

Bazy danych

Projekt zaliczeniowy

Temat projektu: **SKOKI NARCIARSKIE**

Autor projektu: **MARIUSZ JEDRZEJEWSKI**

Numer indeksu : **1 2 8 0 5 9**

Kierunek / rok: **INFORMATYKA I-ego STOPNIA, 2 ROK, 4 SEMESTR**

Grupa lab. : **3**

TEMATYKA PROJEKTU:

Tematem projektu są **skoki narciarskie**. Rozważane są w nim aspekty takie jak spis skoczków biorących udział w różnych zawodach, trenerzy którzy prawują nad nimi pieczę, reprezentacje które są reprezentowane przez skoczków oraz ich trenerzy, skocznie narciarskie, rodzaje zawodów które pojawiają się w skokach narciarskich oraz różne zawody na przestrzeni ostatnich kilku lat.

W projekcie starano się zachować **aspekty rzeczywistości** chociaż jak wiadomo, pewne rzeczy mogą odstawać (np. wpisany do listy skoczków emerytowany polski skoczek, **Adam Małysz**) ale nie sprawiają one kłopotu bo zachowują mimo wszystko **spójność logiczną** i nie ma takich przypadków **wiele**.

Projekt ten został w całości wykonany w programie:

SQL Developer Data Modeler

Został on przygotowany w ramach przedmiotu **Bazy danych** w roku akademickim 2018/2019 odbywającego się w ramach programu kierunku **Informatyka** na semestrze **czwartym, drugiego roku studiów inżynierskich** na **Politechnice Częstochowskiej** przeprowadzonych przez **Pana doktora Krzysztofa Kaczmarka**.

TABELE:

Do zrealizowania zadania stworzone zostały odpowiednie tabele, jesteśmy w stanie wprowadzić do naszej bazy danych takie informacje takie jak: **reprezentacje narodowe, skoczkowie, trenerzy, skocznie, rodzaje zawodów oraz zawody.**

W kolejności jakiej powinny utworzone tabele jest zaprezentowane poniżej:

```
create table Reprezentacje
(
    id_reprezentacji      NUMBER(3) CONSTRAINT id_reprezentacji_pk PRIMARY KEY,
    nazwa_kraju           VARCHAR(30) constraint rep_nazwa_kraju_unique UNIQUE
);

create table Skoczkowie
(
    id_skoczka            NUMBER(3) CONSTRAINT id_skoczka_pk PRIMARY KEY,
    imie                  VARCHAR(40) CONSTRAINT Skoczkowie_imie_notnull NOT NULL,
    nazwisko              VARCHAR(40) CONSTRAINT Skoczkowie_nazwisko_notnull NOT NULL,
    data_urodzenia        DATE CONSTRAINT Skoczkowie_du_check
                        CHECK(data_urodzenia >= to_date('1960.01.01', 'YYYY.MM.DD')),
    narodowosc            VARCHAR(30),
    CONSTRAINT            Skoczkowie_Reprezentacje FOREIGN KEY (narodowosc) REFERENCES Reprezentacje (nazwa_kraju)
);

create table Trenerzy
(
    id_trenera            NUMBER(3) CONSTRAINT id_trenera_pk PRIMARY KEY,
    imie                  VARCHAR(40) CONSTRAINT Trenerzy_imie_notnull NOT NULL,
    nazwisko              VARCHAR(40) CONSTRAINT Trenerzy_nazwisko_notnull NOT NULL,
    data_urodzenia        DATE CONSTRAINT Trenerzy_data_urodzenia_check
                        CHECK(data_urodzenia >= to_date('1940.01.01', 'YYYY.MM.DD')),
    narodowosc            VARCHAR(30),
    CONSTRAINT            Trenerzy_Reprezentacje FOREIGN KEY (narodowosc) REFERENCES Reprezentacje (nazwa_kraju)
    --narodowosc w sensie reprezentacji która reprezentuje!!!
);

create table Skocznie
(
    id_skoczni            NUMBER(3) CONSTRAINT Skocznie_id_primarykey PRIMARY KEY,
    nazwa                  VARCHAR(50) CONSTRAINT Skocznie_nazwa_notnull NOT NULL,
    miejscowosc           VARCHAR(40) CONSTRAINT Skocznie_miejscowosc_notnull NOT NULL,
    panstwo               VARCHAR(30) CONSTRAINT Skocznie_panstwo_notnull NOT NULL,
    punkt_konstrukcyjny   FLOAT(4),
    rekord_skoczni        FLOAT(4),
    id_rekordzisty        NUMBER(3),
    CONSTRAINT            rekord_skoczni_p FOREIGN KEY (id_rekordzisty) REFERENCES Skoczkowie (id_skoczka)
);

create table Zawody_rodzaj -- typ zawodow
(
    id_rodzaj             NUMBER(2) CONSTRAINT id_rodzaj_primarykey PRIMARY KEY,
    nazwa                  VARCHAR2(45 BYTE) CONSTRAINT typ_nazwa_notnull NOT NULL
);

create table Zawody
(
    id_zawodow            NUMBER(2) CONSTRAINT zawody_id_primarykey PRIMARY KEY,
    nazwa                  VARCHAR2(45 BYTE) CONSTRAINT zawody_nazwa_notnull NOT NULL,
    data_roz poczenia     DATE,
    data_zakonczenia      DATE,
    id_rodzaj              NUMBER(2) CONSTRAINT zawody_id_rodzaj_notnull NOT NULL,
    CONSTRAINT            zawody_id_rodzaj FOREIGN KEY (id_rodzaj) REFERENCES Zawody_rodzaj (id_rodzaj)
);
```

```

create table Skoki
(
    -- id_skoki odpowiada za Zawody.id_zawodow
    id_skoki          NUMBER(2),
    id_skoczka_skok   NUMBER(3),
    skok_1            FLOAT(53),
    wiatr_1           FLOAT(53),
    skok_2            FLOAT(53),
    wiatr_2           FLOAT(53),
    punkty_razem      FLOAT(53),
    CONSTRAINT        id_zawody FOREIGN KEY (id_skoki) REFERENCES Zawody (id_zawodow),
    CONSTRAINT        id_skoczka FOREIGN KEY (id_skoczka_skok) REFERENCES Skoczkowie (id_skoczka)
);

```

W tabeli **REPREZENTACJE** mamy dane takie jak:

- **id_reprezentacji** – wartość typu **NUMBER** do 3 cyfr, **klucz główny**
- **nazwa_kraju** – wartość typu **VARCHAR** do 30 znaków, unikalne (**UNIQUE**)

W tabeli **SKOCZKOWIE** mamy dane takie jak:

- **id_skoczka** – wartość typu **NUMBER** do 3 cyfr, **klucz główny**
- **imie** – wartość typu **VARCHAR** do 40 znaków, nie może być puste (**NOT NULL**)
- **nazwisko** – wartość typu **VARCHAR** do 40 znaków, nie może być puste (**NOT NULL**)
- **data_urodzenia** – wartość typu **DATE** z warunkiem sprawdzającym, że skoczek urodził się najwcześniej 1 stycznia 1960 roku (**CHECK**)
- **narodowość** – wartość typu **VARCHAR** do 30 znaków
- **Skoczkowie_Reprezentacje** – **klucz obcy** odwołujący się do tabeli **Reprezentacje**, dotyczy narodowości skoczka

W tabeli **TRENERZY** mamy dane takie jak:

- **id_trenera** – wartość typu **NUMBER** do 3 cyfr, **klucz główny**
- **imie** – wartość typu **VARCHAR** do 40 symboli, nie może być puste (**NOT NULL**)
- **nazwisko** – wartość typu **VARCHAR** do 40 znaków, nie może być puste (**NOT NULL**)
- **data_urodzenia** – wartość typu **DATE** z warunkiem sprawdzającym, że trener urodził się najwcześniej 1 stycznia 1940 roku (**CHECK**)
- **narodowość** – wartość typu **VARCHAR** do 30 symboli
- **Trenerzy_Reprezentacje** – **klucz obcy** odwołujący się do tabeli **Reprezentacje**, dotyczy narodowości trenera

W tabeli **SKOCZNIE** mamy dane takie jak:

- **id_skoczni** – wartość typu **NUMBER** do 3 symboli, **klucz główny**
- **nazwa** – wartość typu **VARCHAR** do 50 symboli, nie może być puste (**NOT NULL**)
- **miejscowość** – wartość typu **VARCHAR** do 40 symboli, nie może być puste (**NOT NULL**)
- **panstwo** – wartość **VARCHAR** do 30 symboli, nie może być puste (**NOT NULL**)
- **punkt_konstrukcyjny** – wartość typu **FLOAT** o precyzji n=53
- **rekord_skoczni** – wartość typu **FLOAT** o precyzji n=53
- **id_rekordzisty** – wartość typu **NUMBER** do 3 cyfr
- **rekord_skoczni_p** – **klucz obcy** odwołujący się do tabeli **Skoczkowie**, dotyczy ID skoczka który jest obecnym rekordzistą na danej skoczni

(w przypadku gdy w wartości id_rekordzisty pojawia się NULL, oznacza to, że dany skoczek albo nie skacze lub nie znalazł się w spisie skoczków zawartych w tej bazie danych)

W tabeli **ZAWODY_RODZAJ** mamy dane takie jak:

- **id_rodzaj** – wartość typu **NUMBER** do 2 cyfr, **klucz główny**
- **nazwa** – wartość typu **VARCHAR** do 45 symboli, nie może być puste (**NOT NULL**)

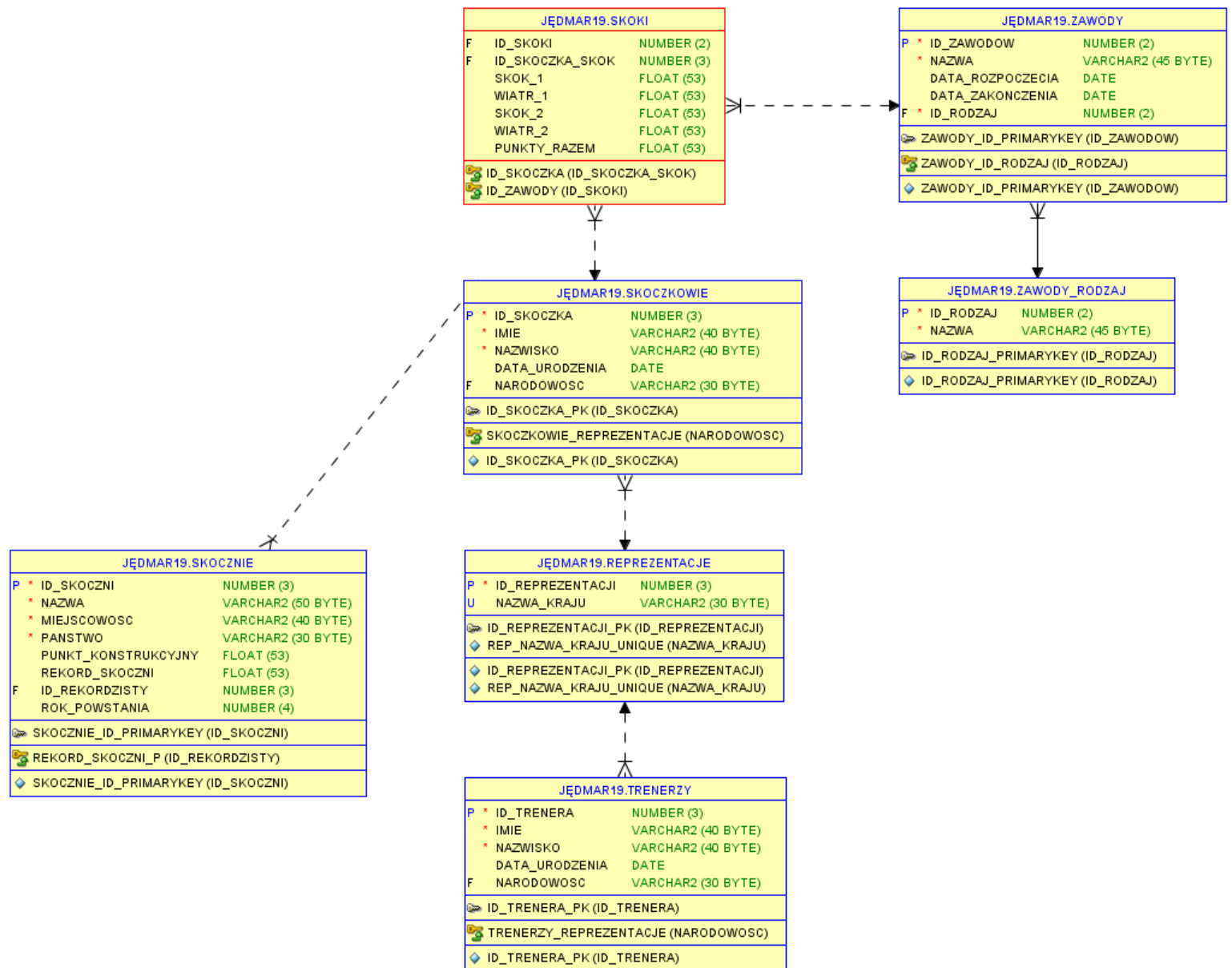
W tabeli **ZAWODY** mamy dane takie jak:

- **id_zawodow** – wartość typu **NUMBER** do 2 cyfr, **klucz główny**
- **nazwa** – wartość typu **VARCHAR** do 45 symboli, nie może być puste (**NOT NULL**)
- **data_rozpoczecia** – wartość typu **DATE**
- **data_zakonczenia** – wartość typu **DATE**
- **id_rodzaj** – wartość typu **NUMBER** do 2 cyfr, nie może być puste (**NOT NULL**)
- **zawody_id_rodzaj** – **klucz obcy** odwołujący się do tabeli **Zawody_rodzaj**, dotyczy ID rodzaju danych zawodów

W tabeli **SKOKI** mamy dane takie jak:

- **id_skoki** – wartość typu **NUMBER** do 2 cyfr
- **id_skoczka_skok** – wartość typu **NUMBER** do 3 cyfr
- **skok_1** – wartość typu **FLOAT** o precyzji n=53
- **wiatr_1** – wartość typu **FLOAT** o precyzji n=53
- **skok_2** – wartość typu **FLOAT** o precyzji n=53
- **wiatr_2** – wartość typu **FLOAT** o precyzji n=53
- **punkty_razem** – wartość typu **FLOAT** o precyzji do 4 cyfr
- **id_zawody** – **klucz obcy** odwołujący się do tabeli **Zawody**, dotyczy numeru ID zawodów w jakich te skoki miały miejsce
- **id_skoczka** – **klucz obcy** odwołujący się do tabeli **Skoczkowie**, dotyczy numeru ID zawodników którzy dany skok oddali
- **id_reprez** – **klucz obcy** odwołujący się do tabeli **Reprezentacje**, dotyczy numeru ID reprezentacji dla jakiej skok był oddawany przez skoczka

MODEL RELACYJNY:



PRZYKŁADOWE DANE WPROWADZANE DO TABEL:

(reszta z nich znajduje się w pliku SQL dołączonym do projektu w ramach odpowiedniego działania i testowania załączonych zagadnień)

...

