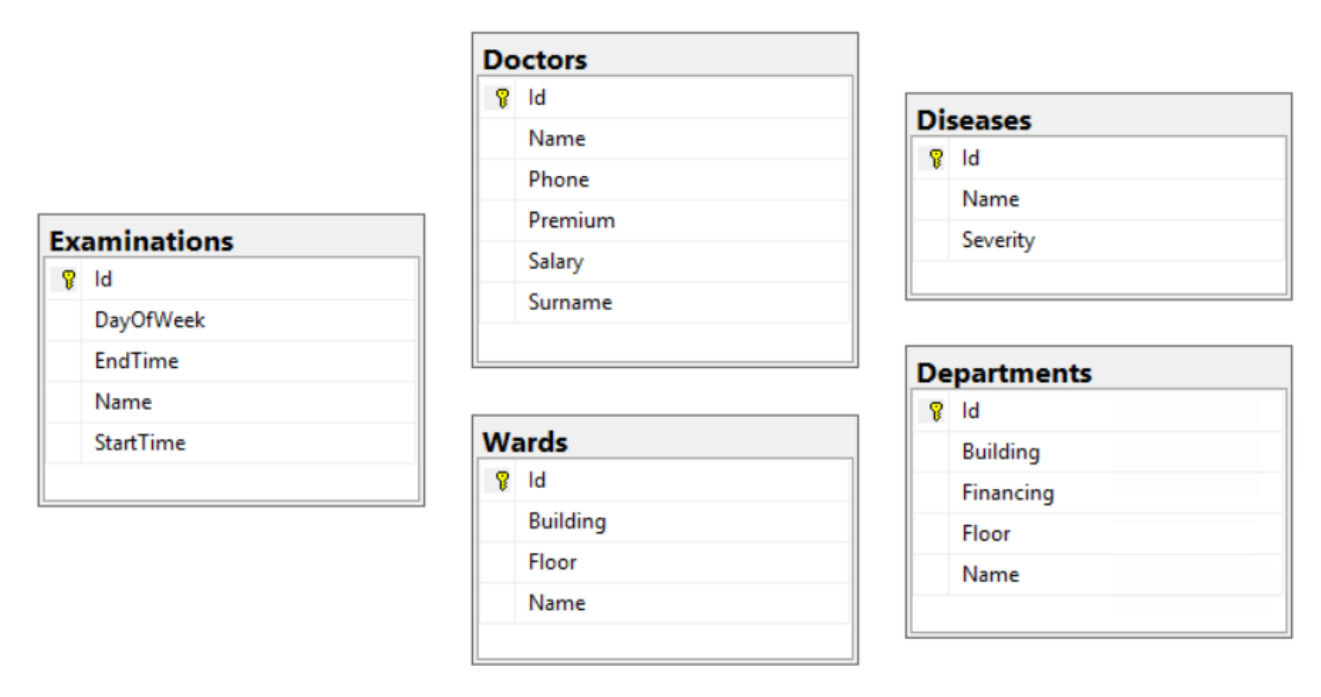


**Практическая работа**

****

Удалить БД **Hospital**, если она существует. Создать БД **Hospital**. Использовать БД **Hospital**.

DROP DATABASE IF EXISTS Hospital;

CREATE DATABASE Hospital;

USE Hospital;

|  |  |
| --- | --- |
| Создать БД | CREATE DATABASE teststep; |
| Изменить наименование БД | ALTER DATABASE teststep  MODIFY NAME = teststep2; |
| Удалить БД | DROP DATABASE teststep; |

■ Создать таблицу Отделения (**Departments**). Идентификатор (**Id**) - уникальный идентификатор отделения, тип данных — **int**, первичный ключ. Корпус (**Building**) - номер корпуса, в котором располагается отделение, тип данных — **int**, не может содержать **null**-значения. Финансирование (**Financing**) - фонд финансирования отделения, тип данных — **money**, не может содержать **null**-значения. Этаж (**Floor**) – этаж отделения, тип данных — **nvarchar(100)**, не может содержать **null**-значения. Название (**Name**) - название отделения, тип данных — **int**, не может содержать **null**-значения.

