МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

**Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования  
 «Кемеровский государственный университет»**

**Математический факультет**

**Кафедра ЮНЕСКО по новым информационным технологиям**

**ОТЧЕТ ПО СЕМЕСТРОВОЙ РАБОТЕ**

**ПО ДИСЦИПЛИНЕ**

**“Базы данных”**

# «Информационная система медицинских организаций города

студента 2 курса

**Кузива Павла Владимировича**

Направление 02.03.03 – Математическое обеспечение и администрирование информационных систем

Преподаватель:

С. Ю. Завозкин

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Работа защищена:

“15 июня”\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2021г.

с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_

Кемерово 2021

СОДЕРЖАНИЕ

[Введение 2](#_Toc310499682)

[1. Постановка задачи 2](#_Toc310499683)

[2. ER-моделирование 3](#_Toc310499684)

[3. Создание БД 6](#_Toc310499685)

[4. Выполнение выборки данных 16](#_Toc310499686)

[Заключение 3](#_Toc310499687)

[Литература 3](#_Toc310499688)

# Введение

Целью работы является разработка информационной системы медицинских организаций города.

# 1. Постановка задачи

Каждая больница города состоит из одного или нескольких корпусов, в каждом из которых размещается одно или несколько отделений, специализирующихся на лечении определенной группы болезней; каждое отделение и имеет некоторое количество палат на определенное число коек. Поликлиники могут административно быть прикрепленными к больницам, а могут быть и нет. Как больницы, так и поликлиники обслуживаются врачебным (хирурги, терапевты, невропатологи, окулисты, стоматологи, рентгенологи, гинекологи и пр.) и обслуживающим персоналом (мед. сестры, санитары, уборщицы и пр.). Врачи любого профиля могут иметь степень кандидата или доктора медицинских наук. Степень доктора медицинских наук дает право на присвоение звания профессора, а степень кандидата медицинских наук на присвоение звания доцента. Разрешено совместительство, так что каждый врач может работать либо в больнице, либо в поликлинике, либо и в одной больнице и в одной поликлинике. Врачи со званием доцента или профессора могут консультировать в нескольких больницах или поликлиниках.

Лаборатории, выполняющие те или иные медицинские анализы, могут обслуживать различные больницы и поликлиники, при условии наличия договора на обслуживание с соответствующим лечебным заведением. При этом каждая лаборатория имеет один или несколько профилей: биохимические, физиологические, химические исследования.

Пациенты амбулаторно лечатся в одной из поликлиник, и по направлению из них могут стационарно лечиться либо в больнице, к которой относится поликлиника, либо в любой другой, если специализация больницы, к которой приписана поликлиника не позволяет провести требуемое лечение. Как в больнице, так и в поликлинике ведется персонифицированный учет пациентов, полная история их болезней, все назначения, операции и т.д. В больнице пациент имеет в каждый данный момент одного лечащего врача, в поликлинике - несколько.

# 2-3. ER-моделирование

При разработке системы были выделены основные сущности предметной области:

- Сотрудники;

- Врачи-отделения;

- Мед учреждения;

- Больницы;

- Отделения;

- Больные;

- Палаты

Проведенный анализ предметной области, а так же подробное исследование информационной потребности позволило выявить необходимый набор атрибутов. Рассмотрим их подробнее ниже.

Сущность «Сотрудники» хранит общую информацию о работниках медицинской сферы города:

- Код сотрудника (Ключевое поле)

- ФИО

- Стаж

-Должность

Сущность «Мед учреждения» хранит общие данные о медицинских организациях города:

- Код учреждения (Ключевое поле)

- Название

- Адрес

Сущность «Больницы» является дочерней для сущности «Мед учреждения» и содержит конкретную информацию о больницах города:

- Код учреждения (Ключевое поле, Внешний ключ)

- Код больницы (Ключевое поле)

- Количество палат

- Количество коек

Сущность «Отделения» содержит информацию об отделениях больниц:

- Код отделения (Ключевое поле)

