Łukasz Belka

Grupa D1  
nr indeksu: 156162  
III semestr, 2. rok  
rok akademicki 2023/2024

Akademia Humanistyczno-Ekonomiczna w Łodzi

Baza danych „Ekstrawynik Mecze Piłkarskie”

**Treści rozpatrywanych zagadnień :**

Stworzona przez mnie baza danych to system do przechowania informacji i wyników meczów piłki nożnej. Głównym założeniem jakie mi przyświecało było możliwe dokładne prześledzenie przebiegu spotkania. Nie chciałem spłycać wydarzenia sportowego jakim jest mecz do zaledwie dwóch cyfr wyniku 2:1 albo 0:0. Głównym założeniem było to by przedstawić dane jak najbardziej bazowe czyli w tym przypadku akcje meczowe, np. gole i na podstawie ich przedstawić np. kto wygrał mecz. Bo przecież do określenia wyniku nie potrzebujemy mieć napisanego zwycięzcą jest „XYZ” jesteśmy to w stanie określić na podstawie matematycznego porównania np. 2 > 1. Dlatego też skupiłem się na pojedynczych wydarzeniach mających miejsce w meczu nie zaś na suchych wynikach.

Owszem projekt można by poszerzyć o te dodatkowe dane jak wynik by były łatwiej dostępne, statystki meczu jak posiadanie piłki, xG, asysty, asysty drugiego poziomu i wiele, wiele innych które możemy znaleźć na profesjonalnych stronach analitycznych takich jak <https://ekstrastats.pl/>.

Oczywiste jest że takie właśnie strony jak wspomniany już ekstrastats, czy flashscore są moimi inspiracjami zaś ujecie analiza danych footbolowych to obecnie bardzo prężnie rozwijający się i ciekawy rynek. Jak grzyby po deszczu powstają firmy analityczne które po analizie setek tysięcy, jak nie milionów danych przedstawiają najbogatszym klubą różne plany rozwojowe czy pomysły na usprawnienie gry. Bardzo ważne w tej mechanice jest dostęp do szczegółowych danych właśnie na bardzo drobnym poziomie. Od ilości kontaktów z piłką, przez skoki presingowe skończywszy na strzałach itd. Mój projekt jedynie ślizga się po powierzchni tej tematyki. Głównym ograniczeniem w rozwinięcie go był czas aktualnie posiada on 17 tabel, więc poszerzając go o więcej danych rozrastał by się w niewiarygodne rozmiary. Generalnie sufitu takiego rozwoju prawie nie widać, gdyż tak naprawdę analizujemy tu coś nie zero-jedynkowego, czym jest zachowanie człowieka.

Biorąc pod uwagę te ograniczenia zdecydowałem się przybliżyć projekt bazy danych do znanego serwisu flashcore.pl lub prostu google. Mamy tam zarówno informacje w skali bardziej makro jak to dla jakiego klubu pracuje zawodnik, oraz bardziej mikro jak informacja o nie trafionym karnym w danym meczu. Oczywiście nawet w takiej skali spotkałem się z wieloma problemami w projekcie bazy:

Chodzi o pewną kategoryzacje zachowań. Dla przykładu dość oczywiste wydaje się dołączenie do zespołu osoby trenera lecz po chwili namysłu okażę się, że może on zostać zmieniony w trakcie sezonu, ba w jednym meczu może nie prowadzić zespołu z uwagi na zawieszenie. Pojawia się nam wtedy pewien paradoks który wymaga głębszej analizy, mamy połączenie zespół > trener ale teraz trzeba go zmienić. Ta zamiana musi zostać odnotowana np. w jednym meczu, zaś połączenie trenera z zespołem zmieni całą tabele i wszystkie jej rekordy. Prowadzi nas to do prostego wniosku ze osoba trenera nie może być na sztywno połączona z zespołem. Podobny problem miałem z osoba zawodnika, najprostszy podział to trener oddzielnie od zawodnika, jednak wystarczy spojrzeć parę lat wstecz (bodaj 1-2 lata) gdzie po zwolnieniu Franciszka Smudy w zespole Wieczysta Kraków, rolę trenera przejął piłkarz esk reprezentant kadry narodowej Sławomir Peszko. Znów dochodzimy do konkluzji, że takie sztywne rozdzielnie jest blednę nawet nie przez to że w realnym życiu pewne rzeczy i granice są dość płynne, ale nawet u tak podstawowego faktu, iż po zakończeniu kariery piłkarskiej ktoś może zostać trenerem. Słabo jakby nasza baza w tym momencie kazała nam tworzyć nowy rekord i nie mielibyśmy dostępu do osiągnięć piłkarskich danego trenera.

Takich problemów i dysonansów w takcie tworzenia bazy miałem naprawdę niezliczone ilości, dlatego też niektóre wyboru w konstrukcji mogą wydawać się na pierwszy rzut oka dziwne, ale jeśli przyjrzymy się im bliżej okaże się że były podstawy ku takiemu działaniu. Więcej szczegółów przedstawię w dalszej części podczas analizy każdej tabeli.

**Przyjęte założenia do projektu :**

Stworzenie bazy danych umożliwiającej śledzenie meczów piłkarskich z ich szczegółami: uczestnikami, sędziami, oraz bardziej ogólnymi danymi, ligami i sezonami. Ułatwienie śledzenia akcji podczas meczów oraz powiązań między zawodnikami, zespołami a poszczególnymi wydarzeniami meczowymi. Zapewnienie narzędzia do analizy statystyk związanych z meczami, drużynami, zawodnikami i akcjami w grze. Rejestracja uczestników (zawodnicy, trenerzy, sędziowie) wraz z ich danymi osobowymi, rolami i przynależnością do zespołów. Zarządzanie zespołami piłkarskimi, w tym informacje o stadionach, historii klubu i związkach z ligami. Śledzenie meczów z informacjami takimi jak frekwencja, akcje podczas meczu, wynik, czas trwania i inne statystyki.

W trakcie wykonywania tych założeń przyjąłem kilka schematów mających na celu zapewnienie przejrzystości w bazie:

Każda tabela zaczyna się od „t\_” pozwala nam to łatwo ja odróżnić.

Analogicznie każda kwerenda jest rozpoczęta od „k\_”.

Każdy identyfikator w bazie zaczyna się od id\_.

Każdy klucz obcy nazywałem od „fk\_ „ (foreign key).

Część id w bazie jest w formie varchar z użyciem specyficznych kodów odpowiadających np. danemu krajowi

W przypadku nazw składających się z wielu słów są one rozdzielane za pomocą „\_”, co bardzo ułatwiło konstytuowanie zapytań do już istniejących tabel.

W bazie znajdziemy 17 tabel, połączonych w 26 relacjach, jest to dość spora ilość, niestety tu wchodzimy znów element wcześniej wspomniany, iż ciężko przedstawić niektóre rzeczy z życia codziennego w formie bazy danych. W końcu zarówno zespół jest polski, jak i osoba ma obywatelstwo polskie, czy też liga jaką jest nasza rodzima ekstraklasa jest polska. Z tego właśnie powodu występuje spora liczba połączeń.

Tabele jak i relacje wykonałem za pomocą komend sql:  
oto komenda do stworzenia pierwszej tabeli jednej z najbardziej podstawowych osoby:

CREATE TABLE "t\_osoby" (

"id\_osoby" INT GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,

"imie" varchar(255),

"nazwisko" varchar(255),

"data\_urodzenia" date,

"fk\_id\_kraj" varchar(255)

