

Narodowa baza danych Covid-19

Michał Kuliński

Jakub Małkowski

1 Podstawowe założenia projektu

1.1 Cel i główne założenia projektu

Celem bazy danych jest umożliwienie przechowywania kluczowych i najważniejszych informacji dotyczących ludzi dotkniętych wirusem oraz systemu z tym związanym. W bazie danych przechowywane są informacje o kwarantannach, szczepieniach, ludziach aktualnie hospitalizowanych oraz przeprowadzanych testach.

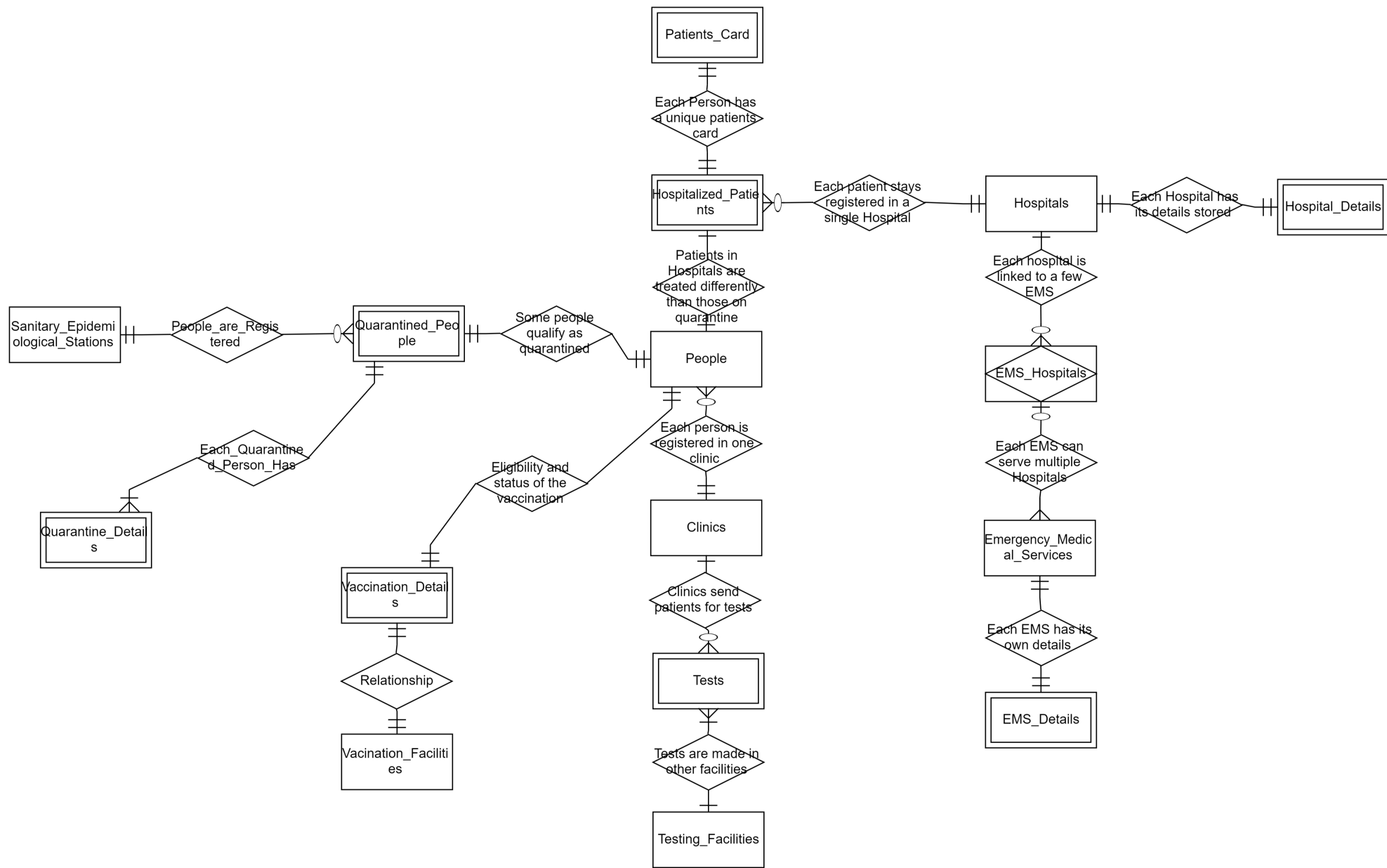
Ponadto w bazie danych znajdują się informacje o przychodniach, szpitalach i pogotowiu ratunkowym. Przechowywane są również informacje o stacjach sanitarno-epidemiologicznych, punktach szczepień i punktach testowania.

1.2 Możliwości

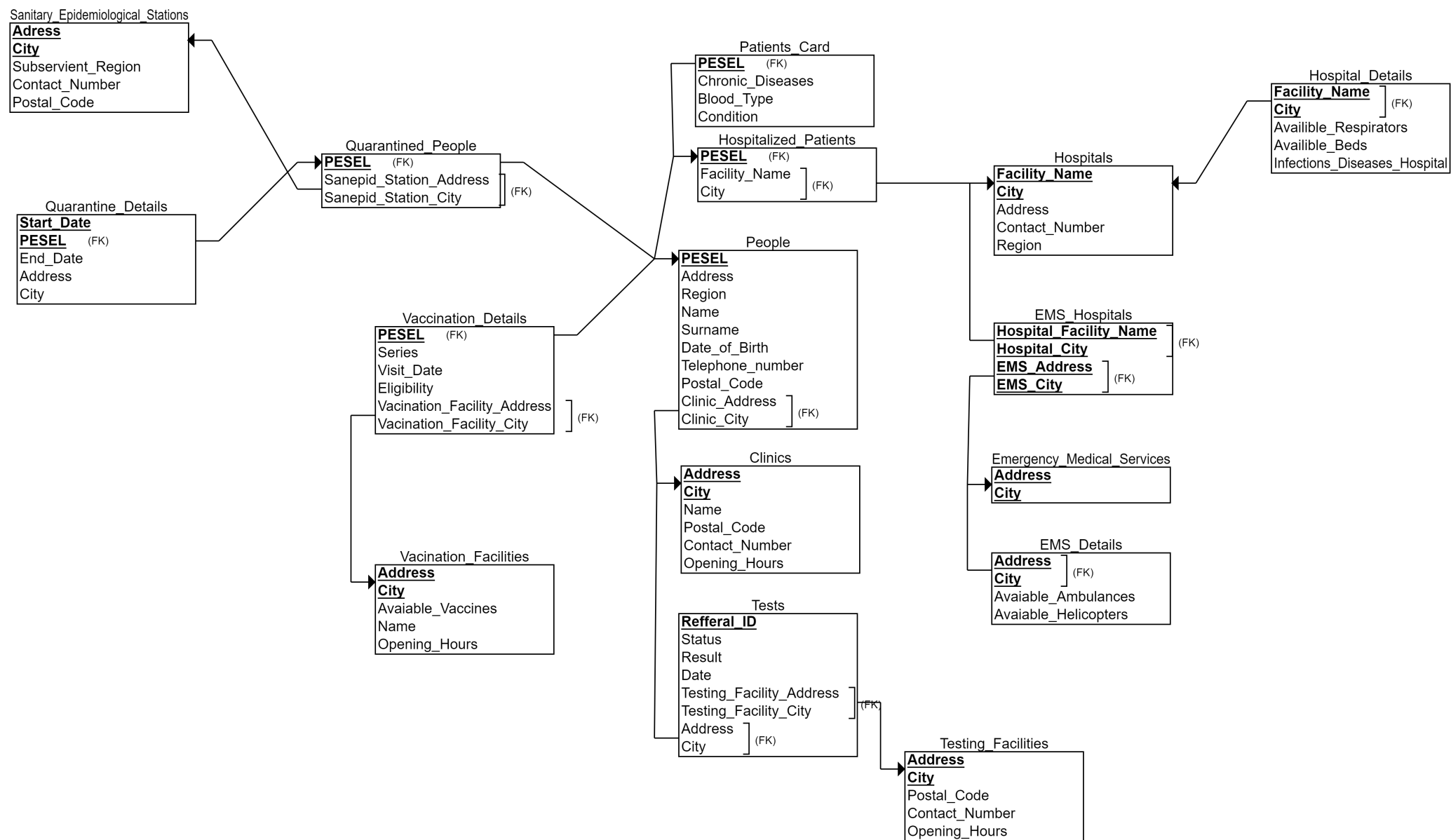
1.2.1 Ludzie

Tabela "People" jest centralnym punktem bazy danych, do którego można dodawać ludzi za pomocą procedury składowanej "AddPerson", która jednocześnie przypisuje im przychodnię oraz punkt szczepień.

2 Diagram ER



3 Schemat bazy danych



4 Widoki

4.1 Vaccination Eligibility

Widok przedstawiający informacje na temat kwalifikowalności do szczepienia.

```
01 | DROP VIEW IF EXISTS dbo.Eligibility;
02 | GO
03 |
04 | CREATE VIEW Eligibility AS
05 |     SELECT Name, Surname, V.Eligibility FROM People P
06 |     INNER JOIN Vaccination_Details V ON V.PESEL = P.PESEL;
07 | GO
```

4.2 Hospital-EMS connections

Widok przedstawiający informacje na temat współpracujących szpitali i pogotowia ratunkowego.

```
01 | DROP VIEW IF EXISTS dbo.Hospital_EMS_Connections;
02 | GO
03 |
04 | CREATE VIEW Hospital_EMS_Connections
05 | AS
06 |     SELECT H.Facility_Name, H.City, E.EMS_Address AS EMS_Address FROM Hospitals H
07 |     INNER JOIN EMS_Hospitals E ON E.Hospital_Facility_Name = H.Facility_Name AND E.Hospital_City = H.City
08 |     INNER JOIN EMS_Details D ON D.Address = E.EMS_Address AND D.City = E.EMS_City
09 | GO
```

4.3 Hospitals in region

Widok przedstawiający informacje na temat szpitali w regionie.

```
01 | DROP VIEW IF EXISTS dbo.HospitalsInRegion;
02 | GO
03 |
04 | CREATE VIEW HospitalsInRegion AS
05 |     SELECT * FROM Hospitals H ORDER BY Region DESC;
06 | GO
```

4.4 Infections per region

Widok przedstawiający liczbę zakażeń na każdy region.

```
01 | DROP VIEW IF EXISTS dbo.Infections_Per_Region;  
02 | GO  
03 |  
04 | CREATE VIEW Infections_Per_Region AS  
05 |     SELECT DISTINCT P.Region, (SELECT DISTINCT COUNT(T.PESEL) FROM Tests T  
06 |         GROUP BY T.Result HAVING T.Result='Positive') Infections FROM People P;  
07 | GO
```

4.5 Patients in hospital

Widok przedstawiający liczbę pacjentów na każdy szpital.

```
01 | DROP VIEW IF EXISTS dbo.Patients_in_Hospital;  
02 | GO  
03 |  
04 | CREATE VIEW Patients_in_Hospital AS  
05 |     SELECT H.Facility_Name, H.Address, (SELECT Count(*) FROM Hospitalized_Patients HP  
06 |         GROUP BY HP.Facility_Name  
07 |         HAVING HP.Facility_Name = H.Facility_Name) AS Patients_Per_Hospital FROM Hospitals H;  
08 | GO
```

5 Procedury składowane

5.1 Add Clinic

Procedura pozwalająca dodać nową przychodnię do tabeli *Clinics*.

```
01 | DROP PROC IF EXISTS dbo.AddClinic;
02 | GO
03 |
04 | CREATE PROC dbo.AddClinic
05 |
06 | @Address VARCHAR(50),
07 | @City VARCHAR(30),
08 | @Name VARCHAR(50),
09 | @Postal_Code VARCHAR(6),
10 | @Contact_Number VARCHAR(20)
11 |
12 | AS
13 |
14 | INSERT INTO Clinics(Address, City, Name, Postal_Code, Contact_Number)
15 | VALUES(@Address, @City, @Name, @Postal_Code, @Contact_Number)
16 |
17 | GO
```

