "Klubs"."Budzet" = "Klubs"."Budzet" + dodatek
    WHERE "Klubs"."KlubId" = :new."KlubId";

END;

create or replace TRIGGER TR_DODANIE_WLASCICIELA_DO_KLUBU_USUN
BEFORE DELETE ON "KlubWlasciels"
FOR EACH ROW DECLARE
dodatek int;
BEGIN

    select "Gotowka" into dodatek from "Wlasciels" where "WlascicielId"= :old."WlascicielId";
    UPDATE "Klubs"
    SET "Klubs"."Budzet" = "Klubs"."Budzet" - dodatek
    WHERE "Klubs"."KlubId" = :old."KlubId";

EXCEPTION
    WHEN OTHERS THEN
        Null;

END;

```

TR_USTAWWYNIK-Jeżeli dodamy mecz do klubu ,budżet klubu się zwiększy o gotówkę .Zależność gotówki zależy od wyniku. W przypadku zwycięstwa klub dostaje 100zł natomiast w przypadku remisu 50zł.

```
create or replace TRIGGER "TR_USTAWWYNIK" BEFORE INSERT ON "Mecze"
FOR EACH ROW BEGIN
    IF :new."Gość_Bramki"=:new."Gospodarz_Bramki" THEN

        :new."Wygrany" :='Remis';
        UPDATE "Klubs"
        SET "Klubs"."Budzet" = "Klubs"."Budzet" + 50
        WHERE "Klubs"."KlubId" = :new."Gospodarz_KlubId" ;
        UPDATE "Klubs"
        SET "Klubs"."Budzet" = "Klubs"."Budzet" + 50
        WHERE "Klubs"."KlubId" = :new."Gość_KlubId" ;

    ELSIF :new."Gość_Bramki" <:new."Gospodarz_Bramki" THEN

        :new."Wygrany" :='Gospodarz';
        UPDATE "Klubs"
        SET "Klubs"."Budzet" = "Klubs"."Budzet" + 100
        WHERE "Klubs"."KlubId" = :new."Gospodarz_KlubId" ;

    ELSE

        :new."Wygrany" :='Gość';
        UPDATE "Klubs"
        SET "Klubs"."Budzet" = "Klubs"."Budzet" + 100
        WHERE "Klubs"."KlubId" = :new."Gość_KlubId" ;

    END IF;
END;
```

Piłkarz

TR_DODAJ__OSIAGNIECIE-Jeżeli dodamy osiągnięcie do piłkarza ,jego pensja się zwiększy o bonus jaki dostanie oraz jego popularność także się zwiększy , analogicznie jeżeli usuniemy to bonus oraz pensja się zmniejszą

```
create or replace TRIGGER TR_DODAJ__OSIAGNIECIE
BEFORE INSERT ON "Osiagniecias"
FOR EACH ROW DECLARE
    dodatek int;
BEGIN

    UPDATE "Pilkarzs"
    SET "Pilkarzs"."Pensja" = "Pilkarzs"."Pensja" + :new."Bonus_Pensja"
    WHERE "Pilkarzs"."PilkarzId" = :new."PilkarzId";

    UPDATE "Pilkarzs"
    SET "Pilkarzs"."Popularnosc" = "Pilkarzs"."Popularnosc" + :new."Bonus_Fani"
    WHERE "Pilkarzs"."PilkarzId" = :new."PilkarzId";

END;

create or replace TRIGGER TR_USUN_OSIAGNIECIE
BEFORE DELETE ON "Osiagniecias"
FOR EACH ROW BEGIN
    UPDATE "Pilkarzs"
    SET "Pilkarzs"."Pensja" = "Pilkarzs"."Pensja" - :old."Bonus_Pensja"
    WHERE "Pilkarzs"."PilkarzId" = :old."PilkarzId";

    UPDATE "Pilkarzs"
    SET "Pilkarzs"."Popularnosc" = "Pilkarzs"."Popularnosc" - :old."Bonus_Fani"
    WHERE "Pilkarzs"."PilkarzId" = :old."PilkarzId";

    EXCEPTION
        WHEN OTHERS THEN
            Null;
END;
```

TR_USTAWPRZERWEPILKARZ-Jeżeli dodamy przerwę do piłkarza ,jego pensja się zwiększy o bonus pomnożony przez ilość dni przerwy oraz jego klub także dostanie bonus do budżetu , analogicznie jeżeli usuniemy to pensja oraz budżet klubu się zmniejszy. Wyzwalacz także automatycznie ustawia koniec przerwy na podstawie początku oraz dni przerwy.

