

Міністерство освіти і науки України
Національний університет “Львівська політехніка”



Звіт
про виконання лабораторної роботи №1
із курсу “Аналітичні та нереляційні бази даних”

Виконав:
ст. гр. ІР-42, ІКТА
Найда Віталій

Прийняв:
Верес З.Є.

Sql script для створення для завдання

24	Найда	Allscript s (Patient registration)	<p>Опишіть <u>варіант(и)</u> використання (use cases) реєстрації пацієнта в клініці. <u>Варіант(и)</u> використання мають містити опис всіх необхідних компонентів варіанту використання, зокрема, основний порядок дій та альтернативні шляхи, винятки.</p> <p>Намалюйте діаграму класів [class diagram], щоб представити множину об'єктів, необхідних для збереження медичної інформації про пацієнта. <i>Примітка - зазвичай діаграма класів служить для представлення статичної структури моделі системи в цілому, однак для цілей цієї лабораторної роботи намалюйте лише ту її частину, яка стосується поставленої задачі</i></p> <p>Намалюйте діаграму активності [activity diagram] реєстрації пацієнта в клініці та направлення його у вільну палату</p> <p>Намалюйте діаграму послідовності [sequence diagram], яка показує пошук вільної палати для пацієнта, перевезення пацієнта у вільну палату та запис інформації про доставку пацієнта у базі даних</p>
----	-------	---	--

USE [master]

GO

IF (db_id(N'lab1db') IS NOT NULL)

DROP DATABASE lab1db;

GO

CREATE DATABASE lab1db;

GO

USE lab1db;

GO

CREATE TABLE Patient (

id INT IDENTITY PRIMARY KEY,

first_name VARCHAR(50) NOT NULL,

last_name VARCHAR(50) NOT NULL,

birth_date DATE NOT NULL,

bio VARCHAR(1000),

country VARCHAR(50),

city VARCHAR(50),

phone CHAR(13),

sex VARCHAR(10) NOT NULL CHECK (sex IN('MALE', 'FEMALE')),

CONSTRAINT CK_PATIENT_PHONE

CHECK (phone LIKE ('[0-9][0-9][0-9]-[0-9][0-9]-[0-9][0-9]-[0-9][0-9][0-9]'))

);

```

CREATE TABLE Doctor (
    id INT IDENTITY PRIMARY KEY,
    first_name VARCHAR(50) NOT NULL,
    last_name VARCHAR(50) NOT NULL,
    birth_date DATE NOT NULL,
    bio VARCHAR(1000),
    sex VARCHAR(10) NOT NULL CHECK (sex IN('MALE', 'FEMALE')),
    salary INT
);

```

---delete location field, add city, country

```

CREATE TABLE Hospital (
    id INT IDENTITY PRIMARY KEY,
    name VARCHAR(50) NOT NULL,
    city VARCHAR(50) NOT NULL,
    country VARCHAR(50) NOT NULL,
    about VARCHAR(1000)
);

```

```

CREATE TABLE [Procedure] (
    id INT IDENTITY PRIMARY KEY,
    name VARCHAR(50) NOT NULL,
    description VARCHAR(1000),
    price REAL NOT NULL,
    duration INT NOT NULL DEFAULT(0),
    CONSTRAINT CK_PROCEDURE_PRICE CHECK (price >= 0)
);

```

```

CREATE TABLE Cabinet (
    id INT IDENTITY PRIMARY KEY,
    number INT NOT NULL,
    description VARCHAR(1000),
    hospital_id INT NOT NULL,
    CONSTRAINT FK_CABINET_HOSPITAL_ID FOREIGN KEY (hospital_id) REFERENCES
Hospital (id)
);

```

```

CREATE TABLE [Resource] (
    id INT IDENTITY PRIMARY KEY,
    name VARCHAR(50) NOT NULL,
    available_quantity INT NOT NULL DEFAULT(0),

```

```
price REAL NOT NULL,  
CONSTRAINT CK_RESOURCE_PRICE CHECK (price >= 0)  
);
```

```
CREATE TABLE PROCEDURE_RESOURCE (  
    procedure_id INT NOT NULL,  
    resource_id INT NOT NULL,  
    quantity INT NOT NULL DEFAULT(0),  
    consumation_date DATE,  
    CONSTRAINT PK_PROCEDURE_RESOURCE PRIMARY KEY (procedure_id, resource_id),  
    CONSTRAINT FK_PROCEDURE_RESOURCE_PROCEDURE_ID FOREIGN KEY  
(procedure_id) REFERENCES [Procedure] (id),  
    CONSTRAINT FK_PROCEDURE_RESOURCE_RESOURCE_ID FOREIGN KEY (resource_id)  
REFERENCES [Resource] (id)  
);
```

```
CREATE TABLE PROCEDURE_PATIENT (  
    procedure_id INT NOT NULL,  
    patient_id INT NOT NULL,  
    date DATE,  
    CONSTRAINT PK_PROCEDURE_PATIENT PRIMARY KEY (procedure_id, patient_id),  
    CONSTRAINT FK_PROCEDURE_PATIENT_PROCEDURE_ID FOREIGN KEY (procedure_id)  
REFERENCES [Procedure] (id),  
    CONSTRAINT FK_PROCEDURE_PATIENT_PATIENT_ID FOREIGN KEY (patient_id)  
REFERENCES Patient (id)  
);
```

---delete position, job_start---

```
CREATE TABLE DOCTOR_HOSPITAL (  
    doctor_id INT NOT NULL,  
    hospital_id INT NOT NULL,  
    job_start DATE,  
    job_end DATE,  
    CONSTRAINT PK_DOCTOR_HOSPITAL PRIMARY KEY (doctor_id, hospital_id),  
    CONSTRAINT FK_DOCTOR_HOSPITAL_DOCTOR_ID FOREIGN KEY (doctor_id)  
REFERENCES Doctor (id),  
    CONSTRAINT FK_DOCTOR_HOSPITAL_HOSPITAL_ID FOREIGN KEY (hospital_id)  
REFERENCES Hospital (id)  
);
```

---add job_end and check CK_RESOURCE_JOB---

```

CREATE TABLE PATIENT_HOSPITAL (
    patient_id INT NOT NULL,
    hospital_id INT NOT NULL,
    position VARCHAR(50),
    CONSTRAINT PK_PATIENT_HOSPITAL PRIMARY KEY (patient_id, hospital_id),
    CONSTRAINT FK_PATIENT_HOSPITAL_PATIENT_ID FOREIGN KEY (patient_id)
REFERENCES Patient (id),
    CONSTRAINT FK_PATIENT_HOSPITAL_HOSPITAL_ID FOREIGN KEY (hospital_id)
REFERENCES Hospital (id)
);

```

---ADD TABLE CABINET_PROCEDURE AND DOCTOR_PROCEDURE

```

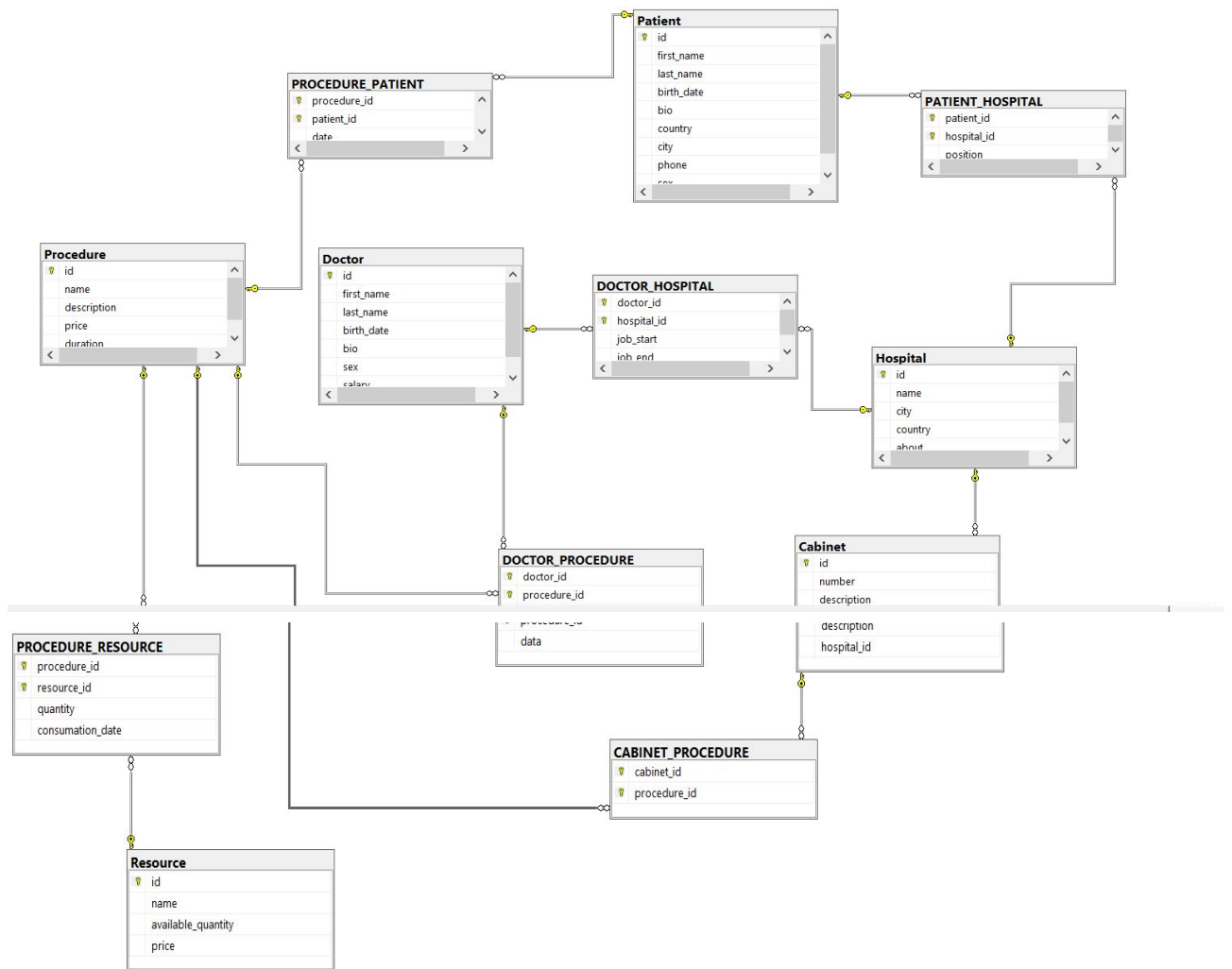
CREATE TABLE CABINET_PROCEDURE (
    cabinet_id INT NOT NULL,
    procedure_id INT NOT NULL,
    CONSTRAINT PK_CABINET_PROCEDURE PRIMARY KEY (cabinet_id, procedure_id),
    CONSTRAINT FK_CABINET_PROCEDURE_CABINET_ID FOREIGN KEY (cabinet_id)
REFERENCES Cabinet (id),
    CONSTRAINT FK_CABINET_PROCEDURE_PROCEDURE_ID FOREIGN KEY (procedure_id)
REFERENCES [Procedure] (id)
);

```

```

CREATE TABLE DOCTOR_PROCEDURE (
    doctor_id INT NOT NULL,
    procedure_id INT NOT NULL,
    data DATE,
    CONSTRAINT PK_DOCTOR_PROCEDURE PRIMARY KEY (doctor_id, procedure_id),
    CONSTRAINT FK_DOCTOR_PROCEDURE_DOCTOR_ID FOREIGN KEY (doctor_id)
REFERENCES Doctor (id),
    CONSTRAINT FK_DOCTOR_PROCEDURE_PROCEDURE_ID FOREIGN KEY (procedure_id)
REFERENCES [Procedure] (id)
);

```



Зв'язки

Один до багатьох:

Лікарня - кабінети

багато до багатьох:

лікарня - пацієнт (лікарня обслуговує багато пацієнтів, а пацієнт може звертатись до різних клінік)

лікарня - лікар (в лікарні працює багато лікарів, лікар може мати кілька робіт в різних закладах)

процедура - ресурс (ресурс - це медикаменти, обладнання, рукавички і тд) (один ресурс може використовуватись під час різноманітних процедур і процедура може використовувати багато ресурсів)

процедура - пацієнт (одна процедура необхідна різним пацієнтам, пацієнту треба кілька процедур)