



МИНОБРНАУКИ РОССИИ
ФГБОУ ВО
РТУ МИРЭА
Колледж программирования и кибербезопасности

Отчет №4
«Задание УП»

По УП.01.01 Учебная практика

Специальность 09.02.07 Информационные системы и программирование

Выполнил студент
Группы ПКС-35
Неструев О. Д.
Проверил преподаватель
Стоколос М. Д.

Москва
2024

Цель: Изменение ER-диаграммы, изменение базы данных в соответствии с ER-диаграммой.

1. Изменение ER-диаграммы

ER-диаграмма была обновлена с использованием новой информации по предметной области. Пример новой ER-диаграммы представлен на рисунке 1.

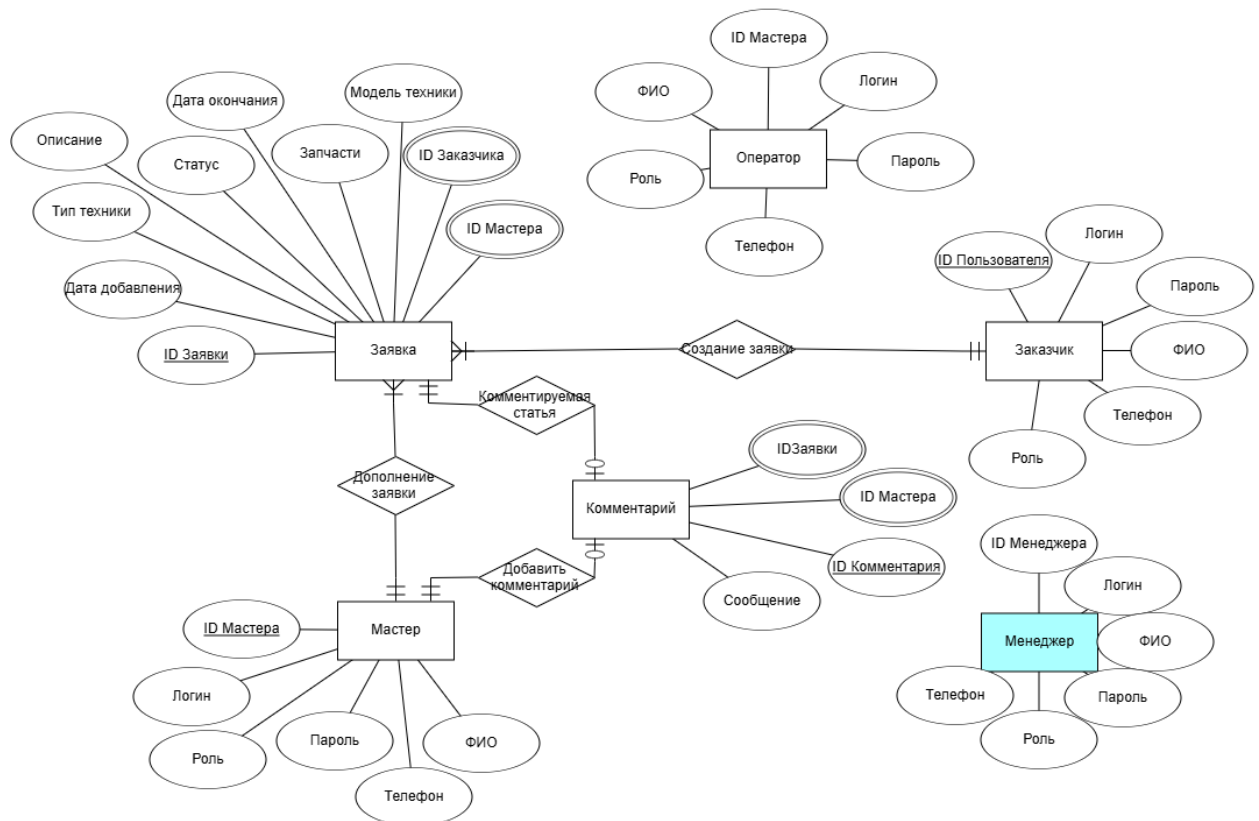


Рис. 1 Пример ER-диаграммы

При разработке ER-диаграммы было выявлено пять сущностей, такие как:

- «Заказчик»;
- «Мастер»;
- «Комментарий»;
- «Оператор»;
- «Заявка».

Сущность «Заказчик» хранит в себе следующие атрибуты:

- «ID Заказчика» (первичный ключ);

- «Логин»;
- «ФИО»;
- «Пароль»;
- «Роль»;
- «Телефон».

Сущность «Мастер» содержит в себе следующие атрибуты:

- «ID Мастера» (первичный ключ);
- «Логин»;
- «ФИО»;
- «Пароль»;
- «Роль»;
- «Телефон».

Сущность «Менеджер» содержит в себе следующие атрибуты:

- «ID Менеджера» (первичный ключ);
- «Логин»;
- «ФИО»;
- «Пароль»;
- «Роль»;
- «Телефон».

Сущность «Комментарий» содержит в себе следующие атрибуты:

- «ID Комментария» (первичный ключ);
- «Сообщение»;
- «ID Мастера» (вторичный ключ);
- «ID Заявки» (вторичный ключ).

Сущность «Оператор» содержит в себе следующие атрибуты:

- «ID Оператора» (первичный ключ);
- «Логин»;
- «ФИО»;
- «Пароль»;
- «Роль»;
- «Телефон».

Сущность «Заявка» содержит в себе следующие атрибуты:

- «ID Заявки» (первичный ключ);
- «Дата добавления»;
- «Описание»;
- «Дата окончания»;
- «Запчасти»;
- «Модель техники»;
- «Тип техники»;
- «Статус»;
- «Комментарий»;
- «ID Заказчика» (вторичный ключ от сущности «Заказчик»);
- «ID Мастера» (вторичный ключ от сущности «Мастер»).

Заказчик может создать несколько заявок, однако заявка может принадлежать только одному заказчик. Это связь «Один ко многим» между сущностями «Заказчик» и «Заявка».

Мастер может работать с несколькими заявками, однако заявка не может принадлежать нескольким мастерам. Это связь «Один ко многим» между сущностями «Мастер» и «Заявка».

Один комментарий не может принадлежать нескольким мастерам. Это связь «Один ко многим» между сущностями «Мастер» и «Комментарий».

Один комментарий не может принадлежать нескольким заявкам. Это связь «Один ко многим» между сущностями «Заявка» и «Комментарий».

Оператор отвечает за назначение Мастера на исполнение. Он не регистрируется в заявке.

2. Создание новой базы данных

Для создания базы данных использовался язык T-SQL, который позволяет создавать базы данных, таблицы и заполнять их, а также изменять и удалять записи.

Создание базы данных «RepairService» представлено в следующем Y-SQL запросе:

```
-- Создание базы данных
create database RepairService;
```

Создание таблицы «Заказчик» представлено в следующем T-SQL запросе:

```
-- Создание таблицы Заказчик
create table Заказчик(
[ID Заказчика] int primary key not null,
Телефон nchar(11) not null,
ФИО nchar(50) not null,
Логин nchar(30) not null,
Пароль nchar(10) not null,
Роль nchar(20) not null
)
```

В таблице определяется первичный ключ primary key, который позволяет различать записи. Тип данных nchar() позволяет хранить текстовую информацию.

Создание таблицы «Мастер» представлено в следующем T-SQL запросе:

```
-- Создание таблицы Мастер
```

```
create table Мастер(  
[ID Мастера] int primary key not null,  
Телефон nchar(11) not null,  
ФИО nchar(50) not null,  
Логин nchar(30) not null,  
Пароль nchar(15) not null,  
Роль nchar(20) not null  
)
```

Таблица создана по аналогии с таблицей «Заказчик».

Создание таблицы «Оператор» представлено в следующем T-SQL запросе:

```
-- Создание таблицы Оператор  
create table Оператор(  
[ID Оператора] int primary key not null,  
Телефон nchar(11) not null,  
ФИО nchar(50) not null,  
Логин nchar(30) not null,  
Пароль nchar(15) not null,  
Роль nchar(20) not null  
)
```

Таблица создана по аналогии с таблицей «Заказчик».

Создание таблицы «Менеджер» представлено в следующем T-SQL запросе:

```
-- Создание таблицы Менеджер  
create table Менеджер(  
[ID Менеджера] int primary key not null,  
Телефон nchar(11) not null,  
ФИО nchar(50) not null,  
Логин nchar(30) not null,  
Пароль nchar(15) not null,  
Роль nchar(20) not null  
)
```

Таблица создана по аналогии с таблицей «Заказчик».

Создание таблицы «Комментарий» представлена в следующем T-SQL запросе:

```
-- Хранит в себе комментарии к заявкам
create table Комментарий(
  [ID Комментария] int primary key not null,
  Сообщение nchar(300) not null,
  [ID Мастера] int not null,
  FOREIGN KEY ([ID Мастера]) REFERENCES Мастер ([ID Мастера]),
  [ID Заявки] int not null,
  FOREIGN KEY ([ID Заявки]) REFERENCES Заявка ([ID Заявки])
)
```

Тип данных int позволяет хранить в себе целочисленные данные.

Таблица хранит в себе вторичные ключи от таблиц «Мастер» и «Заявка»

Создание таблицы «Заявка» представлено в следующем T-SQL зпросе:

```
create table Заявка(
  [ID Заявки] int primary key not null,
  [Дата добавления] date not null,
  [Тип техники] nchar(30) not null,
  [Модель техники] nchar(100),
  Описание nchar(300),
  Статус nchar(20) not null,
  [Дата окончания] date,
  Запчасти nchar(200),

  [ID Мастера] int,
  -- Вторичный ключ на ID мастера
  FOREIGN KEY ([ID Мастера]) REFERENCES Мастер ([ID Мастера]),

  [ID Заказчика] int not null,
  -- Вторичный ключ на ID заказчика
```

```
FOREIGN KEY ([ID Заказчика]) REFERENCES Заказчик ([ID
Заказчика]),
)
```

Команда foreign key создает вторичный ключ, из соседней таблицы. Атрибутам «ID Исполнителя», «Модель техники» «Описание», «Запчасти» и «Дата окончания» предоставлена возможность оставаться пустыми, в случае, если заказчик не будет вводить текст или условия для заполнения не будут выполнены.

Пример диаграммы базы данных представлен на рисунке 2.

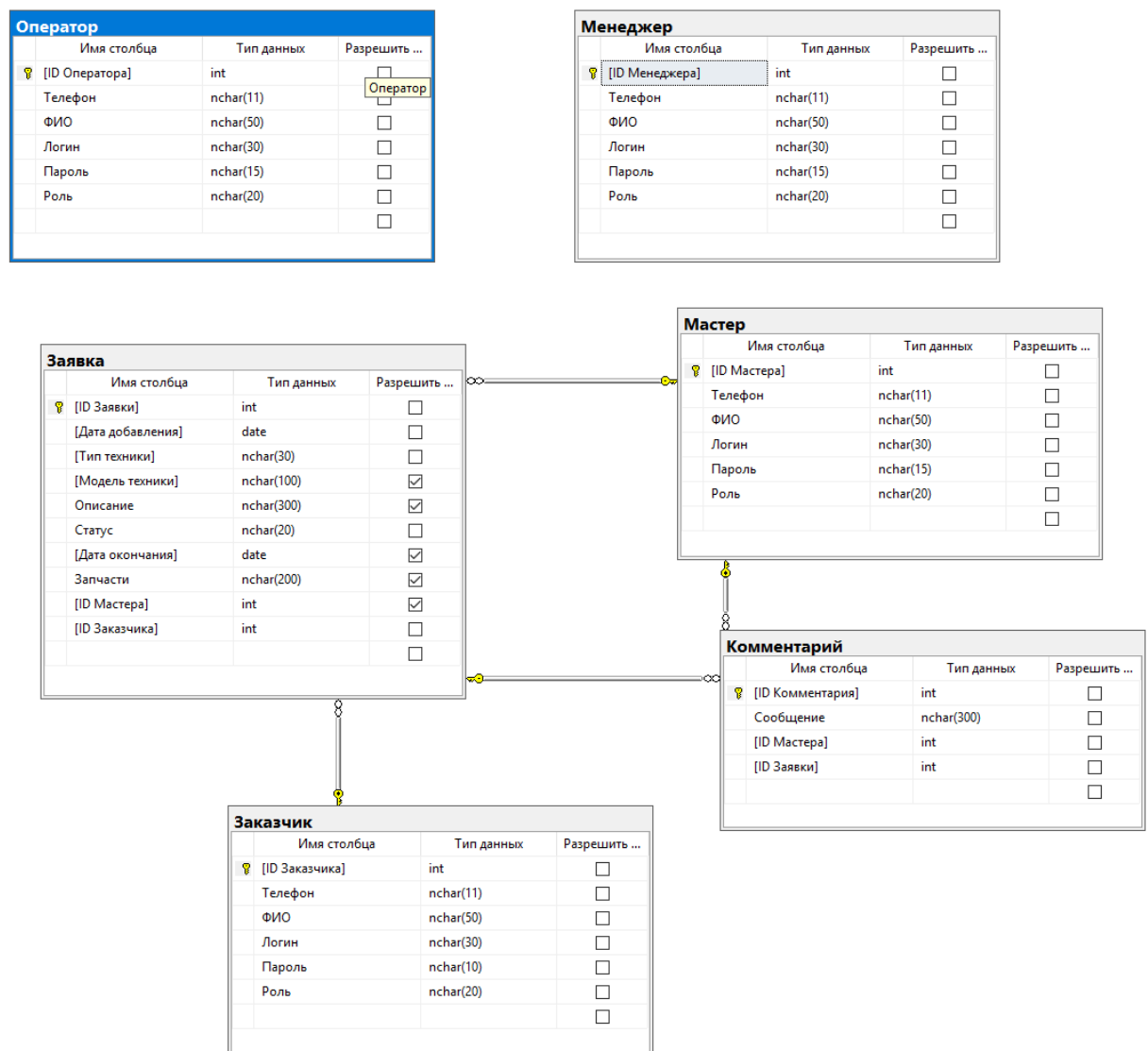


Рис. 2 диаграмма базы данных