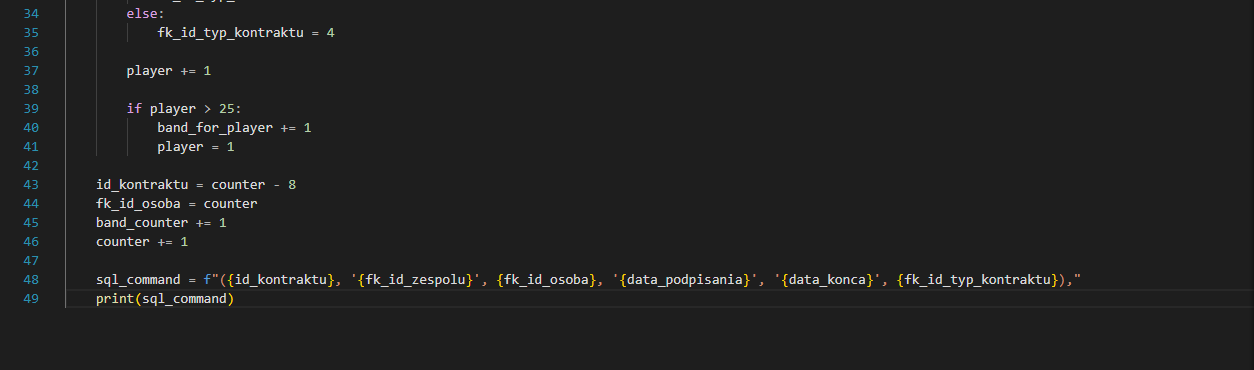
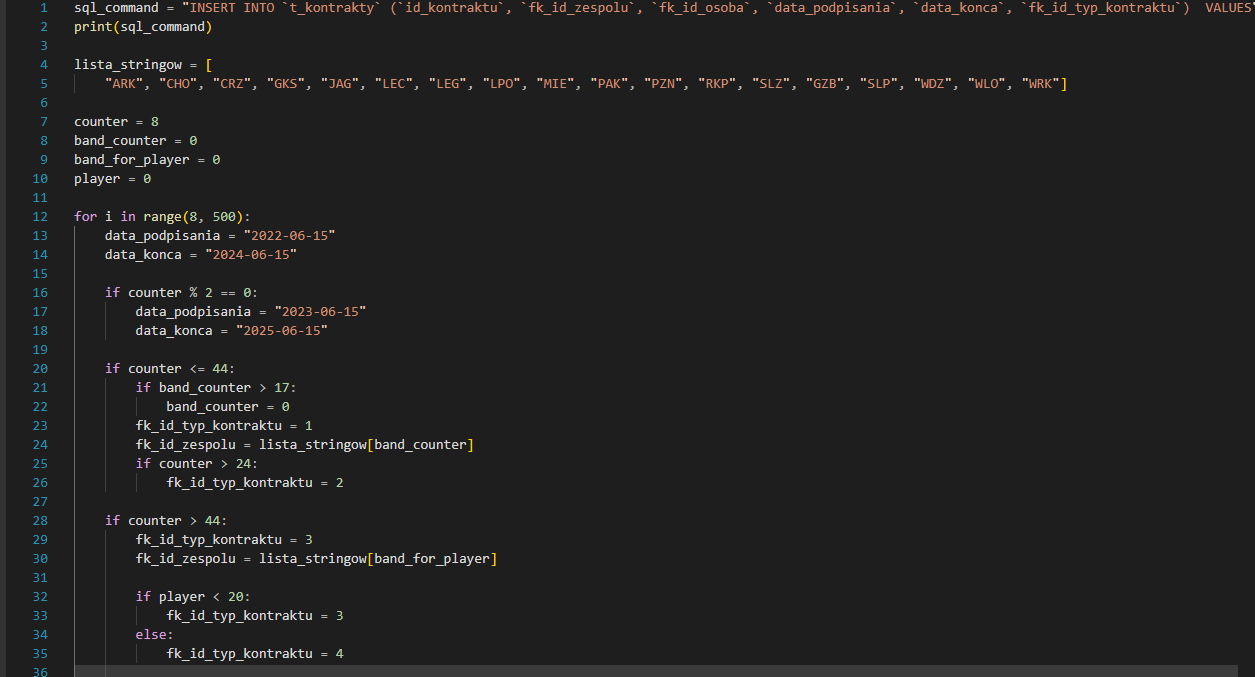
);

kolejna powtarzająca się komendą jest tworzenie relacji

ALTER TABLE "t\_osoby" ADD FOREIGN KEY ("fk\_id\_kraj") REFERENCES "t\_kraje" ("id\_kraj");

Dane w tablach wypełniłem na 3 sposoby:  
1) Ręczne wypełnianie, drobne tabele typy typ\_akcji wypełniłem ręcznie.

2) skrypty w pythonie – Cześć danych dodałem generując je za pomocą spisanych skryptów w Python, dodawały one dane oraz odrobinę je losowały. Nie zawsze kod Python wypełniał dane bezpośrednio w tablicach często używałem go do generowania rekordów w zapytaniu sql. Przykładowy kod Python do generacji kontraktów:



3) Sztuczna inteligencja – Z uwagi na skomplikowanie kodu który generował by np. imiona i nazwiska osób oraz to ze raczej były by one losowe. O wymagane rekordy poprosiłem znany chatGpt to on wylosował imiona i nazwiska osób w bazie.

**Definicję tabel, relacji**

informacje o tablach będę przedstawiać w formie pseudokodu, będzie to informacja o nazwie tabeli i jej kolumnach lecz nie zapytanie sql dla przejrzystości:

**Table t\_kraje** {

id\_kraj varchar(255) [pk]

nazwa\_kraju varchar(255)

}  
Chyba najbardziej podstawowa tabela od której trzeba zacząć wypełnianie naszej bazy z uwagi iż bez kodów krajów cięzko będzie o wypełnienie osób i zespołów. Podjąłem decyzję w niej o id w formie tekstowej nie liczby ponieważ wszystkie kraje na świecie maja swoje oficjalne skróty, dlatego mogłem być spokojny o unikatowość klucza głównego. Dodatkowo zapewnia to dla nas ludzi łatwość w posługiwaniu się bazą łatwiej jest nam zgadnąć ze Polska to pol niż np. 39.  
  
**Table t\_ligi** {

id\_liga varchar(255) [pk]

nazwa\_ligi varchar(255)

rok\_zalozenia int

fk\_id\_kraj varchar(255)

}

Tabela lig piłkarskich, posiada non id też w formie tekstowej z podobnego powodu co wyżej. Łatwiej się nam posługiwać skrótami Ekstraklas = eks niż liczbami. Nazwa ligi to po prostu jej nazwa np. „Bundesliga” oczywistym było wykorzystanie varchar. Rok założenia w formie Int z uwagi iz date nie pasował by, często kluby piłkarskie powydostawały na przełomie XIX i XX wieku dokładne daty nie są znane.   
fk\_id\_kraj to rubryka na klucz obcy wyżej wspomnianego kraju.

**Table t\_sezony** {

id\_sezon varchar(255) [pk]

fk\_id\_liga varchar(255)

przelom\_lat varchar(255)

liczba\_awansujacych int

liczba\_spadajacych int

kwalifikacja\_do\_puchary int

zasady varchar(765)

}

Tabela z sezonami, musiałem taka utworzyć gdyż chcemy móc rozdzielić rozgrywki z jednego roku od drugiego:  
id również w formie varchar podjąłem taka decyzję z podobnego powodu co wyżej, użytkownikowi łatwiej jest zrozumieć id : „eks2023/2024” możemy w takim id zawrzeć trochę więcej informacji niż w samym numerze. Dalej fk\_idliga to varchar klucz obcy do ligi, przelom\_lat varchar to rok w którym jest rozgrywany sezon przez to ze sezony są przeważnie rozgrywane na przełomie alt wybrałem varchar by móc wpisać wartość np. 2023/2024. Idąc dalej liczba\_awansujacych int, liczba\_spadajacych int, kwalifikacja\_do\_puchary int to po prostu liczby określające ile zespołów z czubka i dołu tabli awansuję spada lub dostaje się do pucharów międzynarodowych. Ostatnia rubryka to miejsce na spisanie zasad ligi na dany sezon np. minimum 1 zawodnik u-21 w składzie na mecz etc.

**Table t\_zespoly\_zgloszone** {

id\_zgloszenia int [pk, increment]

fk\_id\_sezon varchar(255)

fk\_id\_zespolu varchar(255)

}

Tabelka pośrednia mająca nam pozwolić co sezon łączyć nowe zespoły z danym sezonem. Wynika to z tego że co sezon skład ligi zmienia się i inne zespoły przystępują do sezonu. Id w formie int oraz dwa klucze obce. fk\_id\_sezon varchar(255) do sezonu o którym pisałem wyżej oraz do zespołu fk\_id\_zespolu varchar(255) do którego zaraz przejdziemy.