```

create or replace TRIGGER TR_USTAWPRZERWEPILKARZ
BEFORE INSERT ON "Przerwas"
FOR EACH ROW DECLARE
TYPE P_Pilkarz IS RECORD(
"PilkarzId" NUMBER(10,0) ,
    "Imie" NCLOB,
    "Nazwisko" NCLOB,
    "Pensja" NUMBER(10,0) ,
    "Wzrost" NUMBER(10,0) ,
    "Wiek" NUMBER(10,0) ,
    "Popularnosc" NUMBER(10,0) ,
    "KlubId" NUMBER(10,0)
);
pilkarz P_Pilkarz;
pensjabonus pilkarz int;

pensjabonus_klub int;
BEGIN
    select * into pilkarz from "Pilkarzs" where "Pilkarzs"."PilkarzId"= :new."PilkarzId";
    :new."Koniec":=:new."Początek" + :new."Dni";
    pensjabonus_pilkarz := :new."Dni" * :new."Bonus_Zawodnik";
    pensjabonus_klub := :new."Dni" * :new."Bonus_Klub";
    UPDATE "Pilkarzs"
    SET "Pilkarzs"."Pensja" = "Pilkarzs"."Pensja" + pensjabonus_pilkarz
    WHERE "Pilkarzs"."PilkarzId" = :new."PilkarzId";

    UPDATE "Klubs"
    SET "Klubs"."Budzet" = "Klubs"."Budzet" + pensjabonus_klub
    WHERE "Klubs"."KlubId" = pilkarz."KlubId";
END;

```

```

create or replace TRIGGER TR_USTAWPRZERWEPILKARZ_USUN
AFTER DELETE ON "Przerwas"
FOR EACH ROW DECLARE
TYPE P_Pilkarz IS RECORD(
"PilkarzId" NUMBER(10,0) ,
    "Imie" NCLOB,
    "Nazwisko" NCLOB,
    "Pensja" NUMBER(10,0) ,
    "Wzrost" NUMBER(10,0) ,
    "Wiek" NUMBER(10,0) ,
    "Popularnosc" NUMBER(10,0) ,
    "KlubId" NUMBER(10,0)
);
pilkarz P_Pilkarz;
pensjabonus_pilkarz int;
pensjabonus_klub int;
BEGIN
    select * into pilkarz from "Pilkarzs" where "Pilkarzs"."PilkarzId"= :old."PilkarzId";

BEGIN
    select * into pilkarz from "Pilkarzs" where "Pilkarzs"."PilkarzId"= :old."PilkarzId";

    pensjabonus_pilkarz := :old."Dni" * :old."Bonus_Zawodnik";
    pensjabonus_klub := :old."Dni" * :old."Bonus_Klub";

    UPDATE "Pilkarzs"
    SET "Pilkarzs"."Pensja" = "Pilkarzs"."Pensja" - pensjabonus_pilkarz
    WHERE "Pilkarzs"."PilkarzId" = :old."PilkarzId";

    UPDATE "Klubs"
    SET "Klubs"."Budzet" = "Klubs"."Budzet" - pensjabonus_klub
    WHERE "Klubs"."KlubId" = pilkarz."KlubId";

    E/CEPTION
    WHEN OTHERS THEN
        Null;
END;

```

TR_PILKARZ_BRAMKI_KARTKI_DODAJ- Jeżeli dodamy bramkę do piłkarza ,jego pensja się zwiększy o bonus pomnożony przez ilość strzelonych bramek X 5 zł oraz jeżeli dodamy czerwoną kartkę piłkarz zostanie ukarany karę 50 zł ,podwójna żółta 25zł ,pojedyncza żółta 10zł. Analogicznie jeżeli usuniemy mecz w którym grał piłkarz jego bonusy/kary zostaną cofnięte.

