WOJSKOWA AKADEMIA TECHNICZNA

im. Jarosława Dąbrowskiego

Laboratorium z przedmiotu

Bazy Danych

Sprawozdanie z projektu

Temat projektu: **LIGA PIŁKARSKA**

Prowadzący: mgr inż. Marcin Dąbkiewicz

Projekt wykonał: Igor Sokół

Grupa: WCY18IY7S1

Data: 28.01.2020 r.

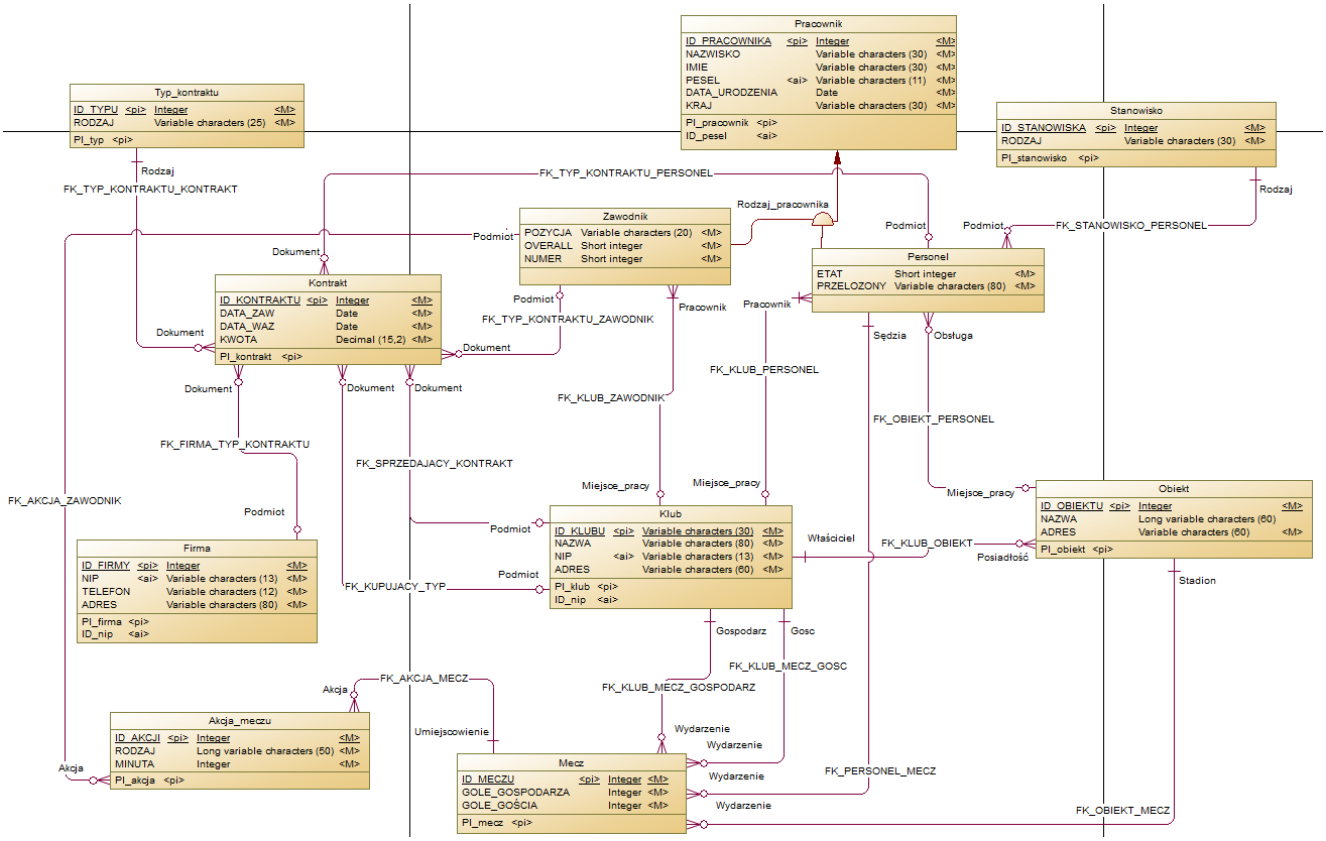
1. Opis problemu:

W lidze piłkarskiej *biorą udział* **kluby**. Każdy z nich *rozgrywa* **mecze** pomiędzy sobą. W klubach *grają* **zawodnicy** i *pracuje* **personel** na różnych **stanowiskach**. Do personelu *należy* również **sędzia**, który *sędziuje* mecze drużyn. W meczu *zdobywane są* bramki, asysty, kartki itd. Czyli **akcje**. Zawodnicy i personel *zawierają* ze swoimi klubami **kontrakty**. Kontrakty mogą zawierać również kluby – pomiędzy sobą lub z **firmami** zewnętrznymi. Mecze *rozgrywane są* na klubowych **obiektach**, na których *pracuje* klubowy personel.

2. Założenia projektowe:

Do ligi piłkarskiej uczęszcza kilkanaście klubów. Każdy klub posiada zawodników, personel oraz obiekty (np. stadion, baza treningowa). Każdy członek personelu posiada konkretne stanowisko, ale nie każdy należy do klubu (np. sędziowie). Zawodnicy odbywają transfery pomiędzy klubami. Praktycznie na każdej płaszczyźnie istotne są kontrakty, które mają swoje rodzaje (inny rodzaj dla transferu, inny dla praw transmisyjnych itd.). Dzięki kontraktom możemy wiązać ze sobą zawodników odbywających transfery z klubu do klubu, członków personelu z klubami oraz kluby z zewnętrznymi firmami. Kluby rozgrywają między sobą mecze. Mecz odbywa się na konkretnym stadionie (obiekcie). Prowadzony jest przez konkretnego sędziego (członka personelu). Tabela „Mecz” przechowuje dane o ilości strzelonych goli w meczu. Na jej podstawie obliczany będzie przydział punktów klubu i jego miejsce w tabeli. Tabela „Akcja_meczu” jest ściśle związana z tabelą „Mecz” – w niej zapisywane są wszystkie boiskowe interakcje związane z zawodnikami występującymi w meczu. Encje „Personel” i „Zawodnik” dziedziczą z encji „Pracownik”. Encja „Pracownik” nie jest generowana w modelu fizycznym.