**Table t\_kolejki** {

id\_kolejka varchar(255) [pk]

numer\_kolejki int

fk\_id\_sezon varchar(255)

fk\_id\_runda int

},

Kolejka to pojedyncza seria gier najczęściej odbywa się od piątku do poniedziałku. Tu również obserwujemy klucz główny varchar, również tez tu istnieje specjalne nazewnictwo dla tych danych mające na celu odwoływać się do sezonu przykładowo „eks2023/2024k1”.   
Zaobserwujemy dwa klucze obce fk\_id\_sezon varchar(255) do wyżej wspomnianych sezonów oraz fk\_id\_runda int odnoszący się do id typu rundy.

**Table t\_typ\_rundy** {

id\_typ\_rundy int [pk, increment]

nazwa\_rundy varchar(255)

}

Tu już id w formie liczbowej, rundę przeważnie dzielimy na jesienna i wiosenną stąd odpowiednie dopasowania 1 i 2 ( potocznie też są nazywane pierwszą i drugą rundą).

**Table t\_stadiony** {

id\_stadionu varchar(255) [pk]

pojemnosc\_max int

wlasciciel varchar(255)

adres varchar(255)

}

Tabela przechowuje informacje o obiektach należących do zespołów jako obiekty domowe oraz o tym na jakim stadionie rozgrywano mecze. Tu dobór typów jest dość jasny jedynie id\_stadionu znów jest w formie tekstowej gdyż jest zakodowany tu główny klub korzystający z danego obiektu dla przykładu stadion Arki Gdynia ma kod „ark-stadion”.

**Table t\_zespoly** {

id\_zespolu varchar(255) [pk]

nazwa\_zespolu varchar(255)

rok\_zalozenia int

fk\_id\_stadion varchar(255)

fk\_id\_kraj varchar(255)

}

Tabela przechowuje wszystkie zespołu zrównanych lig. Id ponownie w formie skrótu pochodzącego od nazwy zespołu Arka Gdynia – „ARK”. Warto zwrócić tu uwagę na dwa klucze: fk\_id\_stadion varchar(255), fk\_id\_kraj varchar(255) odpowiadają one stadionowi którego zespół uzwaja jako gospodarz oraz kraju z którego pochodzi dany zespół (uwaga zespół może pochodzić z kraju a a grać w innej lidze).  
  
 **Table t\_osoby** {

id\_osoby int [pk, increment]

imie varchar(255)

nazwisko varchar(255)

data\_urodzenia date

fk\_id\_kraj varchar(255)

}

Również jedna z najpotrzebniejszych tabel, bo przecież ktoś musi grać. Tu już i w formie po prostu liczby za duża zbieżność imion i nazwisk, oraz dziwnych znaków w nich się znajdujących mogłaby spowodować problemy z indywidualnością klucza głównego, zaś w momencie dodawania liczb na końcu imion używanie jakichś form kodów tekstowych traci raczej sens. Poza tym imię i nazwisko tekowe i finalnie data urodzenia w formie pasującej tu date i klucz obcy do kraju.

**Table t\_sklady** {

id\_skladu int [pk, increment]

fk\_id\_zespolu varchar(255)

}

Tabela pośrednia pozwalająca tworzyć nowe składy dla zespołów. Decyzja o jej stworzeniu przyprawiła mi nie mały bul głowy. Z początku wydaje się niepotrzebna ale jednak potrzebujemy opcji tworzenia składów by dana osoba nie była sztywna połączona z zespołem wszak dany gracz może występować w rożnych zespołach np. Lewandowski występuje w reprezentacji polski i FC Barcelonie. Dodatkowo jeden zespół może wystawić 2 rożne składy gdzie zawodnicy będą prezentować inne zadania. Chciałem się pokusić o też zapisanie uw formacji dla danego składu np. 4-4-2 itd. jednak obecnie te ustawanie w piłce przechodzą tak płynnie z jednego w drugie w trakcie meczu kilkukrotnie ze uznałem to za bezcelowe dodatkowo będzie to określać informacja o zawodniku w składzie do której przejdziemy.

**Table t\_uczestnicy** {

id\_uczestnika int [pk, increment]

fk\_id\_rodzaj\_roli varchar(255)

fk\_id\_osoba int

fk\_id\_sklad int

}

Ta tabela jest pewną forma owinięcia osoby w informacje o tym co robi w meczu na boisku oraz w jakim składzie się znajduje. Nie chciałem powielać np. pod zawodnikiem, sędzią, terenem itd. informacji jak data urodzenia czy kraj. Dlatego dla osób uczestniczących w meczu stworzyłem taka tabele pośrednia która będzie łączyć się z id osoby składu oraz będzie określać rolę danej osoby w tym meczu na boisku (nie wszystkie osoby muszą np. możemy stworzyć osobę właściciela klubu który już z tą tabela nie będzie mieć połączenia). W przypadku gdy w jednym meczu dana osoba będzie np. asystentem to dajemy jej role asystenta i referencje do osoby i składu. W przypadku nagłej zmiany np. konieczności zastąpienia obecnego szkoleniowca tworzony jest nowy uczestnik z id tej samej osoby ale nową rolą trenera w takiej formie może on zostać dołączony do nowego składu i meczu.

**Table t\_role** {

id\_roli varchar(255) [pk]

nazwa\_roli varchar(255)

szczegoly\_roli varchar(255)

}

Jest to tabela przechowująca nazwę i opis danej roli np. trn – trener - kierowanie zespołem. Tu znów mamy id w formie tekstowej skrót „bra” lepiej oddaje rolę bramkarza tych ról na boisku nie będzie w nieskończoność gdyż zawodników na boisku może być max 22 , rezerwowi i sztaby szkoleniowe też są ograniczone.  
  
 **Table t\_mecze** {

id\_mecz int [pk, increment]

fk\_id\_gospodarz varchar(255)

fk\_id\_gosc varchar(255)

fk\_id\_sklad\_gospodarz int

fk\_id\_sklad\_gosc int

fk\_id\_stadion varchar(255)

frekwencja int

czas\_rozpoczecia datetime

fk\_id\_kolejka varchar(255)

kurs\_gosp double

kurs\_gosc double

doliczony\_czas\_1P int

doliczony\_czas\_2P int

fk\_id\_sedzia int

}  
 Najobszerniejsza tabela która zwiera informacje o meczu. Przede wszystkim z uwagi na dużą liczbę spotkań i konieczność tworzenia długich kodów tekstowych wybrałem int jako ID. Omawiając dalej: fk\_id\_gospodarz varchar(255) fk\_id\_gosc varchar(255) są to klucze obce do zespołów biorących udział w spotkaniu. Oraz składów które dane zespoły wystawiły fk\_id\_sklad\_gospodarz int i fk\_id\_sklad\_gosc int, (klucze obce) jak wspominałem wcześniej zespół może wybrać tylko pewna ilość graczy nie zawsze wszystkich np. 11 + 5 = 16 zawodników, z uwagi an rotacje w składach tworzymy nowe połączone z innymi uczestnikami.  
Dalej mamy klucz obcy do stadionu fk\_id\_stadion który łączy nasz mecz z obiektem na którym się odbywał. Frekwencje w formie liczby całkowitej, czas rozpoczęcia dateime. Referencje fk\_id\_kolejka (klucz obcy) do nr kolejki w ramach której rozgrywany jest dany mecz. Dwie wartości dobule do określania kursów bukmacherskich kurs\_gosp double, kurs\_gosc double, dwa inty dla doliczonego czasu : doliczony\_czas\_1P int, doliczony\_czas\_2P int służą do odnotowania i ewentualnego obliczenia realnego całkowitego czasu spotkań. Całą tabelę kończy klucz obcy fk\_id\_sedzia int który jest połączony relacją z id\_uczestnika taki uczestnik powinien mieć wybraną rolę t\_rola = sed - sędzia główny - sędziowanie meczu.

**Table t\_akcje\_meczowe** {

id\_akcji int [pk, increment]

minuta int

fk\_id\_typ\_akcji int

fk\_id\_mecz int

fk\_id\_osoba int (fk\_id\_uczestnika)

}

Tabela przechowująca czas i referencje do danej akcji meczowej. Z uwagi na bardzo dużą ilość i ewentualne powtarzanie się podobnych sytuacji id jest w formie INT z auto inkrementacją. Mamy dalej zapisaną minute int, fk\_id\_typ\_akcji klucz obcy do połączenia z typem akcji który zaraz omówimy. Klucz do id\_meczu w którym dane wydarzenie miało miejsce: fk\_id\_mecz int oraz fk\_id\_osoba która tego zdarzenia dokonała tu drobna pomyłka w nazewnictwie gdyż powinno być fk\_id\_uczestnika odnosi się ona do tabeli t\_uczestnik i jego id. Z uwagi jednak na już wypełnienie tabli danymi (kilkaset o ile nie kilka tysięcy) co pochłania znaczą część czasu pozostawiłem to w takiej formie.

