BAZA DANYCH SIŁOWNI, BASENU I SPA

*JAN SŁABOŃ, JAKUB STĘPIEŃ, MONIKA MIKUŁA*

CEL I ZAŁOŻENIA PROJEKTU

Celem naszego projektu było stworzeniu bazy danych pozwalającej na obsługę, podstawowe zarządzanie i administracje sieci obiektów sportowych. Jednolita struktura wszystkich obiektów pozwala w łatwy i szybki sposób dodać nowy podmiot do kompleksu bez konieczności przebudowy całej struktury bazy.

Śledzenie na bieżąco przychodów i wydatków pozwala określać rentowność obiektu. Realizowane jest to przez trzymanie informacji o zakupionych karnetach przez klientów, pensji pracowników i ich premii oraz koszty konserwacji.

Jak wiadomo bezpieczeństwo powinno być priorytetem. Każde wejście klienta na obiekt jest odnotowywane. Nieautoryzowanie dostanie się na obiekt (na przykład po godzinach otwarcia obiektu) będzie skutkować pojawieniem się komunikatu, po którym mogą zostać podjęte działania względem intruza.

Zastosowany został prosty schemat dziedziczenia:

OSOBA

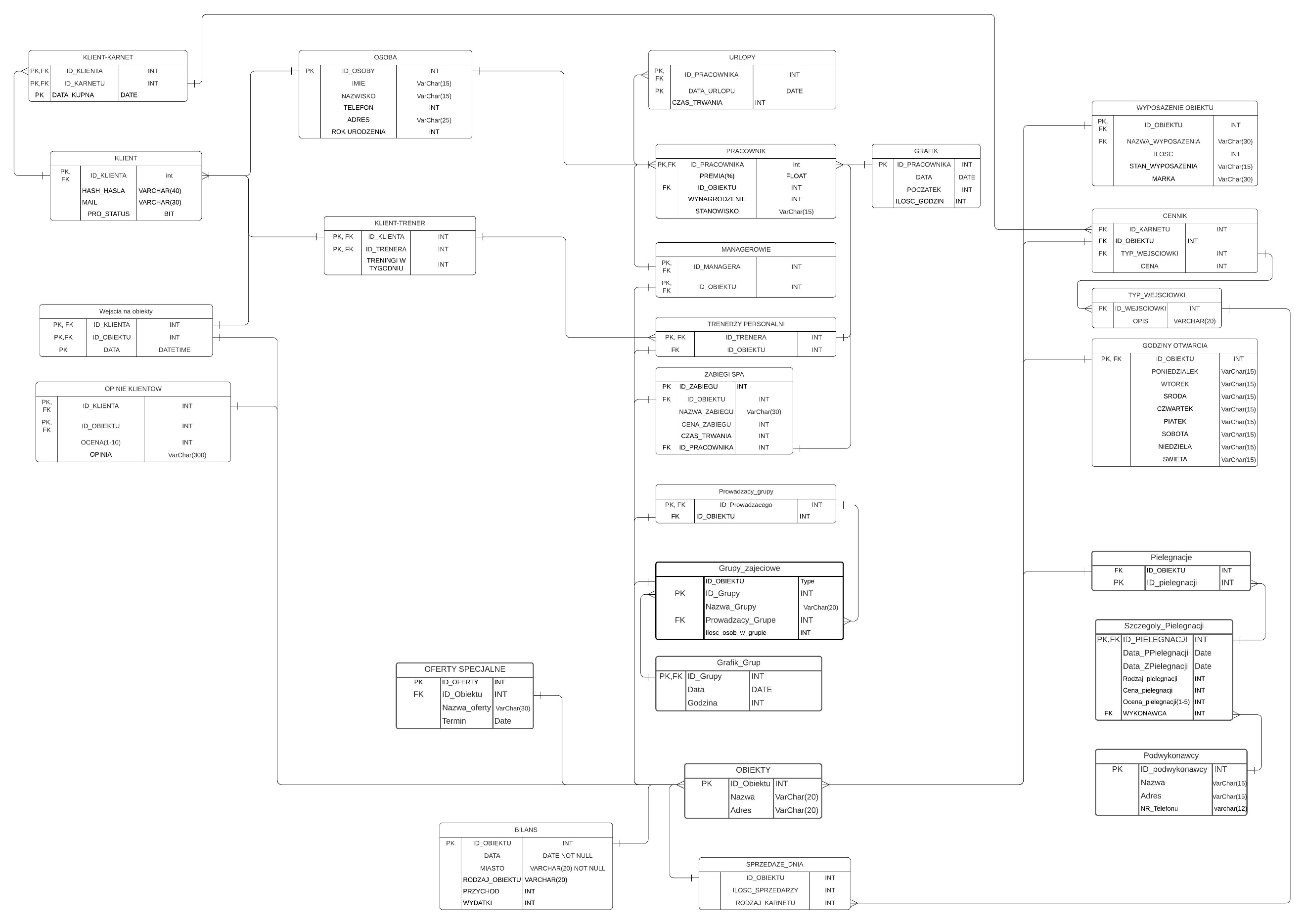
/ \

KLIENT PRACOWNIK

Zarówno wstawianie jak i usuwanie klientów/pracowników odbywa się wyłącznie za pomocą procedur ze względu na strukture bazy.

Szeroka gama procedur i funkcji umożliwia szybki dostęp do różnorodnych statystyk.

SCHEMAT RELACYJNY



Kod tworzacy baze

CREATE TABLE [OSOBA] (

[ID\_OSOBY] INT,

[IMIE] VarChar(25),

[NAZWISKO] VarChar(25),

[TELEFON] INT,

[ADRES] VarChar(40),

[ROK URODZENIA] INT,

PRIMARY KEY ([ID\_OSOBY])

);

CREATE TABLE [Pielegnacje] (

[ID\_OBIEKTU] INT,

[ID\_pielegnacji] INT,

PRIMARY KEY ([ID\_pielegnacji])

);

CREATE TABLE TYP\_WEJSCIOWKI (

[ID\_WEJSCIOWKI] INT PRIMARY KEY,

[OPIS] VARCHAR(60)

);

CREATE TABLE [CENNIK] (

[ID\_KARNETU] INT PRIMARY KEY,

[ID\_OBIEKTU] INT,

[TYP\_WEJSCIOWKI] INT FOREIGN KEY REFERENCES TYP\_WEJSCIOWKI(ID\_WEJSCIOWKI) ON DELETE CASCADE ON UPDATE CASCADE,

[CENA] INT

);

CREATE TABLE KLIENT(

ID\_KLIENTA INT PRIMARY KEY REFERENCES OSOBA(ID\_OSOBY) ON DELETE CASCADE ON UPDATE CASCADE,

PRO\_STATUS BIT NULL,

MAIL VARCHAR(40) NOT NULL,

HASH\_HASLA VARCHAR(60) NOT NULL

)

CREATE TABLE [WYPOSAZENIE OBIEKTU] (

[ID\_OBIEKTU] INT,

[NAZWA\_WYPOSAZENIA] VarChar(30),

[ILOSC] INT,

[STAN\_WYPOSAZENIA] VarChar(15),

[MARKA] VarChar(30),

PRIMARY KEY ([NAZWA\_WYPOSAZENIA], [ID\_OBIEKTU])

);

CREATE TABLE [KLIENT-KARNET] (

[ID\_KLIENTA] INT,

[ID\_KARNETU] INT,

[DATA KUPNA] DATE

PRIMARY KEY ([ID\_KLIENTA], [ID\_KARNETU], [DATA KUPNA])

);

CREATE TABLE [Wejscia na obiekty] (

[ID\_KLIENTA] INT,

[ID\_OBIEKTU] INT,

[DATA] DATETIME,

PRIMARY KEY ([ID\_KLIENTA],[ID\_OBIEKTU],[DATA])

);

CREATE TABLE [URLOPY] (

[ID\_PRACOWNIKA] INT,

[DATA\_URLOPU] DATE,

[CZAS\_TRWANIA] INT

PRIMARY KEY ([DATA\_URLOPU], [ID\_PRACOWNIKA])

);

CREATE TABLE [GODZINY OTWARCIA] (

[ID\_OBIEKTU] INT PRIMARY KEY,

[PONIEDZIALEK] VarChar(15),

[WTOREK] VarChar(15),

[SRODA] VarChar(15),

[CZWARTEK] VarChar(15),

[PIATEK] VarChar(15),

[SOBOTA] VarChar(15),

[NIEDZIELA] VarChar(15),

[SWIETA] VarChar(15)

);

CREATE TABLE [Podwykonawcy] (

[ID\_podwykonawcy] INT,

[Nazwa] VarChar(15),

[Adres] VarChar(15),

[NR\_Telefonu] varchar(12),

PRIMARY KEY ([ID\_podwykonawcy])

);

CREATE TABLE [Prowadzacy\_grupy] (

[ID\_Prowadzacego] INT PRIMARY KEY,

[ID\_OBIEKTU] INT

);

CREATE TABLE [Grafik\_Grup] (

[ID\_Grupy] INT,

[Data] DATE,

[Godzina] INT,

PRIMARY KEY ([ID\_Grupy])

);

CREATE TABLE [Szczegoly\_Pielegnacji] (

[ID\_PIELEGNACJI] INT,

[Data\_PPielegnacji] Date,

[Data\_ZPielegnacji] Date,

[Rodzaj\_pielegnacji] INT,

[Cena\_pielegnacji] INT,

[Ocena\_pielegnacji(1-5)] INT,

[WYKONAWCA] INT,

PRIMARY KEY ([ID\_PIELEGNACJI])

);

CREATE TABLE [KLIENT-TRENER] (

[ID\_KLIENTA] INT,

[ID\_TRENERA] INT,

[TRENINGI W TYGODNIU] INT

PRIMARY KEY([ID\_KLIENTA],[ID\_TRENERA])

);

CREATE TABLE [OBIEKTY] (

[ID\_Obiektu] INT,

[Nazwa] VarChar(20),

[Adres] VarChar(20),

PRIMARY KEY ([ID\_Obiektu])

);

CREATE TABLE [TRENERZY PERSONALNI] (

[ID\_TRENERA] INT PRIMARY KEY,

[ID\_OBIEKTU] INT

);

CREATE TABLE [OPINIE KLIENTOW] (

[ID\_KLIENTA] INT,

[ID\_OBIEKTU] INT,

[OCENA(1-10)] INT,

[OPINIA] VarChar(300)

PRIMARY KEY([ID\_KLIENTA],[ID\_OBIEKTU])

);

CREATE TABLE [ZABIEGI SPA] (

[ID\_ZABIEGU] INT PRIMARY KEY,

[ID\_OBIEKTU] INT,

[NAZWA\_ZABIEGU] VarChar(30),

[CENA\_ZABIEGU] INT,

[CZAS\_TRWANIA] INT,

[ID\_PRACOWNIKA] INT

);

CREATE TABLE [MANAGEROWIE] (

[ID\_MANAGERA] INT,

[ID\_OBIEKTU] INT

PRIMARY KEY([ID\_MANAGERA],[ID\_OBIEKTU])

);

CREATE TABLE [PRACOWNIK] (

[ID\_PRACOWNIKA] int,

[PREMIA(%)] FLOAT,

[ID\_OBIEKTU] INT,

[WYNAGRODZENIE] INT,

[STANOWISKO] VarChar(15),

PRIMARY KEY ([ID\_PRACOWNIKA])

);

CREATE TABLE [OFERTY SPECJALNE] (

[ID\_OFERTY] INT,

[ID\_Obiektu] INT,

[Nazwa\_oferty] VarChar(30),

[Termin] Date,

PRIMARY KEY ([ID\_OFERTY])

);

CREATE TABLE [Grupy\_zajeciowe] (

[ID\_OBIEKTU] INT,

[ID\_Grupy] INT,

[Nazwa\_Grupy] VarChar(50),

[Prowadzacy\_Grupe] INT,

[Ilosc\_osob\_w\_grupie] INT,

PRIMARY KEY ([ID\_Grupy])

);

CREATE TABLE BILANS(

ID\_OBIEKTU INT REFERENCES OBIEKTY(ID\_OBIEKTU) ON DELETE CASCADE ON UPDATE CASCADE,

DATA DATE NOT NULL,

MIASTO VARCHAR(20) NOT NULL,

RODZAJ\_OBIEKTU VARCHAR(20) NOT NULL,

PRZYCHOD INT NOT NULL,

WYDATKI INT NOT NULL

)