a. Model konceptualny:



Główne encje w projekcie:

- Klub – do klubu należą zawodnicy, personel i obiekty, to kluby rozgrywają mecze i podpisują kontrakty.
- Kontrakt – encja dokumentuje powiązania (głównie finansowe) pomiędzy klubami, zawodnikami, personelem, firmami.

b. Tabele (alfabetycznie):

1. Tabela „Akcja meczu” – zawiera dane wydarzeń z konkretnego meczu

Akcja_meczu (DBA)

Columns | Constraints | Referencing Constraints | Indexes | Text Indexes | Triggers | Permissions | Dependent Views | Data

	PK...	Name	...	Obj...	Data Type	Size	Scale	C...	Null	Un...	Value	Comment
1	☑	ID_AKCJI	1	3171	integer				☐	☐		Identyfikator akcji
2	☐	ID_MECZU	2	3172	integer				☐	☐		Identyfikator meczu
3	☐	ID_PRACO...	3	3173	integer				☐	☐		Identyfikator pracownika
4	☐	RODZAJ	4	3174	long varchar			☐	☐	☐		Rodzaj akcji - kartka, gol, asysta itp.
5	☐	MINUTA	5	3175	integer				☐	☐		Minuta akcji

2. Tabela „Firma” – zawiera dane firm związanych z klubami i zawodnikami

Firma (DBA)

Columns | Constraints | Referencing Constraints | Indexes | Text Indexes | Triggers | Permissions | Dependent Views | Data

	PK...	Name	...	Obj...	Data Type	Size	Scale	C...	Null	Un...	Value	Comment
1	☑	ID_FIRMY	1	3181	integer				☐	☐		Identyfikator firmy
2	☐	NIP	2	3182	varchar	13		☐	☐	☑		Numer NIP firmy
3	☐	TELEFON	3	3183	varchar	12		☐	☐	☐		Telefon do firmy
4	☐	ADRES	4	3184	varchar	80		☐	☐	☐		Adres siedziby firmy (zameldowania)

3. Tabela „Klub” – zawiera podstawowe dane klubów grających w lidze

Klub (DBA)

Columns | Constraints | Referencing Constraints | Indexes | Text Indexes | Triggers | Permissions | Dependent Views | Data

	PK...	Name	...	Obj...	Data Type	Size	Scale	C...	Null	Un...	Value	Comment
1	☑	ID_KLUBU	1	3190	varchar	30		☐	☐	☐		Identyfikator klubu grającego w lidze
2	☐	NAZWA	2	3191	varchar	80		☐	☐	☐		Nazwa klubu
3	☐	NIP	3	3192	varchar	13		☐	☐	☑		Numer NIP klubu
4	☐	ADRES	4	3193	varchar	60		☐	☐	☐		Adres siedziby klubu - adres zameld...

4. Tabela „Kontrakt” – zawiera wszystkie dane z kontraktów zawieranych w lidze

Kontrakt (DBA)

Columns | Constraints | Referencing Constraints | Indexes | Text Indexes | Triggers | Permissions | Dependent Views | Data

	PK...	Name	...	Obj...	Data Type	Size	Scale	C...	Null	Un...	Value	Comment
1	☑	ID_KONTR...	1	3199	integer				☐	☐		Identyfikator kontraktu
2	☐	ID_KUPUJ...	2	3200	varchar	30		☐	☑	☐		Identyfikator klubu grającego w lidze
3	☐	ID_SPRZE...	3	3201	varchar	30		☐	☑	☐		Identyfikator klubu grającego w lidze
4	☐	ID_FIRMY	4	3202	integer				☑	☐		Identyfikator firmy
5	☐	ID_TYPU	5	3203	integer				☐	☐		Identyfiktor typu kontraktu
6	☐	ID_PERSO...	6	3204	integer				☑	☐		Identyfikator pracownika
7	☐	ID_ZAWO...	7	3205	integer				☑	☐		Identyfikator pracownika
8	☐	DATA_ZAW	8	3206	date				☐	☐		Data zawarcia kontraktu
9	☐	DATA_WAZ	9	3207	date				☐	☐		Data ważności kontraktu
10	☐	KWOTA	10	3208	decimal	15	2		☐	☐		Kwota na jaką opiewa kontrakt

5. Tabela „Mecz” – zawiera dane na temat odbywanych meczów w lidze

Mecz (DBA)

Columns | Constraints | Referencing Constraints | Indexes | Text Indexes | Triggers | Permissions | Dependent Views | Data

	PK...	Name	...	Obj...	Data Type	Size	Scale	C...	Null	Un...	Value	Comment
1	☑	ID_MECZU	1	3214	integer				☐	☐		Identyfikator meczu
2	☐	ID_OBIEKTU	2	3215	integer				☐	☐		Identyfikator miejsca pracy (budynku)
3	☐	ID_GOSPO...	3	3216	varchar	30		☐	☐	☐		Identyfikator klubu grającego w lidze
4	☐	ID_GOSCIA	4	3217	varchar	30		☐	☐	☐		Identyfikator klubu grającego w lidze
5	☐	ID_SEDZI...	5	3218	integer				☐	☐		Identyfikator pracownika
6	☐	WYNIK	6	3219	integer				☐	☐		Ilość bramek strzelonych przez gosp...
7	☐	GOLE_GOS...	7	3220	integer				☐	☐		Ilość bramek strzelonych przez gościa

6. Tabela „Obiekt” – zawiera dane obiektów należących do klubów z ligi

Obiekt (DBA)

Columns | Constraints | Referencing Constraints | Indexes | Text Indexes | Triggers | Permissions | Dependent Views | Data

	PK...	Name	...	Obj...	Data Type	Size	Scale	C...	Null	Un...	Value	Comment
1	☑	ID_OBIEKTU	1	3228	integer				☐	☐		Identyfikator miejsca pracy (budynku)
2	☐	ID_KLUBU	2	3229	varchar	30		☐	☐	☐		Identyfikator klubu grającego w lidze
3	☐	NAZWA	3	3230	long varchar			☐	☑	☐		Nazwa obiektu
4	☐	ADRES	4	3231	varchar	60		☐	☐	☐		Adres obiektu

7. Tabela „Personel” – zawiera dane personelu poszczególnych klubów

Personel (DBA)

Columns | Constraints | Referencing Constraints | Indexes | Text Indexes | Triggers | Permissions | Dependent Views | Data

	PK...	Name	...	Obj...	Data Type	Size	Scale	C...	Null	Un...	Value	Comment
1	☑	ID_PRACO...	1	3236	integer				☐	☐		Identyfikator pracownika
2	☐	ID_OBIEKTU	2	3237	integer				☑	☐		Identyfikator miejsca pracy (budynku)
3	☐	ID_KLUBU	3	3238	varchar	30		☐	☑	☐		Identyfikator klubu grającego w lidze
4	☐	ID_STANO...	4	3239	integer				☐	☐		Identyfikator rodzaju stanowiska
5	☐	NAZWISKO	5	3240	varchar	30		☐	☐	☐		Nazwisko pracownika
6	☐	IMIE	6	3241	varchar	30		☐	☐	☐		Imię pracownika
7	☐	PESEL	7	3242	varchar	11		☐	☐	☐		Numer PESEL pracownika
8	☐	DATA_UR...	8	3243	date				☐	☐		Data urodzenia pracownika
9	☐	KRAJ	9	3244	varchar	30		☐	☐	☐		Narodowość pracownika
10	☐	ETAT	10	3245	smallint				☐	☐		Etat pracownika (liczba godzin pracy..
11	☐	PRZELOZO...	11	3246	varchar	80		☐	☐	☐		Przełożony pracownika

8. Tabela „Stanowisko” – zawiera rodzaje stanowisk obieranych przez personel klubów

Stanowisko (DBA)

Columns | Constraints | Referencing Constraints | Indexes | Text Indexes | Triggers | Permissions | Dependent Views | Data

	PK...	Name	...	Obj...	Data Type	Size	Scale	C...	Null	Un...	Value	Comment
1	☑	ID_STANO...	1	3253	integer				☐	☐		Identyfikator rodzaju stanowiska
2	☐	RODZAJ	2	3254	varchar	30		☐	☐	☐		Rodzaj stanowiska dla pracownika

9. Tabela „Typ_kontraktu” – zawiera typy zawieranych w lidze kontraktów

Typ_kontraktu (DBA)

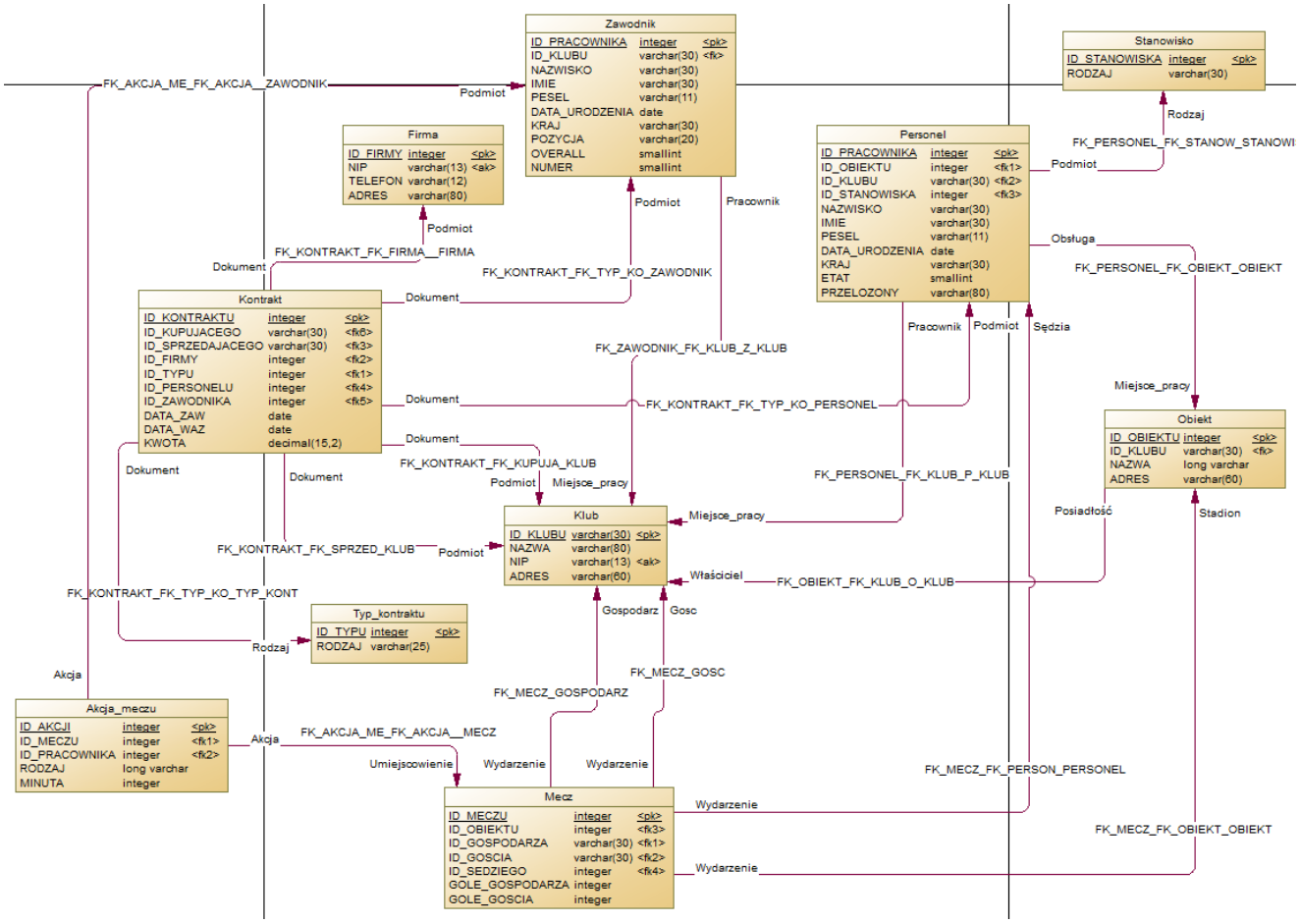
Columns | Constraints | Referencing Constraints | Indexes | Text Indexes | Triggers | Permissions | Dependent Views | Data

