|  |
| --- |
| Карагандинский технический университет имени Абылкаса Сагинова  Кафедра ИВС  **Лабораторная работа №3**  По дисциплине: "Системы управления базами данных"  *(наименование дисциплины)*  Тема: Создание отношений и диаграммы базы данных. Ввод данных.    **Преподаватель:**  Нуртай М.Д.  *(фамилия, инициалы)*  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *(подпись) (дата)*  **Выполнил:**  ст.гр. ITM-20-1 Правдин П.П  *(фамилия, инициалы)*  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *(подпись) (дата)*  Караганда 2022 |

**Цель работы:** освоить способы создания отношений и диаграммы отношений базы данных средствами СУБД MS SQL Server 2012.

**Индивидуальные задания**

Краткое описание

Организация “Presentation

For you” профессионально занимается подготовкой и проведением презентаций для фирм. В фирме за последние несколько кварталов сильно увеличился объем заказов. В результате постоянно стали наблюдаться ситуации, когда презентации задерживались из-за нехватки каких-либо ресурсов (аудиторий, проекторов, досок).В фирме были проведены исследования и было установлено, что ситуация сильно улучшится, если у фирмы появится электронная система распределения ресурсов, а не бумажная как это было раньше. Полная постановка задачи. Предполагается, что система будет функционировать на сервере, к ней будут подключаться клиенты для резервирования ресурсов на определенное время. Объекты системы: сервер, ресурс, расписание использования ресурса, пользователь, администратор,

менеджер ресурсов, клиент

.Сервер: Обладает следующей функциональностью:

• Хранит информацию обо всех ресурсах и пользователях.

• Позволяет управлять пользовательскими записями:

oДобавлять.

oУдалять.

oНазначать уровень доступа.

• Разделяет уровень доступа для различных пользователей на основе ролевых кластеров:

oАдминистратор.

oМенеджер.

oПользователь.

• Выдает информацию о ресурсах в соответствии с уровнем доступа.

Ресурс: название, серийный номер (номер аудитории, номер доски), расписание использования

ресурса.

Расписание использования ресурсов: порождается для каждого ресурса. Включается записи о

времени занятости и цели использования.

Пользователь: ФИО, логин, пароль, информация о дополнительных ролях. Дополнительных

ролей две:

• Администратор.

• Менеджер ресурсов.

У пользователя должны быть следующие функции:

• Запрос на занятие ресурса на определенное время с указанной целью.

• Различные виды просмотров занятости ресурсов.

oКонкретного ресурса.

oГруппы ресурсов.

Администратор:

Выполняет функции менеджера пользователей. Не может управлять ресурсами.

Менеджер ресурсов: Основные функции:

• Добавление и удаление ресурсов.

• Подтверждение или отклонение запросов на занятие ресурсов.

Клиент: Должен быть реализован в виде web-сайта и Windows приложения.

**Листинг кода**

use Pavel

create table Roles(

Id int primary key,

RoleName nvarchar(20),

)

create table Users(

Id int primary key,

FirstName nvarchar(30),

LastName nvarchar(30),

UserLogin nvarchar(30),

UserPassword nvarchar(30),

RoleId int, foreign key(RoleId) references Roles(Id),

)

insert into Users(Id,FirstName,LastName,UserLogin,UserPassword,RoleId)

values

(1,'Alex','Ungefug','X\_1','111',1),

(2,'Bob','Rumor','BR2','222',2),

(3,'Liza','Schmidt','Lis12','333',3),

(4,'Morisa','White','Snow12','444',3)

create table Rooms(

Id int primary key,

RoomNumber int,

DeskNumber int,

SerialNumber int,

KeyHolderId int, foreign key(KeyHolderId) references Users(Id),

Schedule time,

)

insert into Rooms(Id,RoomNumber,DeskNumber,SerialNumber,KeyHolderId,Schedule)

values

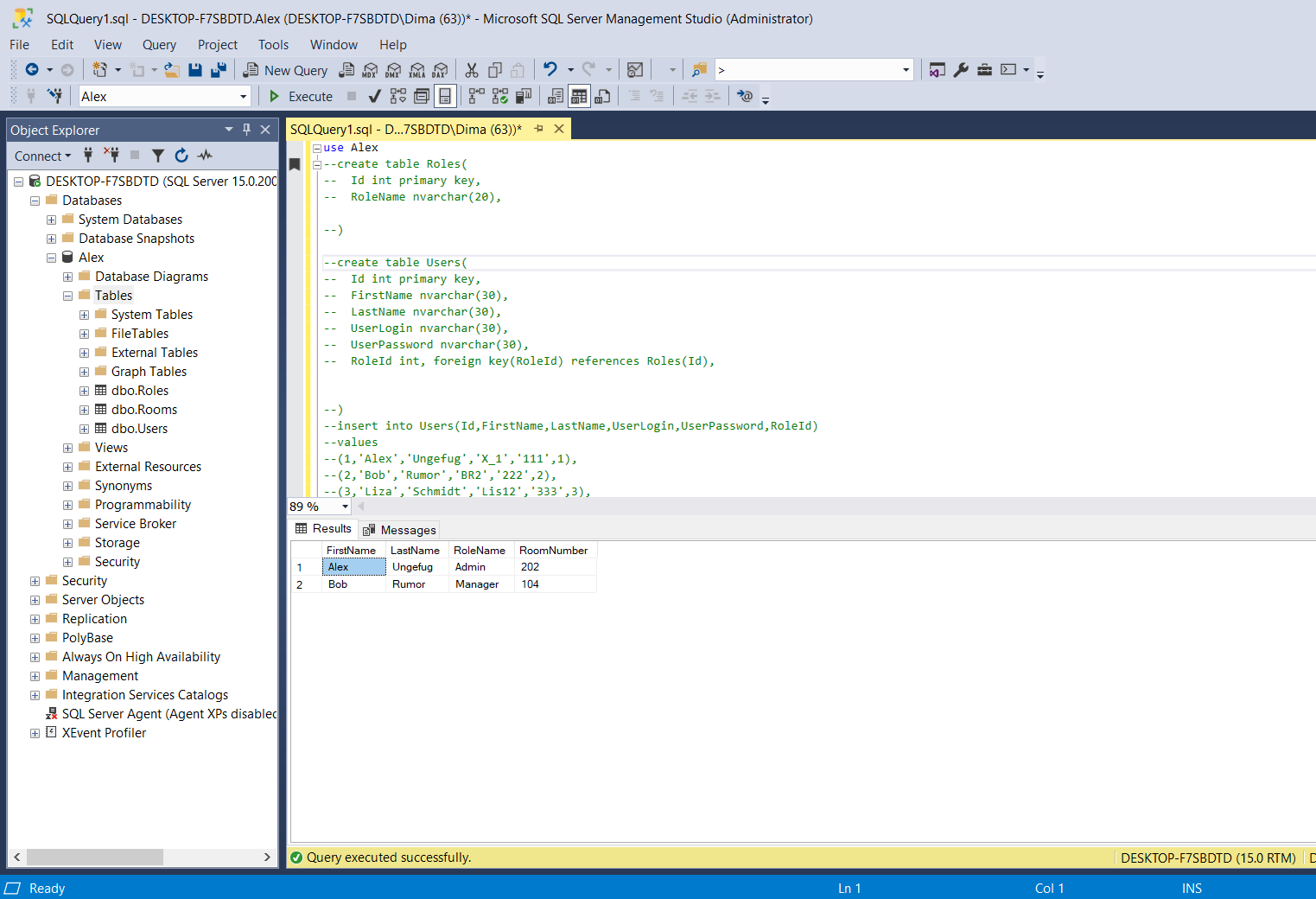
(3,104,7,9771,2,'21:00')

select FirstName,LastName,RoleName,RoomNumber from Users

join Roles on Users.RoleId =Roles.Id

join Rooms on Rooms.KeyHolderId=Users.Id

**Скриншот работы**



**Контрольные вопросы**

1. Назначение механизма отношений. Виды отношений?

Связи делятся на:

1. Многие ко многим.
2. Один ко многим.
   * с обязательной связью;
   * с необязательной связью;
3. Один к одному.
   * с обязательной связью;
   * с необязательной связью;
4. Способы создания отношений

Для этого существуют методы:

INNER JOIN

LEFT JOIN

RIGHT JOIN

FULL OUTER JOIN

1. Ссылочная целостность данных. Способы ее поддержания?

Ссылочная целостность—это **состояние реляционной базы данных в которой записи не могут ссылаться на несуществующие записи в этой базе данных**. FOREIGN KEY—особый вид ограничения (constraint) MySQL, которое позволяет предотвратить нарушение ссылочной целостности при удалении/изменении информации в таблицах предках. Поддержка FOREIGN KEY поддерживается только для таблиц типа InnoDB.

1. Способы отображения зависимостей между таблицами БД

При помощи диаграмм или особых запросов методом SELECT

1. Способы ввода данных в таблицы. Ограничения целостности при вводе данных

Ввод данных в таблицу осуществляется при помощи команды INSERT INTO

1. Удаление данных, удаление таблиц

Для удаления таблицы применяют команду DROP TABLE <table name>

Для удаления данных из таблиц DELETE FROM <table name> WHERE <условие>