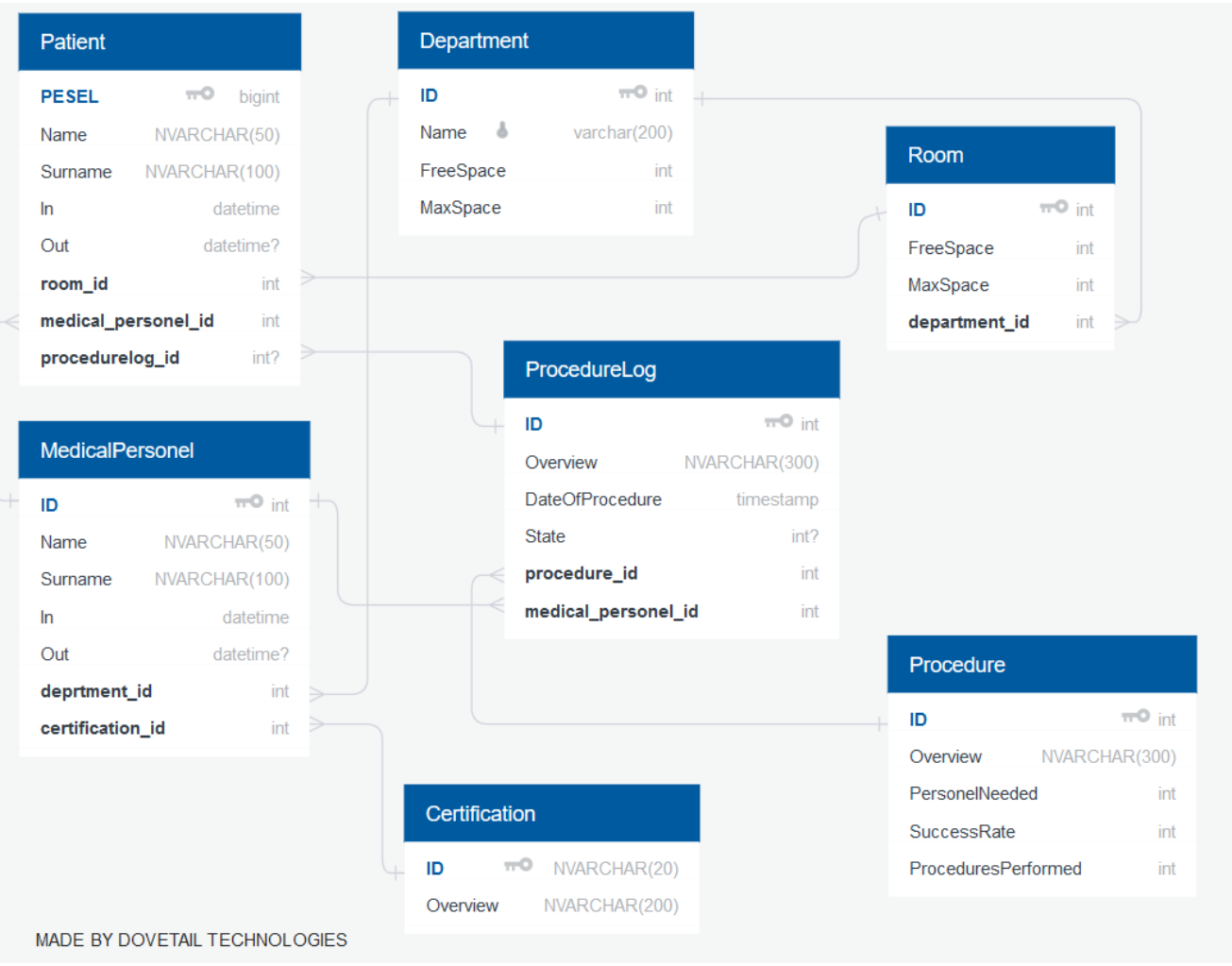
Projekt Bazy Danych

Temat: Baza Danych obsługująca pojedynczy szpital
Autor: Jan Chabik
Nr Albumu: 291060



