

**ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”
Факультет компьютерных наук**

База данных медицинского центра в ErWin

Яковлева Ю.А.

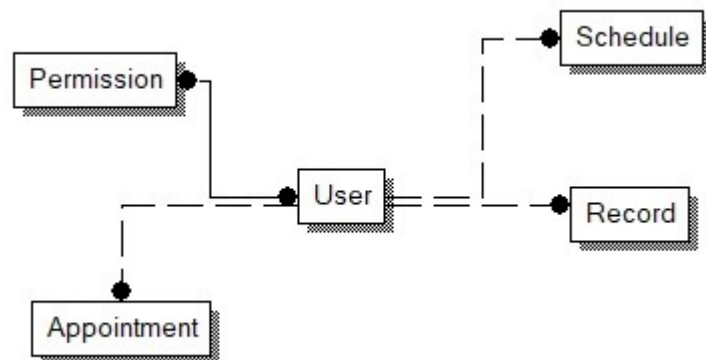
Воронеж 2019

Оглавление

Диаграмма сущность-связь.....	3
Модель данных, основанная на ключах.....	4
Полная атрибутивная модель.....	5
Трансформационная модель.....	7
Модель СУБД.....	8

Диаграмма сущность-связь

Диаграмма сущность-связь представлена на рисунке ниже:



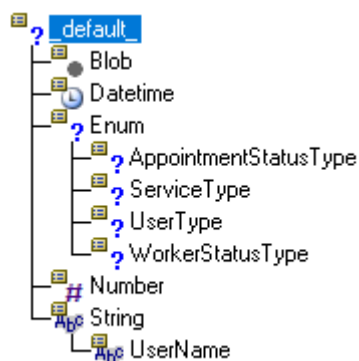
На данной схеме представлены следующие сущности:

- User - пользователь системы
- Permissions - права пользователя в рамках системы
- Appointment - информация о записи клиента на прием к врачу или на сдачу анализов
- Record - результат обследования клиента
- Schedule - расписание пользователей

Связи между сущностями:

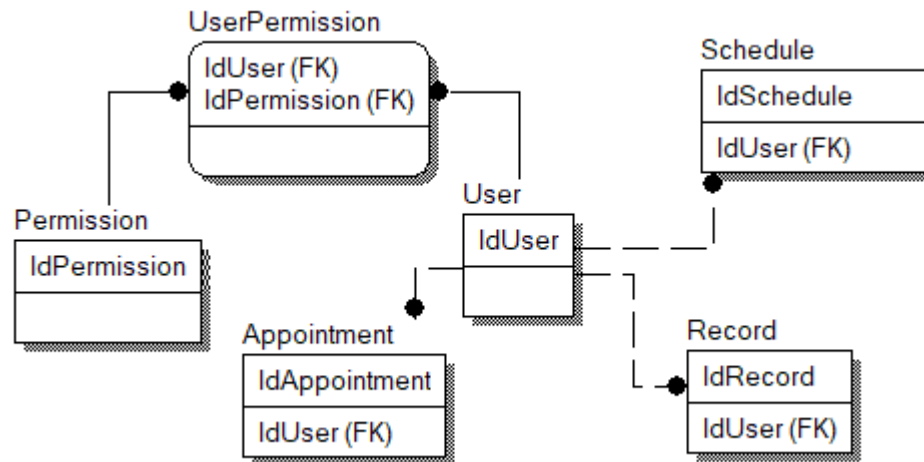
- Как пользователь может иметь несколько разрешений, так и одно и то же разрешение может быть у нескольких пользователей.
- Пользователь может иметь несколько записей на прием
- Пользователь может иметь несколько результатов обследования
- Пользователь может иметь несколько записей расписаний, по которым он работает

Определены были следующие домены:



Модель данных, основанная на ключах

Модель, основанная на ключах, представлена на рисунке ниже:



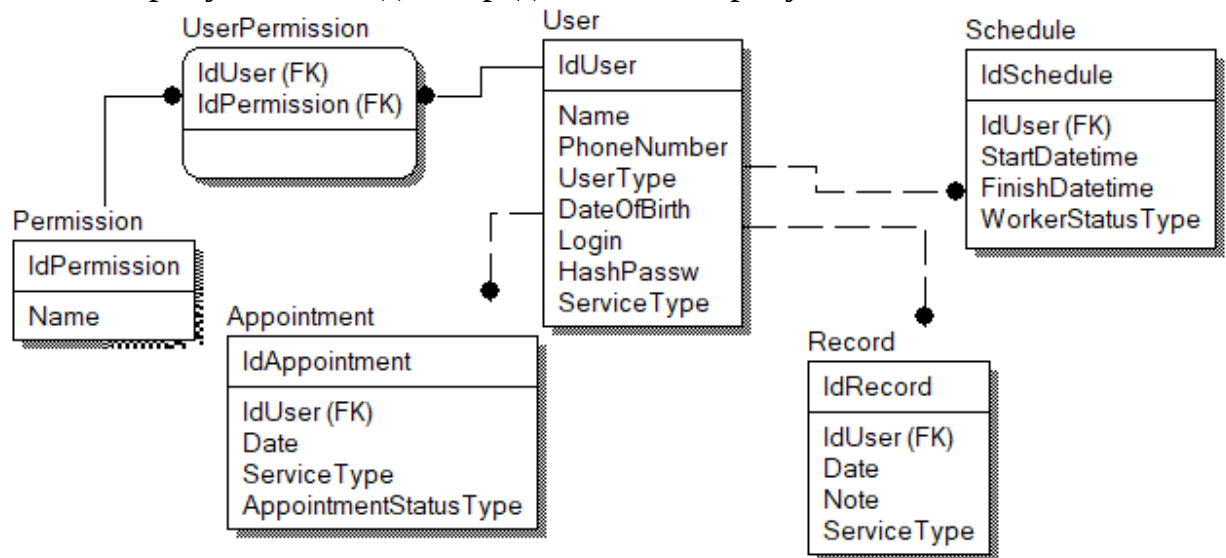
На данной схеме связь многие ко многим между сущностями **Permission** и **User** была заменена двумя связями с новой сущностью **UserPermission**, которая сопоставляет пользователям их права.

На данной схеме были добавлены следующие ключи:

- `IdUser` - первичный ключ сущности **User**
- `IdPermission` - первичный ключ сущности **Permission**
- `IdAppointment` - первичный ключ сущности **Appointment**
- `IdRecord` - первичный ключ сущности **Record**
- `IdSchedule` - первичный ключ сущности **Schedule**
- `IdUser, IdPermission` - составной первичный ключ сущности **UserPermission**, части которого также являются и внешними ключами на первичные ключи `IdUser` и `IdPermission` сущностей **User** и **Permission** соответственно
- `IdUser` - внешний ключ для сущностей **Appointment**, **Record**, **Schedule**, ссылающийся на первичный ключ `IdUser` сущности **User**

Полная атрибутивная модель

Полная атрибутивная модель представлена на рисунке ниже:



На данной схеме к сущностям были добавлены атрибуты .

Атрибуты сущности User

- Name - ФИО пользователя
- PhoneNumber - номер телефона пользователя
- UserType - тип пользователя
- DateOfBirth - дата рождения пользователя
- Login - имя учётной записи, под которой пользователь входит в систему
- HashPassw - хеш пароля учётной записи, под которой пользователь входит в систему
- ServiceType - тип услуги

Атрибуты сущности Permission:

- Name - название роли

Атрибуты сущности Appointment:

- Date - дата записи
- ServiceType - тип услуги
- AppointmentStatusType - статус записи

Атрибуты сущности Record:

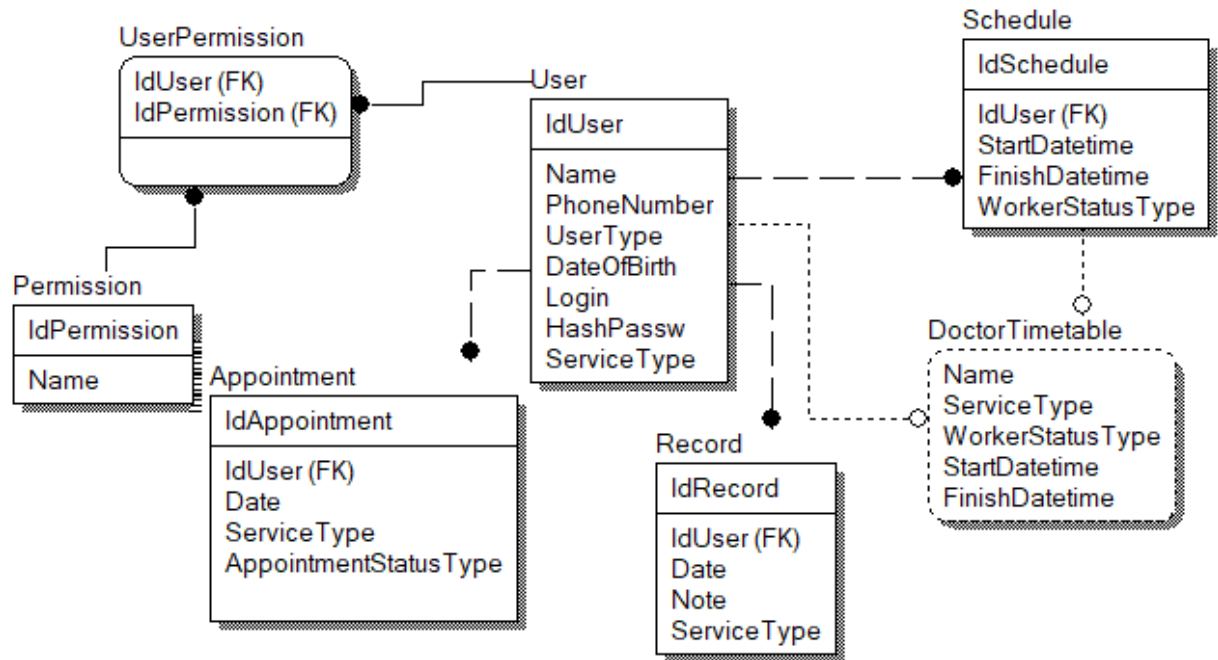
- Date - дата обследования
- Note - информация об обследовании
- ServiceType - тип услуги