5.2 Add Person

Procedura pozwalająca dodać nową osobę do tabeli *People*. Dodatkowo zostaje jej przypisany aktualny status dotyczący szczepienia w tabeli *Vaccination Details*.

```
01 | DROP PROC IF EXISTS dbo.AddPerson;
02 | GO
03 |
04 | CREATE PROC dbo.AddPerson
05 |
06 |     @Address VARCHAR(50),
07 |     @PESEL VARCHAR(11),
08 |     @Region VARCHAR(30),
09 |     @Name VARCHAR(30),
10 |     @Surname VARCHAR(30),
11 |     @Date_of_Birth DATE,
12 |     @Telephone_number VARCHAR(20),
13 |     @Postal_Code VARCHAR(6),
14 |     @Clinic_Address VARCHAR(50),
15 |     @Clinic_City VARCHAR(30),
16 |     @Vaccination_Facility_Address VARCHAR(50),
17 |     @Vaccination_Facility_City VARCHAR(30)
18 |
19 | AS
20 |
21 | INSERT INTO People(Address, PESEL, Region, Name, Surname, Date_of_Birth, Telephone_number, Postal_Code, Clinic_Address, Clinic_City)
22 | VALUES(@Address, @PESEL, @Region, @Name, @Surname, @Date_of_Birth, @Telephone_number, @Postal_Code, @Clinic_Address, @Clinic_City)
23 |
24 | INSERT INTO Vaccination_Details(PESEL, Series, Visit_Date, Eligibility, Vaccination_Facility_Address, Vaccination_Facility_City)
25 | VALUES(@PESEL, NULL, NULL, 0, @Vaccination_Facility_Address, @Vaccination_Facility_City)
26 |
27 | GO
```

5.3 Add Sanepid

Procedura pozwalająca dodać nowy punkt sanitarno-epidemiologiczny do tabeli *Sanitary Epidemiological Stations*.

```
01 | DROP PROC IF EXISTS dbo.AddSanepid
02 | GO
03 |
04 | CREATE PROC dbo.AddSanepid
05 |
06 | @Address VARCHAR(50),
07 | @Subservient_Region VARCHAR(30),
08 | @Contact_Number VARCHAR(20),
09 | @City VARCHAR(30),
10 | @Postal_Code VARCHAR(6)
11 |
12 | AS
13 |
14 | INSERT Sanitary_Epidemiological_Stations(Address, Subservient_Region, Contact_Number, City, Postal_Code)
15 | VALUES(@Address, @Subservient_Region, @Contact_Number, @City, @Postal_Code)
16 |
17 | GO
```

5.4 Add Testing Facility

Procedura pozwalająca dodać nowy punkt przeprowadzania testów do tabeli *Testing Facilities*.

```
01 | DROP PROC IF EXISTS dbo.AddTestingFacility
02 | GO
03 |
04 | CREATE PROC dbo.AddTestingFacility
05 |
06 | @Address VARCHAR(50),
07 | @City VARCHAR(30),
08 | @Postal_Code VARCHAR(6),
09 | @Contact_Number INT
10 |
11 | AS
12 |
13 | INSERT INTO Testing_Facilities(Address, City, Postal_Code, Contact_Number)
14 | VALUES(@Address, @City, @Postal_Code, @Contact_Number)
15 |
16 | GO
```


5.5 Add Vaccination Facility

Procedura pozwalająca dodać nowy punkt przeprowadzania szczepień do tabeli *Vaccination Facilities*.

```
01 | DROP PROC IF EXISTS dbo.AddVaccinationFacility;  
02 | GO  
03 |  
04 | CREATE PROC dbo.AddVaccinationFacility  
05 |  
06 | @Available_Vaccines INT,  
07 | @Address VARCHAR(50),  
08 | @City VARCHAR(30),  
09 | @Name VARCHAR(50)  
10 |  
11 | AS  
12 |  
13 | INSERT INTO Vaccination_Facilities(Available_Vaccines, Address, City, Name)  
14 | VALUES(@Available_Vaccines, @Address, @City, @Name)  
15 |  
16 | GO
```