- Код больницы (Внешний ключ)

- Номер корпуса

- Количество палат

- Количество коек

- Код учреждения (Внешний ключ)

Сущность «Палаты» содержит информацию о палатах, принадлежащих каждому отделению:

- Код палаты (Ключевое поле)

- Номер палаты

- Количество мест

- Код отделения (Внешний ключ)

Сущность «Врачи-отделения» хранит информацию о принадлежности врачебного персонала к отделению определенного учреждения:

- Код (Ключевое поле)

- Код сотрудника(Внешний ключ)

- Код отделения (Внешний ключ)

Сущность «Больные» содержит данные о пациентах, находящихся на стационарном лечении:

* + Код больного (Ключевое поле)
  + Код палаты (Внешний ключ)
  + Дата поступления
  + Диагноз
  + Дата выписки
  + Фамилия
  + Имя
  + Дата рождения

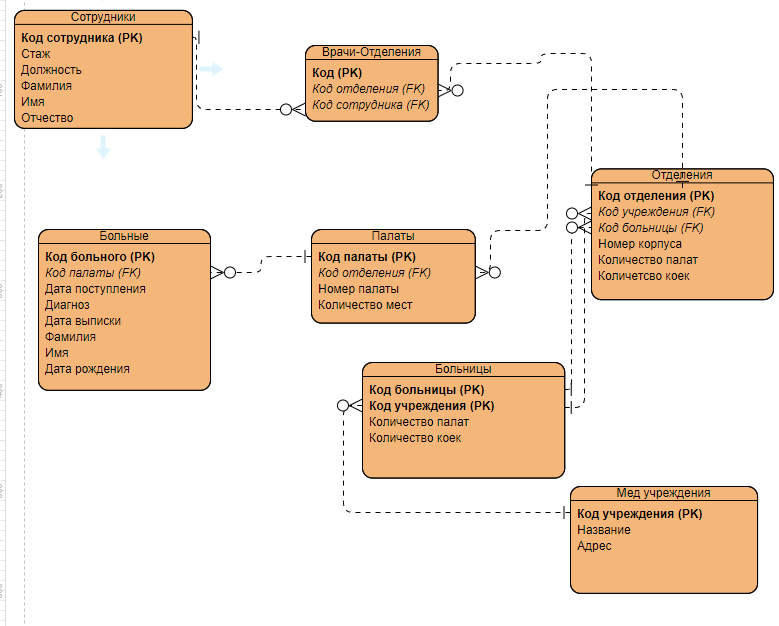


Рисунок 1 – Логическая модель базы данных

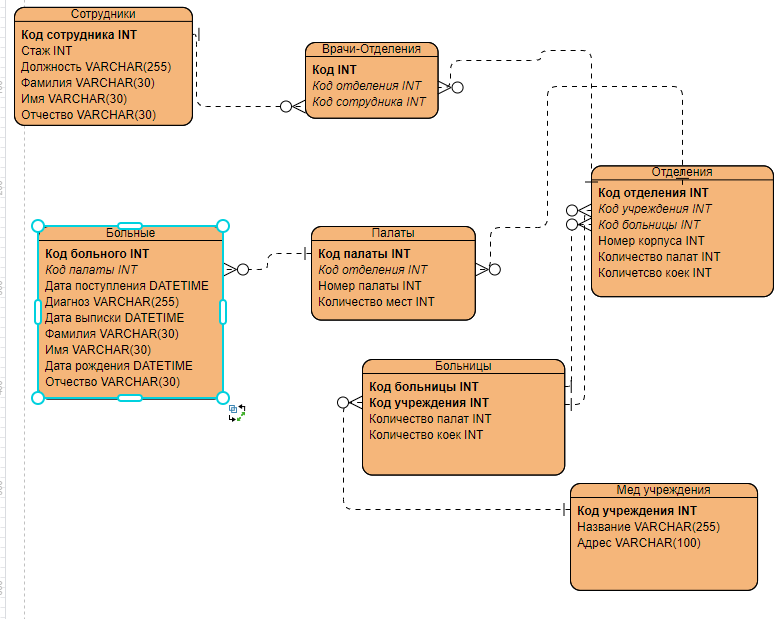


Рисунок 2 – Физическая модель базы данных

# 4. Создание и заполнение БД, запросы

# drop table medical\_employee cascade constraints;

# drop table medical\_establishment cascade constraints;

# drop table hospitals cascade constraints;

# drop table med\_department cascade constraints;

# drop table doctor\_to\_department cascade constraints;

# drop table chambers cascade constraints;

# drop table patients cascade constraints;

# /\*==============================================================\*/

# /\* Table: medical\_employee \*/

# /\*==============================================================\*/

# create table medical\_employee

# (

# EMPLOYEE\_ID NUMBER(7) NOT NULL CONSTRAINT EMPM\_ID\_PK PRIMARY KEY,

# EXPERIENCE NUMBER(7) NOT NULL,

# POSITION VARCHAR(255),

# LAST\_NAME VARCHAR(30),

# FIRST\_NAME VARCHAR(30),

# PATRONYMIC VARCHAR(20)

# );

# /\*==============================================================\*/

# /\* Table: medical\_establishment \*/

# /\*==============================================================\*/

# create table medical\_establishment

# (

# ESTABLISHMENT\_ID NUMBER(7) NOT NULL CONSTRAINT EST\_ID\_PK PRIMARY KEY,

# NAME VARCHAR(255) NOT NULL,

# ADRESS VARCHAR(100) NOT NULL

# );

# /\*==============================================================\*/

# /\* Table: hospitals \*/

# /\*==============================================================\*/

# create table hospitals

# (

# HOSPITAL\_ID NUMBER(7) NOT NULL,

# ESTABLISHMENT\_ID NUMBER(7) NOT NULL CONSTRAINT EST\_ID\_FK REFERENCES medical\_establishment (ESTABLISHMENT\_ID),

# NUMBER\_OF\_CHAMBERS NUMBER(7) NOT NULL,

# NUMBER\_OF\_BEDS NUMBER(7) NOT NULL,

# CONSTRAINT HOST\_PK PRIMARY KEY (HOSPITAL\_ID, ESTABLISHMENT\_ID)

# );

# /\*==============================================================\*/

# /\* Table: med\_department \*/

# /\*==============================================================\*/

# create table med\_department

# (

# DEPARTMENT\_ID NUMBER(7) NOT NULL CONSTRAINT DEP\_ID\_PK PRIMARY KEY,

# ESTABLISHMENT\_ID NUMBER(7) NOT NULL,

# HOSPITAL\_ID NUMBER(7) NOT NULL,

# CASE\_NUMBER NUMBER(7) NOT NULL,

# NUMBER\_OF\_CHAMBERS NUMBER(7) NOT NULL,

# NUMBER\_OF\_BEDS NUMBER(7) NOT NULL,

# CONSTRAINT MED\_DEP\_FK FOREIGN KEY (ESTABLISHMENT\_ID, HOSPITAL\_ID) REFERENCES hospitals (ESTABLISHMENT\_ID, HOSPITAL\_ID)

# );

# /\*==============================================================\*/

# /\* Table: doctor\_to\_department \*/

# /\*==============================================================\*/

# create table doctor\_to\_department

# (

# ID NUMBER(7) NOT NULL CONSTRAINT dct\_id\_pk PRIMARY KEY,

# DEPARTMENT\_ID NUMBER(7) NOT NULL REFERENCES med\_department(DEPARTMENT\_ID) ,

# EMPLOYEE\_ID NUMBER (7) NOT NULL REFERENCES medical\_employee(EMPLOYEE\_ID)

# );

# /\*==============================================================\*/

# /\* Table: chambers \*/

# /\*==============================================================\*/

# create table chambers

# (

# CHAMBER\_ID NUMBER(7) NOT NULL CONSTRAINT CHAM\_ID\_PK PRIMARY KEY,

# DEPARTMENT\_ID NUMBER(7) NOT NULL REFERENCES med\_department (DEPARTMENT\_ID),

# CHAMBER\_NUMBER NUMBER (7) NOT NULL

# );

# /\*==============================================================\*/

# /\* Table: patients \*/

# /\*==============================================================\*/

# create table patients

# (

# PATIENT\_ID NUMBER(7) NOT NULL CONSTRAINT PAT\_ID\_PK PRIMARY KEY,

# CHAMBER\_ID NUMBER (7) NOT NULL REFERENCES chambers (CHAMBER\_ID),

# RECEIPT\_DATE DATE,

# DIAGNOSIS VARCHAR(255),

# LAST\_NAME VARCHAR(30),

# FIRST\_NAME VARCHAR(30),

# PATRONYMIC VARCHAR(20),

# BIRTH\_DAY DATE

# );

# Заполнение данными

INSERT INTO medical\_employee

(EMPLOYEE\_ID, EXPERIENCE, POSITION, LAST\_NAME, FIRST\_NAME, PATRONYMIC)

VALUES

(10001, 12, 'surgeon', 'Petrov', 'Petr', 'Petrovich');

INSERT INTO medical\_employee

(EMPLOYEE\_ID, EXPERIENCE, POSITION, LAST\_NAME, FIRST\_NAME, PATRONYMIC)

VALUES

(10002, 13, 'surgeon', 'Ivanov', 'Ivan', 'Ivanovich');

INSERT INTO medical\_employee

(EMPLOYEE\_ID, EXPERIENCE, POSITION, LAST\_NAME, FIRST\_NAME, PATRONYMIC)

VALUES

(10003, 10, 'nurse', 'Klimova', 'Elena', 'Ivanovna');

INSERT INTO medical\_establishment

(ESTABLISHMENT\_ID, NAME, ADRESS)

VALUES

(11, 'Regional Hospital', 'Gagarina 28');

INSERT INTO medical\_establishment

(ESTABLISHMENT\_ID, NAME, ADRESS)

VALUES

(12, 'Hospital №1', 'Gagarina 50');

INSERT INTO medical\_establishment

(ESTABLISHMENT\_ID, NAME, ADRESS)

VALUES

(13, 'Traumatology', 'Kirova 10');

INSERT INTO hospitals

(HOSPITAL\_ID, ESTABLISHMENT\_ID, NUMBER\_OF\_CHAMBERS, NUMBER\_OF\_BEDS)

VALUES

(101, 11, 80, 100);

INSERT INTO hospitals

(HOSPITAL\_ID, ESTABLISHMENT\_ID, NUMBER\_OF\_CHAMBERS, NUMBER\_OF\_BEDS)

VALUES

(102, 12, 50, 80);

INSERT INTO hospitals

(HOSPITAL\_ID, ESTABLISHMENT\_ID, NUMBER\_OF\_CHAMBERS, NUMBER\_OF\_BEDS)

VALUES

(103, 13, 200, 300);

INSERT INTO med\_department

(DEPARTMENT\_ID, ESTABLISHMENT\_ID, HOSPITAL\_ID, CASE\_NUMBER, NUMBER\_OF\_CHAMBERS, NUMBER\_OF\_BEDS)

VALUES

(1, 11, 101, 1, 80, 100);

INSERT INTO med\_department

(DEPARTMENT\_ID, ESTABLISHMENT\_ID, HOSPITAL\_ID, CASE\_NUMBER, NUMBER\_OF\_CHAMBERS, NUMBER\_OF\_BEDS)

VALUES

(2, 12, 102, 1, 50, 80);

INSERT INTO med\_department

(DEPARTMENT\_ID, ESTABLISHMENT\_ID, HOSPITAL\_ID, CASE\_NUMBER, NUMBER\_OF\_CHAMBERS, NUMBER\_OF\_BEDS)