CREATE TABLE GRAFIK(

ID\_PRACOWNIKA INT NOT NULL REFERENCES PRACOWNIK(ID\_PRACOWNIKA) ON DELETE CASCADE ON UPDATE CASCADE,

DATA DATE NOT NULL,

POCZATEK INT NOT NULL,

ILOSC\_GODZIN INT NOT NULL

)

CREATE TABLE SPRZEDAZE\_DNIA(

ID\_OBIEKTU INT REFERENCES OBIEKTY(ID\_OBIEKTU) ON DELETE CASCADE ON UPDATE CASCADE,

ILOSC\_SPRZEDAZY INT NOT NULL,

RODZAJ\_KARNETU INT NOT NULL

)

ALTER TABLE [TRENERZY PERSONALNI]

ADD FOREIGN KEY (ID\_OBIEKTU) REFERENCES OBIEKTY(ID\_OBIEKTU) ON DELETE CASCADE ON UPDATE CASCADE

ALTER TABLE [TRENERZY PERSONALNI]

ADD FOREIGN KEY (ID\_TRENERA) REFERENCES PRACOWNIK(ID\_PRACOWNIKA) ON DELETE CASCADE ON UPDATE CASCADE

Alter table [Wejscia na obiekty]

ADD FOREIGN KEY (ID\_KLIENTA) REFERENCES KLIENT(ID\_KLIENTA) ON DELETE CASCADE ON UPDATE CASCADE

Alter table [KLIENT-TRENER]

ADD FOREIGN KEY (ID\_KLIENTA) REFERENCES KLIENT(ID\_KLIENTA) ON DELETE CASCADE ON UPDATE CASCADE

Alter table [KLIENT-TRENER]

ADD FOREIGN KEY (ID\_TRENERA) REFERENCES [TRENERZY PERSONALNI](ID\_TRENERA) ON DELETE CASCADE ON UPDATE CASCADE

Alter table [Wejscia na obiekty]

ADD FOREIGN KEY (ID\_OBIEKTU) REFERENCES OBIEKTY(ID\_OBIEKTU) ON DELETE CASCADE ON UPDATE CASCADE

Alter table [KLIENT]

ADD FOREIGN KEY (ID\_KLIENTA) REFERENCES OSOBA(ID\_OSOBY)

Alter table [PRACOWNIK]

ADD FOREIGN KEY (ID\_PRACOWNIKA) REFERENCES OSOBA(ID\_OSOBY)

Alter table [OPINIE KLIENTOW]

ADD FOREIGN KEY (ID\_KLIENTA) REFERENCES KLIENT(ID\_KLIENTA) ON DELETE CASCADE ON UPDATE CASCADE

Alter table [OPINIE KLIENTOW]

ADD FOREIGN KEY (ID\_OBIEKTU) REFERENCES OBIEKTY(ID\_OBIEKTU) ON DELETE CASCADE ON UPDATE CASCADE

Alter table MANAGEROWIE

ADD FOREIGN KEY (ID\_OBIEKTU) REFERENCES OBIEKTY(ID\_OBIEKTU) ON DELETE CASCADE ON UPDATE CASCADE

Alter table MANAGEROWIE

ADD FOREIGN KEY (ID\_MANAGERA) REFERENCES PRACOWNIK(ID\_PRACOWNIKA) ON DELETE CASCADE ON UPDATE CASCADE

Alter table [grupy\_zajeciowe]

ADD FOREIGN KEY (Prowadzacy\_grupe) REFERENCES [prowadzacy\_grupy](ID\_prowadzacego) ON DELETE CASCADE ON UPDATE CASCADE

Alter table [prowadzacy\_grupy]

ADD FOREIGN KEY (ID\_prowadzacego) REFERENCES PRACOWNIK(ID\_PRACOWNIKA) ON DELETE CASCADE ON UPDATE CASCADE

Alter table URLOPY

ADD FOREIGN KEY (ID\_pracownika) REFERENCES PRACOWNIK(ID\_PRACOWNIKA) ON DELETE CASCADE ON UPDATE CASCADE

Alter table [wyposazenie obiektu]

ADD FOREIGN KEY (ID\_OBIEKTU) REFERENCES OBIEKTY(ID\_OBIEKTU) ON DELETE CASCADE ON UPDATE CASCADE

ALTER TABLE SZCZEGOLY\_PIELEGNACJI

ADD CONSTRAINT CHECK\_OCENA\_BETWEEN\_1\_5 CHECK ( [Ocena\_pielegnacji(1-5)] BETWEEN 1 AND 5)

ALTER TABLE CENNIK

ADD FOREIGN KEY (ID\_OBIEKTU) REFERENCES OBIEKTY(ID\_OBIEKTU) ON DELETE CASCADE ON UPDATE CASCADE

ALTER TABLE [OFERTY SPECJALNE]

ADD FOREIGN KEY (ID\_OBIEKTU) REFERENCES OBIEKTY(ID\_OBIEKTU) ON DELETE CASCADE ON UPDATE CASCADE

ALTER TABLE [GRAFIK\_GRUP]

ADD FOREIGN KEY (ID\_GRUPY) REFERENCES GRUPY\_ZAJECIOWE(ID\_GRUPY) ON DELETE CASCADE ON UPDATE CASCADE

ALTER TABLE [GODZINY OTWARCIA]

ADD FOREIGN KEY (ID\_OBIEKTU) REFERENCES OBIEKTY(ID\_OBIEKTU) ON DELETE CASCADE ON UPDATE CASCADE

ALTER TABLE PIELEGNACJE

ADD FOREIGN KEY (ID\_OBIEKTU) REFERENCES OBIEKTY(ID\_OBIEKTU) ON DELETE CASCADE ON UPDATE CASCADE

ALTER TABLE [ZABIEGI SPA]

ADD FOREIGN KEY (ID\_OBIEKTU) REFERENCES OBIEKTY(ID\_OBIEKTU) ON DELETE CASCADE ON UPDATE CASCADE

ALTER TABLE [ZABIEGI SPA]

ADD FOREIGN KEY (ID\_PRACOWNIKA) REFERENCES PRACOWNIK(ID\_PRACOWNIKA) ON DELETE CASCADE ON UPDATE CASCADE

ALTER TABLE SZCZEGOLY\_PIELEGNACJI

ADD FOREIGN KEY (ID\_PIELEGNACJI) REFERENCES PIELEGNACJE(ID\_PIELEGNACJI) ON DELETE CASCADE ON UPDATE CASCADE

ALTER TABLE SZCZEGOLY\_PIELEGNACJI

ADD FOREIGN KEY (WYKONAWCA) REFERENCES PODWYKONAWCY(ID\_PODWYKONAWCY) ON DELETE CASCADE ON UPDATE CASCADE

Wypełnianie bazy rekordami

SELECT \* FROM SPRZEDAZE\_DNIA

USE PROJEKT\_BAZY\_DANYCH

INSERT INTO OBIEKTY VALUES

(1, 'Klub Fitness', 'Ogrodowa 1'),

(2, 'Basen', 'Ogrodowa 2'),

(3, 'SPA', 'Ogrodowa 3')

INSERT INTO [GODZINY OTWARCIA] VALUES

(1, '6-23','6-23','6-23','6-23','6-23','8-22','8-20','9-15'),

(2, '6-23','6-23','6-23','6-23','6-23','8-22','8-20','9-15'),

(3, '6-23','6-23','6-23','6-23','6-23','8-22','8-20','9-15')

--DODAWANIE PRACOWNIKOW

EXEC DODAJ\_PRACOWNIKA 'MAGDALENA', 'WASOWSKA', 123123123, 'BARMANSKA 13, GORLICE', 2000, 1, 2000, 'Kasjer'

EXEC DODAJ\_PRACOWNIKA 'BARTOSZ', 'BOBRZOWSKI', 865786542, 'BAZAROWA 1/3 KRAKOW', 1945, 1, 2000, 'Kasjer'

EXEC DODAJ\_PRACOWNIKA 'KAMIL', 'KALECKI', 465321985, 'LECKA 15, WARSZAWA', 1968, 1, 2000, 'Kasjer'

EXEC DODAJ\_PRACOWNIKA 'ANDRZEJ', 'NOWAK', 798542656, 'KOLEJOWA 33 KRAKOW', 1989, 1, 2000, 'Kasjer'

EXEC DODAJ\_PRACOWNIKA 'ROBERT', 'BIEDRON', 665599321, 'OGRODOWA 4/50 KRAKOW', 2000, 1, 1900, 'Sprzatacz'

EXEC DODAJ\_PRACOWNIKA 'PAWEL', 'ADAMOWSKI', 998844335, 'NADWISLICKA 3/24 KRAKOW', 1977, 1, 1900, 'Sprzatacz'

EXEC DODAJ\_PRACOWNIKA 'NORBERT', 'ADAMOWSKI', 734658979, 'MICKIEWICZA 100 SZCZECIN', 1977, 1, 1900, 'Sprzatacz'

EXEC DODAJ\_PRACOWNIKA 'MAGDALENA', 'WASOWSKA', 113399845, 'LUKASIEWICZA 2/23 POZNAN', 1988, 1, 5000, 'Trener'

EXEC DODAJ\_PRACOWNIKA 'PAWEL', 'ADAMOWSKI', 789456123, 'BRZOZOWA 8 KRAKOW', 1963, 1, 5500, 'Trener'

EXEC DODAJ\_PRACOWNIKA 'JULIA', 'DABROWSKA', 746385926, 'NOWOSADECKA 18/2 KRAKOW', 1984, 1, 4500, 'Trener'

EXEC DODAJ\_PRACOWNIKA 'KAROL', 'MAJCHROWSKI', 839465728, 'CELAROWSKA 33A KRAKOW', 1994, 1, 2300, 'Trener'

EXEC DODAJ\_PRACOWNIKA 'PATRYCJA', 'NADCZUK', 729647256, 'PASTELOWA 10/1 KRAKOW', 1999, 1, 2700, 'Trener'

EXEC DODAJ\_PRACOWNIKA 'TOMASZ', 'MIASIK', 594611090, 'SLOWACKIEGO 13 WIELICZKA', 1992, 1, 3000, 'Trener'

EXEC DODAJ\_PRACOWNIKA 'JOLANTA', 'JAWORSKA', 239849264, 'SZPITALNA 43 MYSLENICE', 1969, 1, 4300, 'Trener'

EXEC DODAJ\_PRACOWNIKA 'AKARA', 'AMARTTAYAKUL', 475384954, 'SZLAK 17/6 KRAKOW', 1987, 1, 5000, 'Manager'

EXEC DODAJ\_PRACOWNIKA 'CHINTARA', 'SUKAPATANA', 475834096, 'SZLAK 17/6 KRAKOW', 1988, 1, 5000, 'Manager'

EXEC DODAJ\_PRACOWNIKA 'ADRIAN', 'KASPRZYK', 849370021, 'KRAKOWSKA 1 SKAWINA', 1990, 2, 2000, 'Kasjer'

EXEC DODAJ\_PRACOWNIKA 'ARTUR', 'RYBOWICZ', 349578267, 'FOCHA 65/3 KRAKOW', 1975, 2, 2000, 'Kasjer'

EXEC DODAJ\_PRACOWNIKA 'BARBARA', 'WIELGUS', 473924759, 'KOSCIELNA 4/3 KRAKOW', 1973, 2, 2000, 'Kasjer'

EXEC DODAJ\_PRACOWNIKA 'DANIEL', 'SYNOS', 834729759, 'AKADEMICKA 8 KRAKOW', 1988, 2, 2000, 'Kasjer'

EXEC DODAJ\_PRACOWNIKA 'DIANA', 'WNEK', 239483750, 'WIEJSKA 10/6 KRAKOW', 1985, 2, 4000, 'Ratownik'

EXEC DODAJ\_PRACOWNIKA 'FILIP', 'NOWAK', 358024003, 'PODMIEJSKA 10/16 KRAKOW', 1989, 2, 2400, 'Ratownik'

EXEC DODAJ\_PRACOWNIKA 'GABRIELA', 'KUSZ', 230047812, 'ZAJECZA 18/1 KRAKOW', 1991, 2, 2500, 'Ratownik'

EXEC DODAJ\_PRACOWNIKA 'HUBERT', 'ZIEBA', 348927493, 'WITOSA 67 KRAKOW', 1978, 2, 3700, 'Ratownik'

EXEC DODAJ\_PRACOWNIKA 'IZABELA', 'KUBICA', 823293039, 'CECHOWA 127 KRAKOW', 1986, 2, 3700, 'Manager'

EXEC DODAJ\_PRACOWNIKA 'JACEK', 'JABLONSKI', 432954029, 'BUKOWA 33/1 KRAKOW', 1992, 2, 4200, 'Manager'

EXEC DODAJ\_PRACOWNIKA 'JULIA', 'AUGUSTYNIAK', 437928049, 'DEBOWA 12/7 KRAKOW', 1991, 2, 3000, 'Instruktor'

EXEC DODAJ\_PRACOWNIKA 'KACPER', 'KANIA', 738293049, 'ORZECHOWA 17 KRAKOW', 1993, 2, 3200, 'Instruktor'

EXEC DODAJ\_PRACOWNIKA 'KAMILA', 'KOT', 347374890, 'WIDOKOWA 13 LUBORZYCA', 1997, 2, 3800, 'Instruktor'

EXEC DODAJ\_PRACOWNIKA 'SZYMON', 'MAJCHROWSKI', 437829483, 'OGRODOWA 8 BALICE', 1995, 2, 1900, 'Sprzatacz'

EXEC DODAJ\_PRACOWNIKA 'KAROLINA', 'DEC', 437829480, 'OGRODOWA 8 BALICE', 1997, 2, 1900, 'Sprzatacz'

EXEC DODAJ\_PRACOWNIKA 'KONRAD', 'KRAWIEC', 428039274, 'SENATORSKA 45 MYSLENICE', 1978, 2, 1900, 'Sprzatacz'

EXEC DODAJ\_PRACOWNIKA 'LILIANA', 'FERSHT', 293047823, 'NOWA 22/2 OLKUSZ', 1984, 3, 2000, 'Kasjer'

EXEC DODAJ\_PRACOWNIKA 'MACIEJ', 'PROKOP', 384926036, 'JASNA 73 KRAKOW', 1991, 3, 2000, 'Kasjer'

EXEC DODAJ\_PRACOWNIKA 'MAGDALENA', 'CHMIEL', 839437056, 'SIEWNA 14/3 KRAKOW', 1990, 3, 2000, 'Kasjer'

EXEC DODAJ\_PRACOWNIKA 'MANNARO', 'VICENZO', 238493054, 'RAJSKA 2/10 KRAKOW', 1990, 3, 2000, 'Kasjer'

EXEC DODAJ\_PRACOWNIKA 'MANUELA', 'KISZKA', 439954823, 'GAJOWA 64/3 WIELICZKA', 1994, 3, 1900, 'Sprzatacz'

EXEC DODAJ\_PRACOWNIKA 'MATEUSZ', 'CZECH', 537389485, 'ZAGAJE 34A KRAKOW', 1987, 3, 1900, 'Sprzatacz'

EXEC DODAJ\_PRACOWNIKA 'MATYLDA', 'LIGACZ', 348924840, 'WICHROWA 2/2 KRAKOW', 1992, 3, 1900, 'Sprzatacz'

EXEC DODAJ\_PRACOWNIKA 'NORBERT', 'PIEKOS', 357923057, 'LUBOSTRON 18/5 KRAKOW', 1996, 3, 4900, 'Manager'

EXEC DODAJ\_PRACOWNIKA 'OLGA', 'GORSKA', 348547937, 'LIPINSKIEGO 3/2 KRAKOW', 1987, 3, 5200, 'Manager'

EXEC DODAJ\_PRACOWNIKA 'PATRYK', 'RUMIANEK', 438039627, 'SZLAK 10/3 KRAKOW', 1993, 3, 3500, 'Kosmetyczka'

EXEC DODAJ\_PRACOWNIKA 'PAULINA', 'KALANDYK', 948374023, 'SADOWA 12/1 KRAKOW', 1988, 3, 3600, 'Kosmetyczka'

EXEC DODAJ\_PRACOWNIKA 'PAWEL', 'MISIEWICZ', 238439728, 'SWOJSKA 3 KRAKOW', 1992, 3, 2900, 'Kosmetyczka'

EXEC DODAJ\_PRACOWNIKA 'POLA', 'MAZUR', 947364873, 'WARZYCKA 13 WIELICZKA', 1994, 3, 3600, 'Kosmetyczka'

EXEC DODAJ\_PRACOWNIKA 'RADOSLAW', 'BURCZYNSKI', 438537039, 'NOWA 26/7 WIELICZKA', 1976, 3, 4600, 'Kosmetyczka'

EXEC DODAJ\_PRACOWNIKA 'RENATA', 'SOLECKA', 438574049, 'MAKOWA 23/3 KRAKOW', 1973, 3, 4000, 'Masazysta'

EXEC DODAJ\_PRACOWNIKA 'SUN WOO', 'NOH', 163829046, 'KRZYWA 7/3 KRAKOW', 1990, 3, 4100, 'Masazysta'

EXEC DODAJ\_PRACOWNIKA 'SYLWIA', 'KLUSKA', 538948367, 'RAJSKA 12/1 KRAKOW', 1987, 3, 3750, 'Masazysta'

EXEC DODAJ\_PRACOWNIKA 'WIKTOR', 'SIKORSKI', 438723943, 'BRONOWICKA 3/1 KRAKOW', 1993, 3, 4900, 'Masazysta'

--DODAWANIE KLIENTOW

EXEC DODAJ\_KLIENTA 'ADAM', 'PTASINSKI', 327837293, 'NALKOWKSIEGO 322/3 KRAKOW', 2001, 'ADAM.P@GMAIL.PL', 'dauHDW#e2', 0

EXEC DODAJ\_KLIENTA 'ADEL', 'BILALOVA', 234438948, 'RADZIKOWSKIEGO 45/1 KRAKOW', 1996, 'ADEL.B@GMAIL.PL', 'RWYyfe@%$', 1

EXEC DODAJ\_KLIENTA 'ADRIAN', 'BAZYLEWSKI', 928374938, 'ZYGMUNTOWSKA 3/3 WIELICZKA', 1982, 'ADRIAN.B@VP.PL', 'fwmIMUFWwD57', 0

EXEC DODAJ\_KLIENTA 'ADRIANNA', 'BEDNARSKA', 748362910, 'OWSIANA 24/2 WADOWICE', 1976, 'ADAADA@ONET.PL', 'fwd65WD5', 1

EXEC DODAJ\_KLIENTA 'AGATA', 'LEONOWICZ', 843928347, 'ZIMOROWICZA 35 KRAKOW', 1968, 'AGALEO@VP.PL', '&Jj8JmM\*m7M', 0

EXEC DODAJ\_KLIENTA 'AGNIESZKA', 'OLIWA', 437854928, 'ZIELINSKA 31/4 KRAKOW', 1972, 'OLIWCIA@GMAIL.PL', '&v%&7Bb\*', 0

EXEC DODAJ\_KLIENTA 'ALA', 'WROBEL', 374283493, 'ZDROWA 75/3 KRAKOW', 1985, 'WROBELALA@GMAIL.PL', 'v^v5&FW6', 0

EXEC DODAJ\_KLIENTA 'ALBERT', 'JAROSZ', 549384239, 'REJA 34/11 WADOWICE', 1999, 'JAROSZJAROSZ@INTERIA.PL', 'm\*&j5%WM68', 0

EXEC DODAJ\_KLIENTA 'ALEKSANDER', 'BANAT', 443234857, 'ZABINIEC 43/10 KRAKOW', 2000, 'BANAT88@VP.PL', 'wfm&FWWF4', 1

EXEC DODAJ\_KLIENTA 'BARBARA', 'MISZTAL', 125364857, 'ROGATKA 28/1 KRAKOW', 1997, 'MISZTALB@GMAIL.PL', 'wf$WF6F4WFW4', 1

EXEC DODAJ\_KLIENTA 'BARTLOMIEJ', 'WILK', 163942849, 'PRANDOTY 19/3 KRAKOW', 1992, 'WILKUU@INTERIA.PL', 'VW5W6FVWw', 0

EXEC DODAJ\_KLIENTA 'BARTOSZ', 'GAWEL', 183049576, 'GROCHOWSKA 10/2 KRAKOW', 1984, 'GAWLIK9@GMAIL.PL', 'vw$W5^wwvEW', 0

EXEC DODAJ\_KLIENTA 'BEATA', 'PIECH', 163827113, 'CIELINSKIEGO 11A KRAKOW', 1987, 'PIECHA33@GMAIL.PL', 'VW##Sv#@WWw', 0

EXEC DODAJ\_KLIENTA 'CYPRIAN', 'REJMAN', 184111029, 'LESISTA 1/12 WADOWICE', 1985, 'CYPEK@GMAIL.PL', 'vcwVww%VEW5', 0

EXEC DODAJ\_KLIENTA 'DAMIAN', 'BATOR', 172109009, 'PODMIEJSKA 12/1 KRAKOW', 1977, 'BATORD@INTERIA.PL', 'wjmcMW#3', 0

EXEC DODAJ\_KLIENTA 'DANIEL', 'SOWA', 843837267, 'MAJOWA 44/4 KRAKOW', 1988, 'SOWA.DANIEL34@VP.PL', 'fw%FWyF6', 1

EXEC DODAJ\_KLIENTA 'DIANA', 'SZELIGA', 748936475, 'WYSOKA 75 KRAKOW', 1999, 'D.SZELIGA77@GMAIL.PL', 'sef%af#wa', 1

EXEC DODAJ\_KLIENTA 'EDYTA', 'LASEK', 273489574, 'NISKA 32 WADOWICE', 1987, 'LASEK55@GMAIL.PL', 'faw#F5W4', 0

EXEC DODAJ\_KLIENTA 'EMILIA', 'JABLONSKA', 293834457, 'AKACJOWA 35/23 KRAKOW', 1989, 'E.JABLONSKA@ONET.PL', 'ace%##w', 0

EXEC DODAJ\_KLIENTA 'FILIP', 'KUZNIAR', 274394857, 'UGOREK 85 KRAKOW', 1982, 'KUZNIAR69@ONET.PL', 'rcwa%rw6', 0

EXEC DODAJ\_KLIENTA 'GABRIEL', 'GAJDA', 294849049, 'LADNA 14 KRAKOW', 1992, 'GAJDA420@VP.PL', 'wrc%awr%cwa', 0

EXEC DODAJ\_KLIENTA 'GRZEGORZ', 'JANKIEWICZ', 225364857, 'MOGILSKA 23/22 KRAKOW', 1992, 'G.JANKIEWICZ2137@GMAIL.PL', 'aw#cwa$cwa4', 0

EXEC DODAJ\_KLIENTA 'HUBERT', 'SAWICKI', 237948364, 'DABSKA 95 KRAKOW', 1993, 'SAWICKI99@WP.PL', 'cwa$#wa3CW3', 1

EXEC DODAJ\_KLIENTA 'IGOR', 'GROCHALA', 299049999, 'JACHOWICZA 4/5 KRAKOW', 1992, 'IGOR.GROCHALA@WP.PL', 'wcW3C33W4#', 0

EXEC DODAJ\_KLIENTA 'IWONA', 'SZARA', 283746580, 'BAJECZNA 5/4 KRAKOW', 1997, 'SZARAAA@ONET.PL', 'vr6AWRRWV3R', 0

EXEC DODAJ\_KLIENTA 'JAKUB', 'BATOR', 374847263, 'SKRZATOW 34B KRAKOW', 1991, 'BATORY43@GMAIL.PL', 'NTe#3TEn', 0

EXEC DODAJ\_KLIENTA 'JAKUB', 'JABLONSKI', 384958333, 'DEKERTA 38 KRAKOW', 1983, 'JAKUBJABLONSKI@GMAIL.PL', 'tnes#E4TN3E', 1

EXEC DODAJ\_KLIENTA 'JOANNA', 'KOWALSKA', 857467938, 'KOZLARSKA 11/3 KRAKOW', 1988, 'KOWALSKAJOANNA@VP.PL', 'setnte$sn44E', 0

EXEC DODAJ\_KLIENTA 'JULIA', 'CZARNEK', 019827364, 'WODNA 7/5 KRAKOW', 1976, 'CZARNEK69@ONET.PL', 'tbeT3e553E', 0

EXEC DODAJ\_KLIENTA 'JULIA', 'KOZLOWSKA', 837462011, 'ZOLNIERSKA 8/7 BALICE', 1970, 'KOZLOWSKAJULKA@GMAIL.PL', 'TEBS$4TES', 1

EXEC DODAJ\_KLIENTA 'JULIA', 'SZELIGA', 827390000, 'KACZA 10/1 KRAKOW', 1980, 'SZELIGAJULECZKA@VP.PL', 'TebT4E#3B', 0

EXEC DODAJ\_KLIENTA 'KACER', 'KRUDYSZ', 162738948, 'KOZIA 3/1 TARNOBRZEG', 1990, 'KKRUDYSZ77@INTERIA.PL', 'TES3TE2B2S34', 0

EXEC DODAJ\_KLIENTA 'KACPER', 'LEWICKI', 843238019, 'BAGROWA 11/2 KRAKOW', 1992, 'LEWICKII8@GMAIL.PL', 'esbt#E3S#b4', 0

EXEC DODAJ\_KLIENTA 'KAMIL', 'KLOC', 283749028, 'MARZANNY 12/3 KRAKOW', 1993, 'KLOCUCH12@VP.PL', 'btes43ETS3TE', 0

EXEC DODAJ\_KLIENTA 'KAMILA', 'DUSZA', 478934029, 'ZABIA 3/12 KRAKOW', 1984, 'DUSZAKK@WP.PL', '3e##3EbTB', 1

EXEC DODAJ\_KLIENTA 'KAROL', 'SZOPA', 548347283, 'KURCZABA 48 KRAKOW', 1984, 'SZOPA.KAROL@VP.PL', 'ebt^eBT44T3EB', 0

EXEC DODAJ\_KLIENTA 'KAROLINA', 'BECZKA', 349238477, 'MEDYCZNA 3 BOCHNIA', 2000, 'BECZKAA@GMAIL.PL', 'Etb$teb%s', 0

EXEC DODAJ\_KLIENTA 'KAROLINA', 'MAJEWSKA', 938465733, 'OBRONNA 97 KRAKOW', 2002, 'KAROLAMAJEWSKA@GMAIL.PL', 'EBTs$%sTE45', 0

EXEC DODAJ\_KLIENTA 'KRYSTIAN', 'BARTON', 940000999, 'MOKRA 12/7 DOBCZYCE', 1989, 'BARTONSKI999@GMAIL.PL', 'btes$3TES3TES', 1

EXEC DODAJ\_KLIENTA 'KRZYSZTOF', 'MACH', 847290000, 'CZAJNA 84 KRAKOW', 2003, 'KRZYSIUMACH@GMAIL.PL', 'ESBT^^T', 0

EXEC DODAJ\_KLIENTA 'KRZYSZTOF', 'DZIEDZIC', 675890467, 'SPORNA 23/5 KRAKOW', 1980, 'DZIEEDZIIC@GMAIL.PL', 'Set6BTES7S', 1

EXEC DODAJ\_KLIENTA 'MONIKA', 'BOCHENEK', 847596046, 'BOCHENKA 22/2 TARNOW', 1971, 'MONIKA\_BOCHENEK@WP.PL', 'rtb$TR3BTR5', 0

EXEC DODAJ\_KLIENTA 'MONIKA', 'NIEZGODA', 374847990, 'ZIMNA 20/2 RZESZOW', 1973, 'MONIKA-NIEZGODA@GMAIL.PL', 'btr$$t4tb$e', 0

EXEC DODAJ\_KLIENTA 'MACIEJ', 'TOMAKA', 748938777, 'KANARKOWA 16/4 KRAKOW', 1965, 'TOTOMAMAKAKA@ONET.PL', 'ebt$4te5tb%e', 0

EXEC DODAJ\_KLIENTA 'MARCEL', 'GORSKI', 374111675, 'TUCHOWSKA 16/2 OLKUSZ', 1968, 'GORSKI\_M@VP.PL', '%#wwr5^wR&R', 0

EXEC DODAJ\_KLIENTA 'MARCIN', 'DUDEK', 883223546, 'WITOSA 54 KRAKOW', 1973, 'DUDUSM@GMAIL.PL', 'Wmw&%M7RWWMRW', 0

EXEC DODAJ\_KLIENTA 'MAREK', 'SLOWIK', 857355222, 'WITOSA 67 KRAKOW', 1979, 'SLOWIKMAREK@GMAIL.PL', 'rbesRE$e', 0

EXEC DODAJ\_KLIENTA 'MARTA', 'SMIALEK', 843473992, 'WSPOLNA 13/3 KRAKOW', 1981, 'SMIALEK-MARTA6@WP.PL', 'RB5E56REA5E7', 1

EXEC DODAJ\_KLIENTA 'MARTYNA', 'BUCIOR', 847001223, 'SANOCKA 36/9 KRAKOW', 1966, 'BUCIOR44@INTERIA.PL', 'teb%$TE5B6', 1

EXEC DODAJ\_KLIENTA 'MATEUSZ', 'GORECKI', 738001243, 'TARNOBRZESKA 44/11 KRAKOW', 1993, 'GORECKIMATI@VP.PL', 'BTEAw$e$', 0

EXEC DODAJ\_KLIENTA 'MALGORZATA', 'KROL', 638799099, 'STRUMIENNA 29/9 WIELICZKA', 1992, 'KROLMALGOSIA33@GMAIL.PL', 'e%^v%rv5WA', 0

--DODAWANIE WEJSC NA OBIEKTY (musi byc dodawane pojedynczo ze wzgledu na sprawdzanie czy dana osoba weszla w czasie otwarcia obiektu)

INSERT INTO [Wejscia na obiekty] VALUES (51, 2, '2020-11-03 11:01:23')

INSERT INTO [Wejscia na obiekty] VALUES (52, 1, '2020-10-04 09:20:55')

INSERT INTO [Wejscia na obiekty] VALUES (52, 3, '2020-12-05 20:10:15')

INSERT INTO [Wejscia na obiekty] VALUES (55, 2, '2020-12-06 13:45:01')

INSERT INTO [Wejscia na obiekty] VALUES (58, 3, '2020-11-08 18:20:11')

INSERT INTO [Wejscia na obiekty] VALUES (59, 3, '2020-11-10 21:09:50')

INSERT INTO [Wejscia na obiekty] VALUES (61, 1, '2020-12-13 08:10:55')

INSERT INTO [Wejscia na obiekty] VALUES (61, 1, '2020-10-14 16:25:34')

INSERT INTO [Wejscia na obiekty] VALUES (69, 1, '2020-09-15 20:20:50')

INSERT INTO [Wejscia na obiekty] VALUES (70, 2, '2020-12-18 14:12:15')

INSERT INTO [Wejscia na obiekty] VALUES (71, 2, '2020-11-20 19:44:52')

INSERT INTO [Wejscia na obiekty] VALUES (84, 3, '2020-12-23 10:34:12')

INSERT INTO [Wejscia na obiekty] VALUES (85, 1, '2020-12-28 13:45:43')

INSERT INTO [Wejscia na obiekty] VALUES (85, 3, '2021-01-03 12:32:44')

INSERT INTO [Wejscia na obiekty] VALUES (85, 2, '2021-01-05 08:11:59')

INSERT INTO [Wejscia na obiekty] VALUES (87, 1, '2021-01-06 14:58:22')

INSERT INTO [Wejscia na obiekty] VALUES (89, 2, '2020-11-10 13:57:33')

INSERT INTO [Wejscia na obiekty] VALUES (90, 3, '2020-11-12 12:00:12')

INSERT INTO [Wejscia na obiekty] VALUES (90, 1, '2020-10-14 17:59:59')

INSERT INTO [Wejscia na obiekty] VALUES (91, 1, '2020-11-15 12:00:01')

INSERT INTO [Wejscia na obiekty] VALUES (92, 3, '2020-09-18 14:08:54')

INSERT INTO [Wejscia na obiekty] VALUES (95, 2, '2020-10-24 17:32:44')

--DODAWANIE DO GRAFIKU PRACOWNIKA (ID\_PRACOWNIKA, OD KIEDY WSTAWIC GRAFIK, NA ILE DNI ROZPISAC, START PRACY (GODZINA), ILOSC GODZIN W PRACY NA DZIEN)

EXEC DODAJ\_GRAFIK\_OKRES 1, '2021-02-10', 100, 8, 7

EXEC DODAJ\_GRAFIK\_OKRES 2, '2021-02-15', 300, 7, 10

EXEC DODAJ\_GRAFIK\_OKRES 3, '2021-02-10', 365, 8, 8

EXEC DODAJ\_GRAFIK\_OKRES 4, '2021-01-11', 100, 8, 8

EXEC DODAJ\_GRAFIK\_OKRES 5, '2021-02-10', 730, 8, 9

EXEC DODAJ\_GRAFIK\_OKRES 6, '2021-02-10', 120, 8, 7

EXEC DODAJ\_GRAFIK\_OKRES 7, '2021-03-20', 90, 8, 7

EXEC DODAJ\_GRAFIK\_OKRES 8, '2021-01-01', 365, 8, 7

EXEC DODAJ\_GRAFIK\_OKRES 9, '2021-02-10', 365, 8, 7

--DODAWANIE URLOPOW (DODANIE URLOPU USUWA AUTOMATYCZNIE REKORDY Z GRAFIKA PRACOWNIKA, NALEŻY WSTAWIAC POJEDYNCZO ZE WZGLĘDU NA DZIAŁANIE WYZWALACZA)

INSERT INTO URLOPY VALUES (1, '2020-12-01', 2)

INSERT INTO URLOPY VALUES (2, '2021-01-10', 5)

INSERT INTO URLOPY VALUES (4, '2020-11-27', 4)

INSERT INTO URLOPY VALUES (8, '2020-10-30', 3)

INSERT INTO URLOPY VALUES (8, '2020-12-03', 2)

INSERT INTO URLOPY VALUES (9, '2020-08-20', 6)

INSERT INTO URLOPY VALUES (9, '2020-10-06', 1)

INSERT INTO URLOPY VALUES (9, '2020-12-27', 1)

INSERT INTO URLOPY VALUES (11, '2020-10-26', 2)

INSERT INTO URLOPY VALUES (12, '2020-11-02', 1)

INSERT INTO URLOPY VALUES (14, '2020-11-12', 3)

INSERT INTO URLOPY VALUES (14, '2020-11-18', 1)

INSERT INTO URLOPY VALUES (15, '2020-12-15', 2)

INSERT INTO URLOPY VALUES (18, '2021-01-03', 3)

INSERT INTO URLOPY VALUES (24, '2020-09-01', 8)

INSERT INTO URLOPY VALUES (25, '2020-10-10', 5)

INSERT INTO URLOPY VALUES (27, '2020-09-04', 4)

INSERT INTO URLOPY VALUES (27, '2020-10-15', 2)

INSERT INTO URLOPY VALUES (28, '2020-12-20', 1)

INSERT INTO URLOPY VALUES (29, '2020-08-01', 7)

INSERT INTO URLOPY VALUES (31, '2020-10-02', 3)

INSERT INTO URLOPY VALUES (33, '2020-11-21', 1)

INSERT INTO URLOPY VALUES (38, '2021-01-05', 4)

INSERT INTO URLOPY VALUES (40, '2020-11-06', 3)

INSERT INTO URLOPY VALUES (40, '2020-12-19', 1)

INSERT INTO URLOPY VALUES (40, '2021-01-10', 1)

INSERT INTO URLOPY VALUES (42, '2020-10-22', 1)

INSERT INTO URLOPY VALUES (42, '2020-01-03', 1)

INSERT INTO URLOPY VALUES (43, '2021-01-02', 6)

INSERT INTO URLOPY VALUES (44, '2020-11-25', 1)

INSERT INTO URLOPY VALUES (45, '2020-12-06', 1)

INSERT INTO URLOPY VALUES (45, '2021-01-15', 1)

INSERT INTO URLOPY VALUES (48, '2020-09-10', 10)

INSERT INTO URLOPY VALUES (49, '2021-01-20', 3)

INSERT INTO URLOPY VALUES (50, '2020-01-09', 4)

--DODAWANIE MANAGEROW DO OBIEKTOW

INSERT INTO MANAGEROWIE VALUES

(15, 1),

(16, 1),

(26, 2),

(25, 2),

(40, 3),

(41, 3)

--DODAWANIE TRENEROW PERSONALNYCH

INSERT INTO [TRENERZY PERSONALNI] VALUES

(8, 1),

(9, 1),

(10, 1),

(11, 1),

(12, 1),

(13, 1),

(14, 1),

(28, 2),

(29, 2)

--DODAWANIE PRZYPISANIA TRENERA DO KLIENTA (NALEŻY WSTAWIAC POJEDYNCZO ZE WZGLĘDU NA DZIAŁANIE WYZWALACZA)

INSERT INTO [KLIENT-TRENER] VALUES (52, 8, 2)

INSERT INTO [KLIENT-TRENER] VALUES (54, 8, 3)

INSERT INTO [KLIENT-TRENER] VALUES (55, 8, 4)

INSERT INTO [KLIENT-TRENER] VALUES (65, 8, 3)

INSERT INTO [KLIENT-TRENER] VALUES (73, 8, 2)

INSERT INTO [KLIENT-TRENER] VALUES (88, 8, 3)

INSERT INTO [KLIENT-TRENER] VALUES (51, 9, 4)

INSERT INTO [KLIENT-TRENER] VALUES (57, 9, 2)

INSERT INTO [KLIENT-TRENER] VALUES (67, 9, 3)

INSERT INTO [KLIENT-TRENER] VALUES (69, 9, 2)

INSERT INTO [KLIENT-TRENER] VALUES (92, 9, 2)

INSERT INTO [KLIENT-TRENER] VALUES (56, 10, 2)

INSERT INTO [KLIENT-TRENER] VALUES (61, 10, 2)

INSERT INTO [KLIENT-TRENER] VALUES (63, 10, 3)

INSERT INTO [KLIENT-TRENER] VALUES (70, 10, 2)

INSERT INTO [KLIENT-TRENER] VALUES (71, 10, 2)

INSERT INTO [KLIENT-TRENER] VALUES (74, 10, 2)

INSERT INTO [KLIENT-TRENER] VALUES (60, 11, 3)

INSERT INTO [KLIENT-TRENER] VALUES (72, 11, 2)

INSERT INTO [KLIENT-TRENER] VALUES (75, 12, 3)

INSERT INTO [KLIENT-TRENER] VALUES (77, 12, 4)

INSERT INTO [KLIENT-TRENER] VALUES (79, 13, 5)

INSERT INTO [KLIENT-TRENER] VALUES (80, 13, 3)

INSERT INTO [KLIENT-TRENER] VALUES (82, 13, 3)

INSERT INTO [KLIENT-TRENER] VALUES (84, 13, 5)

INSERT INTO [KLIENT-TRENER] VALUES (89, 13, 4)

INSERT INTO [KLIENT-TRENER] VALUES (90, 14, 2)

INSERT INTO [KLIENT-TRENER] VALUES (91, 14, 2)

INSERT INTO [KLIENT-TRENER] VALUES (93, 14, 2)

INSERT INTO [KLIENT-TRENER] VALUES (53, 28, 2)

INSERT INTO [KLIENT-TRENER] VALUES (56, 28, 2)

INSERT INTO [KLIENT-TRENER] VALUES (63, 28, 1)

INSERT INTO [KLIENT-TRENER] VALUES (72, 28, 1)

INSERT INTO [KLIENT-TRENER] VALUES (77, 28, 1)

INSERT INTO [KLIENT-TRENER] VALUES (79, 28, 2)

INSERT INTO [KLIENT-TRENER] VALUES (57, 29, 2)

INSERT INTO [KLIENT-TRENER] VALUES (60, 29, 2)

INSERT INTO [KLIENT-TRENER] VALUES (74, 29, 2)

INSERT INTO [KLIENT-TRENER] VALUES (80, 29, 2)

INSERT INTO [KLIENT-TRENER] VALUES (91, 29, 2)

INSERT INTO [KLIENT-TRENER] VALUES (97, 29, 2)

INSERT INTO [KLIENT-TRENER] VALUES (100, 29, 2)

--DODAWANIE PROWADZACYCH GRUPY ZAJECIOWE

EXEC DODAJ\_PROWADZACEGO\_GRUPY 8

EXEC DODAJ\_PROWADZACEGO\_GRUPY 9

EXEC DODAJ\_PROWADZACEGO\_GRUPY 10

EXEC DODAJ\_PROWADZACEGO\_GRUPY 11

EXEC DODAJ\_PROWADZACEGO\_GRUPY 12

EXEC DODAJ\_PROWADZACEGO\_GRUPY 13

EXEC DODAJ\_PROWADZACEGO\_GRUPY 14

EXEC DODAJ\_PROWADZACEGO\_GRUPY 27

EXEC DODAJ\_PROWADZACEGO\_GRUPY 28

EXEC DODAJ\_PROWADZACEGO\_GRUPY 29

--DODAWANIE GRUP ZAJECIOWYCH

INSERT INTO [Grupy\_zajeciowe] VALUES

(1, 1, 'ABT', 8, 20),

(1, 2, 'AKTYWNE 9 MIESIECY', 8, 10),

(1, 3, 'AKTYWNY SENIOR', 8, 10),

(3, 4, 'AQUA AEROBIK', 27, 25),

(3, 5, 'AQUA SENIOR', 27, 20),

(3, 6, 'AQUA BABY', 27, 20),

(3, 7, 'AQUA POWER', 27, 15),

(1, 8, 'BODY BALL', 8, 15),

(1, 9, 'BODY&MIND', 8, 25),

(1, 10, 'BOKS', 9, 16),

(1, 11, 'BUM BUM POSLADKI', 9, 20),

(1, 12, 'CROSS CARDIO', 9, 18),

(1, 13, 'CROSS FITNESS', 9, 18),

(1, 14, 'CROSS TRENING', 9, 18),

(1, 15, 'DANCE DANCE', 10, 25),

(1, 16, 'FAT BURNING', 10, 25),

(1, 17, 'FIT&JUMP', 10, 20),

(1, 18, 'FIZJO TRENING', 11, 20),

(1, 19, 'FIZJO TRENING2', 11, 20),

(1, 20, 'JOGA', 11, 30),

(1, 21, 'KALISTENIKA', 11, 20),

(1, 22, 'LATINO DANCE', 12, 20),

(1, 23, 'POLE DANCE', 12, 10),

(1, 24, 'ZUMBA', 12, 20),

(1, 25, 'SALSA', 12, 20),

(1, 26, 'ZUMBA KIDS', 12, 15),

(1, 27, 'POSLADKI', 13, 15),

(1, 28, 'BRZUCH', 13, 15),

(1, 29, 'NOGI I RAMIONA', 13, 15),

(1, 30, 'TABATA', 14, 20),

(1, 31, 'TRAMPOLINY', 14, 15),

(3, 32, 'BASEN GRUPA PODSTAWOWA', 28, 10),

(3, 33, 'BASEN GRUPA ZAAWANSOWANA', 28, 10),

(3, 34, 'BASEN GRUPA OLIMPIJSKA', 29, 6)

--DODAWANIE ZABIEGOW SPA

INSERT INTO [ZABIEGI SPA] VALUES

(1, 3, 'MANGIR GLEBOKI DETOX', 320, 90, 42),

(2, 3, 'SENSUAL GREEN TEA', 300, 75, 42),

(3, 3, 'DERMO SLIM', 270, 60, 42),

(4, 3, 'AROMATIC HONEY', 260, 90, 42),

(5, 3, 'TONE UP', 350, 75, 42),

(6, 3, 'CELL SLIM', 350, 90, 42),

(7, 3, 'CELL RESHAPE', 350, 75, 42),

(8, 3, 'SENSUAL GREEN TEA', 300, 75, 43),

(9, 3, 'DERMO SLIM', 270, 60, 43),

(10, 3, 'CELL RESHAPE', 350, 75, 43),

(11, 3, 'SNAIL REPAIR', 240, 60, 44),

(12, 3, 'SNAIL REPAIR', 240, 60, 45),

(13, 3, 'SNAIL REPAIR', 240, 60, 46),

(14, 3, 'KOBIDO Z MASKA', 350, 80, 44),

(15, 3, 'KOBIDO Z MASKA', 350, 80, 46),

(16, 3, 'SARANARI', 400, 100, 44),

(17, 3, 'SARANARI', 400, 100, 45),

(18, 3, 'DERMAACNE+', 240, 60, 45),

(19, 3, 'DERMAACNE+', 240, 60, 46),

(20, 3, 'EMBELLIR ZABIEG PODSTAWOWY', 320, 80, 44),

(21, 3, 'EMBELLIR ZABIEG PODSTAWOWY', 320, 80, 45),

(22, 3, 'EMBELLIR ZABIEG PODSTAWOWY', 320, 80, 46),

(23, 3, 'ANTI AGE', 170, 45, 44),

(24, 3, 'ANTI AGE+', 200, 45, 45),

(25, 3, 'ANTI AGE', 170, 45, 46)

--WSTAWIANIE ZAPLANOWANYCH PIELEGNACJI

INSERT INTO Pielegnacje VALUES

(1, 1),

(2, 2),

(3, 3),

(1, 4),

(2, 5),

(3, 6)

--WSTAWIANIE PODWYKONAWCOW PIELEGNACJI

INSERT INTO Podwykonawcy VALUES

(1, 'BOGDANEX', 'SPOKOJNA 132 KRAKOW', 746384928),

(2, 'ZBIGNIEWPOL', 'KROTKA 1 KRAKOW', 537839847),

(3, 'JANUSZEX', 'AKACJOWA 40 KRAKOW', 745839000)

--WSTAWIANIE SZCZEGOLOW PIELEGNACJI OBIEKTOW

INSERT INTO [Szczegoly\_Pielegnacji] VALUES

(1, '2020-09-10', '2020-09-13', 8, 2400, 5, 1),

(2, '2020-09-10', '2020-09-11', 1, 1300, 4, 2),

(3, '2020-10-01', '2020-10-02', 2, 3750, 5, 1),

(4, '2020-10-21', '2020-10-24', 17, 2000, 5, 3),

(5, '2020-11-17', '2020-11-19', 9, 1700, 4, 3),

(6, '2020-12-27', '2020-12-29', 3, 500, 5, 2)

--WSTAWIANIE NOWYCH WYPOSAZEN DO OBIEKTU

INSERT INTO [WYPOSAZENIE OBIEKTU] VALUES

(1, 'BIEZNIA', 10, 'BARDZO DOBRY', 'ABC123'),

(2, 'ROWER STACJONARNY', 8, 'DOBRY', 'XYZ987'),

(1, 'MASZYNA TRENINGOWA #1', 3, 'BARDZO DOBRY', 'ABC123'),

(1, 'MASZYNA TRENINGOWA #2', 2, 'BARDZO DOBRY', 'KLM555'),

(3, 'STOL DO MASAZU WASKI', 5, 'BARDZO DOBRY', 'XYZ987'),

(3, 'STOL DO MASAZU SZEROKI', 3, 'SREDNI', 'XYZ987')

--WSTAWIANIE TYPOW WEJSCIOWEK

INSERT INTO TYP\_WEJSCIOWKI VALUES

(1, 'WSTEP NA OBIEKT SPORTOWY 1 GODZINA'),

(2, 'WSTEP NA OBIEKT SPORTOWY JEDNORAZOWY'),

(3, 'WSTEP NA OBIEKT SPORTOWY KARNET 7 DNI'),

(4, 'WSTEP NA OBIEKT SPORTOWY KARNET 30 DNI'),

(5, 'WSTEP NA OBIEKT SPORTOWY KARNET 30 DNI SENIOR'),

(6, 'WSTEP NA OBIEKT SPORTOWY KARNET 30 DNI STUDENT'),

(7, 'WSTEP NA SPA JEDNORAZOWY'),

(8, 'WSTEP NA SPA 3 DNI')

--WSTAWIANIE DO CENNIKA

INSERT INTO CENNIK VALUES

(1, 1, 1, 10),

(2, 2, 1, 10),

(3, 1, 2, 20),

(4, 2, 2, 20),

(5, 1, 3, 60),

(6, 2, 3, 60),

(7, 1, 4, 140),

(8, 2, 4, 140),

(9, 1, 5, 120),

(10, 2, 5, 120),

(11, 1, 6, 100),

(12, 2, 6, 100),

(13, 3, 7, 50)

--WSTAWIANIE ZAKUPIONYCH KARNETOW PRZEZ KLIENTOW

EXEC DODAJ\_KARNET 51,1

EXEC DODAJ\_KARNET 52,2

EXEC DODAJ\_KARNET 51,3

EXEC DODAJ\_KARNET 54,4

EXEC DODAJ\_KARNET 55,4

EXEC DODAJ\_KARNET 56,2

EXEC DODAJ\_KARNET 58,2

EXEC DODAJ\_KARNET 59,3

EXEC DODAJ\_KARNET 60,7

EXEC DODAJ\_KARNET 61,1

EXEC DODAJ\_KARNET 61,4

EXEC DODAJ\_KARNET 63,4

EXEC DODAJ\_KARNET 66,3

EXEC DODAJ\_KARNET 67,7

EXEC DODAJ\_KARNET 69,1

EXEC DODAJ\_KARNET 51,2

EXEC DODAJ\_KARNET 69,7

EXEC DODAJ\_KARNET 69,8

EXEC DODAJ\_KARNET 71,8

EXEC DODAJ\_KARNET 71,2

EXEC DODAJ\_KARNET 74,4

EXEC DODAJ\_KARNET 77,4

EXEC DODAJ\_KARNET 79,4

EXEC DODAJ\_KARNET 80,6

EXEC DODAJ\_KARNET 81,4

EXEC DODAJ\_KARNET 88,6

EXEC DODAJ\_KARNET 90,4

--WSTAWIANIE OKRESOWYCH OFERT SPECJALNYCH (TERMIN DO KIEDY SA WAZNE)

INSERT INTO [OFERTY SPECJALNE] VALUES

(1, 3, 'Znizka 10% na Dzien Kobiet', '2021-03-08'),

(2, 1, 'Znizka 10% na Dzien Mezczyzn', '2021-03-10'),

(3, 1, 'Znizka 5% na Dzien Wiosny', '2021-03-21'),

(4, 2, 'Znizka 5% na Dzien Wiosny', '2021-03-21'),

(5, 3, 'Znizka 5% na Dzien Wiosny', '2021-03-21')

--WSTAWIANIE DO GRAFIKU GRUPY ZAJECIOWE

INSERT INTO Grafik\_Grup VALUES

(1, '2021-01-06', 16),

(3, '2021-01-06', 17),

(5, '2021-01-07', 18),

(7, '2021-01-07', 16),

(9, '2021-01-07', 16),

(11, '2021-01-07', 19),

(13, '2021-01-08', 20),

(15, '2021-01-08', 20),

(17, '2021-01-08', 18),

(19, '2021-01-08', 19),

(21, '2021-01-09', 15),

(11, '2021-01-09', 16),

(13, '2021-01-09', 15),

(15, '2021-01-09', 15),

(17, '2021-01-11', 17),

(19, '2021-01-11', 17),

(21, '2021-01-11', 18),

(23, '2021-01-11', 19),

(25, '2021-01-12', 18),

(27, '2021-01-12', 18),

(29, '2021-01-12', 17),

(31, '2021-01-13', 20),

(27, '2021-01-14', 21),

(29, '2021-01-14', 17),

(31, '2021-01-15', 21)

Funkcje, procedury, wyzwalacze i ich opisy:

**Wyzwalacz UrlopyCheckDate**

-wyzwalacz sprawdza czy nowo dodany urlop dla pracownika nie koliduje z już istniejącym.

IF OBJECT\_ID('UrlopyCheckDate', 'TR') IS NOT NULL DROP TRIGGER UrlopyCheckDate

GO

CREATE TRIGGER UrlopyCheckDate on URLOPY -- dodawany urlop pracownika nie moze sie nakladac na juz posiadany

AFTER INSERT

AS

IF EXISTS(

SELECT \* FROM URLOPY AS U\_OLD -- urlopy stare

JOIN INSERTED AS U\_NEW -- urlopy wstawione

ON ((DATEADD(DAY,U\_NEW.CZAS\_TRWANIA, U\_NEW.DATA\_URLOPU) >= U\_OLD.DATA\_URLOPU)

AND (U\_NEW.DATA\_URLOPU <= DATEADD(DAY,U\_OLD.CZAS\_TRWANIA, U\_OLD.DATA\_URLOPU))

AND (U\_OLD.ID\_PRACOWNIKA = U\_NEW.ID\_PRACOWNIKA))

WHERE NOT (U\_OLD.ID\_PRACOWNIKA = U\_NEW.ID\_PRACOWNIKA AND U\_OLD.DATA\_URLOPU = U\_NEW.DATA\_URLOPU)

) BEGIN

DELETE FROM URLOPY WHERE ((ID\_PRACOWNIKA IN (SELECT ID\_PRACOWNIKA FROM inserted)) AND (DATA\_URLOPU IN (SELECT DATA\_URLOPU FROM inserted)))

PRINT 'NIE MOZNA WSTAWIC URLOPU! KOLIDUJE Z INNYM URLOPEM!'

END

ELSE

BEGIN

DECLARE @ID INT = (SELECT ID\_PRACOWNIKA FROM INSERTED)

DECLARE @URLOP\_START DATE = (SELECT DATA\_URLOPU FROM INSERTED)

DECLARE @CZAS\_TRWANIA INT = (SELECT CZAS\_TRWANIA FROM INSERTED)

DELETE FROM GRAFIK WHERE ID\_PRACOWNIKA = @ID AND DATA BETWEEN @URLOP\_START AND DATEADD(DAY, @CZAS\_TRWANIA, @URLOP\_START)

END

GO

**Funkcja IloscTrenerów**

-funkcja zwraca ilość trenerów personalnych z wszystkich obietków

IF OBJECT\_ID('IloscTrenerow', 'FN') IS NOT NULL DROP FUNCTION IloscTrenerow

GO

CREATE FUNCTION IloscTrenerow()

RETURNS INT AS BEGIN

DECLARE @ILOSC INT = (SELECT COUNT(\*) FROM [TRENERZY PERSONALNI])

RETURN @ILOSC

END

GO

**Procedura SpisPracownikowObiektu**

-procedura wypisuje wszystkich pracowników obiektu o identyfikatorze danym parametrem

IF OBJECT\_ID('SpisPracownikowObiektu', 'P') IS NOT NULL DROP PROC SpisPracownikowObiektu

GO

CREATE PROCEDURE SpisPracownikowObiektu(@ID\_OBJ INT) AS BEGIN

SELECT IMIE, NAZWISKO FROM OSOBA O JOIN PRACOWNIK P ON (O.ID\_OSOBY = P.ID\_PRACOWNIKA)

WHERE (P.ID\_OBIEKTU = @ID\_OBJ)

END

GO

**Funkcja IloscPodopiecznych**

-funkcja zwraca ilość podopiecznych danego trenera

IF OBJECT\_ID('IloscPodopiecznych', 'FN') IS NOT NULL DROP FUNCTION IloscPodopiecznych

GO

CREATE FUNCTION IloscPodopiecznych(@ID\_TRENERA INT) RETURNS INT

BEGIN

DECLARE @ILOSC INT

SET @ILOSC = (SELECT COUNT(\*) FROM [KLIENT-TRENER] WHERE (ID\_TRENERA = @ID\_TRENERA))

RETURN @ILOSC

END

GO

**Wyzwalacz PremiaIloscPodopiecznych**

-wyzwalacz sprawdza ilość podopiecznych danego trenera poprzez funkcje IloscPodopiecznych. Jeśli ich liczebność jest większa od czterech, to za każdego kolejnego podopiecznego przyznawane jest 10 procent premii do aktualnej stawki.

IF OBJECT\_ID('PremiaIloscPodopiecznych', 'TR') IS NOT NULL DROP TRIGGER PremiaIloscPodopiecznych

GO

-- JESLI ILOSC PODOPIECZNYCH JEST WIEKSZA OD 4 TO ZA KAZDEGO KOLEJNEGO PODOPIECZNEGO TRENER DOSTAJE 10% PREMII

-- JESLI WARTOSC ZMIEJSZA SIE NA MNIEJSZA NIZ 4, ODEJMUJEMY PREMIE

CREATE TRIGGER PremiaIloscPodopiecznych ON [KLIENT-TRENER]

AFTER INSERT, DELETE AS

DECLARE @ID\_TRENERA INT

DECLARE @IloscPodopiecznych INT

IF EXISTS(SELECT \* FROM INSERTED) BEGIN

SET @ID\_TRENERA = (SELECT ID\_TRENERA FROM INSERTED)

SET @IloscPodopiecznych = dbo.IloscPodopiecznych(@ID\_TRENERA)

IF(@IloscPodopiecznych > 4) BEGIN

UPDATE PRACOWNIK SET [PREMIA(%)] += 0.1 WHERE ID\_PRACOWNIKA = @ID\_TRENERA

END

END

IF EXISTS(SELECT \* FROM DELETED) BEGIN

SET @ID\_TRENERA = (SELECT DISTINCT ID\_TRENERA FROM DELETED)

SET @IloscPodopiecznych = dbo.IloscPodopiecznych(@ID\_TRENERA)

IF(@IloscPodopiecznych >= 4) BEGIN

UPDATE PRACOWNIK SET [PREMIA(%)] -= 0.1 WHERE ID\_PRACOWNIKA = @ID\_TRENERA

END

END

GO

**Procedura Dodaj\_klienta**

-Dodanie klienta do bazy danych. Pierwsza część danych dodawana jest do tabeli osoba, a druga do tabeli klient. Należy zauważyć że baza danych nie przechowuje haseł użytkowników przez co potencjalne wykradzenie danych nie skutkuje znajomością haseł. Jak wiemy SHA jest algorytmem bezpiecznym i jego działania nie da się odwrócić zatem hash hasła nie będzie miał żadnej wartości.

create procedure Dodaj\_klienta

(

@imie varchar(20), @nazwisko varchar(20), @telefon varchar(20), @adres varchar(40), @rokUrodzenia INT,

@MAIL VARCHAR(30), @HASLO VARCHAR(40), @PRO BIT

)

as

begin

DECLARE @HASH\_HASLA VARCHAR(40)

SET @HASH\_HASLA = HASHBYTES('SHA2\_256',@HASLO)

DECLARE @ID INT

SET @ID = (SELECT TOP 1 ID\_OSOBY FROM OSOBA ORDER BY ID\_OSOBY DESC)

IF @ID IS NOT NULL

SET @ID = @ID+1

ELSE

SET @ID = 1

INSERT INTO OSOBA (ID\_OSOBY,IMIE,NAZWISKO,TELEFON,ADRES,[ROK URODZENIA])

VALUES (@ID, @IMIE, @NAZWISKO,@TELEFON, @ADRES,@rokUrodzenia)

INSERT INTO KLIENT --(ID\_KLIENTA, [RODZAJ CZLONKOWSTWA],PRO\_STATUS,NR\_SZAFKI,POSIADANY\_KARNET)

VALUES (@ID,@PRO,@MAIL, @HASH\_HASLA );

END

GO

**FUNKCJA WALIDACJA UZYTKOWNIKA**

- FUNKCJA TA SPRAWDZA CZY PODANE LOGIN (EMAIL) I HASLO ODPOWIADAJA PEWNEJ PARZE LOGIN, HASŁO Z BAZY DANYCH. UWAGA! FUNKCJA NIE JEST ODPORNA NA SQL INJECTION, OBSŁUGA OCHRONY PRZED TYM ATAKIEM POWINNA BYĆ ZREALIZOWANA W APLIKACJI KLIENCKIEJ.