|  |  |
| --- | --- |
| Создать таблицу | USE teststep  CREATE TABLE students  (-- PRIMARY KEY = UNIQUE + NOT NULL  id INT PRIMARY KEY,  fio NVARCHAR(50) NOT NULL,  groupNumber NVARCHAR(10)); |
| Изменить наименование таблицы | USE teststep  EXEC sp\_rename 'students', 'Students'; |
| Удалить таблицу | USE teststep  DROP TABLE students; |
| Добавить поле в таблицу | USE teststep  ALTER TABLE students  ADD letter NVARCHAR(5); |
| Изменить тип поля в таблице | USE teststep  ALTER TABLE students  ALTER COLUMN letter NVARCHAR(10); |
| Удалить поле из таблицы | USE teststep  ALTER TABLE students  DROP COLUMN letter; |

CREATE TABLE Departments

(

Id INT PRIMARY KEY,

Building INT NOT NULL,

Financing MONEY NOT NULL,

Floor INT NOT NULL,

Name NVARCHAR(100) NOT NULL

);

Заполнить таблицу данными (10 строк).

|  |  |
| --- | --- |
| Вставить строку с информацией в таблицу | USE teststep  INSERT INTO students (id, fio, groupNumber)  VALUES  (1, N'Иванов Иван Иванович', N'ЕКО-11021'),  (2, N'Петров Петр Петрович', N'ЕКО-11022'),  (3, N'Сидоров Сидор Сидорович', N'ЕКО-11021'); |
| Обновить (редактировать) данные поля в таблице по условию | USE teststep  UPDATE students  SET groupNumber = N'ЕКО-11023'  WHERE id = 3; |
| Удалить строку в таблице по условию | USE teststep  DELETE FROM students  WHERE id = 3; |
| Очистить данные таблицы | USE teststep  TRUNCATE TABLE students; |

INSERT INTO Departments (Id, Building, Financing, Floor, Name)

VALUES

(1, 1, 50000, 1, N'Хирургическое отделение'),

(2, 2, 70000, 1, N'Терапевтическое отделение'),

(3, 3, 100000, 1, N'Гастроэнтерологическое отделение'),

(4, 1, 50000, 2, N'Диагностическое отделение'),

(5, 2, 70000, 2, N'Кардиологическое отделение'),

(6, 3, 100000, 2, N'Эндокринологическое отделение'),

(7, 1, 50000, 1, N'Приемное отделение'),

(8, 2, 70000, 1, N'Неврологическое отделение'),

(9, 3, 100000, 2, N'Нефрологическое отделение'),

(10, 1, 50000, 2, N'Пульмонологическое отделение');

■ Заболевания (**Diseases**). Идентификатор (**Id**) - уникальный идентификатор заболевания, тип данных — **int**, **первичный ключ**. Название (**Name**) - название заболевания, тип данных — **nvarchar(100)**, не может содержать **null**-значения. Степень тяжести (**Severity**) - степень тяжести заболевания, тип данных — **int**, не может содержать **null**-значения.

CREATE TABLE Diseases

(

Id INT PRIMARY KEY,

Name NVARCHAR(100) NOT NULL,

Severity INT NOT NULL

);

Заполнить таблицу информацией.

INSERT INTO Diseases (Id, Name, Severity)

VALUES

(1, N'Аппендицит', 1),

(2, N'Гайморит', 2),

(3, N'Гастрит', 3),

(4, N'Анемия', 1),

(5, N'Атеросклероз', 2),

(6, N'Гипотиреоз', 3),

(7, N'Гепатит', 1),

(8, N'Невралгия', 2),

(9, N'Пиелонефрит', 3),

(10, N'Пневмония', 1);

■ Врачи (**Doctors**). Идентификатор (**Id**) - уникальный идентификатор врача, тип данных — **int**, **первичный ключ**. Имя (**Name**) - имя врача, тип данных — **nvarchar(max)**, не может содержать **null**-значения. Телефон (**Phone**) - телефонный номер врача, тип данных — **char(10)**, может содержать **null**-значения. Премия (**Premium**) - премия врача, тип данных — **money**, не может содержать **null**-значения. Ставка (**Salary**) - ставка врача, тип данных — **money**, не может содержать **null**-значения. Фамилия (**Surname**) - фамилия врача, тип данных — **nvarchar(max)**, не может содержать **null**-значения.