	PK...	Name	...	Obj...	Data Type	Size	Scale	C...	Null	Un...	Value	Comment
1	☑	ID_TYPU	1	3258	integer				☐	☐		Identyfiktor typu kontraktu
2	☐	RODZAJ	2	3259	varchar	25		☐	☐	☐		Rodzaj zawieranego kontraktu

10. Tabela „Zawodnik” – zawiera dane zawodników występujących w lidze

	ID_PRACOWNIKA	ID_KLUBU	NAZWISKO	IMIE	PESEL	ROK_URODZENIA	KRAJ	POZYCJA	OVERALL	NUMER
1	1	1	BECKER	ALLISON	123456789	1 990	Brasil	GK	90	1
2	2	1	ALEXANDER-ARNOLD	TRENT	234567890	1 998	England	DEF	88	66
3	3	1	VAN DIJK	VIRGIL	345678901	1 995	Netherlands	DEF	92	4
4	4	1	ROBERTSON	ANDY	456789012	1 992	England	DEF	89	26
5	5	1	MILNER	JAMES	567890123	1 994	England	MID	85	7
6	6	1	HENDERSON	JORDAN	678901234	1 988	England	MID	84	14
7	7	1	SALAH	MOHAMMED	789012345	1 990	Egypt	FOR	93	11
8	8	1	MANE	SADIO	890123456	1 993	Seneal	FOR	91	10
9	9	1	FIRMINO	ROBERTO	901234567	1 990	Brasil	FOR	90	9
10	10	1	ORIGI	DIVOCK	012345678	1 994	Belgium	FOR	86	27
11	11	1	SHAQIRI	XHERDAN	987654321	1 992	Switzerland	MID	86	32
12	12	3	MAHREZ	RYIAD	324647864	1 994	Algieria	MID	88	27
13	13	2	VARDY	JAMIE	345678351	1 989	England	FOR	88	7
14	14	5	POGBA	PAUL	452389012	1 994	France	MID	89	6
15	15	6	KANE	HARRY	827890123	1 992	England	FOR	91	10
16	16	7	TRAORE	ADAMA	670291234	1 990	Ghana	MID	86	32
17	17	8	AYEW	JORDAN	789888345	1 990	Ghana	FOR	85	7
18	18	9	LUNDSTRAM	JOHN	890903456	1 993	Sweden	MID	82	21
19	19	10	LACAZETTE	ALEXANDRE	901101567	1 993	France	FOR	88	9
20	20	3	DE BRUYNE	KEVIN	020345678	1 996	Belgium	MID	92	17
21	21	2	MADDISON	JAMIE	987632321	1 995	England	MID	89	10

c. Model fizyczny:



d. Zapytania:

1. Zapytanie „Transfery >= 10000000” – Zapytanie przedstawiające zawodników, którzy zostali sprowadzeni za kwotę przynajmniej 10000000 porządkujące po wielkości kwoty.

```
SELECT Z.NAZWISKO, Z.IMIE, Z.PESEL, K.KWOTA
      FROM Zawodnik Z
            INNER JOIN Kontrakt K on K.ID_ZAWODNIKA = Z.ID_PRACOWNIKA
      WHERE K.KWOTA >= 10000000
      ORDER BY 4
```

	NAZWISKO	IMIE	PESEL	KWOTA
1	ROBERTSON	ANDY	456789012	500 000 000,00
2	SALAH	MOHAMMED	789012345	10 000 000,00

2. Zapytanie „Zawodnicy LFC z Overallem >= 90” – Zapytanie przedstawiające wszystkich zawodników klubu LFC z overallem >= 90 porządkujące malejąco po overallu, a następnie rosnąco po nazwisku.

```
SELECT Z.Nazwisko, Z.Imie, K.Nazwa, Z.Pozycja, Z.OVERALL, Z.Numer
      FROM Zawodnik Z
            INNER JOIN Klub K on Z.ID_KLUBU = K.ID_KLUBU
      WHERE Z.ID_KLUBU = 1 and Z.OVERALL >= 90
      ORDER BY 5 DESC, 1 ASC
```

	Nazwisko	Imie	Nazwa	Pozycja	OVERALL	Numer
1	SALAH	MOHAMMED	FC LIVERPOOL	FOR	93	11
2	VAN DIJK	VIRGIL	FC LIVERPOOL	DEF	92	4
3	MANE	SADIO	FC LIVERPOOL	FOR	91	10
4	BECKER	ALLISON	FC LIVERPOOL	GK	90	1
5	FIRMINO	ROBERTO	FC LIVERPOOL	FOR	90	9

3. Zapytanie „Liczba zawodników w klubach” – Zapytanie przedstawiające listę klubów z liczbą zawodników w nich grających porządkowane względem liczby zawodników malejąco.

```
SELECT K.NAZWA, COUNT (Z.ID_PRACOWNIKA) AS LICZBA_ZAWODNIKOW
      FROM Zawodnik Z
            INNER JOIN Klub K on Z.ID_KLUBU = K.ID_KLUBU
      GROUP BY K.NAZWA
      ORDER BY 2 DESC
```

	NAZWA	LICZBA_ZAWODNIKOW
1	FC LIVERPOOL	11
2	LEICESTER CITY	2
3	MANCHESTER CITY	2
4	WOLVERHAMPTON	1
5	ARSENAL LONDON	1
6	CRYSTAL PALACE	1
7	SHEFFIELD UNITED	1
8	MANCHESTER UNITED	1
9	TOTTENHAM HOTSPUR	1

4. Zapytanie „Lista najczęściej wybieranych typów umów” – Zapytanie przedstawiające listę najczęściej wybieranych typów kontraktów porządkowane malejąco.

```
SELECT T.RODZAJ,
SUM(case when K.ID_TYPU is not null then 1 else 0 end) as ILE_RAZY
      FROM Kontrakt K
            INNER JOIN Typ_kontraktu T on K.ID_TYPU = T.ID_TYPU
      GROUP BY T.RODZAJ
      ORDER BY ILE_RAZY DESC
```

	RODZAJ	ILE_RAZY
1	Transfer permanentny	3
2	Wypozyczenie	2
3	Umowa o dzielo	2
4	Umowa zlecenie	2
5	Umowa o prace	1

5. Zapytanie „Lista wydatków klubów” – Zapytanie przedstawiające listę klubów i ich wydatkami na podstawie wszystkich kontraktów porządkowane względem wydatków.

```
SELECT Kl.NAZWA,
      SUM(case when K.kwota is not null then K.kwota end) as WYDANE_PIENIADZE
      FROM Kontrakt K
            INNER JOIN Klub Kl on K.ID_KUPUJACEGO = Kl.ID_KLUBU
      GROUP BY Kl.NAZWA
      ORDER BY 2 DESC
```

	NAZWA	WYDANE_PIENIADZE
1	FC LIVERPOOL	1 519 000 000,00
2	ARSENAL LONDON	100 000 000,00
3	LEICESTER CITY	100 000 000,00
4	SHEFFIELD UNITED	100 000 000,00
5	MANCHESTER UNITED	50 000 000,00

6. Zapytanie „Ostatnie 2,5 roku ważności kontraktów” – Zapytanie przedstawiające listę kontraktów ważnych jeszcze przez 2,5 roku porządkowane względem krótszego terminu ważności. Kontrakty nieaktualne nie są pokazywane.

```
SELECT K.ID_KONTRAKTU, Kl.NAZWA, K.DATA_WAZ
      FROM Kontrakt K
            INNER JOIN Klub Kl on K.ID_KUPUJACEGO = Kl.ID_KLUBU
      WHERE DATEDIFF(month, NOW(), K.DATA_WAZ)<30 and K.DATA_WAZ >= TODAY()
      ORDER BY K.DATA_WAZ
```

	ID_KONTRAKTU	NAZWA	DATA_WAZ
1	6	MANCHESTER UNITED	2020-06-01
2	7	FC LIVERPOOL	2021-12-30
3	8	FC LIVERPOOL	2022-01-01
4	5	LEICESTER CITY	2022-01-01

7. Zapytanie „Lista klubów z liczbą goli strzelonych u siebie większą od 0” – Zapytanie przedstawiające listę klubów które strzeliły więcej niż 0 goli u siebie porządkowane względem ilości strzelonych goli i alfabetycznie.

```
SELECT K.NAZWA,
      SUM(case when M.GOLE_GOSPODARZA is not null then M.GOLE_GOSPODARZA end) as
      GOLE_U_SIEBIE
      FROM Mecz M
            INNER JOIN Klub K on M.ID_GOSPODARZA = K.ID_KLUBU
      GROUP BY K.NAZWA
      HAVING GOLE_U_SIEBIE > 0
      ORDER BY GOLE_U_SIEBIE DESC, K.NAZWA ASC
```

	NAZWA	GOLE_U_SIEBIE
1	FC LIVERPOOL	6
2	MANCHESTER UNITED	6
3	CHELSEA LONDON	4
4	CRYSTAL PALACE	3
5	TOTTENHAM HOTSPUR	2
6	SHEFFIELD UNITED	1
7	WOLVERHAMPTON	1

8. Zapytanie „Trenerzy” – Zapytanie przedstawiające wszystkich członków personelu będących trenerami w klubach porządkujące po nazwisku.

```
SELECT P.Nazwisko, P.Imie, P.PESEL, K.NAZWA as NAZWA_KLUBU
      FROM Personel P
            INNER JOIN Stanowisko S on P.ID_STANOWISKA = S.ID_STANOWISKA
            INNER JOIN Klub K on P.ID_KLUBU = K.ID_KLUBU
      WHERE P.ID_STANOWISKA = 2
      ORDER BY 1
```

	Nazwisko	Imie	PESEL	NAZWA_KLUBU
1	Guardiola	Pep	654876928	MANCHESTER CITY
2	Klopp	Jurgen	098567422	FC LIVERPOOL
3	Lampard	Frank	725638298	CHELSEA LONDON
4	Mourinho	Jose	262839048	TOTTENHAM HOTSPUR
5	Rodgers	Brendan	524387153	LEICESTER CITY
6	Santos	Frederico	728292827	WOLVERHAMPTON
7	Solskjaer	Ole	182739380	MANCHESTER UNITED

9. Zapytanie „Lista wszystkich transferów” – Zapytanie przedstawiające listę wszystkich transferów na podstawie kontraktów między klubami porządkowane względem wysokości kwoty.

```
SELECT K1.NAZWA AS NAZWA_KUPUJACEGO, T.RODZAJ, K2.NAZWA AS NAZWA_SPRZEDAJACEGO,
Z.NAZWISKO, K.KWOTA
      FROM Kontrakt K
            INNER JOIN Klub K1 on K.ID_KUPUJACEGO = K1.ID_KLUBU
            INNER JOIN Klub K2 on K.ID_SPRZEDAJACEGO = K2.ID_KLUBU
            INNER JOIN Zawodnik Z on K.ID_ZAWODNIKA = Z.ID_PRACOWNIKA
            INNER JOIN Typ_kontraktu T on K.ID_TYPU = T.ID_TYPU
      ORDER BY K.KWOTA DESC
```