**Table t\_typ\_akcji**{

id\_typu\_akcji int [pk, increment]

nazwa\_akcji varchar(255)

}

Tabela z typem akcji, dzięki niej wiemy co wydarzyło się w danej minucie, jest to wyciągnięte do oddzielnej tabeli ponieważ w gole, kartki, karne się powtarzają. Mamy tu id typu int inkrementalne i dal przykładu 1 to gol, 2 to karny nie trafiony itd.

**Table t\_kontrakty** {

id\_kontraktu int [pk, increment]

fk\_id\_zespolu varchar(255)

fk\_id\_osoba int

data\_podpisania date

data\_konca date

fk\_id\_typ\_kontraktu int

}

Tabela niezbędna do określenia kto dal kogo pracuje. Skład to jednak forma płynna ktoś morze zagrać w składzie np. gwiazdy europy albo reprezentacji, gdzieś zostać dodanym czysto okazjonalnie ale musimy jakoś określić w jakim klubie gra, grał lub nawet będzie grać. Przychodzi nam z pomocą kontakt który ma klucz obcy do fk\_id\_zespolu varchar(255) by wiedzieć jaki zespół kontaktuje zawodnika. Klucz obcy do fk\_id\_osoba int uwaga nie mylić z t\_uczesnik, tu referencja jest do konkretnej osoby nie uczestnika, tabela składu i uczestnika pozwala nam na płynne zmiany zawodników w kontekście meczów. Zaś kontakt ma być połączeniem danej osoby z zespołem. Dalej data\_podpisania date, data\_konca date to daty pozwalające nam określić status kontaktu czy jest aktywny, czy już wygasł, albo dopiero jest zaplanowany. Ostatni element tabeli to klucz oby do typu składu : fk\_id\_typ\_kontraktu int.  
  
 **Table t\_typy\_kontraktow** {

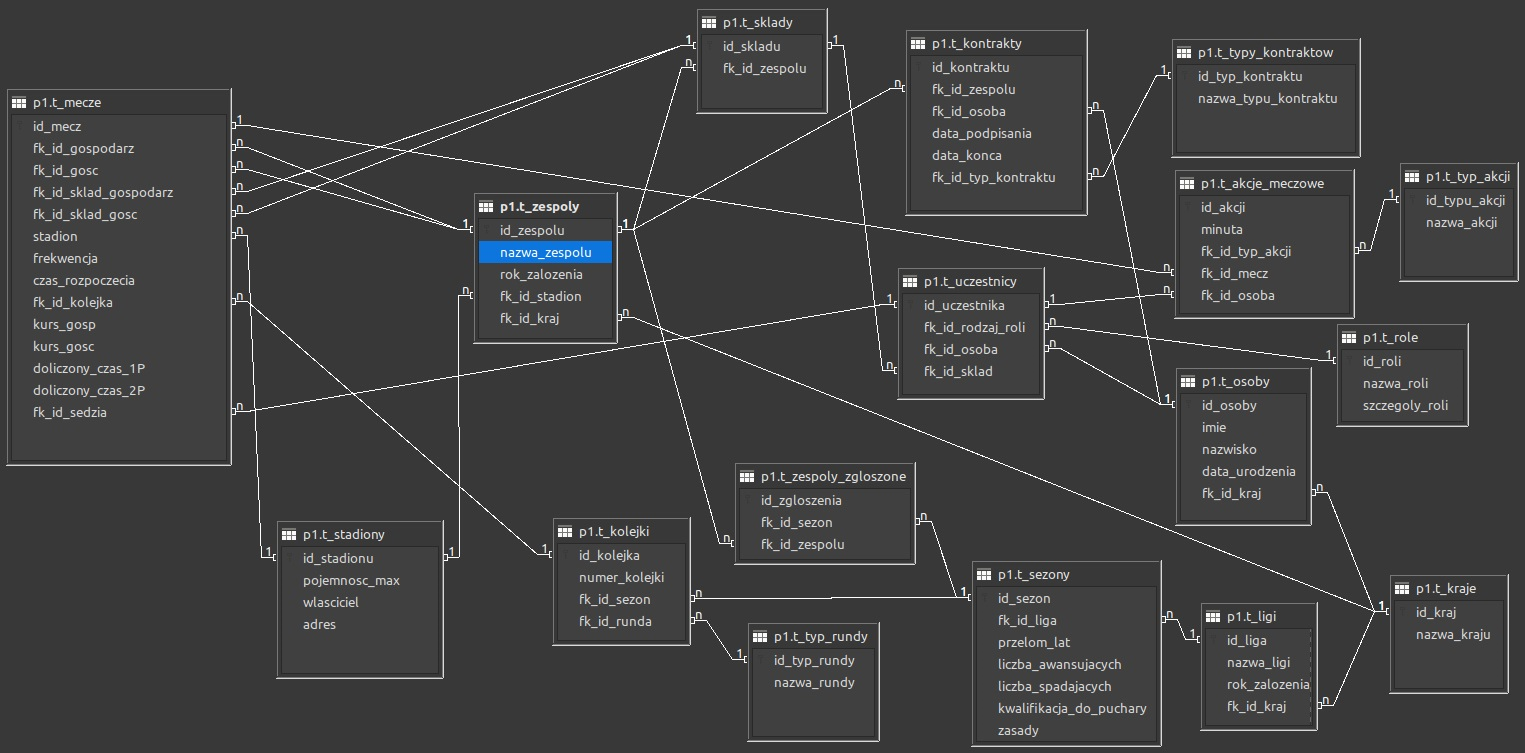
id\_typ\_kontraktu int [pk, increment]

nazwa\_typu\_kontraktu varchar(255)

}

Jest to nasza ostania tabela, może małą ale nie mniej ważna od innych. Pozwala ona nam sprawdzić jaka forma zatrudnienia jest przedstawiona danej osobie i jaka wieź ją łączy z klubem. Id to zwykła liczba auto inkrementowana, pod nazwą zaś string, np. „piłkarski profesjonalny”, „piłkarski juniorski” albo „trnerski” itd.itd.

**Mapa relacji z Libre Office Base.**

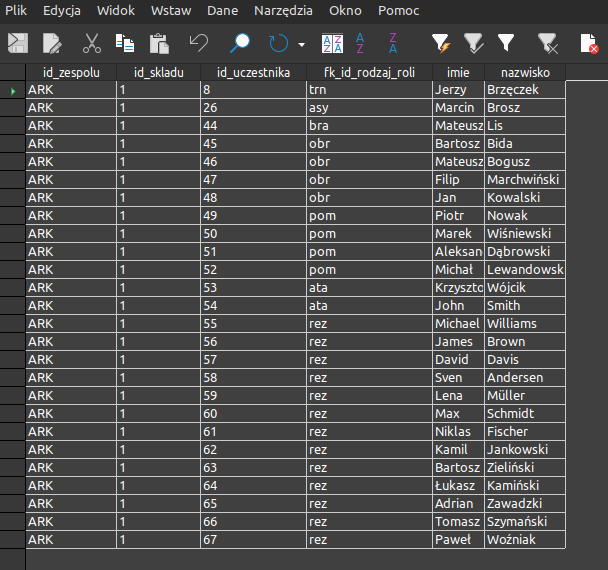
Szczegóły co do relacji opisałem przy okazji tabel odnosząc się wszystkich pul fk\_ które łącza się z innymi tablami. Poniżej przedstawiam mapę z libre office base dal lepszego zwizualizowania sobie połączeń.