Атрибуты сущности Schedule:

- StartDatetime - дата и время начала периода
- FinishDateTime - дата и время окончания периода
- WorkerStatusType - статус в пределах временного интервала

Трансформационная модель

Трансформационная модель представлена ниже:



На данной схеме добавлено представление DoctorTimetable, которое представляет собой расписание врача и содержит:

- Name - имя врача
- ServiceType - тип услуги
- Worker StatusType - статус в пределах временного интервала
- StartDatetime - дата и время начала периода
- FinishDate Time - дата и время окончания периода

Модель СУБД

В качестве СУБД был выбран MySQL 5.0.

Для генерации созданной модели данных был получен следующий скрипт:

```
CREATE TABLE Appointment
(
    IdAppointment INTEGER NOT NULL,
    IdUser INTEGER NOT NULL,
    Date DATE NOT NULL,
    ServiceType ENUM('lor', 'terapevt', 'lab', 'hirurg',
'nevrop', 'endokr', 'gastro') NOT NULL,
    AppointmentStatusType ENUM('cancelled', 'planned', 'ongoing') NOT
NULL
)
;
ALTER TABLE Appointment
    ADD PRIMARY KEY (IdAppointment)
;
CREATE TABLE Permission
(
    IdPermission INTEGER NOT NULL,
    Name VARCHAR(20) NOT NULL
)
;
ALTER TABLE Permission
    ADD PRIMARY KEY (IdPermission)
;
CREATE TABLE Record
(
    IdRecord INTEGER NOT NULL,
    IdUser INTEGER NOT NULL,
    Date DATE NOT NULL,
    Note BLOB NOT NULL,
    ServiceType ENUM('lor', 'terapevt', 'lab', 'hirurg',
'nevrop', 'endokr', 'gastro') NOT NULL
)
;
ALTER TABLE Record
    ADD PRIMARY KEY (IdRecord)
;
CREATE TABLE Schedule
(
    IdSchedule INTEGER NOT NULL,
    IdUser INTEGER NOT NULL,
    StartDatetime DATE NOT NULL,
    FinishDatetime DATE NOT NULL,
    WorkerStatusType ENUM('office', 'vacation') NOT NULL
)
;
ALTER TABLE Schedule
    ADD PRIMARY KEY (IdSchedule)
;
```



```

CREATE TABLE User
(
    IdUser    INTEGER NOT NULL,
    Name      VARCHAR(200) NOT NULL,
    PhoneNumber VARCHAR(20) NOT NULL,
    UserType  ENUM('patient', 'doctor', 'lab', 'rec') NOT NULL,
    DateOfBirth DATE NOT NULL,
    Login     VARCHAR(20) NOT NULL,
    HashPassw VARCHAR(20) NOT NULL,
    ServiceType ENUM('lor', 'terapevt', 'lab', 'hirurg',
'nevrop', 'endokr', 'gastro') NOT NULL
)
;
ALTER TABLE User
    ADD PRIMARY KEY (IdUser)
;
CREATE TABLE UserPermission
(
    IdUser    INTEGER NOT NULL,
    IdPermission INTEGER NOT NULL
)
;
ALTER TABLE UserPermission
    ADD PRIMARY KEY (IdUser, IdPermission)
;
CREATE VIEW DoctorTimetable AS
    SELECT
        User.Name, User.ServiceType, Schedule.WorkerStatusType, Schedule.StartDat
etime, Schedule.FinishDatetime
        FROM Schedule, User
;
ALTER TABLE Appointment
    ADD FOREIGN KEY R_3 (IdUser) REFERENCES User(IdUser)
;
ALTER TABLE Record
    ADD FOREIGN KEY R_4 (IdUser) REFERENCES User(IdUser)
;
ALTER TABLE Schedule
    ADD FOREIGN KEY R_2 (IdUser) REFERENCES User(IdUser)
;
ALTER TABLE UserPermission
    ADD FOREIGN KEY R_6 (IdUser) REFERENCES User(IdUser)
;
ALTER TABLE UserPermission
    ADD FOREIGN KEY R_5 (IdPermission) REFERENCES
Permission(IdPermission)
;

```

Полученная с помощью обратной генерации схема выглядит следующим образом:

