

КУРСОВАЯ РАБОТА

Проектирование и реализация базы
данных “Страховая компания”

Выполнил : Злодеев Э.М.

ЦЕЛИ КУРСОВОЙ РАБОТЫ

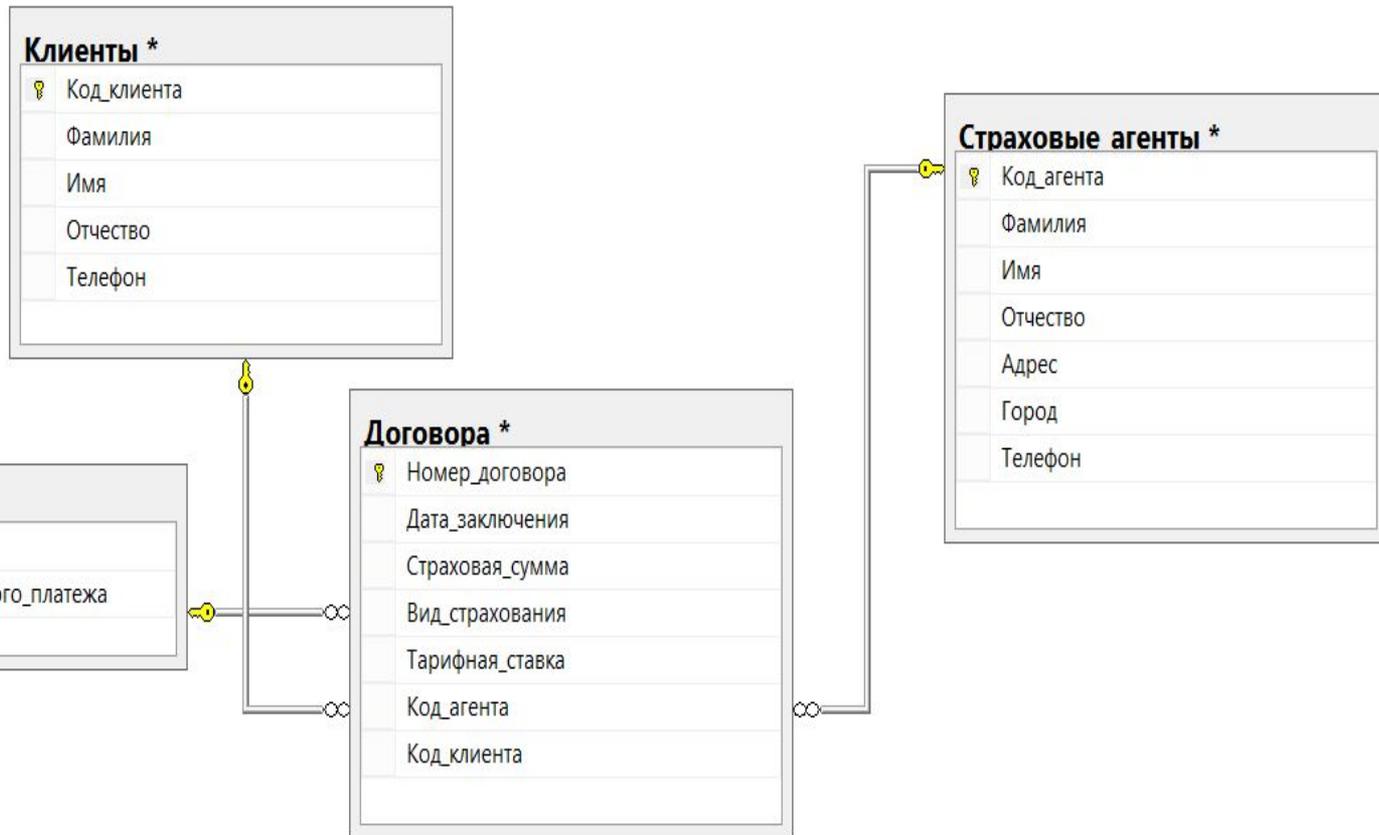
Разработать базу данных для страховой компании , которая будет хранить информацию :

- О страховых агентах
- О клиентах
- О договорах

И выполнять задачи по

- Расчету заработной платы страховым агентам компании

ДИАГРАММА БАЗЫ ДАННЫХ



СОЗДАНИЕ БАЗЫ ДАННЫХ

The screenshot displays the SQL Server Enterprise Manager interface. On the left, the 'Обозреватель объектов' (Object Explorer) pane shows the server instance 'LENOVO-PC\SQLEXPRESS (SQL Server 11)'. Under the 'Базы данных' (Databases) folder, the 'СтраховаяКомпания' database is highlighted. The main pane shows the 'SQLQuery2.sql' script with the following T-SQL code:

```
CREATE DATABASE СтраховаяКомпания
ON PRIMARY
(NAME = 'СтраховаяКомпания', FILENAME='D:\ЛичнаяПапка\СтраховаяКомпания.mdf',
SIZE = 3072kb,
MAXSIZE = UNLIMITED,
FILEGROWTH = 1024kb)
LOG ON
(NAME = 'СтраховаяКомпания_log', FILENAME='D:\ЛичнаяПапка\СтраховаяКомпания_log.ldf',
SIZE = 3072kb,
MAXSIZE = UNLIMITED,
FILEGROWTH = 1024kb)
```

СОЗДАНИЕ ТАБЛИЦ

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Tables' folder is expanded under the 'СтраховаяКомпания' database. On the right, the SQL Query window displays the following T-SQL scripts:

```
CREATE TABLE Страховые_агенты
(
    Шифр_агента int not null,
    Фамилия nvarchar(40) not null,
    Имя nvarchar(20) null,
    Отчество nvarchar(30) null,
    Адрес nvarchar(100) null,
    Город nvarchar(40) default ('Челябинск') null,
    Телефон int not null,
    Primary key (Шифр_агента))

CREATE TABLE Договора
(
    Номер_договора int unique not null,
    Дата_заключения datetime default Getdate() null,
    Страховая_сумма money null,
    Вид_страхования nvarchar(40) not null,
    Тарифная_ставка int null,
    Шифр_агента int not null,
    Шифр_клиента int not null,
    Primary key (Номер_договора),
    Foreign key (Шифр_агента) References Страховые_агенты (Шифр_агента),
    Foreign key (Шифр_клиента) References Клиенты (Шифр_клиента),
    Foreign key (Вид_страхования) References Страхование (Вид_страхования))

CREATE TABLE Страхование
(
    Вид_страхования nvarchar(40) not null,
    Процент_от_страхового_платежа int null,
    Primary key (Вид_страхования))

CREATE TABLE Клиенты
(
    Шифр_клиента int not null,
    Фамилия nvarchar(40) not null,
    Имя nvarchar(20) null,
    Отчество nvarchar(30) null,
    Телефон int null,
    Primary key (Шифр_клиента))
```

ЗАПОЛНЕНИЕ БАЗЫ ДАННЫХ

```
Use СтраховаяКомпания
Insert into Клиенты(Шифр_клиента, Фамилия, Имя, Отчество, Телефон)
Values ('001', 'Иванов', 'Иван', 'Иванович', '23124')
Insert into Страховые_агенты(Шифр_агента, Фамилия, Имя, Отчество, Адрес, Город, Телефон)
Values ('00001', 'Петров', 'Петр', 'Петрович', 'пр Победы д3', 'Москва', '23242')
Insert Into Страхование(Вид_страхования, Процент_от_страхового_платежа)
Values ('Медицинское страхование', '30')
```

СОЗДАНИЕ ИНДЕКСА

- Код

```
Use СтраховаяКомпания  
Create Index Мой_индекс  
on Договора(Тарифная_ставка)
```

100 %



Сообщения

Выполнение команд успешно завершено.

ТРИГГЕР

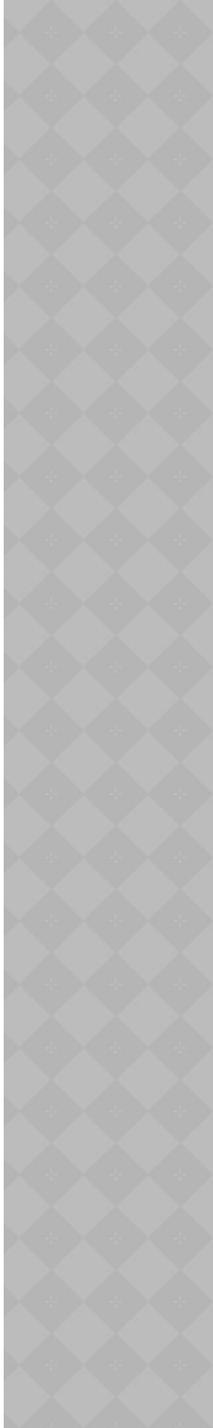
```
Use Lokimo
Go
Create Trigger Ограничение
on Страхование
After Insert
As
If DATEPART (hh, GETDATE()) Not between '07' and '11'
Begin
Print 'Нельзя ввести данные'
Rollback
End
```

100 % <

 Результаты

Выполнение команд успешно завершено.

ЗАПРОСЫ В БАЗЕ ДАННЫХ



ПРЕДСТАВЛЕНИЯ

```
[-] Create View Predstavlenie AS  
Select * From Клиенты  
Where Шифр_клиента = '001'
```

100 % <



Сообщения

Выполнение команд успешно завершено.

ХРАНИМЫЕ ПРОЦЕДУРЫ

```
Use Lokimo
Go
Create Procedure [выбор по коду]
With Execute as Caller
As
Select * From Клиенты Where Шифр_клиента = '001'
```

100 % <

Сообщения

Выполнение команд успешно завершено.

ОБЕСПЕЧЕНИЕ БЕЗОПАСНОСТИ

Processadmin. SQL Server является многозадачной средой. Это значит, что программа может выполнять несколько процессов одновременно. Например, SQL Server может породить один процесс для записи в кеш и другой процесс для считывания из кеша. Член группы processadmin может завершить (или прервать) процесс. Это еще одна хорошая роль для помощника администратора баз данных и разработчиков. Разработчикам в первую очередь необходимо прерывать процессы, инициализированные некорректным запросом или хранимой процедурой.

Dbcreator. Эти пользователи могут создавать и изменять базы данных. Такая роль подходит как помощнику администратора баз данных, так и разработчикам (которых следует предупредить о недопустимости создания ненужных баз данных, попусту занимающих место на сервере).

Diskadmin. Эти пользователи управляют файлами на диске. Они выполняют такие задачи, как зеркальное копирование баз данных и добавление новых устройств резервного копирования. Членами этой группы должны быть помощники администратора баз данных.

public. Назначение этой группы заключается в выдаче пользователям базы данных набора разрешений по умолчанию. Все пользователи базы данных автоматически объединяются в эту группу и не могут быть удалены.

db_owner. Члены этой роли могут выполнять все операции, выполняемые членами других ролей, а также некоторые административные функции

ЗАКЛЮЧЕНИЕ

В ходе выполнения курсового проекта:

Мною была создана база данных для страховой компании , которая позволяет вести