**Kwerendy  
  
 k\_zawodnicy\_ark:** Kwerenda do wypisania wszystkich zawodników danego zespołu, jako przykład użyty id „ARK” (Arka Gdynia):  
SELECT `zesp`.`id\_zespolu`, `sklad`.`id\_skladu`, `ucz`.`id\_uczestnika`, `ucz`.`fk\_id\_rodzaj\_roli`, `osoba`.`imie`, `osoba`.`nazwisko`

FROM `p1`.`t\_zespoly` `zesp`, `p1`.`t\_sklady` `sklad`, `p1`.`t\_uczestnicy` `ucz`, `p1`.`t\_osoby` `osoba`

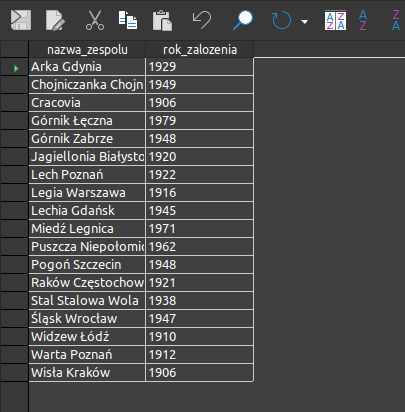
WHERE `zesp`.`id\_zespolu` = `sklad`.`fk\_id\_zespolu` AND `ucz`.`fk\_id\_sklad` = `sklad`.`id\_skladu` AND `osoba`.`id\_osoby` = `ucz`.`fk\_id\_osoba` AND `zesp`.`id\_zespolu` = 'ARK';

efekt:

  
  
 **k\_polskie\_zespoly:** kwerenda pozwalająca odfiltrować polskie zespoły wyświetlić je i przedstawić ich date założenia:  
SELECT `nazwa\_zespolu`, `rok\_zalozenia`

FROM `t\_zespoly`

WHERE `fk\_id\_kraj` = 'pol'

efekt:

**k\_wyniki\_zespolu\_ark** : Kwerenda pozwalająca sprawdzić wyniki danego zespołu w meczach, id można zamienić tu dal przykładu również ark (Arka Gdynia)  
  
SELECT CASE WHEN `Gole\_Zespolu` > `Gole\_Przeciwnika` THEN 'Zwycięstwo' WHEN `Gole\_Zespolu` = `Gole\_Przeciwnika`

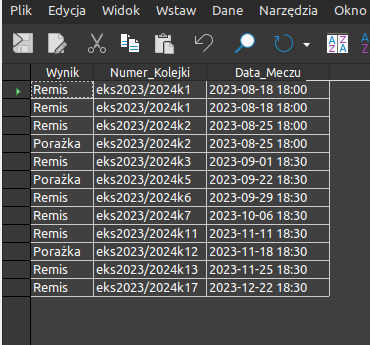
THEN 'Remis' ELSE 'Porażka' END `Wynik`, `Wyniki`.`fk\_id\_kolejka` `Numer\_Kolejki`, `Wyniki`.`czas\_rozpoczecia` `Data\_Meczu`

FROM ( SELECT SUM( CASE WHEN `z`.`id\_zespolu` = 'ID\_Zespołu' AND `ta`.`nazwa\_akcji` = 'Gol' THEN 1 ELSE 0 END ) `Gole\_Zespolu`,

SUM( CASE WHEN `z`.`id\_zespolu` != 'ID\_Zespołu' AND `ta`.`nazwa\_akcji` = 'Gol' THEN 1 ELSE 0 END ) `Gole\_Przeciwnika`, `m`.`fk\_id\_kolejka`, `m`.`czas\_rozpoczecia`

FROM `t\_mecze` `m` INNER JOIN `t\_zespoly` `z` ON `m`.`fk\_id\_gospodarz` = `z`.`id\_zespolu` OR `m`.`fk\_id\_gosc` = `z`.`id\_zespolu` INNER JOIN `t\_akcje\_meczowe` `am` ON `m`.`id\_mecz` = `am`.`fk\_id\_mecz` INNER JOIN `t\_typ\_akcji` `ta` ON `am`.`fk\_id\_typ\_akcji` = `ta`.`id\_typu\_akcji`

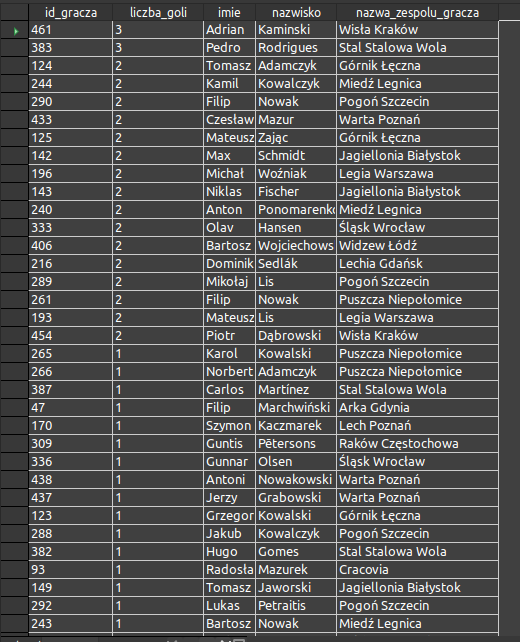
WHERE `z`.`id\_zespolu` = 'ARK' GROUP BY `m`.`id\_mecz` ) AS `Wyniki` ORDER BY `Wyniki`.`czas\_rozpoczecia`

efekt 

**k\_tabela\_strzelcy**: kwerenda pozwalająca nam uzyskać tablice najlepszych strzelców:  
SELECT `t\_uczestnicy`.`fk\_id\_osoba` `id\_gracza`, COUNT( \* ) `liczba\_goli`, `t\_osoby`.`imie`, `t\_osoby`.`nazwisko`, `t\_zespoly`.`nazwa\_zespolu` `nazwa\_zespolu\_gracza`

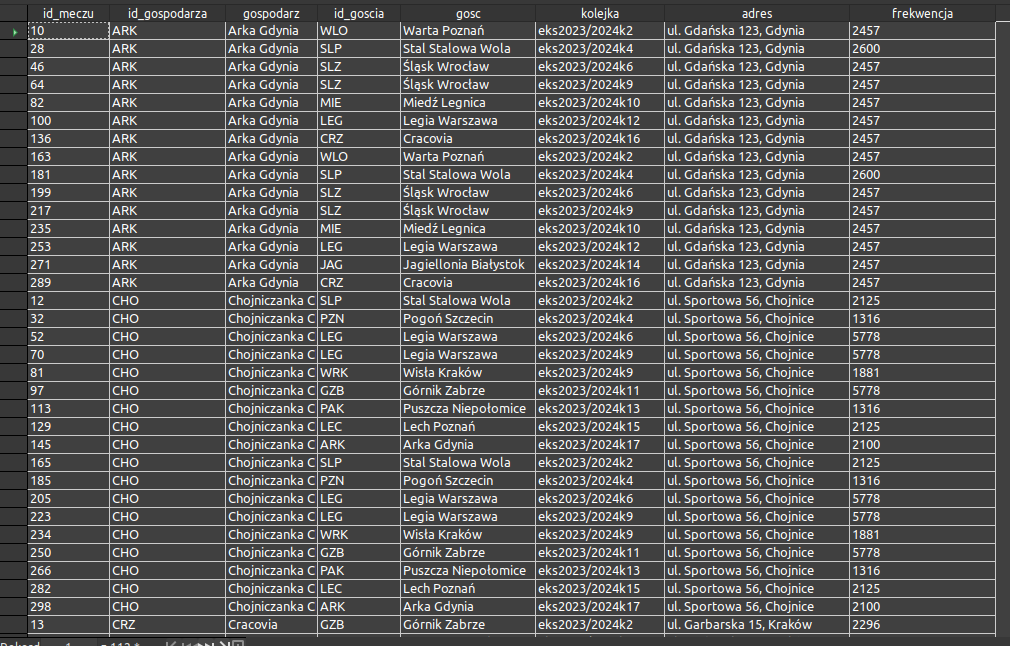
FROM `t\_akcje\_meczowe` JOIN `t\_typ\_akcji` ON `t\_akcje\_meczowe`.`fk\_id\_typ\_akcji` = `t\_typ\_akcji`.`id\_typu\_akcji` JOIN `t\_uczestnicy` ON `t\_akcje\_meczowe`.`fk\_id\_osoba` = `t\_uczestnicy`.`fk\_id\_osoba` JOIN `t\_osoby` ON `t\_uczestnicy`.`fk\_id\_osoba` = `t\_osoby`.`id\_osoby` JOIN `t\_sklady` ON `t\_uczestnicy`.`fk\_id\_sklad` = `t\_sklady`.`id\_skladu` JOIN `t\_zespoly` ON `t\_sklady`.`fk\_id\_zespolu` = `t\_zespoly`.`id\_zespolu`