CREATE TABLE Doctors

(

Id INT PRIMARY KEY,

Name NVARCHAR(MAX) NOT NULL,

Phone CHAR(10) NOT NULL,

Premium MONEY NOT NULL,

Salary MONEY NOT NULL,

Surname NVARCHAR(MAX) NOT NULL

);

Заполнить таблицу информацией.

INSERT INTO Doctors (Id, Name, Phone, Premium, Salary, Surname)

VALUES

(1, N'Михаил', '254569875', 300, 2500, N'Иванов'),

(2, N'Алиса', '295698523', 400, 3000, N'Петрова'),

(3, N'Артем', '445698235', 500, 2400, N'Сидоров'),

(4, N'Мария', '297589632', 300, 2800, N'Смирнова'),

(5, N'Александр', '257894562', 400, 2100, N'Кузнецов'),

(6, N'Ева', '445647895', 500, 3200, N'Соколова'),

(7, N'Марк', '295689452', 300, 3500, N'Михайлов'),

(8, N'Ксения', '445698745', 400, 1800, N'Иваненко'),

(9, N'Матвей', '258965234', 500, 2000, N'Петренко'),

(10, N'София', '258569874', 300, 3600, N'Новикова');

■ Обследования (**Examinations**). Идентификатор (**Id**) - уникальный идентификатор обследования, тип данных — **int**, **первичный ключ**. День недели (**DayOfWeek**) - день недели, в который проводится обследование, тип данных — **int**, не может содержать **null**-значения. Время завершения (**EndTime**) - время завершения обследования, тип данных — **time**, не может содержать **null**-значения. Название (**Name**) - название обследования, тип данных — **nvarchar(100)**, не может содержать **null**-значения. Время начала (**StartTime**) - время начала обследования, тип данных — **time**, не может содержать **null**-значения.

CREATE TABLE Examinations

(

Id INT PRIMARY KEY,

DayOfWeek INT NOT NULL,

EndTime TIME NOT NULL,

Name NVARCHAR(100) NOT NULL,

StartTime TIME NOT NULL

);

Заполнить таблицу информацией.

INSERT INTO Examinations (Id, DayOfWeek, EndTime, Name, StartTime)

VALUES

(1, 1, '09:00:00', N'Электрокардиограмма', '08:00:00'),

(2, 3, '10:00:00', N'Ультразвуковая диагностика', '09:00:00'),

(3, 5, '11:00:00', N'Биохимический анализ крови', '10:00:00'),

(4, 2, '12:00:00', N'Рентгенологическое исследование', '11:00:00'),

(5, 4, '13:00:00', N'Компьютерная томография', '12:00:00'),

(6, 6, '14:00:00', N'Эндоскопическая диагностика', '13:00:00'),

(7, 7, '15:00:00', N'Клинико-диагностические исследования', '14:00:00'),

(8, 1, '16:00:00', N'Флюорография', '15:00:00'),

(9, 3, '17:00:00', N'Молекулярно-генетические исследования', '16:00:00'),

(10, 5, '18:00:00', N'Функциональная диагностика', '17:00:00');

■ Палаты (**Wards**). Идентификатор (**Id**) - уникальный идентификатор палаты, тип данных — **int**, **первичный ключ**. Корпус (**Building**) - номер корпуса, в котором располагается палата, тип данных — **int**, не может содержать **null**-значения. Этаж (**Floor**) – этаж палаты, тип данных — **nvarchar(100)**, не может содержать **null**-значения. Название (**Name**) - название палаты, тип данных — **nvarchar(20)**, не может содержать **null**-значения.

CREATE TABLE Wards

(

Id INT PRIMARY KEY,

Building INT NOT NULL,

Floor INT NOT NULL,

Name NVARCHAR(20) NOT NULL

);

Заполнить таблицу информацией.

INSERT INTO Wards (Id, Building, Floor, Name)

VALUES

(1, 1, 1, N'Электрокардиограмма'),

(2, 2, 1, N'Ультразвуковая диагностика'),

(3, 3, 1, N'Биохимический анализ крови'),

(4, 1, 2, N'Рентгенологическое исследование'),

(5, 2, 2, N'Компьютерная томография'),

(6, 3, 2, N'Эндоскопическая диагностика'),

(7, 1, 1, N'Клинико-диагностические исследования'),

(8, 2, 1, N'Флюорография'),

(9, 3, 1, N'Молекулярно-генетические исследования'),

(10, 1, 2, N'Функциональная диагностика');

Весь листинг

DROP DATABASE IF EXISTS Hospital;

CREATE DATABASE Hospital;

USE Hospital;

CREATE TABLE Departments

(

Id INT PRIMARY KEY,

Building INT NOT NULL,

Financing MONEY NOT NULL,

Floor INT NOT NULL,

Name NVARCHAR(100) NOT NULL

);

INSERT INTO Departments (Id, Building, Financing, Floor, Name)

VALUES

(1, 1, 50000, 1, N'Хирургическое отделение'),

(2, 2, 70000, 1, N'Терапевтическое отделение'),

(3, 3, 100000, 1, N'Гастроэнтерологическое отделение'),

(4, 1, 50000, 2, N'Диагностическое отделение'),

(5, 2, 70000, 2, N'Кардиологическое отделение'),

(6, 3, 100000, 2, N'Эндокринологическое отделение'),

(7, 1, 50000, 1, N'Приемное отделение'),

(8, 2, 70000, 1, N'Неврологическое отделение'),

(9, 3, 100000, 2, N'Нефрологическое отделение'),

(10, 1, 50000, 2, N'Пульмонологическое отделение');

CREATE TABLE Diseases

(

Id INT PRIMARY KEY,

Name NVARCHAR(100) NOT NULL,

Severity INT NOT NULL

);

INSERT INTO Diseases (Id, Name, Severity)

VALUES

(1, N'Аппендицит', 1),

(2, N'Гайморит', 2),

(3, N'Гастрит', 3),

(4, N'Анемия', 1),

(5, N'Атеросклероз', 2),

(6, N'Гипотиреоз', 3),

(7, N'Гепатит', 1),

(8, N'Невралгия', 2),

(9, N'Пиелонефрит', 3),

(10, N'Пневмония', 1);

CREATE TABLE Doctors

(

Id INT PRIMARY KEY,

Name NVARCHAR(MAX) NOT NULL,

Phone CHAR(10) NOT NULL,

Premium MONEY NOT NULL,

Salary MONEY NOT NULL,

Surname NVARCHAR(MAX) NOT NULL

);

INSERT INTO Doctors (Id, Name, Phone, Premium, Salary, Surname)

VALUES

(1, N'Михаил', '254569875', 300, 2500, N'Иванов'),

(2, N'Алиса', '295698523', 400, 3000, N'Петрова'),

(3, N'Артем', '445698235', 500, 2400, N'Сидоров'),

(4, N'Мария', '297589632', 300, 2800, N'Смирнова'),

(5, N'Александр', '257894562', 400, 2100, N'Кузнецов'),

(6, N'Ева', '445647895', 500, 3200, N'Соколова'),

(7, N'Марк', '295689452', 300, 3500, N'Михайлов'),

(8, N'Ксения', '445698745', 400, 1800, N'Иваненко'),

(9, N'Матвей', '258965234', 500, 2000, N'Петренко'),

(10, N'София', '258569874', 300, 3600, N'Новикова');

CREATE TABLE Examinations

(

Id INT PRIMARY KEY,

DayOfWeek INT NOT NULL,

EndTime TIME NOT NULL,

Name NVARCHAR(100) NOT NULL,

StartTime TIME NOT NULL

);

INSERT INTO Examinations (Id, DayOfWeek, EndTime, Name, StartTime)