VALUES

(3, 13, 103, 1, 200, 300);

INSERT INTO doctor\_to\_department

(ID, DEPARTMENT\_ID, EMPLOYEE\_ID)

VALUES

(1, 1, 10001);

INSERT INTO doctor\_to\_department

(ID, DEPARTMENT\_ID, EMPLOYEE\_ID)

VALUES

(2, 2, 10002);

INSERT INTO doctor\_to\_department

(ID, DEPARTMENT\_ID, EMPLOYEE\_ID)

VALUES

(3, 3, 10003);

INSERT INTO chambers

(CHAMBER\_ID, DEPARTMENT\_ID, CHAMBER\_NUMBER)

VALUES

(10, 1, 1);

INSERT INTO chambers

(CHAMBER\_ID, DEPARTMENT\_ID, CHAMBER\_NUMBER)

VALUES

(11, 2, 2);

INSERT INTO chambers

(CHAMBER\_ID, DEPARTMENT\_ID, CHAMBER\_NUMBER)

VALUES

(12, 3, 3);

INSERT INTO patients

(PATIENT\_ID, CHAMBER\_ID, RECEIPT\_DATE, DIAGNOSIS, LAST\_NAME, FIRST\_NAME, PATRONYMIC, BIRTH\_DAY)

VALUES

(111, 10, to\_date('11.05.2020','dd.mm.yyyy'), 'fracture', 'Kirov', 'Mark', 'Petrovich', to\_date('11.05.2000','dd.mm.yyyy') );

INSERT INTO patients

(PATIENT\_ID, CHAMBER\_ID, RECEIPT\_DATE, DIAGNOSIS, LAST\_NAME, FIRST\_NAME, PATRONYMIC, BIRTH\_DAY)

VALUES

(112, 11, to\_date('01.12.2020','dd.mm.yyyy'), 'fracture', 'Krakov', 'Andrey', 'Andreevich', to\_date('13.04.1900','dd.mm.yyyy') );

INSERT INTO patients

(PATIENT\_ID, CHAMBER\_ID, RECEIPT\_DATE, DIAGNOSIS, LAST\_NAME, FIRST\_NAME, PATRONYMIC, BIRTH\_DAY)

VALUES

(113, 12, to\_date('18.11.2020','dd.mm.yyyy'), 'fracture', 'Limonov', 'Kirill', 'Kirillovich', to\_date('29.03.1990','dd.mm.yyyy') );

# Запросы

--1

SELECT last\_name, first\_name

FROM medical\_employee

WHERE position = 'surgeon';

--4

SELECT last\_name, first\_name

FROM medical\_employee

WHERE position = 'surgeon' AND experience > 12;

--6

SELECT last\_name, first\_name, receipt\_date, diagnosis

FROM patients

WHERE CHAMBER\_ID = 10;

--Количество больных в мед. Учреждениях:

select est.name, count(pt.last\_name) "кол-во"

from patients pt, chambers ch, med\_department dep, hospitals hos, medical\_establishment est

where pt.chamber\_id = ch.chamber\_id and ch.department\_id = dep.department\_id

and dep.hospital\_id = hos.hospital\_id and hos.establishment\_id = est.establishment\_id

group by est.name

# Заключение

В результате выполнения работы была реализована информационная система медицинских организаций города.

На первом этапе работы была проанализирована предметная область, возможные средства проектирования базы данных, а так же физические платформы для реализации базы данных.

На втором этапе, проведенный анализ технического задания позволил выделить общие требования к системе, на основании которых было произведено проектирование базы данных. Была построена логическая и физическая модели базы данных.

Разработанная модель является адекватной предметной области и позволяет реализовать систему на выбранных в работе средствах.

# Литература

# Использованная литература

1. Карпова Т. Базы данных: модели, разработка, реализация. - СПб.: Питер, 2005. – 304 с.: ил.
2. Крёнке Д. Теория и практика построения баз данных. 8-е изд. СПб.: Питер, 2004. – 800с.