WHERE `t\_typ\_akcji`.`id\_typu\_akcji` = 1 GROUP BY `t\_uczestnicy`.`fk\_id\_osoba`, `t\_zespoly`.`nazwa\_zespolu` ORDER BY `liczba\_goli` DESC

uwaga nie jest to pełna tabela z uwagi an duża ilość rekordów pozwoliłem sobie pominąć jej dolną cześć:  
  
  
  
 **k\_mecze\_szczegoly:** Kwerenda wybierająca informacje o meczach wraz z ich szczegółami:  
  
SELECT `tm`.`id\_mecz` `id\_meczu`, `tm`.`fk\_id\_gospodarz` `id\_gospodarza`, `tz1`.`nazwa\_zespolu` `gospodarz`, `tm`.`fk\_id\_gosc` `id\_goscia`, `tz2`.`nazwa\_zespolu` `gosc`, `tm`.`fk\_id\_kolejka` `kolejka`, `sta`.`adres`, `tm`.`frekwencja`

FROM `p1`.`t\_mecze` `tm`, `p1`.`t\_zespoly` `tz1`, `p1`.`t\_zespoly` `tz2`, `p1`.`t\_stadiony` `sta` WHERE `tm`.`fk\_id\_gospodarz` = `tz1`.`id\_zespolu` AND `tm`.`fk\_id\_gosc` = `tz2`.`id\_zespolu` AND `tm`.`stadion` = `sta`.`id\_stadionu`

efekt (górna część tabeli):

  
 **k\_mecze\_gole**: Podobna kwerenda do tej z szczegółami meczów ale tym razem zliczająca gole gospodarza i gościa:  
SELECT `m`.`id\_mecz` `ID Meczu`, `z\_gosp`.`nazwa\_zespolu` `Gospodarz`, `z\_gosc`.`nazwa\_zespolu` `Gość`, SUM( CASE WHEN `am`.`fk\_id\_typ\_akcji` = ( SELECT `id\_typu\_akcji`

FROM `t\_typ\_akcji`

WHERE `nazwa\_akcji` = 'Gol' ) AND `u`.`fk\_id\_sklad` = `m`.`fk\_id\_sklad\_gospodarz` THEN 1 ELSE 0 END ) `Gole Gospodarza`, SUM( CASE WHEN `am`.`fk\_id\_typ\_akcji` = ( SELECT `id\_typu\_akcji`

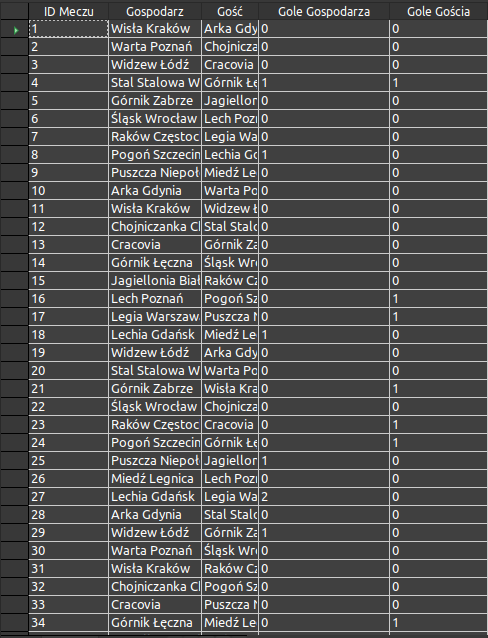
FROM `t\_typ\_akcji`

WHERE `nazwa\_akcji` = 'Gol' ) AND `u`.`fk\_id\_sklad` = `m`.`fk\_id\_sklad\_gosc` THEN 1 ELSE 0 END ) `Gole Gościa`

FROM `t\_mecze` `m` LEFT JOIN `t\_zespoly` `z\_gosp` ON `m`.`fk\_id\_gospodarz` = `z\_gosp`.`id\_zespolu` LEFT JOIN `t\_zespoly` `z\_gosc` ON `m`.`fk\_id\_gosc` = `z\_gosc`.`id\_zespolu` LEFT JOIN `t\_uczestnicy` `u` ON `u`.`fk\_id\_sklad` = `m`.`fk\_id\_sklad\_gospodarz` OR `u`.`fk\_id\_sklad` = `m`.`fk\_id\_sklad\_gosc` LEFT JOIN `t\_akcje\_meczowe` `am` ON `am`.`fk\_id\_osoba` = `u`.`fk\_id\_osoba` AND `am`.`fk\_id\_mecz` = `m`.`id\_mecz`

GROUP BY `m`.`id\_mecz`, `z\_gosp`.`nazwa\_zespolu`, `z\_gosc`.`nazwa\_zespolu`

efekt (górna cześć tabeli):



**k\_mecze\_zwycięscy:** Kwerenda podobna do tej z golami tym razem jednak sprawdzamy zwycięzców:

SELECT `m`.`id\_mecz` `ID Meczu`, `z\_gosp`.`nazwa\_zespolu` `Gospodarz`, `z\_gosc`.`nazwa\_zespolu` `Gość`, SUM( CASE WHEN `am`.`fk\_id\_typ\_akcji` = ( SELECT `id\_typu\_akcji`

FROM `t\_typ\_akcji`

WHERE `nazwa\_akcji` = 'Gol' ) AND `u`.`fk\_id\_sklad` = `m`.`fk\_id\_sklad\_gospodarz` THEN 1 ELSE 0 END ) `Gole Gospodarza`, SUM( CASE WHEN `am`.`fk\_id\_typ\_akcji` = ( SELECT `id\_typu\_akcji`

FROM `t\_typ\_akcji`

WHERE `nazwa\_akcji` = 'Gol' ) AND `u`.`fk\_id\_sklad` = `m`.`fk\_id\_sklad\_gosc` THEN 1 ELSE 0 END ) `Gole Gościa`, CASE WHEN SUM( CASE WHEN `am`.`fk\_id\_typ\_akcji` = ( SELECT `id\_typu\_akcji`

FROM `t\_typ\_akcji`

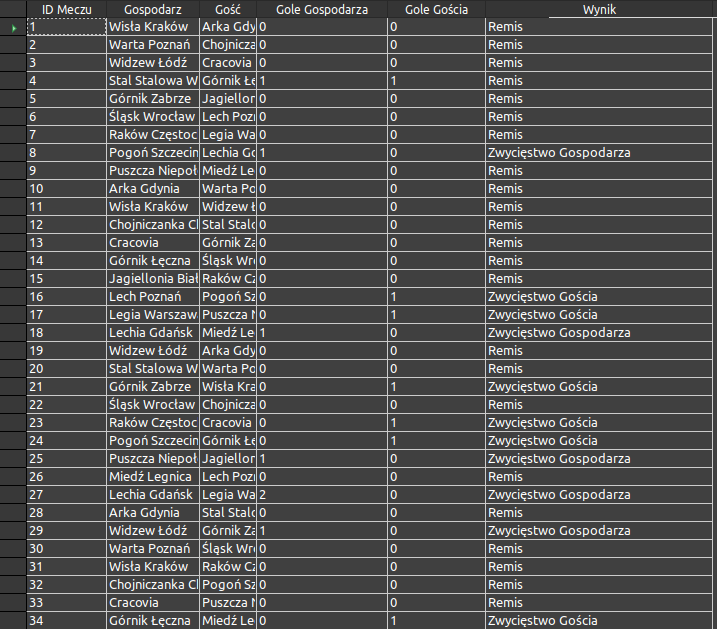
WHERE `nazwa\_akcji` = 'Gol' ) AND `u`.`fk\_id\_sklad` = `m`.`fk\_id\_sklad\_gospodarz` THEN 1 ELSE 0 END ) > SUM( CASE WHEN `am`.`fk\_id\_typ\_akcji` = ( SELECT `id\_typu\_akcji` FROM `t\_typ\_akcji` WHERE `nazwa\_akcji` = 'Gol' ) AND `u`.`fk\_id\_sklad` = `m`.`fk\_id\_sklad\_gosc` THEN 1 ELSE 0 END ) THEN 'Zwycięstwo Gospodarza' WHEN SUM( CASE WHEN `am`.`fk\_id\_typ\_akcji` = ( SELECT `id\_typu\_akcji`

FROM `t\_typ\_akcji`

WHERE `nazwa\_akcji` = 'Gol' ) AND `u`.`fk\_id\_sklad` = `m`.`fk\_id\_sklad\_gospodarz` THEN 1 ELSE 0 END ) = SUM( CASE WHEN `am`.`fk\_id\_typ\_akcji` = ( SELECT `id\_typu\_akcji` FROM `t\_typ\_akcji`

WHERE `nazwa\_akcji` = 'Gol' ) AND `u`.`fk\_id\_sklad` = `m`.`fk\_id\_sklad\_gosc` THEN 1 ELSE 0 END ) THEN 'Remis' ELSE 'Zwycięstwo Gościa' END `Wynik` FROM `t\_mecze` `m` LEFT JOIN `t\_zespoly` `z\_gosp` ON `m`.`fk\_id\_gospodarz` = `z\_gosp`.`id\_zespolu` LEFT JOIN `t\_zespoly` `z\_gosc` ON `m`.`fk\_id\_gosc` = `z\_gosc`.`id\_zespolu` LEFT JOIN `t\_uczestnicy` `u` ON `u`.`fk\_id\_sklad` = `m`.`fk\_id\_sklad\_gospodarz` OR `u`.`fk\_id\_sklad` = `m`.`fk\_id\_sklad\_gosc` LEFT JOIN `t\_akcje\_meczowe` `am` ON `am`.`fk\_id\_osoba` = `u`.`fk\_id\_osoba` AND `am`.`fk\_id\_mecz` = `m`.`id\_mecz`

GROUP BY `m`.`id\_mecz`, `z\_gosp`.`nazwa\_zespolu`, `z\_gosc`.`nazwa\_zespolu`



**k\_krol\_zoltych\_kartek**: kwerenda zliczająca żółte kartki i sortująca od największej do najmniejszej :

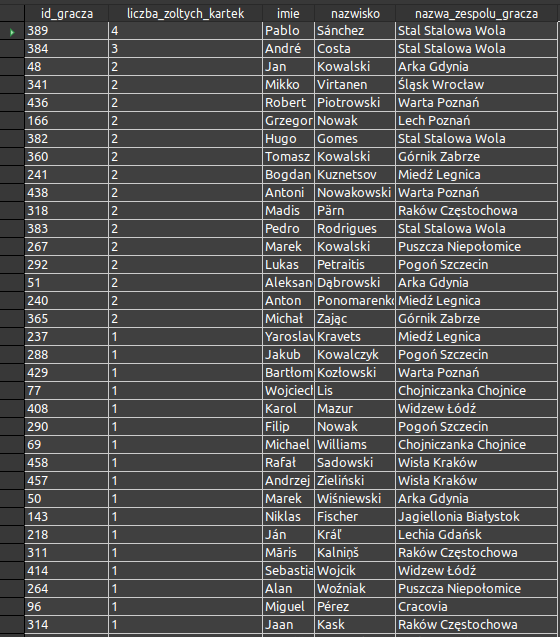
SELECT `t\_uczestnicy`.`fk\_id\_osoba` `id\_gracza`,

COUNT( \* ) `liczba\_zoltych\_kartek`, `t\_osoby`.`imie`, `t\_osoby`.`nazwisko`, `t\_zespoly`.`nazwa\_zespolu` `nazwa\_zespolu\_gracza` FROM `t\_akcje\_meczowe` JOIN `t\_typ\_akcji` ON `t\_akcje\_meczowe`.`fk\_id\_typ\_akcji` = `t\_typ\_akcji`.`id\_typu\_akcji` JOIN `t\_uczestnicy` ON `t\_akcje\_meczowe`.`fk\_id\_osoba` = `t\_uczestnicy`.`fk\_id\_osoba` JOIN `t\_osoby` ON `t\_uczestnicy`.`fk\_id\_osoba` = `t\_osoby`.`id\_osoby` JOIN `t\_sklady` ON `t\_uczestnicy`.`fk\_id\_sklad` = `t\_sklady`.`id\_skladu` JOIN `t\_zespoly` ON `t\_sklady`.`fk\_id\_zespolu` = `t\_zespoly`.`id\_zespolu`

WHERE `t\_typ\_akcji`.`id\_typu\_akcji` = 3 GROUP BY `t\_uczestnicy`.`fk\_id\_osoba`, `t\_zespoly`.`nazwa\_zespolu`

ORDER BY `liczba\_zoltych\_kartek` DESC

poniżej efekt (również wycinek tabeli):



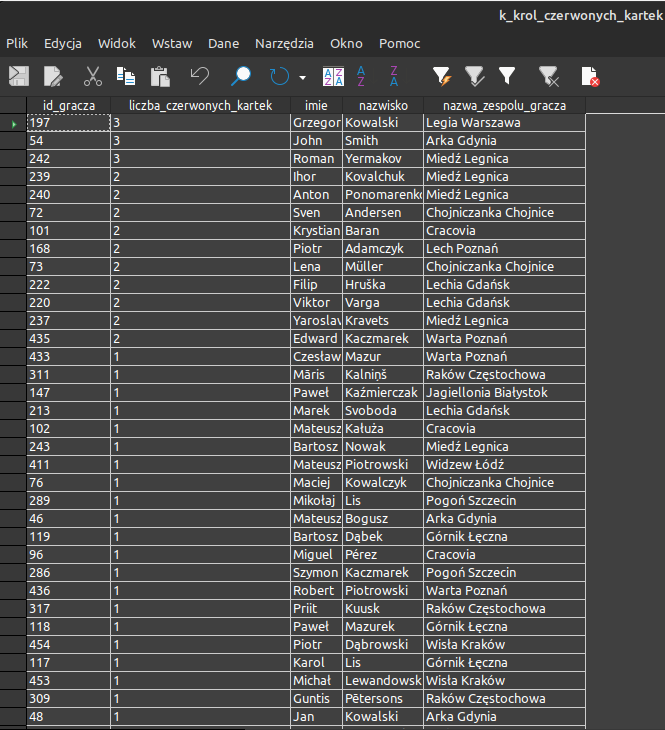
**k\_krol\_czerwonych\_kartek:** podobna do tej co wyżej, zliczamy jednak inna akcję:

SELECT `t\_uczestnicy`.`fk\_id\_osoba` `id\_gracza`,

COUNT( \* ) `liczba\_czerwonych\_kartek`, `t\_osoby`.`imie`, `t\_osoby`.`nazwisko`, `t\_zespoly`.`nazwa\_zespolu` `nazwa\_zespolu\_gracza`

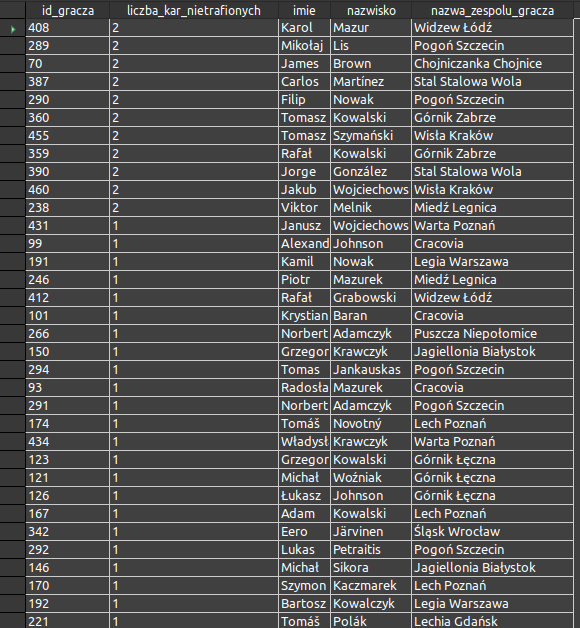
FROM `t\_akcje\_meczowe` JOIN `t\_typ\_akcji` ON `t\_akcje\_meczowe`.`fk\_id\_typ\_akcji` = `t\_typ\_akcji`.`id\_typu\_akcji` JOIN `t\_uczestnicy` ON `t\_akcje\_meczowe`.`fk\_id\_osoba` = `t\_uczestnicy`.`fk\_id\_osoba` JOIN `t\_osoby` ON `t\_uczestnicy`.`fk\_id\_osoba` = `t\_osoby`.`id\_osoby` JOIN `t\_sklady` ON `t\_uczestnicy`.`fk\_id\_sklad` = `t\_sklady`.`id\_skladu` JOIN `t\_zespoly` ON `t\_sklady`.`fk\_id\_zespolu` = `t\_zespoly`.`id\_zespolu`