5.6 Make Eligible For Vaccination

Procedura pozwalająca na zmianę kwalifikowalności do szczepienia poprzez podanie numeru PESEL w tabeli *Vaccination Details*.

```
01 | DROP PROC IF EXISTS dbo.MakeEligibleForVaccination  
02 | GO  
03 |  
04 | CREATE PROC dbo.MakeEligibleForVaccination  
05 |  
06 | @PESEL VARCHAR(11)  
07 |  
08 | AS  
09 |  
10 | UPDATE Vaccination_Details  
11 | SET Eligibility = 1  
12 | WHERE PESEL = @PESEL  
13 |  
14 | GO
```

5.7 Quarantine

Procedura pozwalająca na poddanie osoby kwarantannie poprzez dodanie osoby do tabeli *Quarantined People* oraz dodanie szczegółów dotyczących kwarantanny do tabeli *Quarantine Details*.

```
01 | DROP PROC IF EXISTS dbo.Quarantine
02 | GO
03 |
04 | CREATE PROC dbo.Quarantine
05 |
06 | @Start_Date DATE,
07 | @End_Date DATE,
08 | @Address VARCHAR(50),
09 | @City VARCHAR(30),
10 | @PESEL VARCHAR(11),
11 | @Sanepid_Station_Address VARCHAR(50),
12 | @Sanepid_Station_City VARCHAR(30)
13 |
14 | AS
15 |
16 | INSERT Quarantined_People(PESEL, Sanepid_Station_Address, Sanepid_Station_City)
17 | VALUES(@PESEL, @Sanepid_Station_Address, @Sanepid_Station_City)
18 |
19 | INSERT Quarantine_Details(Start_Date, End_Date, Address, City, PESEL)
20 | VALUES(@Start_Date, @End_Date, @Address, @City, @PESEL)
21 |
22 | GO
```

5.8 Add Patient Card

Procedura pozwalająca na dodanie karty pacjenta do tabeli *Patient Cards*.

```
01 | DROP PROC IF EXISTS dbo.AddPatientCard;  
02 | GO  
03 |  
04 | CREATE PROC dbo.AddPatientCard  
05 |  
06 | @Chronic_Diseases VARCHAR(100),  
07 | @Blood_Type VARCHAR(10),  
08 | @Condition VARCHAR(30),  
09 | @PESEL VARCHAR(11)  
10 |  
11 | AS  
12 |  
13 | INSERT INTO Patients_Card(Chronic_Diseases, Blood_Type, Condition, PESEL)  
14 | VALUES(@Chronic_Diseases, @Blood_Type, @Condition, @PESEL)  
15 |  
16 | GO
```

5.9 Add Patient

Procedura pozwalająca na dodanie pacjenta do tabeli *Patients*.

```
01 | DROP PROC IF EXISTS dbo.AddPatient;  
02 | GO  
03 |  
04 | CREATE PROC dbo.AddPatient  
05 |  
06 | @PESEL VARCHAR(11)  
07 |  
08 | AS  
09 |  
10 | INSERT INTO dbo.Hospitalized_Patients(PESEL, Facility_Name, City)  
11 | SELECT TOP 1 @PESEL, H.Facility_Name, H.City FROM dbo.AVAILABLE_HOSPITALS(@PESEL) H  
12 | WHERE Available_Beds > 0  
13 | ORDER BY NEWID()  
14 | GO
```

5.10 Discharge Patient

Procedura pozwalająca na wypis pacjenta i usunięcie go z tabeli *Hospitalized Patients*.

```
01 | DROP PROC IF EXISTS dbo.DischargePatient;  
02 | GO  
03 |  
04 | CREATE PROC dbo.DischargePatient  
05 |  
06 | @PESEL VARCHAR(11)  
07 |  
08 | AS  
09 |  
10 | BEGIN TRANSACTION  
11 |     DELETE FROM Hospitalized_Patients WHERE PESEL = @PESEL  
12 | COMMIT;
```