	NAZWA_KUPUJACEGO	RODZAJ	NAZWA_SPRZEDAJACEGO	NAZWISKO	KWOTA
1	FC LIVERPOOL	Wypozyczenie	MANCHESTER CITY	ROBERTSON	500 000 000,00
2	FC LIVERPOOL	Transfer permanentny	WOLVERHAMPTON	SALAH	10 000 000,00
3	FC LIVERPOOL	Wypozyczenie	ARSENAL LONDON	VAN DIJK	4 000 000,00
4	FC LIVERPOOL	Transfer permanentny	MANCHESTER CITY	BECKER	3 000 000,00
5	FC LIVERPOOL	Transfer permanentny	MANCHESTER UNITED	ORIGI	2 000 000,00

e. Widoki:

1. Widok „Angole” – Widok przedstawiający listę zawodników i członków personelu pochodzących z Anglii.

```
CREATE VIEW "DBA"."Angole" AS(
SELECT Z.NAZWISKO, Z.IMIE, Z.ROK_URODZENIA
      FROM Zawodnik Z
      WHERE Z.KRAJ = 'England'
UNION
SELECT P.NAZWISKO, P.IMIE, P.ROK_URODZENIA
    FROM Personel P
    WHERE P.KRAJ = 'England')
```

	NAZWISKO	IMIE	ROK_URODZENIA
1	ALEXANDER-ARNOLD	TRENT	1 998
2	ROBERTSON	ANDY	1 992
3	MILNER	JAMES	1 994
4	HENDERSON	JORDAN	1 988
5	VARDY	JAMIE	1 989
6	KANE	HARRY	1 992
7	MADDISON	JAMIE	1 995
8	Bool	Andy	1 960
9	Rodgers	Brendan	1 971
10	Atkinson	Rowan	1 960
11	Cahill	Gary	1 970

2. Widok „ROK_UROD ZENIA <1975;1990>” – Widok przedstawiający listę zawodników i członków personelu urodzonych w latach 1975-1990.

```
CREATE VIEW "DBA"."Rok_urodzenia <1975;1990>" AS(
SELECT Z.NAZWISKO, Z.IMIE, Z.KRAJ, Z.ROK_URODZENIA
      FROM Zawodnik Z
      WHERE Z.ROK_URODZENIA >= 1975 and Z.ROK_URODZENIA <= 1990
UNION
SELECT P.NAZWISKO, P.IMIE, P.KRAJ, P.ROK_URODZENIA
      FROM Personel P
      WHERE P.ROK_URODZENIA >= 1975 and P.ROK_URODZENIA <= 1990)
```

	NAZWISKO	IMIE	KRAJ	ROK_URODZENIA
1	BECKER	ALLISON	Brasil	1 990
2	HENDERSON	JORDAN	England	1 988
3	SALAH	MOHAMMED	Egypt	1 990
4	FIRMINO	ROBERTO	Brasil	1 990
5	VARDY	JAMIE	England	1 989
6	TRAORE	ADAMA	Ghana	1 990
7	AYEW	JORDAN	Ghana	1 990
8	Lampard	Frank	Russia	1 980
9	Solskjaer	Ole	Norway	1 978
10	Mourinho	Jose	Portugal	1 977

3. Widok „Obiekty w Londynie” – Widok przedstawiający listę obiektów należących do klubów znajdujących się w Londynie.

```
CREATE VIEW "DBA"."Obiekty w Londynie" AS(
SELECT O.NAZWA as NAZWA_OBIEKTU, K.NAZWA as KLUB, O.ADRES
      FROM Obiekt O
            INNER JOIN Klub K on O.ID_KLUBU = K.ID_KLUBU
      WHERE O.ADRES like '%London%')
```

	NAZWA_OBIEKTU	KLUB	ADRES
1	Stamford Bridge	CHELSEA LONDON	Stamford Bridge 1, 00-004 London
2	Wembley Stadium	TOTTENHAM HOTSPUR	Wembley Street 43, 00-004 London
3	Emirates Stadium	SHEFFIELD UNITED	Gunners Square 343, 00-004 London
4	Crystal Lake Stadium	CRYSTAL PALACE	Crystal Road 22, 00-004 London

f. Funkcje:

1. Funkcja „ILU_GRA_W_KLUBIE” – Funkcja przyjmująca ID klubu, a zwracająca liczbę zawodników do tego klubu przypisanych.

```
ALTER FUNCTION "DBA"."ILU_GRACZY_W_KLUBIE" (in klub integer)
RETURNS integer
BEGIN
      DECLARE "ile" integer;
      SET ile = (SELECT COUNT(ID_PRACOWNIKA)
                  FROM ZAWODNIK Z
                  WHERE Z.ID_KLUBU = klub);
      RETURN "ile";
END

SELECT ILU_GRA_W_KLUBIE(1) AS ILOSC_DLA_FC_LIVERPOOL
```

	ILOSC_DLA_FC_LIVERPOOL
1	11

2. Funkcja „CZYJ_TO_PESEL” – Funkcja przyjmująca PESEL, a zwracająca nazwisko i imię piłkarza, jeśli jego PESEL jest w bazie lub „brak” jeśli go nie ma.

```
ALTER FUNCTION "DBA"."CZYJ_TO_PESEL" (in pesel varchar(10))
RETURNS varchar(20)
BEGIN
      DECLARE odpowiedz varchar(20);
      DECLARE kursor DYNAMIC SCROLL CURSOR FOR
            SELECT PESEL
                  FROM Zawodnik;
      DECLARE i INTEGER;
      DECLARE temp varchar(10);
      SET i = 0;
      SET odpowiedz = '';
      OPEN kursor;
      WHILE i <(SELECT COUNT(ID_PRACOWNIKA) FROM Zawodnik)
            LOOP
                  SET i = i + 1;
                  FETCH NEXT kursor INTO temp;
                  IF (pesel = temp) THEN SET odpowiedz =
                  (SELECT Nazwisko FROM Zawodnik WHERE ID_PRACOWNIKA = (i))
                  + ' ' + (SELECT Imie FROM Zawodnik WHERE ID_PRACOWNIKA = (i));
                  END IF;
            END LOOP;
      CLOSE kursor;
      IF (odpowiedz = '') THEN SET odpowiedz = 'brak';
      END IF;
      RETURN odpowiedz;
END

SELECT CZYJ_TO_PESEL('123456789') AS ODPOWIEDZ
```

	ODPOWIEDZ
1	BECKER ALLISON

```
SELECT CZYJ_TO_PESEL('000111222') AS ODPOWIEDZ
```

	ODPOWIEDZ
1	brak

3. Funkcja „ILE_GOLI” – Funkcja przyjmująca ID klubu, a zwracająca sumę strzelonych goli we wszystkich rozegranych meczach tego klubu.

```
CREATE FUNCTION ILE_GOLI(IN klub INTEGER)
RETURNS INTEGER
BEGIN
      DECLARE wynik INTEGER;
      DECLARE kursor1 DYNAMIC SCROLL CURSOR FOR SELECT ID_GOSPODARZA FROM Mecz;
      DECLARE kursor2 DYNAMIC SCROLL CURSOR FOR SELECT ID_GOSCIA FROM Mecz;
      DECLARE i INTEGER;
      DECLARE temp1 INTEGER;
      DECLARE temp2 INTEGER;
      SET wynik = 0;
      SET i = 0;
      OPEN kursor1;
      OPEN kursor2;
      WHILE i < (SELECT COUNT(ID_MECZU) FROM Mecz)
            LOOP
                  SET i = i + 1;
                  FETCH NEXT kursor1 INTO temp1;
                  FETCH NEXT kursor2 INTO temp2;
                  IF (temp1 = klub) THEN SET wynik = wynik +
                        (SELECT GOLE_GOSPODARZA FROM Mecz WHERE ID_MECZU = i);
                  END IF;
                  IF (temp2 = klub) THEN SET wynik = wynik +
                        (SELECT GOLE_GOSCIA FROM Mecz WHERE ID_MECZU = i);
                  END IF;
            END LOOP;
      CLOSE kursor2;
      CLOSE kursor1;
      RETURN wynik;
END

SELECT ILE_GOLI(1) AS SUMA
```

	SUMA
1	9

g. Procedury:

1. Procedura „DODAJ_MECZ” – Procedura dodaje mecz do bazy, jeśli mecz rozgrywany jest na boisku gospodarza.

```
CREATE PROCEDURE DODAJ_MECZ(IN ID_m INTEGER, IN ID_o INTEGER, IN ID_gosp INTEGER,
IN ID_gosc INTEGER, IN ID_s INTEGER, IN gole_gosp INTEGER, IN gole_gosc INTEGER)

BEGIN
      IF (ID_gosp=(SELECT ID_KLUBU FROM Obiekt WHERE Obiekt.ID_OBIEKTU=ID_o)) THEN
      INSERT INTO Mecz(ID_MECZU, ID_OBIEKTU, ID_GOSPODARZA, ID_GOSCIA,
            ID_SEDZIEGO, GOLE_GOSPODARZA, GOLE_GOSCIA)
      VALUES(ID_m, ID_o, ID_gosp, ID_gosc, ID_s, gole_gosp, gole_gosc);
      ELSE
            MESSAGE 'zly obiekt!' TO CLIENT;
      END IF;
END

CALL DODAJ_MECZ(11, 3, 2, 10, 9, 2, 0)
```

11	11	3	2	10	9	2	0

```
CALL DODAJ_MECZ(12, 3, 3, 10, 9, 2, 0)
```

zly obiekt!
Execution time: 0.008 seconds
Procedure completed

2. Procedura „TABELA_ZAWODNIKOW” – Procedura tworzy nową tabelę zawierającą zestawienie nazw klubów i liczbie zawodników, którzy w tym klubie grają, posortowanych malejąco względem liczby zawodników. Procedura wykorzystuje funkcję ILU_GRA_W_KLUBIE().

```
CREATE PROCEDURE TABELA_ILOSCI_ZAWODNIKOW()
BEGIN
      DECLARE i INTEGER;
      SET i = 0;
      CREATE TABLE Liczba_Zawodnikow(Klub VARCHAR(30), Liczba INTEGER);
      WHILE i < (SELECT COUNT(ID_KLUBU) FROM Klub) LOOP
            SET i = i + 1;
            INSERT INTO Liczba_Zawodnikow VALUES((SELECT Nazwa
            FROM Klub WHERE ID_KLUBU = i), ILU_GRACZY_W_KLUBIE(i));
      END LOOP;
      SELECT * FROM Liczba_Zawodnikow ORDER BY Liczba DESC, Klub ASC;
      DROP TABLE Liczba_Zawodnikow;
END

CALL TABELA_ILOSCI_ZAWODNIKOW()
```

	Klub	Liczba
1	FC LIVERPOOL	11
2	LEICESTER CITY	2
3	MANCHESTER CITY	2
4	WOLVERHAMPTON	1
5	ARSENAL LONDON	1
6	CRYSTAL PALACE	1
7	SHEFFIELD UNITED	1
8	MANCHESTER UNITED	1
9	TOTTENHAM HOTSPUR	1
10	CHELSEA LONDON	0

Liczba_Zawodnikow DBA Table

	PK...	Name	ID	Obj. ID	Data Type	Size	Scale	C...	Null	Un...
1	☐	Klub	1	3313	varchar	30		☐	☑	☐
2	☐	Liczba	2	3314	integer				☑	☐

3. Procedura „OVERALL_UPGRADE” – Procedura zwiększa Overall zawodników danego klubu o 1, jeśli jest on mniejszy niż 99.

```
CREATE PROCEDURE OVERALL_UPGRADE(IN ID_k INTEGER)

BEGIN
      DECLARE kursor DYNAMIC SCROLL CURSOR FOR SELECT OVERALL FROM Zawodnik ;
      DECLARE i INTEGER;
      DECLARE temp INTEGER;
      SET i = 0;
      OPEN kursor;
      WHILE i < (SELECT COUNT(ID_PRACOWNIKA) FROM Zawodnik)
      LOOP
            SET i = i + 1;
            FETCH NEXT kursor INTO temp;
            IF (temp < 99) THEN
                 UPDATE Zawodnik SET OVERALL = (temp + 1) WHERE ID_PRACOWNIKA = i;
            ELSE
                 MESSAGE 'Zawodnik ma maksymalny OVERALL!' TO CLIENT;
            END IF;
      END LOOP;
    CLOSE kursor;
END
```

PRZED:

	ID_PRACOWNIKA	ID_KLUBU	NAZWISKO	IMIE	PESEL	ROK_URODZENIA	KRAJ	POZYCJA	OVERALL	NUMER
1	1	1	BECKER	ALLISON	123456789	1 990	Brasil	GK	90	1
2	2	1	ALEXANDER-ARNOLD	TRENT	234567890	1 998	England	DEF	88	66
3	3	1	VAN DIJK	VIRGIL	345678901	1 995	Netherlands	DEF	92	4
4	4	1	ROBERTSON	ANDY	456789012	1 992	England	DEF	89	26
5	5	1	MILNER	JAMES	567890123	1 994	England	MID	85	7
6	6	1	HENDERSON	JORDAN	678901234	1 988	England	MID	84	14
7	7	1	SALAH	MOHAMMED	789012345	1 990	Egypt	FOR	93	11
8	8	1	MANE	SADIO	890123456	1 993	Senegal	FOR	91	10
9	9	1	FIRMINO	ROBERTO	901234567	1 990	Brasil	FOR	90	9
10	10	1	ORIGI	DIVOCK	012345678	1 994	Belgium	FOR	86	27
11	11	1	SHAQIRI	XHERDAN	987654321	1 992	Switzerland	MID	86	32

CALL OVERALL_UPGRADE(1)

	ID_PRACOWNIKA	ID_KLUBU	NAZWISKO	IMIE	PESEL	ROK_URODZENIA	KRAJ	POZYCJA	OVERALL	NUMER
1	1	1	BECKER	ALLISON	123456789	1 990	Brasil	GK	91	1
2	2	1	ALEXANDER-ARNOLD	TRENT	234567890	1 998	England	DEF	89	66
3	3	1	VAN DIJK	VIRGIL	345678901	1 995	Netherlands	DEF	93	4
4	4	1	ROBERTSON	ANDY	456789012	1 992	England	DEF	90	26
5	5	1	MILNER	JAMES	567890123	1 994	England	MID	86	7
6	6	1	HENDERSON	JORDAN	678901234	1 988	England	MID	85	14
7	7	1	SALAH	MOHAMMED	789012345	1 990	Egypt	FOR	94	11
8	8	1	MANE	SADIO	890123456	1 993	Senegal	FOR	92	10
9	9	1	FIRMINO	ROBERTO	901234567	1 990	Brasil	FOR	91	9
10	10	1	ORIGI	DIVOCK	012345678	1 994	Belgium	FOR	87	27
11	11	1	SHAQIRI	XHERDAN	987654321	1 992	Switzerland	MID	87	32

h. Triggery:

1. Trigger „ZDUBLOWANE_NUMERY” – Wyzwalacz sprawdzający przy dodaniu zawodnika do bazy, czy ten nie zdublował numeru innego zawodnika z drużyny.

```
CREATE TRIGGER ZDUBLOWANE_NUMERY BEFORE INSERT ON Zawodnik
REFERENCING NEW AS Nowy
FOR EACH ROW
BEGIN
      DECLARE kursor DYNAMIC SCROLL CURSOR FOR (SELECT NUMER
            FROM Zawodnik WHERE Zawodnik.ID_KLUBU = Nowy.ID_KLUBU);
      DECLARE tem INTEGER;
      DECLARE i INTEGER;
      DECLARE holder INTEGER;
      SET tem = 0;
      SET i = 0;
      OPEN kursor;
      WHILE i < (SELECT COUNT(*) FROM Zawodnik
            WHERE Zawodnik.ID_KLUBU = Nowy.ID_KLUBU)
      LOOP
            SET i = i + 1;
            FETCH NEXT kursor INTO holder;
            IF (holder = Nowy.NUMER) THEN SET tem = tem + 1;
            END IF;
      END LOOP;
      CLOSE kursor;
      IF (tem <> 0) THEN
            MESSAGE 'Dodany zawodnik zdublowal numer na koszulce!' TO CLIENT;
      END IF;
END;
```

PO DODANIU:

22	22	3	SILVA	DAVID	927329102	1 990	Spain	FOR	90	17

KTÓRY DUBLUJE NUMER:

20	20	3	DE BRUYNE	KEVIN	020345678	1 996	Belgium	MID	93	17

Dodany zawodnik zdublowal numer na koszulce!
1 row(s) inserted
Execution time: 0.013 seconds

2. Trigger „SPRAWDZ_DATE” – Wyzwalacz sprawdzający czy dodawany kontrakt nie jest dodawany z datą późniejsza niż aktualna.

```
CREATE TRIGGER SPRAWDZ_DATE
BEFORE INSERT ON Kontrakt
REFERENCING NEW AS NOWY
FOR EACH ROW
BEGIN
      IF NOWY.DATA_ZAW > now() THEN
      MESSAGE 'Dodajesz kontrakt z data pozniejsza niz dzisiejsza!' TO CLIENT;
      END IF;
END
```

PO DODANIU:

11	11	7	(NULL)	5	8	1	(NULL)	2020-02-01	2024-11-11	50 000 000,00

Dodajesz kontrakt z data pozniejsza niz dzisiejsza!
1 row(s) inserted
Execution time: 0.01 seconds

3. Trigger „SPRAWDZ_MINUTE” – Wyzwalacz sprawdzający czy dodawana akcja meczu dzieje się w regulaminowych 90 minutach.

```
CREATE TRIGGER SPRAWDZ_MINUTE
BEFORE INSERT ON Akcja_meczu
REFERENCING NEW AS NOWY
FOR EACH ROW
BEGIN
      IF NOWY.MINUTA > 90 THEN
      MESSAGE 'Akcja dzieje sie w nieregulaminowym czasie!' TO CLIENT;
      END IF;
END
```

PO DODANIU:

11	11	1	3	FAUL	92

Akcja dzieje sie w nieregulaminowym czasie!
1 row(s) inserted
Execution time: 0.013 seconds