WHERE `t\_typ\_akcji`.`id\_typu\_akcji` = 4 GROUP BY `t\_uczestnicy`.`fk\_id\_osoba`, `t\_zespoly`.`nazwa\_zespolu` ORDER BY `liczba\_czerwonych\_kartek` DESC

  
  
 **k\_karne\_nietrafione:** tabela zliczajaca kto najcześciej nie trafiał z karnych:  
SELECT `t\_uczestnicy`.`fk\_id\_osoba` `id\_gracza`, COUNT( \* ) `liczba\_kar\_nietrafionych`, `t\_osoby`.`imie`, `t\_osoby`.`nazwisko`, `t\_zespoly`.`nazwa\_zespolu` `nazwa\_zespolu\_gracza` FROM `t\_akcje\_meczowe` JOIN `t\_typ\_akcji` ON `t\_akcje\_meczowe`.`fk\_id\_typ\_akcji` = `t\_typ\_akcji`.`id\_typu\_akcji` JOIN `t\_uczestnicy` ON `t\_akcje\_meczowe`.`fk\_id\_osoba` = `t\_uczestnicy`.`fk\_id\_osoba` JOIN `t\_osoby` ON `t\_uczestnicy`.`fk\_id\_osoba` = `t\_osoby`.`id\_osoby` JOIN `t\_sklady` ON `t\_uczestnicy`.`fk\_id\_sklad` = `t\_sklady`.`id\_skladu` JOIN `t\_zespoly` ON `t\_sklady`.`fk\_id\_zespolu` = `t\_zespoly`.`id\_zespolu`

WHERE `t\_typ\_akcji`.`id\_typu\_akcji` = 2 GROUP BY `t\_uczestnicy`.`fk\_id\_osoba`, `t\_zespoly`.`nazwa\_zespolu`

ORDER BY `liczba\_kar\_nietrafionych` DESC



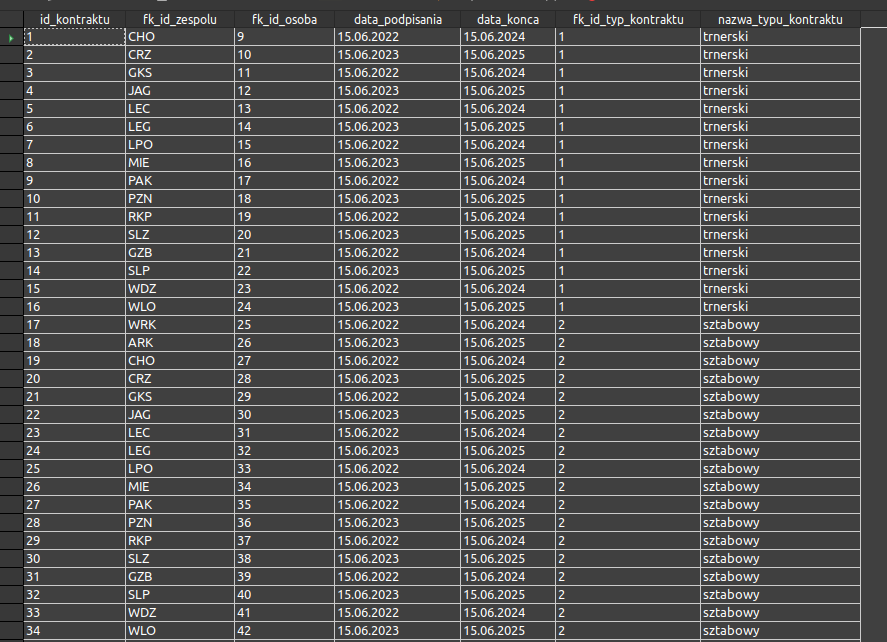
**k\_gole\_zespolow:** Kwerenda zliczająca gole każdego zespołu

SELECT `z`.`nazwa\_zespolu` `Zespol`,

SUM( CASE WHEN `typ\_akcji`.`nazwa\_akcji` = 'Gol' THEN 1 ELSE 0 END ) `Suma\_Goli` FROM `t\_mecze` `m` INNER JOIN `t\_zespoly` `z` ON `m`.`fk\_id\_gospodarz` = `z`.`id\_zespolu` OR `m`.`fk\_id\_gosc` = `z`.`id\_zespolu` INNER JOIN `t\_akcje\_meczowe` `am` ON `m`.`id\_mecz` = `am`.`fk\_id\_mecz` INNER JOIN `t\_typ\_akcji` `typ\_akcji` ON `am`.`fk\_id\_typ\_akcji` = `typ\_akcji`.`id\_typu\_akcji` GROUP BY `z`.`nazwa\_zespolu`

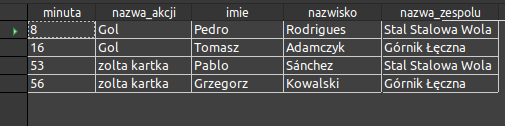
ORDER BY `Suma\_Goli` DESC  
  

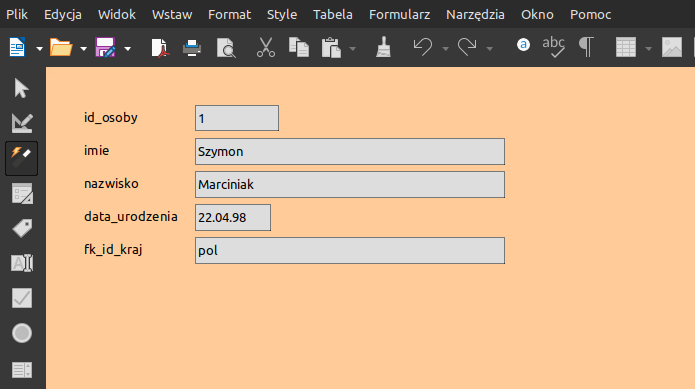

**k\_aktywne\_kontakty:** Kwerenda sprawdzająca jakie są obecnie aktywne kontakty opiera si.e o porównanie dat końca i początku z wprowadzoną datą:  
  
SELECT `t\_kontrakty`.\*, `t\_typy\_kontraktow`.`nazwa\_typu\_kontraktu`   
FROM `p1`.`t\_kontrakty` `t\_kontrakty`, `p1`.`t\_typy\_kontraktow` `t\_typy\_kontraktow`   
WHERE `t\_kontrakty`.`fk\_id\_typ\_kontraktu` = `t\_typy\_kontraktow`.`id\_typ\_kontraktu` AND `t\_kontrakty`.`data\_podpisania` < {d '2024-01-05' } AND `t\_kontrakty`.`data\_konca` > {d '2024-01-05' }

  
 **k\_akcje\_w\_meczu\_id\_4:** kwerenda pozwalając nam sprawdzić wszystkie akcje w danym meczu oraz ich autorów dal przykładu wybrałem id 4 bo było tam kilka rekordów  
  
SELECT `t\_akcje\_meczowe`.`minuta`, `t\_typ\_akcji`.`nazwa\_akcji`, `t\_osoby`.`imie`, `t\_osoby`.`nazwisko`, `t\_zespoly`.`nazwa\_zespolu`

FROM `t\_akcje\_meczowe` JOIN `t\_typ\_akcji` ON `t\_akcje\_meczowe`.`fk\_id\_typ\_akcji` = `t\_typ\_akcji`.`id\_typu\_akcji` JOIN `t\_uczestnicy` ON `t\_akcje\_meczowe`.`fk\_id\_osoba` = `t\_uczestnicy`.`id\_uczestnika` JOIN `t\_osoby` ON `t\_uczestnicy`.`fk\_id\_osoba` = `t\_osoby`.`id\_osoby` JOIN `t\_sklady` ON `t\_uczestnicy`.`fk\_id\_sklad` = `t\_sklady`.`id\_skladu` JOIN `t\_zespoly` ON `t\_sklady`.`fk\_id\_zespolu` = `t\_zespoly`.`id\_zespolu`

WHERE `t\_akcje\_meczowe`.`fk\_id\_mecz` = '4'

ORDER BY `t\_akcje\_meczowe`.`minuta`  
  
  
  
**Formularze:**

na zakończenie przygotowałem 2 formularze:  
Formularz t\_osoba do wprowadzania nowych osób, bazuje na tablicy t\_osoby:  
  
  
oraz formularz t\_stadion do stadionów bazujący na tablicy stadionów:

