

Paweł Buchajczuk 298743

Kacper Przydatek 298778

5TZT

Pawel.buchajczuk.stud@pw.edu.pl

Kacper.przydatek.stud@pw.edu.pl

BAZY DANYCH I BIG DATA

PROJEKT – CZĘŚĆ 1

Operator telekomunikacyjny

Oświadczamy, że niniejsza praca, stanowiąca podstawę do uznania osiągnięcia efektów uczenia z przedmiotu „Bazy danych i Big Data została wykonana przez nas samodzielnie.

1. Zakres i cel projektu

Celem projektu jest stworzenie relacyjnej bazy danych na poziomie koncepcyjnym oraz logicznym. Następnie należy zaimplementować bazę w sposób fizyczny. Baza zawiera dane na temat operatora telekomunikacyjnego.

Do stworzenia projektu skorzystano z następujących narzędzi:

- TOAD for Oracle
- TOAD Data Modeler
- Oracle Database 19c

i. Założenia funkcjonalne bazy danych

Operator telekomunikacyjny jest pewną firmą, która ma swoje oddziały: biura i hangary techniczne. Do każdego oddziału przynależą pracownicy: monterzy i pracownicy biurowi. Firma zapewnia klientom usługi: Internet 300Mbps, Internet 900Mbps i Internet ADSL 120 20, z których klienci korzystają. Firma jest w relacji z klientami na zasadzie „Firma ma klienta”. Oprócz tego firma ma także punkty komunikacyjne, żeby jakoś usługi móc rozprowadzać. Te, dzielą się na anteny i węzły komutacyjne.

2. Definicja systemu

i. Perspektywy użytkowników

Administrator – posiada dostęp do wszystkich funkcjonalności bazy

Pracownik biurowy – posiada dostęp do wszystkich informacji, nie może modyfikować danych wrażliwych – danych pozostałych pracowników, stanowisk, wypłat, danych firmy

Monter – posiada informacje o klientach, adresach, punktach komunikacyjnych. Może modyfikować wszystkie te zakresy po dokonaniu zmian w sieci

Klient – posiada dostęp do informacji komercyjnych – usług, oraz do informacji na temat operatora – poszczególnych jej oddziałów i głównego członu firmy

Uprawnienia	Administrator	Pracownik biurowy	Monter	Klient
Modyfikacja danych pracowników	+	-	-	-
Podgląd danych pracowników	+	+	-	-
Modyfikacja danych klientów	+	+	+	-
Podgląd danych klientów	+	+	+	-
Modyfikacja usług	+	+	-	-
Podgląd usług	+	+	-	+
Modyfikacja punktów komunikacyjnych	+	+	+	-
Podgląd punktów komunikacyjnych	+	+	+	-
Modyfikacja tablicy adresów	+	+	+	-
Podgląd tablicy adresów	+	+	+	-
Modyfikacja tablicy stanowisk	+	-	-	-
Podgląd tablicy stanowisk	+	+	-	-
Modyfikacja tablicy wynagrodzeń	+	-	-	-
Podgląd tablicy wynagrodzeń	+	+	-	-
Modyfikacja oddziałów	+	+	-	-

Podgląd oddziałów	+	+	-	+
Modyfikacja Firm	+	-	-	-
Podgląd Firm	+	+	-	+

3. Model konceptualny

- i. Definicja zbiorów encji określonych w projekcie (decyzje projektowe)

Firma – główna encja opisująca operatora

Oddział – zawiera informacje na temat oddziałów firmy

Pracownik - zawiera informacje o pracownikach firmy

Klient - zawiera informacje o klientach firmy

Usługa – opisuje usługi oferowane przez operatora

Punkt komunikacyjny - zawiera anteny i węzły komutacyjne

Biuro – opisuje oddział firmy – biuro

Hangar techniczny – opisuje oddział firmy – hangar techniczny

Monter – opisuje pracownika firmy o specjalizacji – monter

Pracownik biurowy – opisuje pracownika firmy o specjalizacji – pracownik biurowy

Antena – zawiera informacje o parametrach anteny

Węzeł komutacyjny – opisuje parametry węzła komutacyjnego

- ii. Ustalenie związków między encjami i ich typów

Relacja	Krotność
Firma – Oddział	1..1 – 0..m
Firma – Klient	1..1 – 0..m
Firma – Punkt Komunikacyjny	1..1 – 0..m
Oddział – Pracownik	0..1 – 0..m
Firma – Usługa	1..1 – 0..m
Firma – Pracownik	1..1 – 0..m
Usługa – Klient	0..n – 0..m
Punkt Komunikacyjny – Usługa	1..n – 0..m

iii. Określenie atrybutów i ich dziedzin

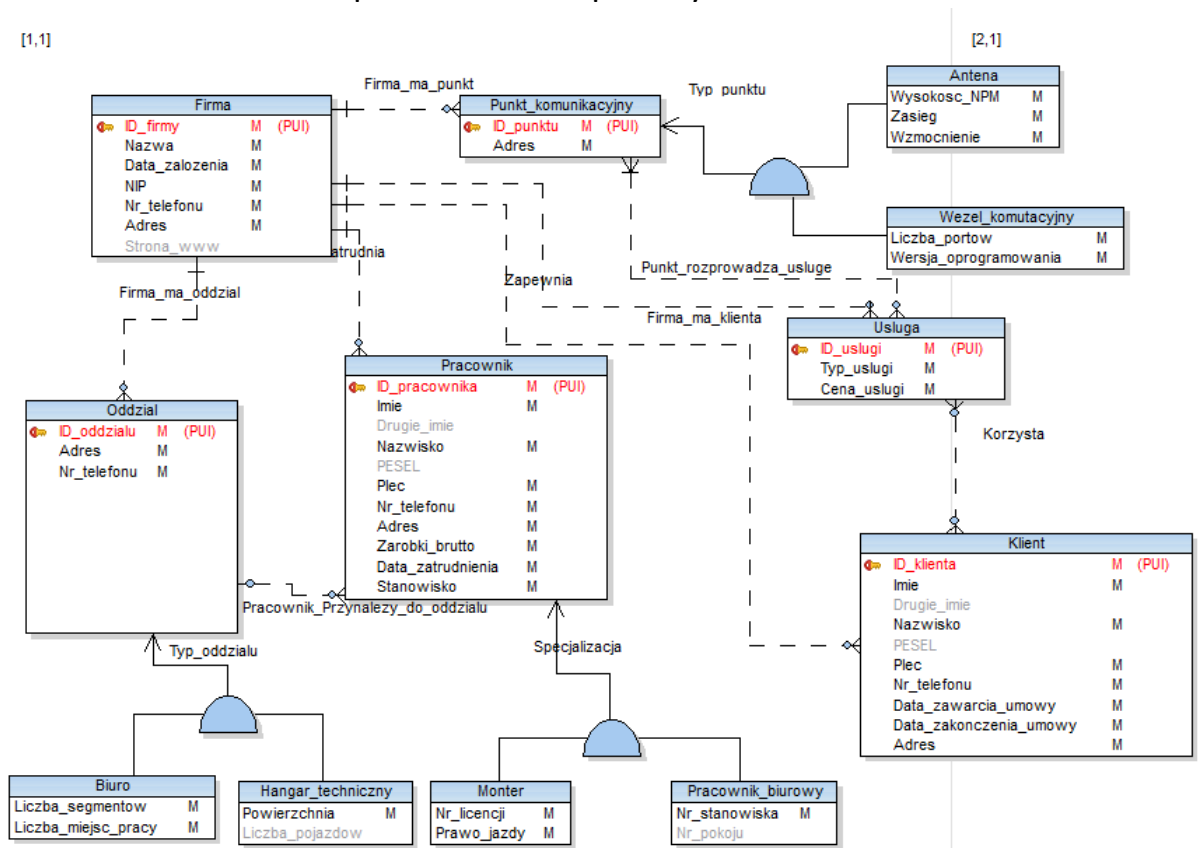
Nazwa atrybutu	Typ i dziedzina	Mandatory?
Firma		
ID firmy	Integer	Tak (klucz główny)
Nazwa	VarChar(50)	Tak
Data zalozenia	Date	Tak
NIP	VarChar(11)	Tak
Nr telefonu	VarChar(9)	Tak
Adres	VarChar(300)	Tak
Strona www	VarChar(300)	Nie
Oddział		
ID oddziału	Integer	Tak (klucz główny)
Adres	VarChar(300)	Tak
Nr telefonu	VarChar(9)	Tak
Biuro		
Liczba segmentow	SmallInt	Tak
Liczba miejsc pracy	Integer	Tak
Hangar techniczny		
Powierzchnia	Integer	Tak
Liczba pojazdow	Integer	Nie
Pracownik		
ID pracownika	Integer	Tak (klucz główny)
Imie	VarChar(20)	Tak
Drugie imie	VarChar(20)	Nie
Nazwisko	VarChar(30)	Tak
PESEL	Character(11)	Nie
Plec	PlecD	Tak
Nr telefonu	VarChar(9)	Tak
Adres	VarChar(300)	Tak
Zarobki brutto	Money	Tak
Data zatrudnienia	Date	Tak
Stanowisko	VarChar(30)	Tak
Monter		
Nr licencji	Integer	Tak
Prawo jazdy	Boolean	Tak
Pracownik biurowy		
Nr stanowiska	Integer	Tak
Nr pietra	VarChar(6)	Nie
Usługa		
ID usługi	Integer	Tak (klucz główny)
Typ usługi	Usługa_telekomunikacyjnaD	Tak
Cena usługi	Money	Tak
Klient		
ID klienta	Integer	Tak (klucz główny)
Imie	VarChar(20)	Tak

Drugie imie	VarChar(20)	Nie
Nazwisko	VarChar(30)	Tak
PESEL	VarChar(11)	Nie
Plec	PlecD	Tak
Nr telefonu	VarChar(9)	Tak
Data zawarcia umowy	Date	Tak
Data zakończenia umowy	Date	Tak
Adres	VarChar(300)	Tak
Punkt komunikacyjny		
ID punktu	Integer	Tak (klucz główny)
Adres	VarChar(300)	Tak
Antena		
Wysokosc NPM	SmallInt	Tak
Zasieg	Integer	Tak
Wzmocnienie	SmallInt	Tak
Węzeł komutacyjny		
Liczba portow	Integer	Tak
Wersja oprogramowania	VarChar(40)	Tak

iv. Dodatkowe reguły integralnościowe (reguły biznesowe)

- a) Hangar techniczny zawsze jest budynkiem własnym Firmy.
- b) Biuro może być zarówno budynkiem firmy, jak i piętrem/oddziałem wynajmowanym w wieżowcu innego właściciela.
- c) Klient może posiadać wiele umów z firmą, z czego ma możliwość posiadania dwóch (i więcej) umów podpisanych na wykup tej samej usługi.
- d) Punkty komunikacyjne (zarówno Anteny jak i Węzły komutacyjne) mogą być własnością Firmy, lub też firma może dzierżawić takie punkty od innego operatora w ramach usługi MVNO.

v. Schemat ER na poziomie konceptualnym



vi. Problem pułapek szczelinowych i wachlarzowych – analiza

Pułapki szczelinowe występują, gdy model sugeruje istnienie związku pomiędzy zbiorami encji, ale nie istnieje ścieżka łącząca pewne wystąpienia tych encji, w przypadku, gdy w modelu znajduje się co najmniej jeden związek o minimalnej krotności zero, który jest elementem ścieżki pomiędzy powiązanymi encjami.

Pułapki wachlarzowe występują w sytuacji, gdy model przedstawia związek pomiędzy pewnymi zbiorami encji, ale wynikające z tego ścieżki pomiędzy wystąpieniami encji nie są jednoznaczne, w przypadku, gdy co najmniej dwa związki typu 1..N wychodzą z tej samej encji.

4. Model logiczny

i. Charakterystyka modelu relacyjnego

Narzędzie Toad Data Modeler udostępnia nam możliwość przekonwertowania modelu konceptualnego w model relacyjny. Przekonwertowanie wywołało zmianę relacji wiele do wielu w dwie relacje jeden do wielu połączone tabelą.

ii. Usunięcie właściwości niekompatybilnych z modelem relacyjnym – przykłady

Zamieniono pole Adres na osobną encję Adresy o polach: Ulica, Miasto, Kod Pocztowy

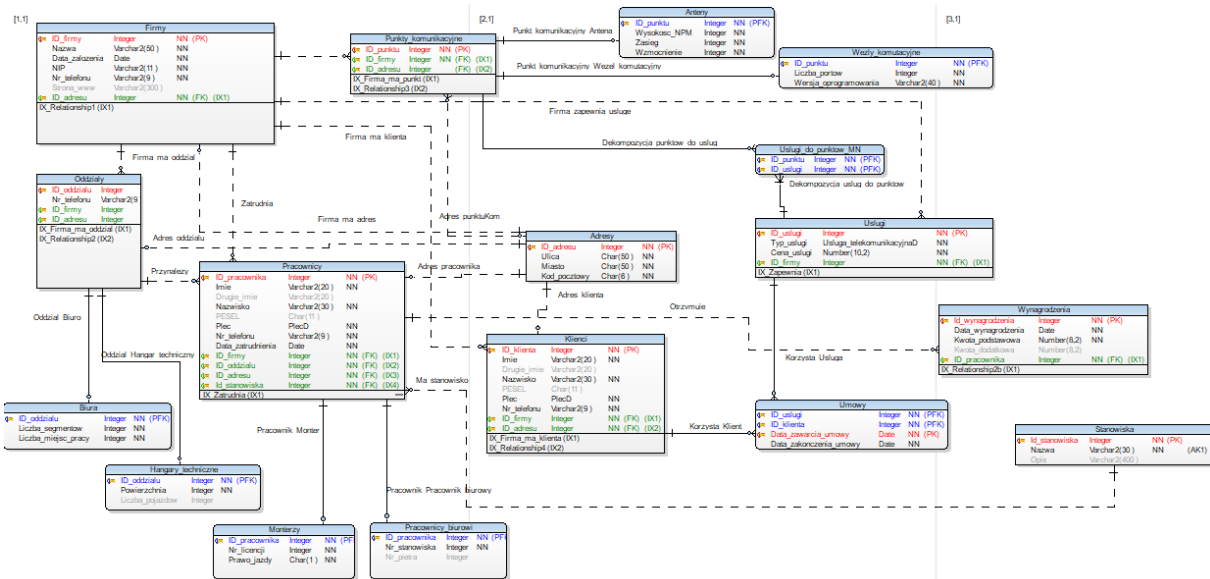
Zostały usunięte związki wiele do wielu. Zamiast nich powstały encje Usługi do punktów pomiędzy encjami Punkty komunikacyjne a Usługi oraz Umowy pomiędzy encjami Klienci a Usługi.

iii. Proces normalizacji – analiza

Proces normalizacji polega na doprowadzeniu relacji do postaci normalnej.

Dla nas pożądana jest trzecia postać normalna, czyli każda relacja musi spełniać pierwszą i drugą postać normalną, a dodatkowo żaden atrybut niekluczowy nie jest zależny funkcyjnie od innych atrybutów niekluczowych.

iv. Schemat ER na poziomie modelu logicznego



v. Więzy integralności

Każdy klucz jest unikatowy, wszystkie pola są atomowe i niewiele kluczy przyjmuje wartość NULL

vi. Proces denormalizacji – analiza

Denormalizacja bazy polega na stworzeniu redundantnych danych. Pozwala to na przyspieszenie wykonywania operacji na bazie.

5. Faza fizyczna

i. Strojenie bazy danych – dobór indeksów

- Wyszukiwanie firm
CREATE INDEX "IX_Relationship1" ON "Firma" ("ID_adresu")
- Wyszukiwanie oddziałów
CREATE INDEX "IX_Firma_ma_oddzial" ON "Oddzialy" ("ID_firmy")
- Wyszukiwanie adresu oddziału
CREATE INDEX "IX_Relationship2" ON "Oddzialy" ("ID_adresu")
- Wyszukiwanie usług
CREATE INDEX "IX_Zapewnia" ON "Uslugi" ("ID_firmy")
- Wyszukiwanie pracowników
CREATE INDEX "IX_Zatrudnia" ON "Pracownicy" ("ID_firmy")
- Wyszukiwanie klientów
CREATE INDEX "IX_Firma_ma_klienta" ON "Klienci" ("ID_firmy")

ii. Skrypt SQL zakładający bazę danych

```
/*
Created: 25.04.2021
Modified: 29.04.2021
Model: Logical model
Database: Oracle 19c
*/

-- Create sequences section -----
-----

CREATE SEQUENCE "FirmaSeq1"
  INCREMENT BY 1
  START WITH 1
  NOMAXVALUE
  NOMINVALUE
  CACHE 20
/

CREATE SEQUENCE "OddzialySeq1"
  INCREMENT BY 1
  START WITH 1
  NOMAXVALUE
  NOMINVALUE
  CACHE 20
/

CREATE SEQUENCE "PracownicySeq1"
  INCREMENT BY 1
  START WITH 1
  NOMAXVALUE
  NOMINVALUE
  CACHE 20
/

CREATE SEQUENCE "PktKomunikacyjneSeq1"
```

```

    INCREMENT BY 1
    START WITH 1
    NOMAXVALUE
    NOMINVALUE
    CACHE 20
/

CREATE SEQUENCE "AdresySeq1"
    INCREMENT BY 1
    START WITH 1
    NOMAXVALUE
    NOMINVALUE
    CACHE 20
/

CREATE SEQUENCE "KlienciSeq1"
    INCREMENT BY 1
    START WITH 1
    NOMAXVALUE
    NOMINVALUE
    CACHE 20
/

CREATE SEQUENCE "UslugiSeq1"
    INCREMENT BY 1
    START WITH 1
    NOMAXVALUE
    NOMINVALUE
    CACHE 20
/

CREATE SEQUENCE "WynagrodzeniaSeq1"
    INCREMENT BY 1
    START WITH 1
    NOMAXVALUE
    NOMINVALUE
    CACHE 20
/

CREATE SEQUENCE "StanowiskaSeq1"
    INCREMENT BY 1
    START WITH 1
    NOMAXVALUE
    NOMINVALUE
    CACHE 20
/

-- Create tables section -----
---

-- Table Firmy

CREATE TABLE "Firmy"(
    "ID_firmy" Integer NOT NULL,
    "Nazwa" Varchar2(50 ) NOT NULL,
    "Data_zalozenia" Date NOT NULL,
    "NIP" Varchar2(13 ) NOT NULL,
    "Nr_telefonu" Varchar2(9 ) NOT NULL,

```

```

        "Strona_www" Varchar2(300 ),
        "ID_adresu" Integer NOT NULL
    )
/

-- Create indexes for table Firmy

CREATE INDEX "IX_Relationship1" ON "Firmy" ("ID_adresu")
/

-- Add keys for table Firmy

ALTER TABLE "Firmy" ADD CONSTRAINT "Firma_PK" PRIMARY KEY ("ID_firmy")
/

-- Table Oddzialy

CREATE TABLE "Oddzialy"(
    "ID_oddzialu" Integer NOT NULL,
    "Nr_telefonu" Varchar2(9 ) NOT NULL,
    "ID_firmy" Integer NOT NULL,
    "ID_adresu" Integer NOT NULL
)
/

-- Create indexes for table Oddzialy

CREATE INDEX "IX_Firma_ma_oddzial" ON "Oddzialy" ("ID_firmy")
/

CREATE INDEX "IX_Relationship2" ON "Oddzialy" ("ID_adresu")
/

-- Add keys for table Oddzialy

ALTER TABLE "Oddzialy" ADD CONSTRAINT "Oddzial_PK" PRIMARY KEY
("ID_oddzialu")
/

-- Table Biura

CREATE TABLE "Biura"(
    "ID_oddzialu" Integer NOT NULL,
    "Liczba_segmentow" Integer NOT NULL,
    "Liczba_miejsc_pracy" Integer NOT NULL
)
/

-- Add keys for table Biura

ALTER TABLE "Biura" ADD CONSTRAINT "Unique_Identifier1" PRIMARY KEY
("ID_oddzialu")
/

-- Table Hangary_techiczne

CREATE TABLE "Hangary_techiczne"(
    "ID_oddzialu" Integer NOT NULL,

```

```

        "Powierzchnia" Integer NOT NULL,
        "Liczba_pojazdow" Integer
    )
/

-- Add keys for table Hangary_techiczne

ALTER TABLE "Hangary_techiczne" ADD CONSTRAINT "Unique_Identifier2"
PRIMARY KEY ("ID_oddzialu")
/

-- Table Usługi

CREATE TABLE "Usługi"(
    "ID_uslugi" Integer NOT NULL,
    "Typ_uslugi" Varchar2(300 ) NOT NULL
        CHECK ("Typ_uslugi" IN ('Internet_300Mbps', 'Internet_900Mbps',
'Internet_ADSL_120_20')),
    "Cena_uslugi" Number(10,2) NOT NULL,
    "ID_firmy" Integer NOT NULL
)
/

-- Create indexes for table Usługi

CREATE INDEX "IX_Zapewnia" ON "Usługi" ("ID_firmy")
/

-- Add keys for table Usługi

ALTER TABLE "Usługi" ADD CONSTRAINT "Usługa_PK" PRIMARY KEY
("ID_uslugi")
/

-- Table Pracownicy

CREATE TABLE "Pracownicy"(
    "ID_pracownika" Integer NOT NULL,
    "Imie" Varchar2(20 ) NOT NULL,
    "Drugie_imie" Varchar2(20 ),
    "Nazwisko" Varchar2(30 ) NOT NULL,
    "PESEL" Char(11 ),
    "Plec" Varchar2(30 ) NOT NULL
        CONSTRAINT "CheckConstraintA1" CHECK ("Plec" IN ('Kobieta',
'Mezczyzna')),
        CHECK ("Plec" IN ('Kobieta', 'Mezczyzna')),
    "Nr_telefonu" Varchar2(9 ) NOT NULL,
    "Data_zatrudnienia" Date NOT NULL,
    "ID_firmy" Integer NOT NULL,
    "ID_oddzialu" Integer NOT NULL,
    "ID_adresu" Integer NOT NULL,
    "ID_stanowiska" Integer NOT NULL
)
/

-- Create indexes for table Pracownicy

CREATE INDEX "IX_Zatrudnia" ON "Pracownicy" ("ID_firmy")

```

```

/

CREATE INDEX "IX_Przynalezy" ON "Pracownicy" ("ID_oddzialu")
/

CREATE INDEX "IX_Relationship1a" ON "Pracownicy" ("ID_adresu")
/

CREATE INDEX "IX_Relationship3c" ON "Pracownicy" ("Id_stanowiska")
/

-- Add keys for table Pracownicy

ALTER TABLE "Pracownicy" ADD CONSTRAINT "Pracownik_PK" PRIMARY KEY
("ID_pracownika")
/

-- Table Monterzy

CREATE TABLE "Monterzy"(
    "ID_pracownika" Integer NOT NULL,
    "Nr_licencji" Integer NOT NULL,
    "Prawo_jazdy" Char(1 ) NOT NULL
)
/

-- Add keys for table Monterzy

ALTER TABLE "Monterzy" ADD CONSTRAINT "Unique_Identifier5" PRIMARY KEY
("ID_pracownika")
/

-- Table Pracownicy_biurowi

CREATE TABLE "Pracownicy_biurowi"(
    "ID_pracownika" Integer NOT NULL,
    "Nr_stanowiska" Integer NOT NULL,
    "Nr_pietra" Integer
)
/

-- Add keys for table Pracownicy_biurowi

ALTER TABLE "Pracownicy_biurowi" ADD CONSTRAINT "Unique_Identifier6"
PRIMARY KEY ("ID_pracownika")
/

-- Table Klienci

CREATE TABLE "Klienci"(
    "ID_klienta" Integer NOT NULL,
    "Imie" Varchar2(20 ) NOT NULL,
    "Drugie_imie" Varchar2(20 ),
    "Nazwisko" Varchar2(30 ) NOT NULL,
    "PESEL" Char(11 ),
    "Plec" Varchar2(30 ) NOT NULL
        CHECK ("Plec" IN ('Kobieta', 'Mezczyzna')),
    "Nr_telefonu" Varchar2(9 ) NOT NULL,

```

```

        "ID_firmy" Integer NOT NULL,
        "ID_adresu" Integer NOT NULL
    )
/

-- Create indexes for table Klienci

CREATE INDEX "IX_Firma_ma_klienta" ON "Klienci" ("ID_firmy")
/

CREATE INDEX "IX_Relationship4" ON "Klienci" ("ID_adresu")
/

-- Add keys for table Klienci

ALTER TABLE "Klienci" ADD CONSTRAINT "Klient_PK" PRIMARY KEY
("ID_klienta")
/

-- Table Punkty_komunikacyjne

CREATE TABLE "Punkty_komunikacyjne"(
    "ID_punktu" Integer NOT NULL,
    "ID_firmy" Integer NOT NULL,
    "ID_adresu" Integer
)
/

-- Create indexes for table Punkty_komunikacyjne

CREATE INDEX "IX_Firma_ma_punkt" ON "Punkty_komunikacyjne" ("ID_firmy")
/

CREATE INDEX "IX_Relationship3" ON "Punkty_komunikacyjne" ("ID_adresu")
/

-- Add keys for table Punkty_komunikacyjne

ALTER TABLE "Punkty_komunikacyjne" ADD CONSTRAINT
"Punkt_komunikacyjny_PK" PRIMARY KEY ("ID_punktu")
/

-- Table Anteny

CREATE TABLE "Anteny"(
    "ID_punktu" Integer NOT NULL,
    "Wysokosc_NPM" Integer NOT NULL,
    "Zasieg" Integer NOT NULL,
    "Wzmocnienie" Integer NOT NULL
)
/

-- Add keys for table Anteny

ALTER TABLE "Anteny" ADD CONSTRAINT "Unique_Identifier7" PRIMARY KEY
("ID_punktu")
/

```

```

-- Table Wezly_komutacyjne

CREATE TABLE "Wezly_komutacyjne"(
  "ID_punktu" Integer NOT NULL,
  "Liczba_portow" Integer NOT NULL,
  "Wersja_oprogramowania" Varchar2(40 ) NOT NULL
)
/

-- Add keys for table Wezly_komutacyjne

ALTER TABLE "Wezly_komutacyjne" ADD CONSTRAINT "Unique_Identifier8"
PRIMARY KEY ("ID_punktu")
/

-- Table Umowy

CREATE TABLE "Umowy"(
  "ID_uslugi" Integer NOT NULL,
  "ID_klienta" Integer NOT NULL,
  "Data_zawarcia_umowy" Date NOT NULL,
  "Data_zakonczenia_umowy" Date NOT NULL
)
/

-- Table Adresy

CREATE TABLE "Adresy"(
  "ID_adresu" Integer NOT NULL,
  "Ulica" Char(50 ) NOT NULL,
  "Miasto" Char(50 ) NOT NULL,
  "Kod_pocztowy" Char(6 ) NOT NULL
)
/

-- Add keys for table Adresy

ALTER TABLE "Adresy" ADD CONSTRAINT "PK_Adresy" PRIMARY KEY
("ID_adresu")
/

-- Table Uslugi_do_punktow_MN

CREATE TABLE "Uslugi_do_punktow_MN"(
  "ID_punktu" Integer NOT NULL,
  "ID_uslugi" Integer NOT NULL
)
/

-- Add keys for table Uslugi_do_punktow_MN

ALTER TABLE "Uslugi_do_punktow_MN" ADD CONSTRAINT
"PK_Uslugi_do_punktow_MN" PRIMARY KEY ("ID_punktu","ID_uslugi")
/

-- Table Wynagrodzenia

CREATE TABLE "Wynagrodzenia"(

```



```

        "Id_wynagrodzenia" Integer NOT NULL,
        "Data_wynagrodzenia" Date NOT NULL,
        "Kwota_podstawowa" Number(8,2) NOT NULL,
        "Kwota_dodatkowa" Number(8,2),
        "ID_pracownika" Integer NOT NULL
    )
/

-- Create indexes for table Wynagrodzenia

CREATE INDEX "IX_Relationship2b" ON "Wynagrodzenia" ("ID_pracownika")
/

-- Add keys for table Wynagrodzenia

ALTER TABLE "Wynagrodzenia" ADD CONSTRAINT "PK_Wynagrodzenia" PRIMARY
KEY ("Id_wynagrodzenia")
/

-- Table Stanowiska

CREATE TABLE "Stanowiska"(
    "Id_stanowiska" Integer NOT NULL,
    "Nazwa" Varchar2(30 ) NOT NULL,
    "Opis" Varchar2(400 )
)
/

-- Add keys for table Stanowiska

ALTER TABLE "Stanowiska" ADD CONSTRAINT "PK_Stalowiska" PRIMARY KEY
("Id_stanowiska")
/

ALTER TABLE "Stanowiska" ADD CONSTRAINT "Nazwa" UNIQUE ("Nazwa")
/

-- Trigger for sequence FirmySeq1 for column ID_firmy in table Firmy --
-----
CREATE OR REPLACE TRIGGER "ts_Firmy_FirmySeq1" BEFORE INSERT
ON "Firmy" FOR EACH ROW
BEGIN
    :new."ID_firmy" := "FirmySeq1".nextval;
END;
/

CREATE OR REPLACE TRIGGER "tsu_Firmy_FirmySeq1" AFTER UPDATE OF
"ID_firmy"
ON "Firmy" FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010,'Cannot update column "ID_firmy" in
table "Firmy" as it uses sequence.');
```

```

END;
/

-- Trigger for sequence OddzialySeq1 for column ID_oddzialu in table
Oddzialy -----
CREATE OR REPLACE TRIGGER "ts_Oddzialy_OddzialySeq1" BEFORE INSERT
ON "Oddzialy" FOR EACH ROW
```

```

BEGIN
    :new."ID_oddzialu" := "OddzialySeq1".nextval;
END;
/
CREATE OR REPLACE TRIGGER "tsu_Oddzialy_OddzialySeq1" AFTER UPDATE OF
"ID_oddzialu"
ON "Oddzialy" FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010,'Cannot update column "ID_oddzialu" in
table "Oddzialy" as it uses sequence.');
```

```

END;
/

-- Trigger for sequence UslugiSeq1 for column ID_uslugi in table Uslugi
-----
CREATE OR REPLACE TRIGGER "ts_Uslugi_UslugiSeq1" BEFORE INSERT
ON "Uslugi" FOR EACH ROW
BEGIN
    :new."ID_uslugi" := "UslugiSeq1".nextval;
END;
/
CREATE OR REPLACE TRIGGER "tsu_Uslugi_UslugiSeq1" AFTER UPDATE OF
"ID_uslugi"
ON "Uslugi" FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010,'Cannot update column "ID_uslugi" in
table "Uslugi" as it uses sequence.');
```

```

END;
/

-- Trigger for sequence PracownicySeq1 for column ID_pracownika in
table Pracownicy -----
CREATE OR REPLACE TRIGGER "ts_Pracownicy_PracownicySeq1" BEFORE INSERT
ON "Pracownicy" FOR EACH ROW
BEGIN
    :new."ID_pracownika" := "PracownicySeq1".nextval;
END;
/
CREATE OR REPLACE TRIGGER "tsu_Pracownicy_PracownicySeq1" AFTER UPDATE
OF "ID_pracownika"
ON "Pracownicy" FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010,'Cannot update column "ID_pracownika"
in table "Pracownicy" as it uses sequence.');
```

```

END;
/

-- Trigger for sequence KlienciSeq1 for column ID_klienta in table
Klienci -----
CREATE OR REPLACE TRIGGER "ts_Klienci_KlienciSeq1" BEFORE INSERT
ON "Klienci" FOR EACH ROW
BEGIN
    :new."ID_klienta" := "KlienciSeq1".nextval;
END;
/
CREATE OR REPLACE TRIGGER "tsu_Klienci_KlienciSeq1" AFTER UPDATE OF
"ID_klienta"
ON "Klienci" FOR EACH ROW
```

```

BEGIN
    RAISE_APPLICATION_ERROR(-20010,'Cannot update column "ID_klienta" in
table "Klienci" as it uses sequence.');
```

END;

/

```

-- Trigger for sequence PktKomunikacyjneSeq1 for column ID_punktu in
table Punkty_komunikacyjne -----
CREATE OR REPLACE TRIGGER
"ts_Punkty_komunikacyjne_PktKomunikacyjneSeq1" BEFORE INSERT
ON "Punkty_komunikacyjne" FOR EACH ROW
BEGIN
    :new."ID_punktu" := "PktKomunikacyjneSeq1".nextval;
END;
```

/

```

CREATE OR REPLACE TRIGGER
"tsu_Punkty_komunikacyjne_PktKomunikacyjneSeq1" AFTER UPDATE OF
"ID_punktu"
ON "Punkty_komunikacyjne" FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010,'Cannot update column "ID_punktu" in
table "Punkty_komunikacyjne" as it uses sequence.');
```

END;

/

```

-- Trigger for sequence AdresySeq1 for column ID_adresu in table Adresy
-----
CREATE OR REPLACE TRIGGER "ts_Adresy_AdresySeq1" BEFORE INSERT
ON "Adresy" FOR EACH ROW
BEGIN
    :new."ID_adresu" := "AdresySeq1".nextval;
END;
```

/

```

CREATE OR REPLACE TRIGGER "tsu_Adresy_AdresySeq1" AFTER UPDATE OF
"ID_adresu"
ON "Adresy" FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010,'Cannot update column "ID_adresu" in
table "Adresy" as it uses sequence.');
```

END;

/

```

-- Trigger for sequence WynagrodzeniaSeq1 for column Id_wynagrodzenia
in table Wynagrodzenia -----
CREATE OR REPLACE TRIGGER "ts_Wynagrodzenia_WynagrodzeniaSeq1" BEFORE
INSERT
ON "Wynagrodzenia" FOR EACH ROW
BEGIN
    :new."Id_wynagrodzenia" := "WynagrodzeniaSeq1".nextval;
END;
```

/

```

CREATE OR REPLACE TRIGGER "tsu_Wynagrodzenia_WynagrodzeniaSeq1" AFTER
UPDATE OF "Id_wynagrodzenia"
ON "Wynagrodzenia" FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010,'Cannot update column
"Id_wynagrodzenia" in table "Wynagrodzenia" as it uses sequence.');
```

END;

```

/

-- Trigger for sequence StanowiskaSeq1 for column Id_stanowiska in
table Stanowiska -----
CREATE OR REPLACE TRIGGER "ts_Stanowiska_StanowiskaSeq1" BEFORE INSERT
ON "Stanowiska" FOR EACH ROW
BEGIN
    :new."Id_stanowiska" := "StanowiskaSeq1".nextval;
END;
/

CREATE OR REPLACE TRIGGER "tsu_Stanowiska_StanowiskaSeq1" AFTER UPDATE
OF "Id_stanowiska"
ON "Stanowiska" FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010,'Cannot update column "Id_stanowiska"
in table "Stanowiska" as it uses sequence.');
```

```

-- Create foreign keys (relationships) section -----
-----

ALTER TABLE "Oddzialy" ADD CONSTRAINT "Firma_ma_oddzial" FOREIGN KEY
("ID_firmy") REFERENCES "Firma" ("ID_firmy")
/

ALTER TABLE "Pracownicy" ADD CONSTRAINT "Zatrudnia" FOREIGN KEY
("ID_firmy") REFERENCES "Firma" ("ID_firmy")
/

ALTER TABLE "Punkty_komunikacyjne" ADD CONSTRAINT "Firma_ma_punkt"
FOREIGN KEY ("ID_firmy") REFERENCES "Firma" ("ID_firmy")
/

ALTER TABLE "Uslugi" ADD CONSTRAINT "Firma_zapewnia_usluge" FOREIGN KEY
("ID_firmy") REFERENCES "Firma" ("ID_firmy")
/

ALTER TABLE "Klienci" ADD CONSTRAINT "Firma_ma_klienta" FOREIGN KEY
("ID_firmy") REFERENCES "Firma" ("ID_firmy")
/

ALTER TABLE "Pracownicy" ADD CONSTRAINT "Przynalezy" FOREIGN KEY
("ID_oddzialu") REFERENCES "Oddzialy" ("ID_oddzialu")
/
```

```
ALTER TABLE "Pracownicy" ADD CONSTRAINT "Adres_pracownika" FOREIGN KEY  
("ID_adresu") REFERENCES "Adresy" ("ID_adresu")  
/
```

```
ALTER TABLE "Oddzialy" ADD CONSTRAINT "Adres_oddzialu" FOREIGN KEY  
("ID_adresu") REFERENCES "Adresy" ("ID_adresu")  
/
```

```
ALTER TABLE "Punkty_komunikacyjne" ADD CONSTRAINT "Adres_punktuKom"  
FOREIGN KEY ("ID_adresu") REFERENCES "Adresy" ("ID_adresu")  
/
```

```
ALTER TABLE "Klienci" ADD CONSTRAINT "Adres_klienta" FOREIGN KEY  
("ID_adresu") REFERENCES "Adresy" ("ID_adresu")  
/
```

```
ALTER TABLE "Uslugi_do_punktow_MN" ADD CONSTRAINT  
"Dekompozycja_punktow_do_uslug" FOREIGN KEY ("ID_punktu") REFERENCES  
"Punkty_komunikacyjne" ("ID_punktu")  
/
```

```
ALTER TABLE "Uslugi_do_punktow_MN" ADD CONSTRAINT  
"Dekompozycja_uslug_do_punktow" FOREIGN KEY ("ID_uslugi") REFERENCES  
"Uslugi" ("ID_uslugi")  
/
```

```
ALTER TABLE "Firmy" ADD CONSTRAINT "Firma_ma_adres" FOREIGN KEY  
("ID_adresu") REFERENCES "Adresy" ("ID_adresu")  
/
```

```
ALTER TABLE "Wynagrodzenia" ADD CONSTRAINT "Otrzymuje" FOREIGN KEY  
("ID_pracownika") REFERENCES "Pracownicy" ("ID_pracownika")  
/
```

```
ALTER TABLE "Pracownicy" ADD CONSTRAINT "Ma_stanowisko" FOREIGN KEY  
("Id_stanowiska") REFERENCES "Stanowiska" ("Id_stanowiska")  
/
```

iii. Przykłady zapytań i poleceń SQL odnoszących się do bazy danych

- SELECT * FROM "Adresy"

ID_adresu	Ulica	Miasto	Kod_pocztowy
1	pl. Politechniki 1	Warszawa	00-661
22	Nowowiejska 15/19	Warszawa	00-665
23	Wilanowska 27	Warszawa	02-765
25	Kraszewskiego 13	Pruszków	05-803
28	Żelazna 15	Katowice	40-851
30	3 maja 1/3	Łódź	90-001

- SELECT "Nr_telefonu" FROM "Oddzialy"

Nr_telefonu
122334455
233445566