```

create or replace TRIGGER TR_PILKARZ_BRAMKI_KARTKI_DODAJ
BEFORE UPDATE ON "MeczPilkarzs"
FOR EACH ROW
DECLARE
TYPE P_Pilkarz IS RECORD(
"PilkarzId" NUMBER(10,0) ,
    "Imie" NCLOB,
    "Nazwisko" NCLOB,
    "Pensja" NUMBER(10,0) ,
    "Wzrost" NUMBER(10,0) ,
    "Wiek" NUMBER(10,0) ,
    "Popularnosc" NUMBER(10,0) ,
    "KlubId" NUMBER(10,0)
);
pilkarz P_Pilkarz;
pensjabonus_bramki int;
karapensja_kartki int;
BEGIN
    select * into pilkarz from "Pilkarzs" where "Pilkarzs"."PilkarzId"= :new."PilkarzId";

    pensjabonus_bramki := :new."Bramki" * 5;
    karapensja_kartki := :old."CzerwonaKartka" * 50 + :old."DoubleZoltaKartka" * 25 + :old."ZoltaKartka" * 10 ;

    UPDATE "Pilkarzs"
    SET "Pilkarzs"."Pensja" = "Pilkarzs"."Pensja" + pensjabonus_bramki - karapensja_kartki
    WHERE "Pilkarzs"."PilkarzId" = :new."PilkarzId";

END;

```


Mecz

TR_USTAWKARTKI – wyzwalacz sprawdza czy użytkownik dobrze zaznaczył ilość kartek.(Jeżeli w meczu zawodnik dostał dwie żółte kartki ,to automatycznie dostał jedną czerwoną oraz jedną żółtą)

```
create or replace TRIGGER TR_USTAWKARTKI
BEFORE INSERT ON "MeczePilkarskie"
for each row BEGIN
  IF :new."DoubleZolteKartka"=1 THEN
    :new."ZolteKartka":=1;
    :new."CzerwonaKartka":=1;
  END IF;
END;
```

TR_USTAWWYNIK – wyzwalacz który częściowo modyfikuje tabelę Mecz . W przypadku wygranej gospodarza ustawia on wartość kolumny Wygrany na „Gospodarz” ,wygrana gości „Gość” natomiast w przypadku remisu „Remis”.
*Zdjęcie wyzwalacza powyżej

2. Procedury

GetTop3Wiek– procedura która ustawia wartość wyjściowa listą właścicieli(kursor) którzy są poniżej 21 roku oraz posiadają ilość gotówki powyżej 100000zł

```
create or replace package body WLASCICIEL
as
  procedure GetTop3Wiek(kursor_wlasciciel out kursor)
  is
  begin
    open kursor_wlasciciel for SELECT
      "A1"."Imie"          "Imie",
      "A1"."Nazwisko"      "Nazwisko",
      "A1"."Gotowka"       "Gotowka"
    FROM
      "MOLIK"."Wlasciciels" "A1"
    WHERE "A1"."Wiek" <=21 and "A1"."Gotowka" >=100000;
  end GetTop3Wiek;
end WLASCICIEL;
```

3.Funkcje

POBIERZBRAMKOSTRZELNYZAWODNIK– funkcja która zwraca ilość strzelonych bramek oraz imie i nazwisko najbardziej bramkostrzelnego zawodnika.

```

create or replace FUNCTION POBIERZBRAMKOSTRZELNYZAWODNIK (bramkiilos OUT long)
RETURN VARCHAR2 AS
sawodnik_imie NCLOB;
sawodnik_nazwisko NCLOB;
calosc VARCHAR2(100);
sawodnikid NUMBER(10,0);
bramki long;
BEGIN
    SELECT "MecePilkarss"."PilkarssId", SUM("MecePilkarss"."Bramki") as "Bramki"
    into sawodnikid,bramki FROM "MecePilkarss" GROUP BY "MecePilkarss"."PilkarssId"
    ORDER BY SUM("MecePilkarss"."Bramki") DESC fetch first 1 rows only;
    bramkiilos := bramki;
    SELECT "Pilkarss"."Imie","Pilkarss"."Nazwisko" into sawodnik_imie,sawodnik_nazwisko from "Pilkarss" where "Pilkarss"."PilkarssId" = sawodnikid;
    sawodnik_nazwisko := CONCAT(' ',sawodnik_nazwisko);
    calosc := CONCAT(sawodnik_imie,sawodnik_nazwisko);
    RETURN calosc;
END POBIERZBRAMKOSTRZELNYZAWODNIK;

```

PobierzNajbogatszyKlub– funkcja która zwraca najbogatsza klub ligi (największy budżet)

```

create or replace function PobierzNajbogatszyKlub (budzet OUT long)
return VARCHAR2
as
klub VARCHAR2(100);
begin
    select "Klubs"."Nazwa","Klubs"."Budzet" into klub,budzet from "Klubs" ORDER BY "Klubs"."Budzet" DESC fetch first 1 rows only;
    return klub;
end;

```

POBIERZNAJLEPIEJOPLACALNYZAWODNIK– funkcja która zwraca najbogatsza opłacalnego zawodnik(największa pensja).

```

create or replace FUNCTION POBIERZNAJLEPIEJOPLACALNYZAWODNIK (pensja OUT long) RETURN
VARCHAR2 AS
sawodnik_imie NCLOB;
sawodnik_nazwisko NCLOB;
calosc VARCHAR2(100);
sawodnikid NUMBER(10,0);
pensjawartosc long;
BEGIN
    SELECT "Pilkarss"."PilkarssId", "Pilkarss"."Pensja"
    into sawodnikid,pensjawartosc FROM "Pilkarss"
    ORDER BY "Pilkarss"."Pensja" DESC fetch first 1 rows only;
    pensja := pensjawartosc;
    SELECT "Pilkarss"."Imie","Pilkarss"."Nazwisko" into sawodnik_imie,sawodnik_nazwisko from "Pilkarss" where "Pilkarss"."PilkarssId" = sawodnikid;
    sawodnik_nazwisko := CONCAT(' ',sawodnik_nazwisko);
    calosc := CONCAT(sawodnik_imie,sawodnik_nazwisko);

    RETURN calosc;
END POBIERZNAJLEPIEJOPLACALNYZAWODNIK;

```

POBIERZNAJLEPIEJOPLACALNYZAWODNIK– funkcja która zwraca najmłodszego zawodnika ligi.

```

create or replace FUNCTION POBIERZNAJMLODSZYZAWODNIK (wiek OUT long) RETURN
VARCHAR2 AS
sawodnik_imie NCLOB;
sawodnik_nazwisko NCLOB;
calosc VARCHAR2(100);
sawodnikid NUMBER(10,0);
lata long;
BEGIN

    SELECT "Pilkarss"."PilkarssId", "Pilkarss"."Wiek"
    into sawodnikid,lata FROM "Pilkarss"
    ORDER BY "Pilkarss"."Wiek" ASC fetch first 1 rows only;

    wiek := lata;

    SELECT "Pilkarss"."Imie","Pilkarss"."Nazwisko" into sawodnik_imie,sawodnik_nazwisko from "Pilkarss" where "Pilkarss"."PilkarssId" = sawodnikid;
    sawodnik_nazwisko := CONCAT(' ',sawodnik_nazwisko);
    calosc := CONCAT(sawodnik_imie,sawodnik_nazwisko);

    RETURN calosc;
END POBIERZNAJMLODSZYZAWODNIK;

```

POBIERZNAJWIECEJBAMEKGOSC– funkcja która zwraca najbardziej skuteczny klub na wyjeździe(największa ilość bramek strzelonych w roli gości)

```

create or replace FUNCTION POBIERZNAJWIECEJBRAKEGOSC (bramkiilosc OUT long) RETURN
VARCHAR2 AS
klub Varchar2(100);
klubid NUMBER;
bramki NUMBER;
BEGIN

    SELECT "Mecss"."Gość_KlubId", SUM("Mecss"."Gość_Bramki") as "Bramki"
    into klubid,bramki FROM "Mecss" GROUP BY "Mecss"."Gość_KlubId"
    ORDER BY SUM("Mecss"."Gość_Bramki") DESC fetch first 1 rows only;

    bramkiilosc := bramki;

    select "Klubs"."Nazwa" into klub from "Klubs" WHERE "Klubs"."KlubId"=klubid;
    RETURN klub;
END POBIERZNAJWIECEJBRAKEGOSC.

```

POBIERZNAJWIECEJBRAKEGOSPODARZKLUB– funkcja która zwraca najbardziej skuteczny klub w roli gospodarza(największa ilość bramek strzelonych w roli gospodarza)

```

create or replace FUNCTION POBIERZNAJWIECEJBRAKEGOSPODARZKLUB (bramkiilosc OUT long) RETURN
VARCHAR2 AS
klub Varchar2(100);
klubid NUMBER;
bramki NUMBER;
BEGIN

    SELECT "Mecss"."Gospodarz_KlubId", SUM("Mecss"."Gospodarz_Bramki") as "Bramki"
    into klubid,bramki FROM "Mecss" GROUP BY "Mecss"."Gospodarz_KlubId"
    ORDER BY SUM("Mecss"."Gospodarz_Bramki") DESC fetch first 1 rows only;

    bramkiilosc := bramki;

    select "Klubs"."Nazwa" into klub from "Klubs" WHERE "Klubs"."KlubId"=klubid;
    RETURN klub;
END POBIERZNAJWIECEJBRAKEGOSPODARZKLUB;

```

POBIERZNIEBEZPIECZNYZAWODNIK– funkcja która zwraca najbardziej niebezpiecznego zawodnika (największa ilość czerwonych kartek)

```

create or replace FUNCTION POBIERZNIEBEZPIECZNYZAWODNIK (kartkiilosc OUT long) RETURN
VARCHAR2 AS
sawodnik_imie NCLOB;
sawodnik_nazwisko NCLOB;
calosc Varchar2(100);
sawodnikid NUMBER(10,0);
kartki long;
BEGIN

    SELECT "Mecspilkarss"."PilkarssId", COUNT(CASE WHEN "Mecspilkarss"."CzerwonaKartka" = 1 THEN 1 END)
    into sawodnikid,kartki FROM "Mecspilkarss" GROUP BY "Mecspilkarss"."PilkarssId"
    ORDER BY COUNT(CASE WHEN "Mecspilkarss"."CzerwonaKartka" = 1 THEN 1 END) DESC fetch first 1 rows only;

    kartkiilosc := kartki;

    SELECT "Pilkarss"."Imie","Pilkarss"."Nazwisko" into sawodnik_imie,sawodnik_nazwisko from "Pilkarss" where "Pilkarss"."PilkarssId" = sawodnikid;
    sawodnik_nazwisko := CONCAT(' ',sawodnik_nazwisko);
    calosc := CONCAT(sawodnik_imie,sawodnik_nazwisko);

    RETURN calosc;
END POBIERZNIEBEZPIECZNYZAWODNIK;

```

POBIERZPOPULARNYZAWODNIK– funkcja która zwraca najbardziej popularnego zawodnika (największa ilość fanów)

```

create or replace FUNCTION POBIERZPOPULARNYZAWODNIK (fani OUT long) RETURN
VARCHAR2 AS
zawodnik_imie NCLOB;
zawodnik_nazwisko NCLOB;
calosc VARCHAR2(100);
zawodnikid NUMBER(10,0);
popularnosc long;
BEGIN

