**Лабораторная работа № 3**

**Логическая модель данных. Правила целостности.**

**Задание**

1.На основе концептуальной модели предметной области построить логическую модель данных. Использовать реляционную модель. Модель реализовать в среде MS SQL Server. Построить схему (диаграмму) базы данных.

2.Спроектировать и реализовать необходимые правила (ограничения) целостности.

3.В базу необходимо ввести некоторое количество тестовых данных. В каждой таблице должно быть не менее 7 записей. В ассоциациях, не менее 20 записей.

4.В пояснительную записку надо включить:

* логическую модель в виде схемы (диаграммы) базы данных;
* логическую модель в виде текста с подробным описанием правил (ограничений) целостности;
* анализ модели на соответствие принципам нормализации.

**Учебный пример разработки базы данных**

**Построение логической модели данных**

Логическая модель строится на основе концептуальной модели.

Из всех возможных видов логических моделей используем реляционную логическую модель. Это самый распространенный вид логических моделей.

Необходимо выполнить следующие шаги процедуры проектирования логической модели.

1. Представить каждую сущность как таблицу базы данных и специфицировать первичный ключ этой таблицы. При необходимости специфицировать внешний ключ и ограничения на внешний ключ этой таблицы.

2. Представить каждую ассоциацию как таблицу. Использовать в этой таблице внешние ключи для идентификации участников ассоциации и специфицировать ограничения, связанные с каждым из этих внешних ключей.

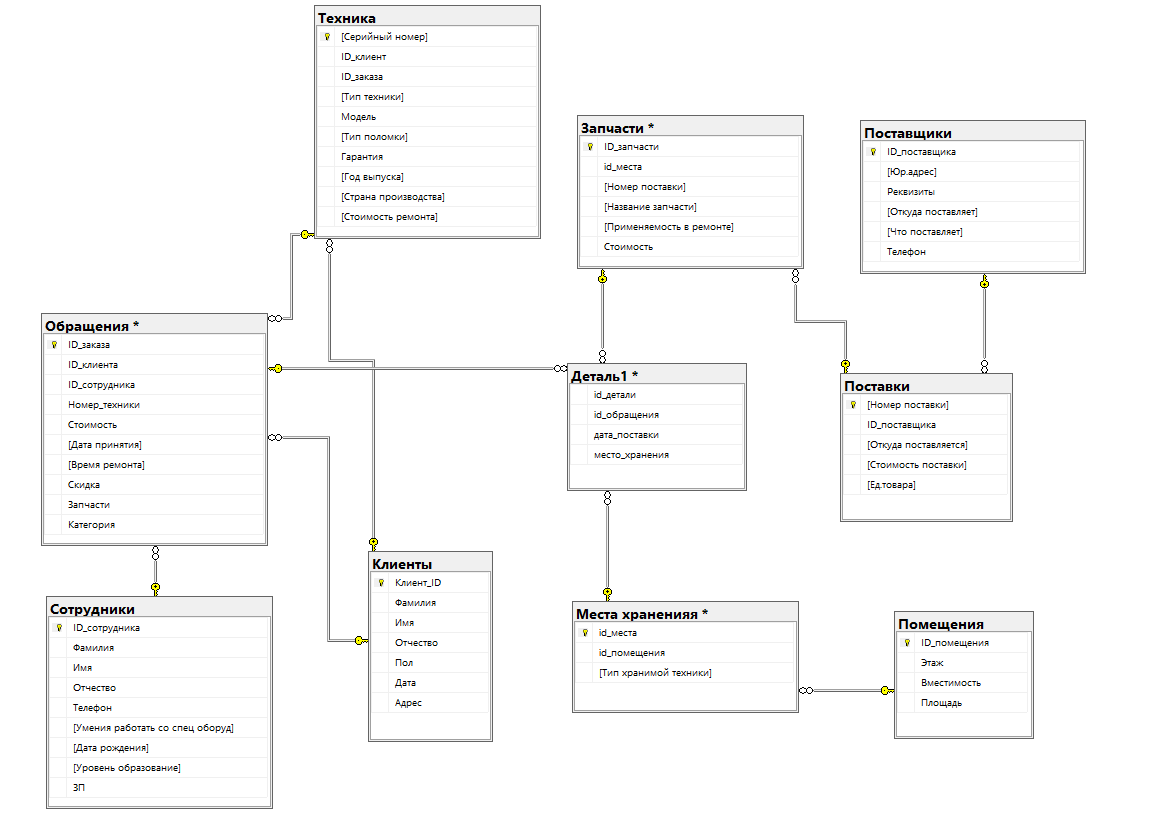
5. Представить каждый атрибут сущности как поле в таблице, представляющей сущность.

6. Для того чтобы исключить в проекте непреднамеренные нарушения каких-либо принципов нормализации, выполнить процедуру проверки отношений на соответствие 3-й нормальной форме (3НФ) или нормальной форме Бойса-Кодда (НФБК), описание процедуры см. в курсе лекций.

7. Если в процессе проверки было выявлено несоответствие и произведено разделение каких-либо таблиц, то следует модифицировать концептуальную модель базы данных и повторить перечисленные шаги.

8. Кроме уже введенных, ввести необходимые правила целостности, в частности правила доменной целостности. Дать краткое описание полученных таблиц и их полей.

Для нашей учебной базы данных «Сервисный центр» логическая модель в виде схемы (диаграммы) базы данных будет выглядеть так:



Логическая модель в виде текста с подробным описанием правил(ограничений) целостности описывается так:

USE [master]

CREATE DATABASE [АфанасьевИС]

CREATE TABLE [dbo].[Заказы](

[ID\_заказа] [int] IDENTITY(1,1) NOT NULL,

[ID\_клиента] [int] NULL,

[ID\_cотрудника] [int] NULL,

[Стоимость] [money] NULL,

[Дата принятия] [date] NULL,

[Время ремонта] [date] NULL,

[Скидка] [money] NOT NULL,

[Запчасти] [int] NULL,

CONSTRAINT [PK\_Заказы] PRIMARY KEY CLUSTERED

(

[ID\_заказа] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

CREATE TABLE [dbo].[Запчасти](

[ID\_запчасти] [int] IDENTITY(1,1) NOT NULL,

[id\_места] [int] NOT NULL,

[Номер поставки] [int] NOT NULL,

[Название запчасти] [varchar](50) NOT NULL,

[Применяемость в ремонте] [varchar](50) NULL,

[Стоимость] [money] NULL,

CONSTRAINT [PK\_Запчасти] PRIMARY KEY CLUSTERED

(

[ID\_запчасти] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

CREATE TABLE [dbo].[Клиенты](

[Клиент\_ID] [int] IDENTITY(1,1) NOT NULL,

[Фамилия] [varchar](50) NOT NULL,

[Имя] [varchar](50) NOT NULL,

[Отчество] [varchar](50) NULL,

[Пол] [char](10) NOT NULL,

[Дата] [date] NOT NULL,

[Адрес] [varchar](50) NULL,

CONSTRAINT [PK\_Клиент] PRIMARY KEY CLUSTERED

(

[Клиент\_ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

CREATE TABLE [dbo].[Места храненияя](

[id\_места] [int] NOT NULL,

[id\_помещения] [int] IDENTITY(1,1) NOT NULL,

[Тип хранимой техники] [varchar](50) NULL,

CONSTRAINT [PK\_Места храненияя] PRIMARY KEY CLUSTERED

(

[id\_места] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Помещения] Script Date: 27.10.2021 12:11:14 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Помещения](

[ID\_помещения] [int] IDENTITY(1,1) NOT NULL,

[Этаж] [int] NOT NULL,

[Вместимость] [int] NULL,

[Площадь] [int] NULL,

CONSTRAINT [PK\_Помещения] PRIMARY KEY CLUSTERED

(

[ID\_помещения] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

CREATE TABLE [dbo].[Поставки](

[Номер поставки] [int] NOT NULL,

[ID\_поставщика] [int] NOT NULL,

[Откуда поставляется] [varchar](50) NULL,

[Стоимость поставки] [int] NOT NULL,

[Ед.товара] [int] NOT NULL,

CONSTRAINT [PK\_Поставки] PRIMARY KEY CLUSTERED

(

[Номер поставки] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

CREATE TABLE [dbo].[Поставщики](

[ID\_поставщика] [int] IDENTITY(1,1) NOT NULL,

[Юр.адрес] [varchar](50) NULL,

[Реквизиты] [varchar](50) NULL,

[Откуда поставляет] [varchar](50) NULL,

[Что поставляет] [varchar](50) NULL,

[Телефон] [varchar](10) NULL,

CONSTRAINT [PK\_Поставщики] PRIMARY KEY CLUSTERED

(

[ID\_поставщика] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

CREATE TABLE [dbo].[Сотрудники](

[ID\_сотрудника] [int] NOT NULL,

[Фамилия] [varchar](50) NOT NULL,

[Имя] [varchar](50) NOT NULL,

[Отчество] [varchar](50) NULL,

[Телефон] [int] NOT NULL,

[Умения работать со спец оборуд] [char](10) NULL,

[Дата рождения] [date] NOT NULL,

[Уровень образование] [varchar](50) NOT NULL,

[ЗП] [money] NOT NULL,

CONSTRAINT [PK\_Сотрудники] PRIMARY KEY CLUSTERED

(

[ID\_сотрудника] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

CREATE TABLE [dbo].[Техника](

[Серийный номер] [int] IDENTITY(201200,1) NOT NULL,

[ID\_клиент] [int] NOT NULL,

[ID\_заказа] [int] NOT NULL,

[Тип техники] [varchar](30) NULL,

[Модель] [varchar](30) NOT NULL,

[Тип поломки] [varchar](max) NOT NULL,

[Гарантия] [char](10) NOT NULL,

[Год выпуска] [int] NULL,

[Страна производства] [varchar](30) NULL,

[Стоимость ремонта] [money] NULL,

CONSTRAINT [PK\_Техника] PRIMARY KEY CLUSTERED

(

[Серийный номер] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

GO

ALTER TABLE [dbo].[Заказы] ADD CONSTRAINT [DF\_Заказы\_Скидка] DEFAULT ((0)) FOR [Скидка]

GO

ALTER TABLE [dbo].[Заказы] WITH CHECK ADD CONSTRAINT [FK\_Заказы\_Клиент] FOREIGN KEY([Запчасти])

REFERENCES [dbo].[Запчасти] ([ID\_запчасти])

ON UPDATE CASCADE

ALTER TABLE [dbo].[Заказы] WITH CHECK ADD CONSTRAINT [FK\_Заказы\_Клиенты] FOREIGN KEY([ID\_клиента])

REFERENCES [dbo].[Клиенты] ([Клиент\_ID])

ON UPDATE CASCADE

ALTER TABLE [dbo].[Заказы] WITH CHECK ADD CONSTRAINT [FK\_Заказы\_Сотрудники] FOREIGN KEY([ID\_cотрудника])

REFERENCES [dbo].[Сотрудники] ([ID\_сотрудника])

ON UPDATE CASCADE

ALTER TABLE [dbo].[Запчасти] WITH CHECK ADD CONSTRAINT [FK\_Запчасти\_Места храненияя] FOREIGN KEY([id\_места])

REFERENCES [dbo].[Места храненияя] ([id\_места])

ON UPDATE CASCADE

ALTER TABLE [dbo].[Запчасти] WITH CHECK ADD CONSTRAINT [FK\_Запчасти\_Номер\_поставки] FOREIGN KEY([Номер поставки])

REFERENCES [dbo].[Поставки] ([Номер поставки])

ON UPDATE CASCADE

ALTER TABLE [dbo].[Поставки] WITH CHECK ADD CONSTRAINT [FK\_Поставки\_Поставки] FOREIGN KEY([ID\_поставщика])

REFERENCES [dbo].[Поставщики] ([ID\_поставщика])

ON UPDATE CASCADE

ALTER TABLE [dbo].[Техника] WITH CHECK ADD CONSTRAINT [FK\_Техника\_Заказы] FOREIGN KEY([ID\_заказа])

REFERENCES [dbo].[Заказы] ([ID\_заказа])

ALTER TABLE [dbo].[Техника] WITH CHECK ADD CONSTRAINT [FK\_Техника\_Клиент] FOREIGN KEY([ID\_клиент])

REFERENCES [dbo].[Клиенты] ([Клиент\_ID])

ON UPDATE CASCADE

ALTER TABLE [dbo].[Заказы] WITH CHECK ADD CONSTRAINT [CK\_Заказы] CHECK (([Скидка]>(0) AND [Скидка]<(100)))

ALTER TABLE [dbo].[Заказы] WITH CHECK ADD CONSTRAINT [CK\_Заказы\_Стоимость] CHECK (([Стоимость]>(0) AND [Стоимость]<(100000)))

ALTER TABLE [dbo].[Запчасти] WITH CHECK ADD CONSTRAINT [CK\_Запчасти] CHECK (([Название запчасти] like '[A-Za-zА-Яа-я.,/]%'))

ALTER TABLE [dbo].[Запчасти] WITH CHECK ADD CONSTRAINT [CK\_Запчасти\_1] CHECK (([Применяемость в ремонте] like '[A-Za-zА-Яа-я.,/]%'))

ALTER TABLE [dbo].[Запчасти] WITH CHECK ADD CONSTRAINT [CK\_Запчасти\_Стоимость] CHECK (([Стоимость]>(0)))

ALTER TABLE [dbo].[Клиенты] WITH CHECK ADD CONSTRAINT [CK\_Клиенты\_Адрес] CHECK ((NOT [Адрес] like '%[^A-Za-zА-Яа-я]%'))

ALTER TABLE [dbo].[Клиенты] WITH CHECK ADD CONSTRAINT [CK\_Клиенты\_Имя] CHECK ((NOT [Имя] like '%[^A-Za-zА-Яа-я]%'))

ALTER TABLE [dbo].[Клиенты] WITH CHECK ADD CONSTRAINT [CK\_Клиенты\_Пол] CHECK (([Пол] like '[МЖ]'))

ALTER TABLE [dbo].[Клиенты] WITH CHECK ADD CONSTRAINT [CK\_Клиенты\_Фамилия] CHECK ((NOT [Фамилия] like '%[^A-Za-zА-Яа-я]%'))

ALTER TABLE [dbo].[Места храненияя] WITH CHECK ADD CONSTRAINT [CK\_Места храненияя] CHECK ((NOT [Тип хранимой техники] like '%[^A-Za-zА-Яа-я0-9]%'))

ALTER TABLE [dbo].[Помещения] WITH CHECK ADD CONSTRAINT [CK\_Помещения\_Вместимость] CHECK (([Вместимость]>(0)))

ALTER TABLE [dbo].[Помещения] WITH CHECK ADD CONSTRAINT [CK\_Помещения\_Площадь] CHECK (([Площадь]>(0)))

ALTER TABLE [dbo].[Помещения] WITH CHECK ADD CONSTRAINT [CK\_Помещения\_этаж] CHECK (([Этаж]>=(0)))

ALTER TABLE [dbo].[Поставщики] WITH CHECK ADD CONSTRAINT [CK\_Поставщики\_Юр\_адрес] CHECK ((NOT [Юр.адрес] like '%[^A-Za-zА-Яа-я1-9./,]%'))

ALTER TABLE [dbo].[Сотрудники] WITH CHECK ADD CONSTRAINT [CK\_Сотрудники] CHECK ((NOT [Уровень образование] like '%[^A-Za-zА-Яа-я]%'))

ALTER TABLE [dbo].[Сотрудники] WITH CHECK ADD CONSTRAINT [CK\_Сотрудники\_1] CHECK (([ЗП]>(0)))

ALTER TABLE [dbo].[Сотрудники] WITH CHECK ADD CONSTRAINT [CK\_Сотрудники\_Имя] CHECK ((NOT [Имя] like '%[^A-Za-zА-Яа-я]%'))

ALTER TABLE [dbo].[Сотрудники] WITH CHECK ADD CONSTRAINT [CK\_Сотрудники\_Фамилия] CHECK ((NOT [Фамилия] like '%[^A-Za-zА-Яа-я]%'))

USE [master]