1. Opis bazy

Baza danych służy obsłudze informatycznej jednego szpitala. Można dodawać nowych pacjentów i personel medyczny, przypisywać ich do poszczególnych sal i oddziałów. Można zapisywać informację o nadchodzących i przeszłych zabiegach medycznych oraz pacjentach i lekarzach którzy w nich uczestniczą/będą uczestniczyć. Baza pozwala prowadzić archiwum przeszłych pacjentów, personelu medycznego i zabiegów. Automatycznie prowadzone są statystyki na temat powodzenia zabiegów/operacji

2. Procedury specjalne
    a. Dodawanie pacjenta i pokoju (ARG : Imię, nazwisko, id_sali, nazwisko lekarza opiekuna, nr pokoju, liczba miejsc). Pokój musi należeć do oddziału do którego przypisany jest lekarz opiekun. Jednocześnie wykonywany jest update miejsc w pokoju i na oddziale.
    b. Dodawanie lekarza i certyfikacji (ARG : Imię, nazwisko, id_certyfikacji, opis cerfyfikatu, id_oddziału). Stotowane w sytuacji, gdy dodajemy lekarza, który ma certyfikacje, której opisu nie mamy w bazie danych.
    c. Dodawanie procedureLog i Procedury(ARG : opis zabiegu z tabelki procedureLog, data, PESEL pacjenta, nazwisko lekarza, nazwa procedury z tabelki Procedure, ilość potrzebnego personelu). Stosowane w sytuacji, kiedy wykonujemy zabieg, który nigdy nie odbył się w tym szpitalu.

SKRYPT

DROPY

```
/*DROPY TABEL*/
DROP TABLE patient
DROP TABLE procedurelog
DROP TABLE medicalpersonel
DROP TABLE room
DROP TABLE department
DROP TABLE proceduretype
DROP TABLE certification


/*DROPY PROCEDUR*/
DROP PROCEDURE add_certification
DROP PROCEDURE add_proceduretype
DROP PROCEDURE add_department1
DROP PROCEDURE add_room2
DROP PROCEDURE add_medicalpersonel
DROP PROCEDURE add_procedurelog
DROP PROCEDURE add_patient6


/*DROPY PROCEDUR NA UPDATE*/
DROP PROCEDURE sni_certification
DROP PROCEDURE sni_proceduretype
DROP PROCEDURE sni_department
DROP PROCEDURE sni_room1
DROP PROCEDURE sni_medicalpersonel2
DROP PROCEDURE sni_procedurelog
DROP PROCEDURE sni_patient

/*DROPY PROCEDUR SPECJALNYCH*/
DROP PROCEDURE add_patient_room1
DROP PROCEDURE add_medicalpersonel_certification
DROP PROCEDURE add_procedurelog_procedure

/*DROPY TRIGGERÓW*/
DROP TRIGGER patient_show_room
DROP TRIGGER del_patient
```

```
DROP TRIGGER procedue_update
DROP TRIGGER room_show_dep
DROP TRIGGER patient_leave
DROP TRIGGER del_room
DROP TRIGGER doc_leave
DROP TRIGGER add_personel
DROP TRIGGER insert_prclog
```

NOWE TABELE

```
/*NOWE TABELE*/
CREATE TABLE certification(
	id NVARCHAR(50) PRIMARY KEY,
	overview NVARCHAR(200)
)
CREATE TABLE proceduretype (
	id INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
	overview NVARCHAR(300),
	personel INT,
	successrate FLOAT,
	no_procedures INT

)
CREATE TABLE department (
	id INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
	dep_name NVARCHAR(100),
	free_space INT NOT NULL,
	max_space INT NOT NULL
)
CREATE TABLE room (
	id INT NOT NULL PRIMARY KEY,
	free_space INT NOT NULL,
	max_space INT NOT NULL,
	department_id INT FOREIGN KEY REFERENCES department(id)
)
CREATE TABLE medicalpersonel (
	id INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
	firstname NVARCHAR(100) NOT NULL,
	surname NVARCHAR(150) NOT NULL,
	inside DATETIME NOT NULL,
	outside DATETIME,
	department_id INT FOREIGN KEY REFERENCES department(id),
	certification_id NVARCHAR(50) FOREIGN KEY REFERENCES certification(id)
)

CREATE TABLE procedurelog (
	id INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
	overview NVARCHAR(300),
	prc_date DATETIME,
	prc_state INT,
	procedure_id INT FOREIGN KEY REFERENCES proceduretype(id),
	medical_personel_id INT FOREIGN KEY REFERENCES medicalpersonel(id)
)
CREATE TABLE patient (
	pesel BIGINT PRIMARY KEY,
	firstname NVARCHAR(100) NOT NULL,
	surname NVARCHAR(150) NOT NULL,
	inside DATETIME NOT NULL,
	outside DATETIME,
	room_id INT FOREIGN KEY REFERENCES room(id),
```

```
	medical_personel_id INT FOREIGN KEY REFERENCES medicalpersonel(id),
	procedurelog_id INT FOREIGN KEY REFERENCES procedurelog(id)
)
```

POPULUJEMY TABELE

```
INSERT INTO certification
VALUES('cardio1','Mlodszy kardiolog'),
('cardio2','Sredni kardiolog'),
('cardio3','starszy kardiolog'),
('dentist1','Mlodszy dentysta'),
('dentist2','Sredni dentysta'),
('dentist3','Starszy dentysta'),
('uro1','Mlodszy urolog'),
('uro2','Sredni urolog'),
('uro3','Starszy urolog'),
('endo1','Mlodszy endokrynolog'),
('endo2','Sredni endokrynolog'),
('endo3','Starszy endokrynolog')


INSERT INTO proceduretype
VALUES('Przeszczep serca i wstawienie zywego serca',6,0.0,0),
('Przeszczep serca i wstawienie sztucznego serca',6,0.0,0),
('Wstawienie bypassow',10,0.0,0),
('Wyciecie nerki',3,0.0,0),
('Przeszczep pluca',4,0.0,0),
('Wstawienie sztucznej kosci',2,0.0,0),
('Usuniecie drzazgi',1,0.0,0),
('Wyciagniecie kleszcza',1,0.0,0),
('Zalozenie gipsu',1,0.0,0),
('Usuniecie raka mozgu',8,0.0,0),
('Terapia ultradzwiekami i krysztalami X',0,0.0,0),
('Rezonans magnetyczny i wyciecie wyroska robaczkowego',6,0.0,0)

INSERT INTO department
VALUES('Kardiologia', 40, 40),
('OIOM', 20, 20),
('Urologia', 30, 30),
('Stomatologia', 20, 20),
('Medycyna alternatywna', 10, 10),
('Onkologia', 10, 10)

INSERT INTO room
VALUES(101,15,15,1),
(102,15,15,1),
(103,10,10,1),
(201,10,10,2),
(202,10,10,2),
(301,15,15,3),
(302,15,15,3),
(401,20,20,4),
(501,10,10,5),
(601,10,10,6)

INSERT INTO medicalpersonel
VALUES('Jan','Chabik',GETDATE(), NULL,1,'cardio1'),
('Jakub','Chabik',GETDATE(), NULL,1,'cardio2'),
('Anna','Chabik',GETDATE(), NULL,1,'cardio2'),
('Emilia','Chabik',GETDATE(), NULL,1,'cardio3'),
('Stanisław','Chabik',GETDATE(), NULL,2,'dentist1'),
```

```
('Ewa','Chabik',GETDATE(), NULL,2,'dentist2'),
('Magda','Kowalski',GETDATE(), NULL,3,'dentist3'),
('Konrad','Kowalsi',GETDATE(), NULL,4,'uro1'),
('Marcin','Malinowski',GETDATE(), NULL,4,'uro2'),
('Jan','Malinowski',GETDATE(), NULL,4,'uro3'),
('Anna','Malinowski',GETDATE(), NULL,5,'uro3'),
('Dominika','Rydz',GETDATE(), NULL,5,'endo1'),
('Samuel','Rydz',GETDATE(), NULL,5,'endo2'),
('Ryszard','Wojak',GETDATE(), NULL,6,'endo1'),
('Monika','Wojak',GETDATE(), NULL,6,'endo3')


INSERT INTO procedurelog
VALUES('Zabieg na kobiecie 80L. Zalecana szczegolna ostroznosc','20180619 10:00:00
AM',0,1,1),
('Zabieg na kobiecie 80L. Zalecana szczegolna ostroznosc','20180619 10:00:00 AM',0,1,2),
('Zabieg na kobiecie 84L. Zalecana szczegolna ostroznosc','20180619 10:00:00 AM',0,2,3),
('Zabieg na kobiecie 30L. Dobry stan pacjenta.','20180619 10:00:00 AM',0,2,3),
('Zabieg na kobiecie 50L. Duze zmiany w prawym plucu','20180619 10:00:00 AM',0,3,4),
('Zabieg na kobiecie 40L. Dobry stan pacjenta','20180619 10:00:00 AM',0,4,5),
('Zabieg na kobiecie 18L. Zły stan lewej zrenicy','20180619 10:00:00 AM',0,5,6),
('Zabieg na kobiecie 10L. Zalecana szczegolna ostroznosc','20180619 10:00:00 AM',0,6,7),
('Zabieg na mezczyznie 80L. Zalecana szczegolna ostroznosc','20180619 10:00:00 AM',0,6,8),
('Zabieg na mezczyznie 90L. Zalecana szczegolna ostroznosc','20180619 10:00:00 AM',0,6,9),
('Zabieg na mezczyznie 50L. Zly stan trzustki','20180619 10:00:00 AM',0,7,10),
('Zabieg na mezczyznie 53L. Dobry stan pacjenta','20180619 10:00:00 AM',0,8,11),
('Zabieg na mezczyznie 81L. Zalecana szczegolna ostroznosc. Bardzo zly stan
tchawicy','20180619 10:00:00 AM',0,9,12),
('Zabieg na mezczyznie 15L. Dobry stan pacjenta','20180619 10:00:00 AM',0,10,13),
('Zabieg na mezczyznie 40L. Zly stan jedynek i siekaczy','20180619 10:00:00 AM',0,11,14),
('Zabieg na mezczyznie 42L. Krzywe zeby','20180619 10:00:00 AM',0,11,14)

INSERT INTO patient
VALUES(1234567890,'Jan','Czarny', GETDATE(), NULL, 101, 1, 1),
(123,'Grzegorz','Riv', GETDATE(), NULL, 102, 2, 2),
(12345,'Apu','Humbo', GETDATE(), NULL, 103, 3, 3),
(12345678,'Homer','Simpson', GETDATE(), NULL, 201, 4, 4),
(123456,'Marge','Simpson', GETDATE(), NULL, 201, 4, 6),
(1234567,'Bart','Simpson', GETDATE(), NULL, 201, 5, 8),
(12,'Lisa','Simpson', GETDATE(), NULL, 202, 6, 9),
(1234,'Seymour','Skinner', GETDATE(), NULL, 301, 7, 10),
(123456711,'Maggie','Simpson', GETDATE(), NULL, 302, 8, 11),
(123450000,'Jimbo','Jones', GETDATE(), NULL, 401, 9, 12),
(1200000,'Crabapple','Dog', GETDATE(), NULL, 501, 10, 13),
(123400,'Millhouse','Manastorm', GETDATE(), NULL, 601, 11, 15)


SELECT * FROM certification
SELECT * FROM proceduretype
SELECT * FROM department
SELECT * FROM room
SELECT * FROM medicalpersonel
SELECT * FROM procedurelog
SELECT * FROM patient
```

TWORZYMY PROCEDURY NA INSERT

```
CREATE PROCEDURE add_certification @id nvarchar(50), @overview nvarchar(200)
AS
```

```
INSERT INTO certification
VALUES(@id, @overview)
GO

CREATE PROCEDURE add_proceduretype @overview nvarchar(300), @personel int
AS
INSERT INTO proceduretype
VALUES(@overview, @personel, 0.0, 0)
GO

CREATE PROCEDURE add_department1 @dep_name nvarchar(100)
AS
INSERT INTO department
VALUES(@dep_name,0 , 0)
GO
CREATE PROCEDURE add_room2 @room_number int, @dep_id int, @maxspace int
AS
UPDATE department
SET max_space = max_space + @maxspace, free_space = free_space + @maxspace
WHERE id = @dep_id
INSERT INTO room
VALUES(@room_number,@maxspace,@maxspace,@dep_id)
GO
CREATE PROCEDURE add_medicalpersonel @firstname NVARCHAR(100), @surname NVARCHAR(150),
@dep_id int, @cert_id nvarchar(50)
AS
INSERT INTO medicalpersonel
VALUES(@firstname,@surname,GETDATE(), NULL, @dep_id, @cert_id)
GO

CREATE PROCEDURE add_procedurelog @overview NVARCHAR(300), @date DATETIME ,@state int,
@prc_id int, @doctor_id int
AS
INSERT INTO procedurelog
VALUES(@overview,@date,@state,@prc_id,@doctor_id)
GO

CREATE PROCEDURE add_patient6 @pesel bigint, @firstname NVARCHAR(100), @surname
NVARCHAR(150), @room_id int, @doctor_id int, @prclg_id int
AS
INSERT INTO patient
VALUES(@pesel,@firstname,@surname,GETDATE(), NULL, @room_id, @doctor_id,@prclg_id)
UPDATE room
SET free_space = free_space-1
WHERE id = @room_id
GO
```

TWORZYMY PROCEDURY NA SEARCH AND UPDATE

```
CREATE PROCEDURE sni_certification @id nvarchar(50), @overview nvarchar(200)
AS
UPDATE certification
SET overview = @overview
WHERE id = @id
GO

CREATE PROCEDURE sni_proceduretype @id int, @overview nvarchar(300), @personel int
AS
UPDATE proceduretype
```

```
SET overview = @overview, personel = @personel
WHERE id = @id
GO

CREATE PROCEDURE sni_department @dep_id int, @dep_name nvarchar(100), @maxspace int,
@freespace int
AS
UPDATE department
SET dep_name = @dep_name, max_space = @maxspace, free_space = @freespace
WHERE id = @dep_id
GO

CREATE PROCEDURE sni_room1 @room_number int, @dep_id int, @maxspace int, @freespace int
AS
UPDATE room
SET department_id = @dep_id, max_space = @maxspace, free_space = @freespace
WHERE @room_number = id
GO

CREATE PROCEDURE sni_medicalpersonel2 @firstname NVARCHAR(100), @surname NVARCHAR(150),
@dep_id int, @cert_id nvarchar(50), @out_time datetime
AS
UPDATE medicalpersonel
SET department_id = @dep_id, certification_id = @cert_id, outside = @out_time
WHERE firstname = @firstname AND surname = @surname
GO

CREATE PROCEDURE sni_procedurelog @overview NVARCHAR(300), @date DATETIME ,@state int,
@prc_id int, @doctor_id int, @id int
AS
UPDATE procedurelog
SET overview = @overview,prc_date = @date,prc_state = @state, procedure_id =
@prc_id,medical_personel_id = @doctor_id
WHERE id = @id
GO

CREATE PROCEDURE sni_patient @pesel bigint, @room_id int, @doctor_id int, @prclg_id int,
@out_time datetime
AS
UPDATE patient
SET room_id = @room_id, medical_personel_id = @doctor_id, procedurelog_id = @prclg_id,
outside = @out_time
WHERE @pesel = pesel
```