SELECT "Pilkars"."PilkarsId", "Pilkars"."Popularnosc"
into zawodnikid,popularnosc FROM "Pilkars"
ORDER BY "Pilkars"."Popularnosc" DESC fetch first 1 rows only;

fani := popularnosc;

SELECT "Pilkars"."Imie","Pilkars"."Nazwisko" into zawodnik_imie,zawodnik_nazwisko from "Pilkars" where "Pilkars"."PilkarsId" = zawodnikid;
zawodnik_nazwisko := CONCAT(' ',zawodnik_nazwisko);
calosc := CONCAT(zawodnik_imie,zawodnik_nazwisko);

RETURN calosc;

```

4.Kursor

Kursor– REF Cursor tak jak zwykły kursor jest typem danych, który wskazuje na konkretny wiersz w zbiorze wierszy zwróconym przez zapytanie.

```

create or replace PACKAGE WLASCICIEL AS

```

```

type kursor is ref cursor;
procedure GetTop3Wiek ( kursor_wlasciciel out kursor);

```

```

END WLASCICIEL;

```

5.Pakiety

KLUBY– pakiet przechowujący trzy funkcje dotyczące klubów(PobierzNajbogatszyKlub, POBIERZNAJWIECEJBAMEKGOSPODARZKLUB, POBIERZNAJWIECEJBAMEKGOSC).

```

create or replace PACKAGE KLUBY AS

```

```

function PobierzNajbogatszyKlub (budzet OUT long) return Varchar2;
function POBIERZNAJWIECEJBAMEKGOSPODARZKLUB (bramkiilosc OUT long) RETURN VARCHAR2 ;
function POBIERZNAJWIECEJBAMEKGOSC (bramkiilosc OUT long) RETURN VARCHAR2 ;

```

```

END KLUBY;

```

PILKARZE– pakiet przechowujący pięć funkcji dotyczące piłkarzy(POBIERZBRAMKOSTRZELNYZAWODNIK, POBIERZNIEBEZPIECZNYZAWODNIK, POBIERZNAJLEPIEJOPLACALNYZAWODNIK, POBIERZNAJMLODSZYZAWODNIK, POBIERZPOPULARNYZAWODNIK).

```
create or replace PACKAGE PILKARZE AS
```

```
function POBIERZBRAMKOSTRZELNYZAWODNIK (bramkiilosc OUT long) RETURN VARCHAR2;
function POBIERZNIEBEZPIECZNYZAWODNIK (kartkiilosc OUT long) RETURN VARCHAR2;
function POBIERZNAJLEPIEJOPLACALNYZAWODNIK (pensja OUT long) RETURN VARCHAR2;
function POBIERZNAJMLODSZYZAWODNIK (wiek OUT long) RETURN VARCHAR2;
function POBIERZPOPULARNYZAWODNIK (wiek OUT long) RETURN VARCHAR2;
```

```
END PILKARZE;
```

Użyte technologie

- .Net/ASP.NET MVC 5
- Bazy Danych Oracle
- CSS/HTML/ Jquery
- Entity Framework – ORM
- -U

Diagram ERD



OPIS: Diagram ERD zawiera 8 tabel (6 tabel normalnych + 2 tabele łączące). Tabela Klub zawiera informacje podstawowe o klubie (Nazwa klubu, Budżet, Data założenia itp.) Każdy Klub może posiadać właściciela/właścicieli (relacja wiele do wielu, klub – właściciel). Każdy klub także może posiadać piłkarza (relacja jeden do wielu – jeden piłkarz może grać w jednym klubie). Piłkarz jest połączony z trzema tabelami (Przerwa, Osiągnięcie oraz MeczPilkarz). Pierwsze dwa połączenia to relacja jeden do wielu natomiast

ostatnie to połączenie wiele do wielu. Tabela Klub jest także połączona z tabelą mecz relacją jeden do wielu(jeden klub może grać wiele meczy jako gospodarz lub gość).