INSERT INTO Reprezentacje VALUES(130, 'POLSKA');

INSERT INTO Reprezentacje VALUES(131, 'ROSJA');

INSERT INTO Reprezentacje VALUES(132, 'RUMUNIA');

...

...

INSERT INTO Trenerzy VALUES(10, 'MICHAL', 'DOLEZAL', '1978.03.11', 'POLSKA');

INSERT INTO Skoczkowie VALUES(101, 'KAMIL', 'STOCH', '1987.05.25', 'POLSKA');

INSERT INTO Skoczkowie VALUES(102, 'ADAM', 'MALYSZ', '1977.12.03', 'POLSKA');

...

...

INSERT INTO Skocznie VALUES(19, 'MIDTSTUBAKKEN K95', 'OSLO', 'NORWEGIA', 95, 110, 176);

INSERT INTO Skocznie VALUES(20, 'VIKERSUNDBAKKEN K200', 'VIKERSUND', 'NORWEGIA', 200, 253.5, 174);

INSERT INTO Skocznie VALUES(21, 'SNIEZYINKA K125', 'CZAJKOWSKIJ', 'ROSJA', 125, 147, NULL);

...

...

INSERT INTO Zawody_rodzaj VALUES(1, 'TURNIEJ CZTERECH SKOCZNI');

INSERT INTO Zawody_rodzaj VALUES(2, 'MISTRZOSTWA SWIATA');

...

...

INSERT INTO Zawody VALUES(1, '63. TURNIEJ CZTERECH SKOCZNI', '2014.12.27', '2015.01.06', 1);

INSERT INTO Zawody VALUES(2, '64. TURNIEJ CZTERECH SKOCZNI', '2015.12.28', '2016.01.06', 1);

...

USUWANIE TABEL / PERSPEKTYW W CELACH NAPRAWY NASZEJ BAZY DANYCH:

Gdyby doszło do pewnych pomyłek lub koniecznej poprawy danych w tabelach, można zastosować komendy **DROP TABLE** wraz z dodatkową komendą **CASCADE CONSTRAINTS** która usunie automatycznie powstałe **indeksy** oraz ograniczenia narzucone przez **klucze obce lub ograniczenia**.

DODATKOWA INFORMACJA: Język SQL w programie **SQL Developer Data Modeler** samemu tworzy **indeksy** które podlegają i mają powiązania z **kluczami głównymi**.

KOMENDY DROP TABLE WYKORZYSTANE W NASZEJ BAZIE DANYCH:

drop table	Zawody	CASCADE constraints;
drop table	Zawody_rodzaj	CASCADE constraints;
drop table	Skocznie	CASCADE constraints;
drop table	Skoczkowie	CASCADE constraints;
drop table	Trenerzy	CASCADE constraints;
drop table	Reprezentacje	CASCADE constraints;
drop table	Skoki	CASCADE constraints;

KOMENDY DROP WYKORZYSTANE W NASZEJ BAZIE DANYCH:

```
drop view Zawody_w_skokach;  
drop view Rekordzisci_skoczni;  
drop view Ilosc_skoczkow_reprezentacji;  
...  
drop trigger usun_skocznie_BRAK_REK;  
...
```

KOMENDY ALTER oraz UPDATE WYKORZYSTANE W NASZEJ BAZIE DANYCH:

```
alter table Skocznie ADD Rok_powstania NUMBER(4);  
  
alter table Skoki ADD Miejsce Number(2);  
  
update Skocznie SET Rok_powstania = 1950 WHERE id_skoczni = 1  
update Skocznie SET Rok_powstania = 1925 WHERE id_skoczni = 24;  
...
```


UTWORZONE PERSPEKTYWY / VIEWS:

Perspektywa **Zawody_w_skokach**:

„Stwórz perspektywę w której zdefiniowane jest ile zawodów danego typu się odbyło lub ma się odbyć.”

SKŁADNIA SQL:

```
82 --PERSPEKTYWA [1]
83 --Stworz perspektywe w ktorej zdefiniowane jest ile
84 --zawodow danego typu sie odbylo lub ma sie odbyc.
85 CREATE OR REPLACE VIEW Zawody_w_skokach AS
86 SELECT zawody_rodzaj.nazwa, count(*) as "ILOSC ZAWODOW TEGO TYPU" FROM zawody_rodzaj
87 INNER JOIN zawody ON zawody_rodzaj.id_rodzaj = zawody.id_rodzaj
88 GROUP BY zawody_rodzaj.nazwa;
```

Rezultat po skompilowaniu kodu:

	NAZWA	ILOSC ZAWODOW TEGO TYPU
1	PLANICA7	2
2	WILLINGEN5	2
3	TURNIEJ CZTERECH SKOCZNI	5
4	RAW AIR	1
5	MISTRZOSTWA SWIATA W LOTACH	4
6	PUCHAR SWIATA W LOTACH	4

Perspektywa Rekordzisci_skoczni:

„Stwórz perspektywę która pokaże skoczków posiadających w swoim dorobku rekordy skoczni, gdzie je ustanowili oraz ile ten rekord wynosi.
Jeśli skoczek nie ma rekordu skoczni to nie będzie tutaj wyświetlony.”

Składnia SQL:

```
90 --PERSPEKTYWA [2]
91 --Stworz perspektywe ktora pokaże skoczków posiadających w swoim dorobku
92 --rekordy skoczni, gdzie je ustanowili i ile on wynosi.
93 --Jesli skoczek nie ma rekordu to nie jest tu wyswietlony.
94 CREATE OR REPLACE VIEW Rekordzisci_skoczni AS
95 SELECT id_skoczka, imie, nazwisko, narodowosc, rekord_skoczni, nazwa FROM Skoczkowie
96 INNER JOIN skocznie ON skoczkowie.id_skoczka = skocznie.id_rekordzisty;
```

Rezultat po skompilowaniu kodu:

ID_SKOCZKA	IMIE	NAZWISKO	NARODOWOSC	REKORD_SKOCZNI	NAZWA
1	101 KAMIL	STOCH	POLSKA	150	OKURAYAMA K123
2	101 KAMIL	STOCH	POLSKA	140	TREMLIN DU PRAZ K125
3	102 ADAM	MALYSZ	POLSKA	140	TRAMPOLINO DAL BEN K120
4	102 ADAM	MALYSZ	POLSKA	100	MIYANOMORI K90
5	105 DAWID	KUBACKI	POLSKA	140	WIELKA KROKIEW IM. STANISLAWA MARUSARZA K125
6	105 DAWID	KUBACKI	POLSKA	150	SEPP BRADL-SKISTADION K125
7	109 JAKUB	KOT	POLSKA	96	KRUCZA SKALA W LUBAWCE K85
8	111 SEVERIN	FREUND	NIEMCY	140	LUGNET K120
9	121 JOHANN ANDRE	FORFANG	NORWEGIA	140	SALPAUSSELKA K116
10	127 DANIEL-ANDRE	TANDE	NORWEGIA	240	HEINI-KLOPFER-SKIFLUGSCHANZE K200
11	130 JANNE	AHONEN	FINLANDIA	150	MUHLENKOPFSCHANZE K130
12	140 RYOYU	KOBAYASHI	JAPONIA	140	GROSS-TITLIS-SCHANZE K125
13	140 RYOYU	KOBAYASHI	JAPONIA	250	LETALNICA K200
14	160 PETER	PREVC	SLOWENIA	110	BAUHENK K100
15	160 PETER	PREVC	SLOWENIA	240	KULM K200
16	170 MICHAEL	HAYBOCK	AUSTRIA	140	BERGISEL K120
17	174 STEFAN	KRAFT	AUSTRIA	250	VIKERSUNDBAKKEN K200
18	174 STEFAN	KRAFT	AUSTRIA	100	TEHVANDI K90
19	176 GREGOR	SCHLIERENZAUER	AUSTRIA	110	MIDTSTUBAKKEN K95
20	177 THOMAS	MORGENSTERN	AUSTRIA	140	JESTED A K120
21	177 THOMAS	MORGENSTERN	AUSTRIA	210	CERTAK K185

Perspektywa Ilosc_skoczkow_reprezentacji:

„Stwórz perspektywę która wyświetla kraje które posiadają chociaż jednego zawodnika w reprezentacji narodowej.

Rezultat jest posortowany malejąco względem ilości skoczków.”

Składnia SQL:

```
98 | --PERSPEKTYWA [3]
99 | --Stworz perspektywe ktora wyswietla kraje ktore posiadaja
100 | --chociaz jednego zawodnika w reprezentacji narodowej oraz rezultat
101 | --jest posortowany malejaco wzgledem ilosci skoczkow.
102 | CREATE OR REPLACE VIEW Ilosc_skoczkow_reprezentacji AS
103 | SELECT narodowosc, count(*) as "ILOSC SKOCZKOW W TYM KRAJU" FROM Skoczkanie
104 | INNER JOIN Reprezentacje ON Skoczkanie.narodowosc = Reprezentacje.nazwa_kraju
105 | GROUP BY Skoczkanie.narodowosc
106 | ORDER BY 2 DESC;
```

Rezultat po skompilowaniu kodu:

	NARODOWOSC	ILOSC SKOCZKOW W TYM KRAJU
1	POLSKA	9
2	JAPONIA	9
3	AUSTRIA	8
4	NIEMCY	8
5	NORWEGIA	8
6	SLOWENIA	6
7	FINLANDIA	6
8	CZECHY	6
9	ROSJA	5
10	SZWAJCARIA	4
11	ESTONIA	4
12	KANADA	4
13	KAZACHSTAN	4
14	WLOCHY	3
15	STANY ZJEDNOCZONE	3
16	UKRAINA	3
17	FRANCJA	2

Perspektywa ZAWODY_Z_WARUNKAMI

„Stwórz perspektywę która wyświetli nazwy zawodów dot. skoków narciarskich w taki sposób, że wyniki skoczków przekraczały lub wynosiły 275 punktów, ale nie wynosiły więcej lub nie były równe 300 punktów, te wyniki nie dawały zawodnikom podium (czyli miejsc 1, 2 lub 3), a punkty za wiatr (zsumowane) odejmowały od wyniku (były ujemne) lub były większe lub równe 10 punktów na plusie.”