TWORZYMY PROCEDURY NA DODAWANIE DO WIELU TABEL

```
CREATE PROCEDURE add_patient_room1 @pesel bigint, @firstname NVARCHAR(100), @surname
NVARCHAR(150), @room_id int, @prclg_id int, @doc_surname NVARCHAR(150), @max_space int,
@dep_id int
AS
DECLARE @doc_id int
SET @doc_id = (SELECT id FROM medicalpersonel
WHERE surname = @doc_surname)
INSERT INTO room
VALUES(@room_id, (@max_space - 1), @max_space, @dep_id)
INSERT INTO patient
VALUES(@pesel,@firstname,@surname,GETDATE(), NULL, @room_id, @doc_id,@prclg_id)
GO
```

```
CREATE PROCEDURE add_medicalpersonel_certification @firstname NVARCHAR(100), @surname
NVARCHAR(150), @dep_id int, @cert_id nvarchar(50), @cert_overview nvarchar(200)
AS
INSERT INTO certification
VALUES(@cert_id,@cert_overview)
INSERT INTO medicalpersonel
VALUES(@firstname,@surname,GETDATE(), NULL, @dep_id, @cert_id)
GO

CREATE PROCEDURE add_procedurelog_procedure @overview NVARCHAR(300), @date DATETIME ,@state
int, @doctor_id int, @prc_oview nvarchar(300), @doc_needed int
AS
INSERT INTO proceduretype
VALUES(@prc_oview, @doc_needed, 0.0, 0)
DECLARE @prc_id int
SET @prc_id = (SELECT MAX(p.id)
FROM proceduretype AS p)
INSERT INTO procedurelog
VALUES(@overview,@date,@state,@prc_id,@doctor_id)
GO
```

WYWOŁANIA PROCEDUR NA INSERT

Certyfikacja

```
SELECT * FROM certification WHERE id = 'kidney1'
EXEC add_certification @id = ‘kidney1’, @overview = ‘lekarz specjalizuje się w leczeniu
chorob nerek’
SELECT * FROM certification WHERE id = 'kidney1'
```

Typy zabiegow

```
SELECT * FROM proceduretype WHERE personel = 5
EXEC add_proceduretype @overview = 'Ucinanie reki zarazonej gangrena', @personel = 5
SELECT * FROM proceduretype WHERE personel = 5
```

Oddziały

```
SELECT * FROM department WHERE dep_name = 'Stomatologia'
EXEC add_department1 @dep_name = 'Stomatologia'
SELECT * FROM department WHERE dep_name = 'Stomatologia'
```

Pokoje

```
SELECT * FROM room r JOIN department d ON r.department_id = d.id WHERE r.id = 999
EXEC add_room2 @room_number = 999, @dep_id = 2, @maxspace = 6
SELECT * FROM room r JOIN department d ON r.department_id = d.id WHERE r.id = 999
```

Lekarze

```
SELECT * FROM medicalpersonel WHERE firstname = 'TEST'
EXEC add_medicalpersonel @firstname = 'TEST', @surname = 'TESTOWY', @dep_id = 1, @cert_id =
'uro1'
SELECT * FROM medicalpersonel WHERE firstname = 'TEST'
```

Zabiegi

```
SELECT * FROM procedurelog WHERE overview = 'TESTOWY'
EXEC add_procedurelog @overview = 'TESTOWY', @date = '20180619 10:00:00 AM', @state = 0,
@prc_id = 1, @doctor_id = 2
SELECT * FROM procedurelog WHERE overview = 'TESTOWY'
```

Pacjenci

```
SELECT * FROM patient WHERE firstname = 'TEST_PACJENT'
```

```
EXEC add_patient6 @pesel = 99999, @firstname = 'TEST_PACJENT', @surname = 'TEST_PACJENT',
@room_id = 101, @doctor_id = 2, @prclg_id = 2
SELECT * FROM patient WHERE firstname = 'TEST_PACJENT'
```

WYWOŁANIA PROCEDUR NA SEARCH AND UPDATE

Certyfikacja

```
SELECT * FROM certification WHERE id = 'cardio1'
EXEC sni_certification @id = 'cardio1', @overview = 'SNU_TEST'
SELECT * FROM certification WHERE id = 'cardio1'
```

Typy zabiegow

```
SELECT * FROM proceduretype WHERE id = 1
EXEC sni_proceduretype @id = 1, @overview = 'SNU_TEST', @personel = 5
SELECT * FROM proceduretype WHERE id = 1
```

Oddziały

```
SELECT * FROM department WHERE id = 1
EXEC sni_department @dep_id = 1, @dep_name = 'SNU_TEST', @maxspace = 40, @freespace = 40
SELECT * FROM department WHERE id = 1
```

Pokoje

```
SELECT * FROM room WHERE id = 1
EXEC sni_room1 @room_number = 1, @dep_id = 1, @maxspace = 33, @freespace = 33
SELECT * FROM room WHERE id = 1
```

Lekarze

```
SELECT * FROM medicalpersonel WHERE firstname = 'TEST'
EXEC sni_medicalpersonel2 @firstname = 'TEST', @surname = 'TESTOWY', @dep_id = 1, @cert_id =
'uro1', @out_time = '20180619 10:00:00 AM'
SELECT * FROM medicalpersonel WHERE firstname = 'TEST'
```

Zabiegi

```
SELECT * FROM procedurelog WHERE overview = 'SNU_TEST'
EXEC sni_procedurelog @overview = 'SNU_TEST', @date = '20180619 10:00:00 AM', @state = 2,
@prc_id = 1, @doctor_id = 2, @id = 1
SELECT * FROM procedurelog WHERE overview = 'SNU_TEST'
```

Pacjenci

```
SELECT * FROM patient WHERE pesel = 99999
EXEC sni_patient @pesel = 99999, @room_id =1, @doctor_id = 3, @prclg_id = 1, @out_time =
'20181212 10:44:44 AM'
SELECT * FROM patient WHERE pesel = 99999
```

WYWOŁANIA PROCEDUR NA WIELE INSERTÓW

Pacjent + pokoj

```
SELECT *
FROM patient JOIN room
ON patient.room_id = room.id
WHERE pesel = 11111
EXEC add_patient_room1 @pesel = 11111, @firstname = 'TEST_SUPER', @surname = 'Kowalski',
@room_id = 2, @prclg_id = 1, @doc_surname = 'Blaszkowska', @max_space = 8, @dep_id = 1
```

```
SELECT *
FROM patient JOIN room
ON patient.room_id = room.id
WHERE pesel = 11111
```

Personel + certyfikacja

```
SELECT *
FROM medicalpersonel JOIN certification
ON medicalpersonel.certification_id = certification.id
WHERE firstname = 'TEST_SUPER1'
EXEC add_medicalpersonel_certification @firstname = 'TEST_SUPER1', @surname = 'Khalifa',
@dep_id = 1, @cert_id = 'gyne1', @cert_overview = 'TEST_SUPER1'
SELECT *
FROM medicalpersonel JOIN certification
ON medicalpersonel.certification_id = certification.id
WHERE firstname = 'TEST_SUPER1'
```

Zabieg + typ zabiegu

```
SELECT *
FROM procedurelog JOIN proceduretype
ON procedurelog.procedure_id = proceduretype.id
WHERE procedurelog.overview = 'TEST_SUPER2'
EXEC add_procedurelog_procedure @overview = 'TEST_SUPER2', @date = '20180619 10:00:00 AM'
,@state = 0, @doctor_id = 4, @prc_oview = 'TEST_SUPER2', @doc_needed = 1
SELECT *
FROM procedurelog JOIN proceduretype
ON procedurelog.procedure_id = proceduretype.id
WHERE procedurelog.overview = 'TEST_SUPER2'
```