CREATE FUNCTION WALIDACJA\_UZYTKOWNIKA (@LOGIN VARCHAR(30), @HASLO VARCHAR(40))

RETURNS BIT

AS

BEGIN

DECLARE @HASH\_HASLA VARCHAR(40)

SET @HASH\_HASLA = HASHBYTES('SHA2\_256', @HASLO)

DECLARE @RETURN\_VALUE BIT

IF EXISTS(SELECT \* FROM KLIENT WHERE MAIL = @LOGIN AND HASH\_HASLA = @HASH\_HASLA)

SET @RETURN\_VALUE = 1

ELSE SET @RETURN\_VALUE = 0

RETURN @RETURN\_VALUE

END

GO

**Procedura DODAJ\_PRACOWNIKA**

-procedura dodaje pracownika (analogicznie do procedury DODAJ\_KLIENTA)

IF OBJECT\_ID('DODAJ\_PRACOWNIKA', 'P') IS NOT NULL DROP PROC DODAJ\_PRACOWNIKA

GO

create procedure DODAJ\_PRACOWNIKA

(

@imie varchar(20), @nazwisko varchar(20), @telefon varchar(20), @adres varchar(40), @rokUrodzenia INT,

@ID\_OBIEKTU INT, @WYNAGRODZENIE INT, @STANOWISKO VarChar(15)

)

as

begin

DECLARE @ID INT

SET @ID = (SELECT TOP 1 ID\_OSOBY FROM OSOBA ORDER BY ID\_OSOBY DESC)

IF @ID IS NOT NULL

SET @ID = @ID+1

ELSE

SET @ID = 1

INSERT INTO OSOBA (ID\_OSOBY,IMIE,NAZWISKO,TELEFON,ADRES,[ROK URODZENIA])

VALUES (@ID, @IMIE, @NAZWISKO,@TELEFON, @ADRES,@rokUrodzenia)

INSERT INTO PRACOWNIK

VALUES (@ID, 1.0, @ID\_OBIEKTU, @WYNAGRODZENIE, @STANOWISKO);

END

GO

**Procedura USUN\_PRACOWNIKA**

-procedura usuwa pracownika z tabeli PRACOWNIK jeżeli istnieje pracownik o identyfikatorze danym parametrem, następnie jego dane z tabeli OSOBA

IF OBJECT\_ID('USUN\_PRACOWNIKA', 'P') IS NOT NULL DROP PROC USUN\_PRACOWNIKA

GO

create procedure USUN\_PRACOWNIKA (@ID INT)

AS BEGIN

IF EXISTS (SELECT \* FROM PRACOWNIK WHERE ID\_PRACOWNIKA = @ID) BEGIN

DELETE FROM PRACOWNIK WHERE ID\_PRACOWNIKA = @ID

DELETE FROM OSOBA WHERE ID\_OSOBY = @ID

END

ELSE

BEGIN

PRINT 'BRAK PRACOWNIKA O PODANYM ID'

END

END

GO

**Procedura USUN\_KLIENTA**

-procedura usuwa klienta z tabeli KLIENT jeżeli istnieje klient o identyfikatorze danym parametrem, następnie jego dane z tabeli OSOBA

IF OBJECT\_ID('USUN\_KLIENTA', 'P') IS NOT NULL DROP PROC USUN\_KLIENTA

GO

create procedure USUN\_KLIENTA (@ID INT)

AS BEGIN

IF EXISTS (SELECT \* FROM KLIENT WHERE ID\_KLIENTA = @ID) BEGIN

DELETE FROM KLIENT WHERE ID\_KLIENTA = @ID

DELETE FROM OSOBA WHERE ID\_OSOBY = @ID

END

ELSE

BEGIN

PRINT 'BRAK KLIENTA O PODANYM ID'

END

END

GO

**Funkcja WYDATKI\_Z\_DNIA**

-Funkcja wydatki z dnia podsumowuje całkowitą ilość wydatków, to znaczy wypłaty pracowników oraz koszty wszelkiego rodzaju prac konserwacyjnych, zakładając że płatność następuje w dniu zakończenia prac.

IF OBJECT\_ID('WYDATKI\_Z\_DNIA', 'FN') IS NOT NULL DROP FUNCTION WYDATKI\_Z\_DNIA

GO

CREATE FUNCTION WYDATKI\_Z\_DNIA (@ID\_OBIEKTU INT, @DATA DATE)

RETURNS INT

AS

BEGIN

DECLARE @Wydatki INT

SET @Wydatki = (SELECT SUM(WYNAGRODZENIE\*ILOSC\_GODZIN\*[PREMIA(%)])

FROM GRAFIK G JOIN PRACOWNIK P ON G.ID\_PRACOWNIKA = P.ID\_PRACOWNIKA

WHERE P.ID\_OBIEKTU = @ID\_OBIEKTU) +

(SELECT SUM(S.Cena\_pielegnacji)

FROM Pielegnacje P JOIN Szczegoly\_Pielegnacji S ON P.ID\_pielegnacji = S.ID\_PIELEGNACJI

WHERE P.ID\_OBIEKTU = @ID\_OBIEKTU AND S.Data\_ZPielegnacji = @DATA)

RETURN @Wydatki

END

GO

**Funkcja PRZYCHODY\_Z\_DNIA**

-Funkcja ta sumuje całość przychodów z danego dnia, przychody są umieszczone w tabeli przychody dnia co pozwala uniknąć konieczności przeglądania wszystkich sprzedaży z całej historii

IF OBJECT\_ID('PRZYCHODY\_Z\_DNIA', 'FN') IS NOT NULL DROP FUNCTION PRZYCHODY\_Z\_DNIA

GO

CREATE FUNCTION PRZYCHODY\_Z\_DNIA (@ID\_OBIEKTU INT)

RETURNS INT

AS

BEGIN

DECLARE @PRZYCHODY INT

SET @PRZYCHODY = (SELECT SUM(CENA\*ILOSC\_SPRZEDAZY)

FROM SPRZEDAZE\_DNIA S JOIN CENNIK C ON S.RODZAJ\_KARNETU = C.TYP\_WEJSCIOWKI

WHERE S.ID\_OBIEKTU = @ID\_OBIEKTU)

RETURN @PRZYCHODY

END

GO

**Procedura PODSUMOWANIE\_DNIA**

-Procedura ta generuje wpis do tabeli bilans która może służyć później do tworzenia podsumowań oraz analizy pracy obiektów. Procedura ta usuwa z tabeli przychody dnia wszystkie rekordy które zostały zawarte w posumowaniu zapewniając w ten sposób że w tabeli tej znajdują się wyłącznie przychody otrzymane w danym dniu co powoduje że procedura wykonuje się szybciej. Jest ona zaprojektowana tak aby używać jej codziennie przed zamknięciem lub jeżeli klub jest czynny dłużej to przed północą. Należy ją wykonywać tak jak zamknięcie kasy w sklepie aby wygenerować podsumowanie.

IF OBJECT\_ID('PODSUMOWANIE\_DNIA', 'P') IS NOT NULL DROP PROC PODSUMOWANIE\_DNIA

GO

CREATE PROCEDURE PODSUMOWANIE\_DNIA(@ID\_OBIEKTU INT) AS -- W ADRESIE MIASTO MUSI BYĆ PIERWSZE

BEGIN

DECLARE @WYDATKI INT = dbo.WYDATKI\_Z\_DNIA(@ID\_OBIEKTU, GETDATE())

DECLARE @PRZYCHODY INT = dbo.PRZYCHODY\_Z\_DNIA(@ID\_OBIEKTU)

DELETE FROM SPRZEDARZE\_DNIA WHERE ID\_OBIEKTU = @ID\_OBIEKTU

DECLARE @ADRES VARCHAR(20)

SET @ADRES = (SELECT ADRES FROM OBIEKTY WHERE ID\_OBIEKTU = @ID\_OBIEKTU)

INSERT INTO BILANS VALUES(@ID\_OBIEKTU, GETDATE(), SUBSTRING(@ADRES, 1, CHARINDEX(' ', @ADRES) - 1), (SELECT NAZWA FROM OBIEKTY WHERE ID\_OBIEKTU = @ID\_OBIEKTU), @PRZYCHODY, @WYDATKI)

END

GO

**Wyzwalacz NIEAUTORYZOWANY\_DOSTEP**

-wyzwalacz sprawdza godzinę wejścia klienta na dany obiekt, następnie weryfikuje czy nie dostał się on tam poza godzinami otwarcia danego obiektu. Jeśli tak, to następuje wyświetlenie się komunikatu.

IF OBJECT\_ID('NIEAUTORYZOWANY\_DOSTEP', 'TR') IS NOT NULL DROP TRIGGER NIEAUTORYZOWANY\_DOSTEP

GO

CREATE TRIGGER NIEAUTORYZOWANY\_DOSTEP ON [Wejscia na obiekty]

AFTER INSERT AS

DECLARE @ID\_KLIENTA INT = (SELECT ID\_KLIENTA FROM INSERTED)

DECLARE @ID\_OBIEKTU INT = (SELECT ID\_OBIEKTU FROM INSERTED)

DECLARE @DATA DATETIME = (SELECT DATA FROM INSERTED)

DECLARE @DZIEN\_WEJSCIA VARCHAR(20) = (SELECT DATENAME(WEEKDAY, @Data))

DECLARE @GODZINA\_WEJSCIA INT = (SELECT DATEPART(HOUR, @DATA))

SET @DZIEN\_WEJSCIA = (

CASE @DZIEN\_WEJSCIA

WHEN 'Monday' THEN 'Poniedzialek'

WHEN 'Tuesday' THEN 'Wtorek'

WHEN 'Wednesday' THEN 'Sroda'

WHEN 'Thursday' THEN 'Czwartek'

WHEN 'Friday' THEN 'Piatek'

WHEN 'Saturday' THEN 'Sobota'

WHEN 'Sunday' THEN 'Niedziela'

END)

DECLARE @ID VARCHAR(30) = (SELECT CAST(@ID\_OBIEKTU as varchar(10)))

DECLARE @GODZINA\_OTWARCIA INT

DECLARE @GODZINA\_ZAMKNIECIA INT

DECLARE @QUERY NVARCHAR(150) = 'SELECT @G\_O = CAST(SUBSTRING('+@DZIEN\_WEJSCIA +',1,1) AS INT), @G\_Z = CAST(SUBSTRING('+@DZIEN\_WEJSCIA +',3,2) AS INT) FROM [GODZINY OTWARCIA] WHERE ID\_OBIEKTU=' + @ID

DECLARE @PARAMS NVARCHAR(150) = '@G\_O INT OUTPUT, @G\_Z INT OUTPUT, @DZIEN\_WEJSCIA VARCHAR(20), @ID INT'

EXECUTE SP\_EXECUTESQL @QUERY, @PARAMS,@G\_O = @GODZINA\_OTWARCIA OUTPUT, @G\_Z = @GODZINA\_ZAMKNIECIA OUTPUT, @DZIEN\_WEJSCIA = @DZIEN\_WEJSCIA, @ID = @ID

IF @GODZINA\_WEJSCIA NOT BETWEEN @GODZINA\_OTWARCIA AND @GODZINA\_ZAMKNIECIA

BEGIN

PRINT 'WYKRYTO NIEAUTORYZOWANY DOSTEP!!!'

PRINT CONCAT('GODZINY OTWARCIA OBIEKTU:', (@GODZINA\_OTWARCIA), '-', @GODZINA\_ZAMKNIECIA)

PRINT CONCAT('GODZINA WEJSCIA: ',@GODZINA\_WEJSCIA,' ID\_KLIENTA: ', @ID\_KLIENTA)

END

GO

**Wyzwalacz ZLA\_OPINIA**

-wyzwalacz sprawdza ocenę wystawioną przez klienta. Jeśil jest ona mniejsza bądź równa 3, wysyła do niego maila.

IF OBJECT\_ID('ZLA\_OPINIA', 'TR') IS NOT NULL DROP TRIGGER ZLA\_OPINIA

GO

CREATE TRIGGER ZLA\_OPINIA ON [OPINIE KLIENTOW]

AFTER INSERT AS

DECLARE @OCENA INT = (SELECT [OCENA(1-10)] FROM INSERTED)

DECLARE @MAIL VARCHAR(30) = (SELECT K.MAIL FROM INSERTED I JOIN KLIENT K ON I.ID\_KLIENTA = K.ID\_KLIENTA)

IF @OCENA <= 3

BEGIN

EXEC msdb.dbo.sp\_send\_dbmail

@PROFILE\_NAME = 'myEmail',

@recipients = @MAIL,

@body = 'dziękujemy za Twoja opinie. Przykro nam ze nie spelnilismy oczekiwan, stale dazymy do tego by byc coraz lepszym.',

@subject = 'Dziękujemy za Twoja opinie'

END

GO

**Funkcja SPIS\_KLIENTOW\_OBIEKT**

-funkcja zwraca klientów posiadających karnet w obiekcie o identyfikatorze danym parametrem.

IF OBJECT\_ID('SPIS\_KLIENTOW\_OBIEKT', 'IF') IS NOT NULL DROP FUNCTION SPIS\_KLIENTOW\_OBIEKT

GO

CREATE FUNCTION SPIS\_KLIENTOW\_OBIEKT (@ID\_KLIENTA INT, @ID\_OBIEKTU INT)

RETURNS TABLE AS RETURN(

SELECT O.ID\_OSOBY, O.IMIE, O.NAZWISKO, A.ID\_OBIEKTU FROM OSOBA O JOIN

(SELECT KK.ID\_KLIENTA, C.ID\_OBIEKTU FROM [KLIENT-KARNET] KK JOIN CENNIK C ON KK.ID\_KARNETU = C.ID\_KARNETU WHERE KK.ID\_KLIENTA = @ID\_KLIENTA AND C.ID\_OBIEKTU = @ID\_OBIEKTU) A ON

A.ID\_KLIENTA = O.ID\_OSOBY

)

GO

**Funkcja SPIS\_PRACOWNIKOW\_OBIEKT**

-funkcja zwraca pracowników zatrudniowych w obiekcie o identyfikatorze danym parametrem

IF OBJECT\_ID('SPIS\_PRACOWNIKOW\_OBIEKT', 'IF') IS NOT NULL DROP FUNCTION SPIS\_PRACOWNIKOW\_OBIEKT

GO

CREATE FUNCTION SPIS\_PRACOWNIKOW\_OBIEKT (@ID\_OBIEKTU INT)

RETURNS TABLE AS RETURN(

SELECT O.ID\_OSOBY, O.IMIE, O.NAZWISKO FROM OSOBA O JOIN PRACOWNIK K ON O.ID\_OSOBY = K.ID\_OBIEKTU WHERE K.ID\_OBIEKTU = @ID\_OBIEKTU

)

GO

**Procedura DODAJ\_GRAFIK\_OKRES**

-procedura dodaje grafik dla danego pracownika w danym przedziale czasowym (@START\_PRACY - @ILOSC\_GODZIN) na ilość dni daną parametrem, począwszy od @OD\_DNIA

IF OBJECT\_ID('DODAJ\_GRAFIK\_OKRES', 'P') IS NOT NULL DROP PROC DODAJ\_GRAFIK\_OKRES

GO

CREATE PROCEDURE DODAJ\_GRAFIK\_OKRES (@ID\_PRACOWNIKA INT, @OD\_DNIA DATE, @ILOSC\_DNI INT, @START\_PRACY INT, @ILOSC\_GODZIN INT) AS

BEGIN

DECLARE @I INT = 0

WHILE @I < @ILOSC\_DNI

BEGIN

INSERT INTO GRAFIK VALUES

(@ID\_PRACOWNIKA, DATEADD(DAY, @I, @OD\_DNIA), @START\_PRACY, @ILOSC\_GODZIN)

SET @I = @I+1

END

END

GO

**Procedura DODAJ\_PROWADZACEGO\_GRUPY**

-procedura mająca na celu przypisanie pracownika do danej grupy jako jej prowadzący

IF OBJECT\_ID('DODAJ\_PROWADZACEGO\_GRUPY', 'P') IS NOT NULL DROP PROC DODAJ\_PROWADZACEGO\_GRUPY

GO

CREATE PROCEDURE DODAJ\_PROWADZACEGO\_GRUPY (@ID\_PRACOWNIKA INT) AS

BEGIN

INSERT INTO Prowadzacy\_grupy VALUES

(@ID\_PRACOWNIKA, (SELECT ID\_OBIEKTU FROM PRACOWNIK WHERE ID\_PRACOWNIKA = @ID\_PRACOWNIKA))

END

GO

**Procedura GRAFIK\_GRUPY\_NA\_KOLEJNY\_MIESIAC**

-procedura wyświetla grafik grupy o identyfikatorze danym parametrem na kolejny miesiąc począwszy od obecnego dnia

IF OBJECT\_ID('GRAFIK\_GRUPY\_NA\_KOLEJNY\_MIESIAC', 'P') IS NOT NULL DROP PROC GRAFIK\_GRUPY\_NA\_KOLEJNY\_MIESIAC

GO

CREATE PROCEDURE GRAFIK\_GRUPY\_NA\_KOLEJNY\_MIESIAC (@ID\_GRUPY INT) AS

BEGIN

SELECT B.ID\_GRUPY, B.NAZWA\_GRUPY, A.DATA, A.GODZINA FROM Grupy\_zajeciowe B JOIN

(SELECT ID\_GRUPY, DATA, GODZINA FROM Grafik\_Grup WHERE ID\_Grupy = @ID\_GRUPY AND DATA BETWEEN GETDATE() AND DATEADD(MONTH, 1, GETDATE())) AS A

ON A.ID\_Grupy = B.ID\_Grupy

END

GO

**Procedura TOP\_10\_NAJBARDZIEJ\_AKTYWNYCH\_KLIENTOW\_MIESIACA**

-procedura wyświetla dziesięciu klientów którzy mieli odnotowane najwięcej wejść na obiekty w ostatnim miesiącu (licząc od obecnego dnia)

IF OBJECT\_ID('TOP\_10\_NAJBARDZIEJ\_AKTYWNYCH\_KLIENTOW\_MIESIACA', 'P') IS NOT NULL DROP PROC TOP\_10\_NAJBARDZIEJ\_AKTYWNYCH\_KLIENTOW\_MIESIACA

GO

CREATE PROCEDURE TOP\_10\_NAJBARDZIEJ\_AKTYWNYCH\_KLIENTOW\_MIESIACA AS

BEGIN

SELECT O.ID\_OSOBY, O.IMIE, O.NAZWISKO, [ILOSC WEJSC] FROM

(SELECT TOP 10 ID\_KLIENTA, COUNT(\*) AS [ILOSC WEJSC] FROM [Wejscia na obiekty] WHERE DATA BETWEEN DATEADD(MONTH, -1, GETDATE()) AND GETDATE() GROUP BY ID\_KLIENTA) A

JOIN OSOBA O ON A.ID\_KLIENTA = O.ID\_OSOBY ORDER BY [ILOSC WEJSC] DESC

END

GO

**Procedura DODAJ\_KARNET**

-procedura dodaje karnet zakupiony przez klienta, następnie odnotowuje to w tabeli sprzedaże dnia. (jeśli taki sam typ karnetu został kupiony przez innego klienta w ten sam dzień, wartość rekordu jest inkrementowana)

IF OBJECT\_ID('DODAJ\_KARNET', 'P') IS NOT NULL DROP PROC DODAJ\_KARNET

GO

CREATE PROCEDURE DODAJ\_KARNET (@ID\_KLIENTA INT, @ID\_KARNETU INT) AS

BEGIN

INSERT INTO [KLIENT-KARNET] VALUES (@ID\_KLIENTA, @ID\_KARNETU, GETDATE())

DECLARE @ID\_OBIEKTU INT = (SELECT ID\_OBIEKTU FROM CENNIK WHERE ID\_KARNETU = @ID\_KARNETU)

IF EXISTS (SELECT \* FROM SPRZEDAZE\_DNIA WHERE ID\_OBIEKTU = @ID\_OBIEKTU AND RODZAJ\_KARNETU = @ID\_KARNETU)

UPDATE SPRZEDAZE\_DNIA SET ILOSC\_SPRZEDAZY += 1 WHERE ID\_OBIEKTU = @ID\_OBIEKTU AND RODZAJ\_KARNETU = @ID\_KARNETU

ELSE

INSERT INTO SPRZEDAZE\_DNIA VALUES (@ID\_OBIEKTU, 1, @ID\_KARNETU)

END

GO

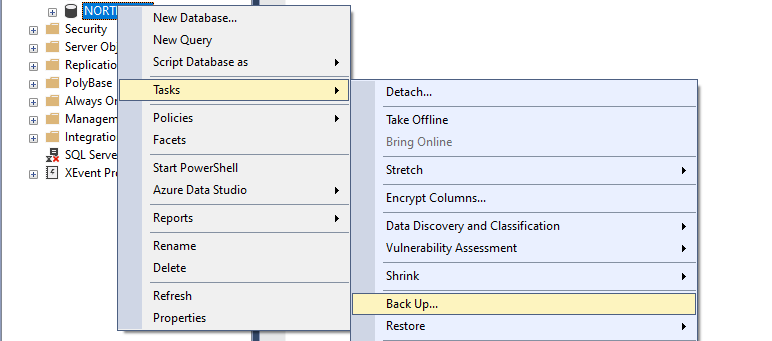
Strategia pielęgnacji bazy danych

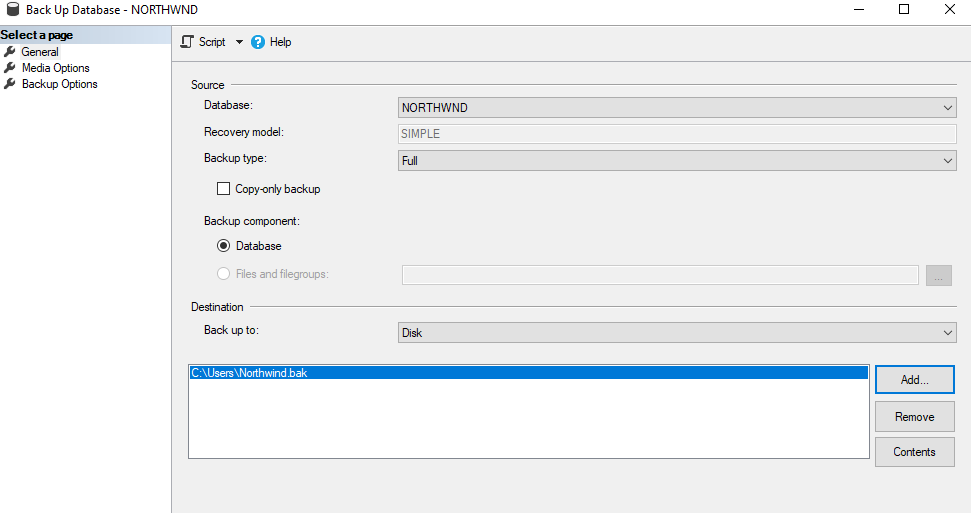
**Kopia zapasowa – w jakim celu?**

Aby uniknąć trudności w funkcjonowaniu przedsiębiorstwa na wypadek awarii, uszkodzenia dysku twardego lub innych zdarzeń, w ramach których utracimy naszą bazę danych, musimy zadbać o wykonanie jej backupu – kopii zapasowej.

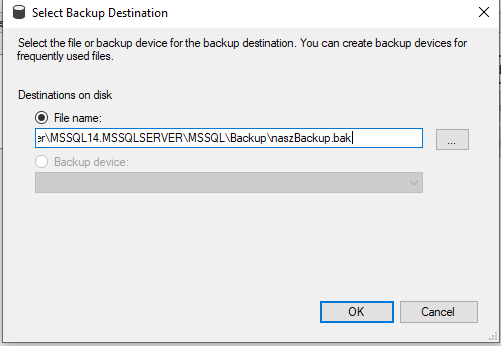
**Jak utworzyć kopię zapasową?**

Zaczynamy od wybrania funkcji back up na interesującej nas bazie danych:

W wyskakującym oknie, w polu Back up to możemy zmienić ścieżkę do folderu, w którym kopia zapasowa bazy danych zostanie zapisana poprzez usunięcie istniejącej i dodatnie własnej (np. aby stworzyć nowe pliki, zamiast jedynie nadpisywać istniejące).



W przyadku tworzenia nowej ścieżki nadajemy nazwę tworzonej kopii (np. Nazwa\_bazy Data\_utworzenia) I musimy pamiętać o rozszerzeniu \*.bak



Właśnie utworzyliśmy kopię zapasową!

**Jaką strategię pięlęgnacji bazy danych przyjmiemy?**

Zacznijmy od tego, że serwer SQL umożliwia tworzenie trzech typów kopii zapasowych: pełnych kopii zapasowych baz danych, kopii dziennika transakcyjnego oraz kopii różnicowych baz danych.

Pełna kopia – zawiera wszystkie dane znajdujące się w bazie,

Kopia różnicowa - zawiera tylko zakresy zmodyfikowane od czasu wykonania ostatniej pełnej kopii plus wymaganą do odtworzenia bazy część dziennika transakcyjnego,

Kopia dziennika transakcyjnego — zawiera wszystkie wpisy dziennika transakcyjnego zapisane od momentu wykonania jego poprzedniej kopii

Z uwagi na profil działania obiektu, nie możemy pozwolić sobie na utratę danych z dłuższego okresu. Nasza baza danych musi więc działać w trybie FULL.

Aby zmienić tryb odtwarzania bazy danych wykonujemy instrukcję:

*ALTER DATABASE nazwa\_bazy*

*SET RECOVERY FULL*

Dzięki temu zabiegowi możemy przyjąć strategię pełnych kopii, kopii różnicowych i kopii dziennika.

Tworzymy więc regularnie pełne kopie bazy danych, kopie dziennika (np. co godzinę) oraz kopie różnicowe (np. codziennie).

W takiem podejściu, w razie awarii, wykorzystamy ostatnią pełną kopię, ostatnią różnicową kopię oraz wszystkie kopie dziennika transakcyjnego, które zostały wykonane po ostatniej kopii różnicowej.

Kopia różnicowa zawierać będzie dane, które należy bezpośrednio przenieść do pełnej kopii, a nastęnie wykonać ponownie wszystkie operacje zawarte w wybranych dziennikach.

Przyjęcie proponowanej strategii zagwarantuje nam, że nie utracimy danych z dłuższego okresu (np. aż do momentu poprzedniej pełnej kopii).