Składnia SQL:

```
111 --PERSPEKTYWA [4]
112 --Stwórz perspektywę która wyświetli nazwy zawodów dot. skoków narciarskich w taki sposób,
113 --że wyniki skoczków przekraczały lub wynosiły 275 punktów, ale nie wynosiły więcej lub nie były równe 300 punktów,
114 --te wyniki nie dawały zawodnikom podium (czyli miejsc 1, 2 lub 3),
115 --a punkty za wiatr (zsumowane) odejmowały od wyniku (były ujemne)
116 --lub były większe lub równe 10 punktów na plusie
117
118 CREATE OR REPLACE VIEW ZAWODY_Z_WARUNKAMI AS
119 SELECT nazwa ZAWODY, id_skoczka_skok "ID SKOCZKA", imie, nazwisko, wiatr_1+wiatr_2 WIATR_OGOLEM, punkty_razem WYNIK, Miejsce from Skoki s
120 left join Zawody z on (z.id_zawodow = s.id_skoki)
121 left join Skoczkowie m on (m.id_skoczka = s.id_skoczka_skok)
122 where (s.punkty_razem between 275 and 300)
123 and (S.miejsce > 3) and
124 (s.wiatr_1+s.wiatr_2 < 0 or s.wiatr_1+s.wiatr_2 > 10) and (s.wiatr_1+s.wiatr_2 < 0 or s.wiatr_1+s.wiatr_2 > 10);
125
```

Rezultat po wyświetleniu perspektywy:

	ZAWODY	ID SKOCZKA	IMIE	NAZWISKO	WIATR_OGOLEM	WYNIK	MIEJSCE
1	66. TURNIEJ CZTERECH SKOCZNI	113	RICHARD	FREITAG	17,6	275	5

UTWORZONE WYZWALACZE / TRIGGERS:

Wyzwalacz **usun_skocznie_BRAK_REK**:

„Wyzwalacz ten usuwa skocznie które nie mają w rekordach zarejestrowanego rekordzisty.”

Domyślnie zakomentowany w pliku SQL.

Składnia SQL:

```
71 --WYZWALACZ / TRIGGER [1]
72 --Usuwa te skocznie ktore nie maja zarejestrowanego
73 --rekordzisty
74 CREATE OR REPLACE TRIGGER usun_skocznie_BRAK_REK
75 AFTER DELETE ON Skocznie
76 FOR EACH ROW
77 BEGIN
78     delete from Skocznie WHERE id_rekordzisty IS NULL;
79 END;
```

Wyzwalacz **skoczkowie_DELETE_FRANCJA**

„Ten wyzwalacz usuwa z tabeli **Skoczkowie** zawodników którzy mają narodowość francuską.”

Składnia SQL:

```
29 --WYZWALACZ / TRIGGER [2]
30 -- usuwa z listy skoczków
31 -- zawodników Francuskich
32 CREATE OR REPLACE TRIGGER skoczkowie_DELETE_FRANCJA
33 AFTER INSERT OR UPDATE ON Skoczkowie
34 FOR EACH ROW
35 BEGIN
36     DELETE FROM Skoczkowie WHERE narodowosc = 'FRANCJA';
37 END;
38 ALTER TRIGGER skoczkowie delete FRANCJA DISABLE;
```

Wyzwalacz trenerzy_max30

„Ten wyzwalacz usuwa tych trenerów w tabeli **Trenerzy** którzy mają więcej niż 30 lat.”

Składnia SQL:

```
39 --WYZWALACZ / TRIGGER [3] ktory usuwa z listy trenerów
40 -- tych trenerów, którzy maja więcej niz 30 lat.
41 CREATE OR REPLACE TRIGGER trenerzy_max30
42 AFTER INSERT OR UPDATE ON Trenerzy
43 FOR EACH ROW
44 BEGIN
45     DELETE FROM Trenerzy WHERE
46         extract(YEAR from SYSDATE) - extract(YEAR from data_urodzenia) <= 30;
47 END;
48 ALTER TRIGGER trenerzy_max30 DISABLE;
```


ZAPYTANIA SQL:

Zapytanie SQL #1:

„Za pomocą odpowiedniego polecenia **SELECT** pogrupuj ilość skoczków z każdego państwa posortowane krajami alfabetycznie od **Z** do **A**.

Składnia SQL:

```
127 --ZAPYTANIE SQL [1]
128 --Za pomocą odpowiedniego polecenia SELECT pogrupuj ilość skoczków z każdego państwa
129 --posortowane krajami alfabetycznie od Z do A.
130
131 SELECT narodowosc, COUNT(*) AS "Ilosc zawodnikow"
132 FROM Skoczkowie
133 GROUP BY narodowosc ORDER BY 1 desc;
```

Rezultat po skompilowaniu kodu:

NARODOWOSC	Ilosc zawodnikow
1 WLOCHY	3
2 UKRAINA	3
3 SZWAJCARIA	4
4 STANY ZJEDNOCZONE	3
5 SLOWENIA	6
6 ROSJA	5
7 POLSKA	9
8 NORWEGIA	8
9 NIEMCY	8
10 KAZACHSTAN	4
11 KANADA	4
12 JAPONIA	9
13 FRANCJA	2
14 FINLANDIA	6
15 ESTONIA	4
16 CZECHY	6
17 AUSTRIA	8

Zapytanie SQL #2:

„Za pomocą polecenia **SELECT** pokaż rodzaje zawodów które na końcu swojej nazwie mają liczbę **naturalną** (traktujemy „0” jako naturalną)”

Składnia SQL:

```
134 --ZAPYTANIE SQL [2]
135 --Za pomocą polecenia SELECT pokaz rodzaje zawodów które
136 --swoją nazwę kończą na liczbie naturalnej.
137 select nazwa from Zawody_rodzaj WHERE
138     nazwa LIKE '%0' OR nazwa LIKE '%1'
139 OR nazwa LIKE '%2' OR nazwa LIKE '%3'
140 OR nazwa LIKE '%4' OR nazwa LIKE '%5'
141 OR nazwa LIKE '%6' OR nazwa LIKE '%7'
142 OR nazwa like '%8' OR nazwa LIKE '%9';
```

Rezultat po skompilowaniu kodu:

	NAZWA
1	PLANICA7
2	WILLINGEN5

Zapytanie SQL #3:

„Za pomocą polecenia **SELECT** wyświetl kraje które posiadają **minimum** dwie skocznie narciarskie. Posortuj wynik od największej ilości do najmniejszej.”

Składnia SQL:

```
144 --ZAPYTANIE SQL [3]
145 --Za pomocą polecenia SELECT wyświetl kraje
146 --posiadające minimum 2 skocznie narciarskie,
147 --posortuj wynik od największej ilości skoczni do najmniejszej.
148 SELECT panstwo, count(*) as "ILOSC SKOCZNI" FROM Skocznie
149 INNER JOIN Reprezentacje ON skocznie.panstwo = reprezentacje.nazwa_kraju
150 GROUP BY skocznie.panstwo
151 HAVING count(*) >= 2
152 ORDER BY count(*) DESC;
```

Rezultat po skompilowaniu kodu:

	⚡ PANSTWO	⚡ ILOSC SKOCZNI
1	JAPONIA	4
2	AUSTRIA	3
3	NORWEGIA	3
4	CZECHY	2
5	POLSKA	2
6	NIEMCY	2
7	FRANCJA	2
8	ESTONIA	2
9	SLOWENIA	2
10	SZWECJA	2

Zapytanie SQL #4:

„Za pomocą polecenia **SELECT** wyświetl te skocznie które mają **ustalony** rekord skoczni, **ALE** informacje na temat rekordzisty nie są w tej bazie danych **zarejestrowane**. Pamiętaj o tym, żeby wziąć pod uwagę te skocznie które **NIE** mają rekordu skoczni!

Składnia SQL:

```
154 --ZAPYTANIE SQL [4]
155 --Za pomoca polecenia SELECT wyswietl te skocznie
156 --majace ustalony rekord skoczni ale informacji na temat
157 --rekordzisty nie sa zarejestrowane w tej bazie danych
158 --oraz te skocznie ktore NIE maja rekordu skoczni.
159 SELECT * from Skocznie WHERE id_rekordzisty IS NULL;
160
161 --Samo id_rekordzisty zalatwia sprawe, bo skoro nie bylo rekordzisty skoczni,
162 --to nie ma takze rekordu_skoczni! ;)
```

Pytanie zostało specjalnie tak skonstruowane by wydawało się być bardziej skomplikowane niż jest w rzeczywistości, jest to analogiczne, jeżeli skocznia **nie posiada** rekordzisty to jednoznacznie **nie posiada** rekordu skoczni!

Rezultat pod skompilowaniu kodu:

ID_SKOCZNI	NAZWA	MIEJSCOWOSC	PANSTWO	PUNKT_KONSTRUKCYJNY	REKORD_SKOCZNI	ID_REKORDZYSTY	ROK_POWSTANIA
1	4 IGMAN K112	SARAJEWO	BOSNIA I HERCEGOWINA	110	120	(null)	(null)
2	5 DRAGON HILL K125	YABULI	CHINY	130	140	(null)	(null)
3	6 JAPLENSKY VRH K70	DELNICE	CHORWACJA	70	73	(null)	(null)
4	10 MUSTAMAE K50	TALLINN	ESTONIA	50	59	(null)	(null)
5	13 LA COTE FEUILLEE K106	CHAUX-NEUVE	FRANCJA	110	120	(null)	(null)
6	14 SAKARTWELO K115	BAKURIANI	GRUZJA	120	120	(null)	(null)
7	21 SNIEZYINKA K125	CZAJKOWSKIJ	ROSJA	130	150	(null)	(null)
8	24 SUICIDE HILL K90	ISHPEMING	STANY ZJEDNOCZONE	90	100	(null)	1925
9	27 PARADISKULLEN K90	ORNSKOLDSVIK	SZWECJA	90	100	(null)	(null)
10	31 ARAIYAMA K55	SAPPORO	JAPONIA	55	59	(null)	2004
11	32 BOYODAI K80	OTARU	JAPONIA	80	(null)	(null)	(null)
12	33 HANNIBALBAKKEN K80	KONGSBERG	NORWEGIA	80	89	(null)	(null)

Zapytanie SQL #5:

„Za pomocą polecenia **SELECT** wyświetl z naszej bazy danych informacje na temat **NAJMŁODSZEGO** oraz na temat **NAJSTARSZEGO** skoczka narciarskiego.”

Składnia SQL:

```
164 --ZAPYTANIE SQL [5]
165 --Za pomocą polecenia SELECT wyświetl z naszej bazy danych
166 --informacje na temat NAJMŁODSZEGO oraz na temat NAJSTARSZEGO
167 --skoczka narciarskiego.
168 SELECT zad.id_skoczek, zad.imie, zad.nazwisko, zad.data_urodzenia, zad.narodowosc FROM Skoczkowie
169 zad JOIN
170 (
171     SELECT MAX(data_urodzenia) AS NAJMŁODSZY,
172            MIN(data_urodzenia) AS NAJSTARSZY
173     from Skoczkowie
174 )
175 oznaczenie ON (zad.data_urodzenia = NAJMŁODSZY OR zad.data_urodzenia = NAJSTARSZY);
```

Rezultat po skompilowaniu kodu:

	ID_SKOCZKA	IMIE	NAZWISKO	DATA_URODZENIA	NARODOWOSC
1	132	MICO	AHONEN	01/11/27	FINLANDIA
2	143	NORIAKI	KASAI	72/06/06	JAPONIA

Zapytanie SQL #6:

„Wyświetl skoczków którzy posiadają chociaż jeden rekord skoczni, przynajmniej raz znaleźli się w czołówce zawodów (miejsca 1, 2 lub 3), a ich wiek mieści się w przedziale 28 a 34 lata (do wykonania obliczeń przyjmij wartość YEAR).”

Składnia SQL:

```
233 --ZAPYTANIE SQL [6]
234 --Wyświetl skoczków którzy posiadają chociaż jeden rekord skoczni,
235 --chociaż raz znaleźli się w czołówce zawodów (1-3) a ich wiek znajduje się
236 --pomiędzy 28 a 34 lata.
237 SELECT DISTINCT id_rekordzisty, imie, nazwisko, narodowosc, data_urodzenia,
238 nazwa as "NAZWA SKOCZNI Z REKORDEM", rekord_skoczni, Miejsce as "MIEJSCE NA ZAWODACH" FROM Skoczkowie S
239 LEFT JOIN Skocznie M ON (S.id_skoczka = M.id_rekordzisty)
240 LEFT JOIN Skoki T ON (T.id_skoczka_skok = S.id_skoczka)
241 WHERE T.Miejsce BETWEEN 1 AND 3
242 AND extract(YEAR from SYSDATE) - extract(YEAR from S.data_urodzenia) BETWEEN 28 AND 34;
243 --AND select S.id_skoczka from Skoki WHERE count(*) > 2;
244 --select * from Skoki where Miejsce BETWEEN 1 AND 2;
```

Rezultat po skompilowaniu kodu:

ID_REKORDZISTY	IMIE	NAZWISKO	NARODOWOSC	DATA_URODZENIA	NAZWA SKOCZNI Z REKORDEM	REKORD_SKOCZNI	MIEJSCE NA ZAWODACH
1	101 KAMIL	STOCH	POLSKA	87/05/25	TREMP LIN DU PRAZ K125	140	1
2	101 KAMIL	STOCH	POLSKA	87/05/25	OKURAYAMA K123	150	1
3	112 MARKUS	EISENBICHLER	NIEMCY	91/04/03	GRANASEN K124	150	2

PODSUMOWANIE:

6 tabel, 3 perspektywy, 1 wyzwalacz, indeksy stworzone na bazie kluczy głównych przez środowisko **Oracle** oraz spora ilość rekordów która pozwoli odpowiednio przetestować bazę danych pod tytułem **Skoki narciarskie**.

By móc wykonać ten projekt korzystano z serwisów takich jak:

Wikipedia.org
Stackoverflow.com
W3Schools.com
ICIS.PCZ.PL/~Kaczmarek
docs.Oracle.com

Dane wykorzystywane w projekcie w 99% zgadzają się z rzeczywistością oprócz wspomnianych kilku skoczków którzy są już emerytowani.

Cały wymagany kod jest zawarty w dołączonym pliku .sql

**PROJEKT WYKONANY NA KONCIE
JĘDMAR19**