VALUES

(1, 1, '09:00:00', N'Электрокардиограмма', '08:00:00'),

(2, 3, '10:00:00', N'Ультразвуковая диагностика', '09:00:00'),

(3, 5, '11:00:00', N'Биохимический анализ крови', '10:00:00'),

(4, 2, '12:00:00', N'Рентгенологическое исследование', '11:00:00'),

(5, 4, '13:00:00', N'Компьютерная томография', '12:00:00'),

(6, 6, '14:00:00', N'Эндоскопическая диагностика', '13:00:00'),

(7, 7, '15:00:00', N'Клинико-диагностические исследования', '14:00:00'),

(8, 1, '16:00:00', N'Флюорография', '15:00:00'),

(9, 3, '17:00:00', N'Молекулярно-генетические исследования', '16:00:00'),

(10, 5, '18:00:00', N'Функциональная диагностика', '17:00:00');

CREATE TABLE Wards

(

Id INT PRIMARY KEY,

Building INT NOT NULL,

Floor INT NOT NULL,

Name NVARCHAR(20) NOT NULL

);

INSERT INTO Wards (Id, Building, Floor, Name)

VALUES

(1, 1, 1, N'Электрокардиограмма'),

(2, 2, 1, N'Ультразвуковая диагностика'),

(3, 3, 1, N'Биохимический анализ крови'),

(4, 1, 2, N'Рентгенологическое исследование'),

(5, 2, 2, N'Компьютерная томография'),

(6, 3, 2, N'Эндоскопическая диагностика'),

(7, 1, 1, N'Клинико-диагностические исследования'),

(8, 2, 1, N'Флюорография'),

(9, 3, 1, N'Молекулярно-генетические исследования'),

(10, 1, 2, N'Функциональная диагностика');

**Задания по запросам, необходимо сохранить для проверки**

1. Вывести содержимое таблицы палат.

|  |  |
| --- | --- |
| Выборка данных из таблицы | USE teststep  SELECT \*  FROM students; |

1. Вывести фамилии и телефоны всех врачей.

|  |  |
| --- | --- |
| Выборка данных из таблицы | USE teststep  SELECT id, fio  FROM students; |

1. Вывести все этажи без повторений, на которых располагаются палаты.

|  |  |
| --- | --- |
| Выборка уникальных данных из таблицы | USE teststep  SELECT DISTINCT \*  FROM students; |

1. Вывести названия заболеваний под именем “Name of Disease” и степень их тяжести под именем “Severity of Disease”.

|  |  |
| --- | --- |
| Выборка данных из таблицы с псевдонимами | USE teststep  SELECT id AS ИД, fio AS ФИО  FROM students; |

1. Использовать выражение FROM для любых трех таблиц базы данных, используя для них псевдонимы.

|  |  |
| --- | --- |
| Выборка данных из таблицы с псевдонимами | USE teststep  SELECT id AS ИД, fio AS ФИО  FROM students; |

1. Вывести названия отделений, расположенных в корпусе 5 и имеющих фонд финансирования менее 30000.

|  |  |
| --- | --- |
| Выборка данных из таблицы по составному условию | USE teststep  SELECT \*  FROM students  WHERE id > 1 AND id < 3;  -- AND OR NOT = != > < <= >= |

1. Вывести названия отделений, расположенных в 3-м корпусе с фондом финансирования в диапазоне от 12000 до 15000.

|  |  |
| --- | --- |
| Выборка данных из таблицы по условию вхождения в диапазон | USE teststep  SELECT \*  FROM students  WHERE id BETWEEN 1 AND 3; |

1. Вывести названия палат, расположенных в корпусах 4 и 5 на 1-м этаже.

|  |  |
| --- | --- |
| Выборка данных из таблицы по условию вхождения в множество | USE teststep  SELECT \*  FROM students  WHERE id IN (1);  -- IN NOT IN |

1. Вывести названия, корпуса и фонды финансирования отделений, расположенных в корпусах 3 или 6 и имеющих фонд финансирования меньше 11000 или больше 25000.

|  |  |
| --- | --- |
| Выборка данных из таблицы по составному условию | USE teststep  SELECT \*  FROM students  WHERE id > 1 AND id < 3;  -- AND OR NOT = != > < <= >= |
| Выборка данных из таблицы по условию вхождения в множество | USE teststep  SELECT \*  FROM students  WHERE id IN (1);  -- IN NOT IN |