UTWORZENIE TRIGGERÓW

```
CREATE TRIGGER patient_show_room ON patient FOR UPDATE
AS
SELECT i.firstname, i.surname,    i.room_id FROM inserted i
WHERE NOT EXISTS (SELECT * FROM inserted i JOIN deleted d ON i.room_id = d.room_id)
GO
CREATE TRIGGER del_patient ON patient FOR DELETE
AS
BEGIN TRAN T1

IF EXISTS(SELECT * FROM deleted d
                    WHERE d.outside is null)
BEGIN
       RAISERROR('nie wolno zmieniac outside gdy jest nullem', 16, 3)
       ROLLBACK TRAN
END
GO
CREATE TRIGGER procedure_update ON procedurelog FOR UPDATE
AS
UPDATE proceduretype
SET no_procedures = no_procedures+1
WHERE id IN (SELECT d.procedure_id FROM deleted d JOIN
                           inserted i ON d.id = i.id JOIN
                           proceduretype p ON i.procedure_id = p.id)
UPDATE proceduretype
SET successrate = successrate+ 5.0
WHERE id IN (SELECT d.procedure_id FROM deleted d JOIN
                           inserted i ON d.id = i.id JOIN
```

```
                        proceduretype p ON i.procedure_id = p.id)
GO
CREATE TRIGGER room_show_dep ON room FOR UPDATE
AS
SELECT i.department_id FROM inserted i
WHERE NOT EXISTS (SELECT i.id FROM inserted i JOIN deleted d ON i.department_id =
d.department_id)
GO
CREATE TRIGGER patient_leave ON patient FOR UPDATE
AS
UPDATE procedurelog
SET prc_date = NULL, prc_state = 0
WHERE id IN (SELECT p.id FROM inserted i JOIN procedurelog p ON p.id = i.procedurelog_id)
GO
CREATE TRIGGER del_room ON room FOR DELETE
AS
SELECT d.department_id FROM deleted d
GO
CREATE TRIGGER doc_leave ON medicalpersonel FOR UPDATE
AS
SELECT * FROM procedurelog
WHERE id IN (SELECT pl.id FROM inserted i JOIN
                        procedurelog pl ON i.id = pl.medical_personel_id WHERE i.outside
IS NOT NULL)
GO
CREATE TRIGGER add_personel ON medicalpersonel FOR INSERT
AS
SELECT c.overview FROM inserted i JOIN
certification c ON c.id = i.certification_id
GO
CREATE TRIGGER insert_prclg ON procedurelog FOR INSERT
AS
SELECT * FROM proceduretype
WHERE id IN (SELECT p.id FROM inserted i JOIN proceduretype p ON p.id = i.procedure_id)
```

WYWOŁANIA TRIGGERÓW

1. Gdy robimy UPDATE pacjenta pokazujemy w których salach się coś zmieniło

```
UPDATE patient
SET room_id = 999
WHERE pesel = 99999
```

2. Gdy chcemy usunąć pacjenta będącego jeszcze w szpitalu nie możemy tego zrobić

```
DELETE FROM patient WHERE firstname = 'Lisa'
```

3. Gdy zmieniamy stan zabiegu w ProcedureLog odpowiednio zmienia się ProceduresPerformed i SuccessRate w Procedure oraz ScheduledProcedure w Patient.

```
SELECT * FROM procedurelog pl JOIN proceduretype p ON pl.procedure_id = p.id
UPDATE procedurelog
SET prc_state = 2
WHERE id = 1
    SELECT * FROM procedurelog pl JOIN proceduretype p ON pl.procedure_id = p.id
```

4. Gdy przypisujemy sale do innego oddziału wyświetlamy odziały w których przybyło osób

```
UPDATE room
SET department_id = 4
WHERE id = 101
```

5. Gdy Pacjent opuszcza szpital, a jest zapisany na zabieg, zostaje oznaczony w specjalny sposób

```
SELECT * FROM procedurelog pl JOIN patient p ON p.procedurelog_id = pl.id WHERE p.pesel = 123
UPDATE patient
SET outside = '20180619 10:00:00 AM'
WHERE pesel = 123
    SELECT * FROM procedurelog pl JOIN patient p ON p.procedurelog_id = pl.id WHERE p.pesel =
    123
```

6. Gdy usuwamy sale, pokazujemy oddziały na których ubyło sal

```
   DELETE room
WHERE id = 501
```

7. Gdy lekarz przestaje pracować w szpitalu wyświetlana jest lista zabiegów którym miał przewodzić.

```
      UPDATE medicalpersonel
SET outside = '20180619 10:00:00 AM'
WHERE id = 3
```

8. Gdy dodajemy personel medyczny wyświetlana jest certyfikacja personelu (INSERT)

```
INSERT INTO medicalpersonel
VALUES('KOWAL','KOWALSKI', GETDATE(), NULL, 3, 'endo1')
```

9. Gdy dodajemy nowy Log Procedury wyświetlane są informację o liczbie wykonanych zabiegów danego typu, liczbie wykonanych zabiegów i ilości potrzbnego personelu (INSERT)

```
INSERT INTO procedurelog
VALUES('TEST_ZABIEG','20180619 10:00:00 AM',0,1,4)
```