1. Вывести фамилии врачей, чья зарплата (сумма ставки и надбавки) превышает 1500.

|  |  |
| --- | --- |
| Выборка данных из таблицы по условию | USE teststep  SELECT \*  FROM students  WHERE id = 1; |

1. Вывести фамилии врачей, у которых половина зарплаты превышает троекратную надбавку.

|  |  |
| --- | --- |
| Выборка данных из таблицы по условию | USE teststep  SELECT \*  FROM students  WHERE id = 1; |

1. Вывести названия обследований без повторений, проводимых в первые три дня недели с 12:00 до 15:00.

|  |  |
| --- | --- |
| Выборка уникальных данных из таблицы | USE teststep  SELECT DISTINCT \*  FROM students; |
| Выборка данных из таблицы по условию вхождения в множество | USE teststep  SELECT \*  FROM students  WHERE id IN (1);  -- IN NOT IN |
| Выборка данных из таблицы по условию вхождения в диапазон | USE teststep  SELECT \*  FROM students  WHERE id BETWEEN 1 AND 3; |

1. Вывести названия и номера корпусов отделений, расположенных в корпусах 1, 3, 8 или 10.

|  |  |
| --- | --- |
| Выборка данных из таблицы по условию вхождения в множество | USE teststep  SELECT \*  FROM students  WHERE id IN (1);  -- IN NOT IN |

1. Вывести названия заболеваний всех степеней тяжести, кроме 1-й и 2-й.

|  |  |
| --- | --- |
| Выборка данных из таблицы по условию вхождения в множество | USE teststep  SELECT \*  FROM students  WHERE id IN (1);  -- IN NOT IN |

1. Вывести названия отделений, которые не располагаются в 1-м или 3-м корпусе.

|  |  |
| --- | --- |
| Выборка данных из таблицы по условию вхождения в множество | USE teststep  SELECT \*  FROM students  WHERE id IN (1);  -- IN NOT IN |

1. Вывести названия отделений, которые располагаются в 1-м или 3-м корпусе.

|  |  |
| --- | --- |
| Выборка данных из таблицы по условию вхождения в множество | USE teststep  SELECT \*  FROM students  WHERE id IN (1);  -- IN NOT IN |

1. Вывести фамилии врачей, начинающиеся на букву “N”.

|  |  |
| --- | --- |
| Выборка данных из таблицы по условию вхождения в фильтр | USE teststep  SELECT \*  FROM students  WHERE fio LIKE N'%Иван%';  -- % \_ |
| Выборка данных из таблицы по условию вхождения в фильтр регулярного выражения | USE teststep  SELECT \*  FROM students  WHERE id LIKE '[1-2]';  -- [abc] [!abc] [начало-конец] |

1. Вывести 5 первых отделений из таблицы Отделения.

|  |  |
| --- | --- |
| Выборка ограниченного количества записей (строк) из таблицы | USE teststep  SELECT TOP 2 \*  FROM students; |

1. Вывести таблицу с информацией о докторах с сортировкой по убыванию.

|  |  |
| --- | --- |
| Выборка данных из таблицы с сортировкой | USE teststep  SELECT \*  FROM students  ORDER BY id DESC;  -- ASC DESC |

1. Вывести из каждой таблицы 3 первых уникальных значения с псевдонимами, условием (исходя из данных таблицы), сортировкой по убыванию.

|  |  |
| --- | --- |
| Выборка уникальных данных из таблицы | USE teststep  SELECT DISTINCT \*  FROM students; |
| Выборка ограниченного количества записей (строк) из таблицы | USE teststep  SELECT TOP 2 \*  FROM students; |
| Выборка данных из таблицы с псевдонимами | USE teststep  SELECT id AS ИД, fio AS ФИО  FROM students; |
| Выборка данных из таблицы по составному условию | USE teststep  SELECT \*  FROM students  WHERE id > 1 AND id < 3; |
| Выборка данных из таблицы с сортировкой | USE teststep  SELECT \*  FROM students  ORDER BY id DESC; |