

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ИМЕНИ В.Ф. УТКИНА»

КАФЕДРА «ЭЛЕКТРОННЫЕ ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ»

МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Б1.В.02 «Проектирование моделей данных»

Направление подготовки

38.03.05 Бизнес-информатика

ОПОП академического бакалавриата

«Бизнес-информатика»

Квалификация (степень) выпускника — бакалавр

Форма обучения — очная

1. МЕТОДИЧЕСКИЕ УКАЗАНИЯ К САМОСТОЯТЕЛЬНОЙ РАБОТЕ

Самостоятельная работа студентов по дисциплине «Проектирование моделей данных» играет важную роль в ходе всего учебного процесса. Самостоятельная работа способствует закреплению знаний, умений и навыков, приобретаемых в ходе различных видов аудиторных занятий.

Основными видами самостоятельной работы по дисциплине являются: подготовка к лабораторным и практическим занятиям (доработка конспекта лекции с применением учебника, методической и дополнительной литературы; подбор иллюстраций (примеров) к теоретическим положениям; подготовка доклада на заданную тему; самостоятельное изучение отдельных вопросов и тем курса) и подготовка к процедуре промежуточной аттестации.

Работа с конспектом лекций предполагает просмотр конспекта в тот же день после занятий. При этом необходимо пометить материалы конспекта, которые вызывают затруднения для понимания. При этом обучающийся должен стараться найти ответы на затруднительные вопросы, используя рекомендуемую литературу. Если ему самостоятельно не удалось разобраться в материале, необходимо сформулировать вопросы и обратиться за помощью к преподавателю на консультации или ближайшей лекции.

Лабораторные работы и практические занятия составляют важную часть профессиональной подготовки студентов. Они направлены на экспериментальное подтверждение теоретических положений и формирование учебных и профессиональных практических умений.

Выполнение студентами лабораторных работ направлено на следующие цели:

- обобщение, систематизацию, углубление, закрепление полученных теоретических знаний по конкретным темам дисциплин;
- формирование необходимых профессиональных умений и навыков.

Выполнению лабораторной работы предшествует проверка знаний студентов – их теоретической готовности к выполнению задания.

Помимо выполнения работы для каждой лабораторной работы предусмотрена процедура защиты, в ходе которой преподаватель проводит устный или письменный опрос студентов для контроля понимания и правильной интерпретации полученных результатов и усвоения ими основных теоретических и практических знаний по теме лабораторной работы.

Практические занятия направлены на закрепление основных теоретических знаний и положений курса, полученных обучающимися в рамках лекционных и самостоятельных занятий на практике. Практическому занятию предшествует предварительная подготовка обучающегося в соответствии с тематикой занятия.

При подготовке к экзамену в дополнение к изучению конспектов лекций, учебных пособий, слайдов и другого раздаточного материала предусмотренной рабочей программой дисциплины, необходимо пользоваться учебной литературой, рекомендованной к настоящей рабочей программе. При подготовке к зачету нужно изучить теорию: определения всех понятий и подходы к оцениванию до состояния понимания материала и самостоятельно решить по несколько типовых задач из каждой темы (в том случае если тема предусматривает решение задач). При решении задач необходимо наличие умений пояснить получаемые результаты и ход решения.

Теоретическая составляющая курса «Проектирование моделей данных» становится более понятной, когда дополнительно к прослушиванию лекции и изучению конспекта, обучающимся изучается дополнительная рекомендованная литература.

Типовые задания в рамках самостоятельной работы студентов для укрепления теоретических знаний, развития умений и навыков, предусмотренных компетенциями, закрепленными за дисциплиной:

1. Формирование семантического описания предметной области в соответствии с заданной предметной областью.
2. Выявление потребностей пользователя информационной системы.
3. Анализ существующих программных решений для данной предметной области.
4. Разработка общей структуры информационной системы с клиент-серверной архитектурой, используя двухзвенную модель DBS (сервер баз данных).
5. Построение схемы БД в СУБД MS SQL Server.
6. Разработка основных объектов структуры БД: задание частных ограничений целостности данных, разработка представлений, разработка запросов к БД, разработка хранимых процедур, триггеров.
7. Тестирование объектов БД: тестирование частных ограничений целостности данных, тестирование представлений, тестирование хранимых процедур и триггеров.
8. Разработка сценария развертывания базы данных, включающего сценарий создания объектов базы данных, сценарий заполнения таблиц базы данных исходными данными.

2. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ПОДГОТОВКЕ РЕФЕРАТОВ

Реферат представляет собой краткий доклад по определённой теме, в котором собрана информация из одного или нескольких источников. Данный вид работ направлен на более глубокое самостоятельное изучение студентами лекционного материала или рассмотрения вопросов для дополнительного изучения.

Типовые темы рефератов по разделам курса «Проектирование моделей данных»:

1. Типы современных баз данных.
2. NoSQL базы данных.
3. Типы существующих СУБД.
4. Современные технологии доступа к данным в БД.
5. Математические аппараты БД.
6. Примеры конкретных операций, нарушающих целостность данных.
7. Существующие стратегии поддержания ссылочной целостности.
8. Выбор архитектуры информационной системы.
9. Преимущества двухуровневой архитектуры.
10. Пример модели распределенного представления.
11. Пример модели удаленного представления.
12. Пример модели распределения бизнес-логики.
13. Пример модели распределенного управления данными.
14. Пример модели удаленного управления данными.
15. Основные принципы распределенных БД.

Основные требования к оформлению:

1. Общий объем работы от 30 до 40 страниц. Реферат должен содержать введение, основную часть с анализом и выводами по рассматриваемому вопросу и обоснованное заключение. Список используемых источников – не менее 15 наименований.

2. Оформление основного текста в соответствии с ГОСТ 7.32-2017 «Отчет о научно-исследовательской работе. Структура и правила оформления». Оформление библиографического списка в соответствии с ГОСТ 7.1-2003 «Библиографическая запись».

3. Дата отправки на проверку устанавливается преподавателем.

3. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ПОДГОТОВКЕ РЕФЕРАТОВ

Целью курсового проекта является освоение методов проектирования, программирования и сопровождения баз данных получение навыков инсталляции, настройки и отладки СУБД и систем программирования общего назначения, а также получение навыков самостоятельной работы при изучении технической документации в области проектирования, программирования и сопровождения БД.

Задание содержит название конкретной предметной области, для которой необходимо выполнить проектирование модели БД, разработать схему БД в СУБД MS SQL Server, включающую все необходимые для работы пользователя объекты (ограничения целостности, запросы др.). БД должна охватывать все информационные потребности пользователя заданной предметной области.

Общие требования к работе:

В БД должно быть не менее 10 таблиц. Главные таблицы должны содержать не менее 10 строк, подчиненные — не менее 20.

Требуется проверить все таблицы БД на соответствие нормальной форме Бойса-Кодда.

Клиентская часть ИС должна охватывать все информационные потребности пользователя. При выполнении курсового проекта необходимо использовать материал лекций и рекомендуемую учебно-методическую литературу.

К защите **курсового проекта** студент должен представить:

- работоспособную БД, установленную на компьютере в лабораториях кафедры;
- пояснительную записку в распечатанном виде;
- электронный каталог, содержащий пояснительную записку, дистрибутив БД, исходные тексты, файлы БД, сценарий инсталляции БД,

В ходе выполнения курсовой работы рекомендуется придерживаться календарного плана, приведённого в таблице.

Содержание этапа	Продолжительность этапа
1. Выбор темы и утверждение технического задания.	2 недели
2. Описание предметной области	2 недели
3. Разработка инфологической модели.	1 неделя
4. Разработка даталогической модели.	2 недели
5. Создание схемы базы данных и заполнение таблиц данными.	4 недели
5. Оформление пояснительной записки.	1 неделя
6. Сдача курсового проекта на проверку.	1 неделя
7. Защита курсовой работы.	1 неделя

По результатам второго, третьего и четвертого этапов проводится промежуточный просмотр курсового проекта.

Варианты предметных областей для курсового проекта:

1. Разработка для ИС «Парковка».
2. Разработка для ИС «Выездные медосмотры».
3. Разработка для ИС «Приют для животных».
4. Разработка для ИС «Выставочный комплекс».
5. Разработка для ИС «Услуги SPA-салона».
6. Разработка для ИС «Птицефабрика».
7. Разработка для ИС «Справочная система театра».
8. Разработка для ИС «Отдел автоматизации и сопровождения ПО на предприятии».
9. Разработка для ИС «Зоомагазин».

10. Разработка для ИС «Независимая экологическая экспертиза».
11. Разработка для ИС «Адресное бюро».
12. Разработка для ИС «Аквапарк».
13. Разработка для ИС «Дизайн-студия интерьера».
14. Разработка для ИС «Услуги по пошиву и ремонту трикотажных изделий».
15. Разработка для ИС «Услуги прачечных».
16. Разработка для ИС «Рыболовная база».
17. Разработка для ИС «Санитарная обработка помещений».
18. Разработка для ИС «Обустройство детских площадок».
19. Разработка для ИС «Комиссионный магазин».
20. Разработка для ИС «Телекоммуникационный центр».
21. Разработка для ИС «Прокат спортивного инвентаря».
22. Разработка для ИС «Выставка-продажа сельскохозяйственной продукции».
23. Разработка для ИС «Дневной стационар».
24. Разработка для ИС «Бюро трудоустройства».
25. Разработка для ИС «Показ-продажа моделей высокой моды».
26. Разработка ИС «Видеопрокат».
27. Разработка ИС «Чемпионат мира по футболу».
28. Разработка ИС «Риэлторская фирма».
29. Разработка ИС «Расписание занятий».
30. Разработка ИС «Поликлиника».
31. Разработка ИС «Больница».
32. Разработка ИС «Агентство недвижимости».
33. Разработка ИС «Деканат».
34. Разработка ИС «Учет товаров на складе».
35. Разработка ИС «Автосервис».
36. Разработка ИС «Продажа легковых автомобилей».
37. Разработка ИС «Хозяйственный магазин».
38. Разработка ИС «Мебельный магазин».
39. Разработка ИС «Обувная мастерская».
40. Разработка для ИС «Тренажерный зал».
41. Разработка для ИС «Салон красоты».
42. Разработка для ИС «Автовокзал».
43. Разработка для ИС «Туристическая фирма».
44. Разработка для ИС «Продажа компьютерной техники».
45. Разработка для ИС «Автоматизация».
46. Разработка для ИС «Спортивные товары».
47. Разработка для ИС «Аэропорт».
48. Разработка для ИС «Курьерские службы».
49. Разработка для ИС «Ресторан».
50. Разработка для ИС «Автошкола».

Типовые вопросы при защите курсового проекта

1. Сформулируйте задачу проектирования БД.
2. Опишите основные понятия и определения метод ER-диаграмм.
3. Покажите изображение диаграмм ER-типов.
4. Дайте определение кардинальности связей.
5. Опишите порядок проектирования баз данных методом ER-диаграмм.
6. Что такое избыточное дублирование и аномалии,
7. Что определяют зависимости между атрибутами отношения?
8. Что такое диаграммы функциональных зависимостей,
9. Дайте определение 1НФ.

10. Дайте определение 2НФ.
11. Дайте определение 3НФ.
12. Приведите пример проектирования базы данных методом нормальных форм.
13. Дайте определение БКНФ.
14. Выполните обзор моделей информационных систем с архитектурой клиент-сервер.
15. Приведите пример двухзвенной модели клиент-сервер.
16. Приведите пример трехзвенной модели клиент-сервер.

Требования к содержанию пояснительной записки:

Введение.

1. Постановка задачи и технико-экономическое обоснование разработки.
 - 1.1. Анализ предметной области, выявление необходимой пользователю функциональности.
 - 1.2. Обзор существующих аналогов.
 - 1.3. Выводы об актуальности разработки.
2. Разработка общей архитектуры.
3. Разработка моделей данных.
 - 3.1. Разработка инфологической модели данных.
 - 3.1.1. Выявление сущностей и связей.
 - 3.1.2. Построение ER-диаграммы.
 - 3.2. Разработка даталогической модели данных.
 - 3.2.1. Переход от ER-диаграммы к предварительным отношениям.
 - 3.2.2. Заполнение предварительных отношений атрибутами.
 - 3.2.3. Проверка предварительных отношений на соответствие нормальным формам.
 - 3.2.4. Построение схемы данных.
4. Разработка основных объектов БД.
 - 4.1. Создание файлов БД и таблиц БД
 - 4.2. Задание частных ограничений целостности данных.
 - 4.3. Разработка запросов к БД.

Заключение.

Список используемой литературы.

Приложение: SQL-скрипт для создания БД (должны быть включены все объекты БД – таблицы, ограничения целостности и др.).

4. ПРИМЕР ОФОРМЛЕНИЯ НЕКОТОРЫХ РАЗДЕЛОВ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ К КУРСОВОМУ ПРОЕКТУ

В качестве примера рассматривается предметная область «Предприятие по экспорту, продаже и сервисному обслуживанию российских автомобилей в странах Европы».

Перед проектированием моделей данных требуется провести тщательный анализ заданной предметной области, выявить функции, которые необходимы конечному пользователю системы.

4.1. Анализ предметной области, выявление необходимой пользователю функциональности

Предметная область – закупка автомобилей у российских автозаводов, экспорт, продажа и сервисное обслуживание автомобилей, проданных за границей.

В базе данных требуется хранить следующую *информацию*:

- данные о заводах-изготовителях: название (уникально), адрес, телефон;
- данные о представительствах фирмы – центральный офис (располагается в России), автосалоны (за рубежом) (автосалоны обеспечивают продажу и сервисное обслуживание автомобилей): тип (офис/автосалон), адрес (уникален), телефон;
- данные о сотрудниках фирмы: ФИО, телефон, должность, место работы;
- данные о клиентах фирмы – покупателях автомобилей: ФИО, телефон;
- данные об автомобилях, закупленных фирмой: IDАвтомобиля (уникальный числовой код), завод-изготовитель, модель, комплектация;
- данные о комплектациях автомобилей, предоставляемых фирмой: название комплектации (уникально, например Minimum, Medium, Maximum), цвет, объём двигателя, КПП, электропакет, число подушек безопасности, музыкальное оборудование;
- данные о закупках автомобилей у заводов-изготовителей (в России): IDАвтомобиля (уникально), ФИО сотрудника (осуществившего закупку), дата закупки, стоимость закупки;
- данные об экспорте автомобилей за границу: IDАвтомобиля (уникально), дата отправки, сотрудник (осуществивший отправку), автосалон (в который направлен автомобиль);
- данные о продажах автомобилей: IDАвтомобиля (уникально), сотрудник (продавец), клиент, дата продажи, стоимость продажи;
- данные о сервисном обслуживании автомобилей: IDАвтомобиля, сотрудник, дата обслуживания, причина обращения.

Анализ ограничений, накладываемых на модель

При этом на предметную область задачи, а следовательно, и на разрабатываемую базу данных накладываются следующие ограничения, или *бизнес-правила*:

- сотрудник может работать только в одном офисе/автосалоне;
- сотрудник может занимать только одну должность;
- автомобиль изготовлен лишь одним заводом, но каждый завод производит множество автомобилей;

- автомобиль может иметь только одну комплектацию, но у различных автомобилей может быть одинаковая комплектация;
- автомобиль может быть закуплен/экспортирован/продан лишь одним сотрудником, но каждый сотрудник может закупить/экспортировать/продать/ множество автомобилей;
- сотрудник может обслужить несколько автомобилей, равно как и автомобиль может быть обслужен несколькими сотрудниками (поочередно);
- закупку и экспорт одного и того же автомобиля могут производить разные сотрудники;
- закупку и экспорт автомобилей могут производить только сотрудники центрального офиса с должностью «Менеджер», продажи автомобилей могут производить сотрудники представительств только с должностью «Менеджер», сервисное обслуживание могут производить только сотрудники с должностью «Автомеханик»;
- автомобиль может быть закуплен, экспортирован и продан лишь единожды, но обслуживание может как производиться многократно в одном или различных центрах, так и вообще не производиться;
- автомобиль может принадлежать лишь одному клиенту, но клиент может иметь несколько автомобилей;
- автомобиль не может быть экспортирован раньше, чем закуплен; продан раньше, чем экспортирован; обслужен раньше, чем продан;
- автомобиль не может быть экспортирован в центральный офис (так как он находится в России);
- автомобиль может быть продан только сотрудником того офиса, в который он был экспортирован.

4.2. Проектирование моделей данных

Процесс проектирования моделей данных должен включать следующие этапы:

1. Выявление сущностей и связей.
2. Построение ER – диаграммы.
3. Формирование предварительных отношений.
4. Подготовка списка атрибутов, которые были выделены в предметной области.
5. Распределение атрибутов по предварительным отношениям.
6. Проверка отношений на соответствие БКНФ.
7. Пересмотр ER-диаграммы.
8. Построение схемы базы данных.

Рассмотрим примеры оформления результатов каждого из этапов проектирования моделей данных.

4.2.1. Выявление сущностей и связей

В предметной области можно выделить следующие *сущности*:

1. Завод-изготовитель;
2. Офис;
3. Сотрудник;
4. Клиент;

5. Автомобиль;
6. Комплектация.

В предметной области можно выделить следующие *связи* между сущностями:

1. Сотрудник работает в офисе;
2. Сотрудник закупает автомобиль некоторой комплектации у завода-изготовителя;
3. Сотрудник экспортирует автомобиль в салон;
4. Сотрудник продаёт автомобиль клиенту;
5. Сотрудник обслуживает автомобиль.

4.2.2. Построение ER–диаграмм

На рисунках отображены степени связей между сущностями и классы принадлежности. Рассмотрим их подробнее.

1. Сотрудник работает в офисе:



Рисунок 1 – ER-диаграмма связи «сотрудник работает в офисе»

Тип связи: один ко многим. Сотрудник работает только в одном офисе, в офисе работает множество сотрудников.

Класс принадлежности: обязательный со стороны сотрудника (сотрудник обязательно работает в офисе) и необязательный со стороны офиса (в офисе не обязательно работают сотрудники).

2. Сотрудник закупает автомобиль некоторой комплектации у завода-изготовителя:

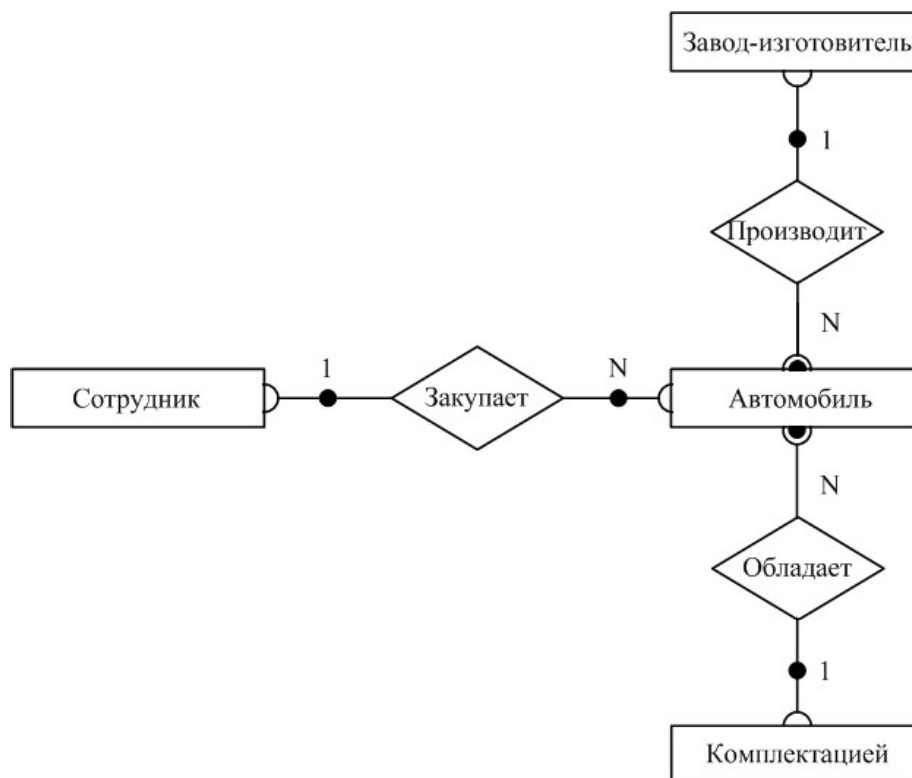


Рисунок 2 – ER-диаграмма связи

«сотрудник закупает автомобиль некоторой комплектации у завода-изготовителя»

Типы связей: один ко многим (сотрудник закупает множество автомобилей, автомобиль закуплен одним сотрудником; автомобиль обладает одной комплектацией, множество

автомобилей могут обладать одинаковой комплектацией; завод производит множество автомобилей, автомобиль произведён одним заводом).

Классы принадлежности: в связи «сотрудник закупает автомобиль» – необязательные (сотрудник может не закупать автомобилей, автомобиль может быть не закуплен), в связях «завод производит автомобиль» и «автомобиль обладает комплектацией» - обязательные со стороны автомобиля (автомобиль обязательно произведён заводом, автомобиль обязательно обладает комплектацией).

3. Сотрудник экспортирует автомобиль в салон:

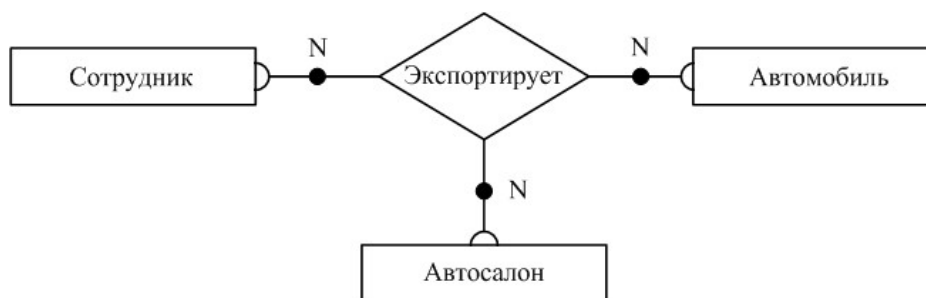


Рисунок 3 – ER-диаграмма связи «Сотрудник экспортирует автомобиль в салон»

Типы связей: многие ко многим (сотрудник экспортирует множество автомобилей в множество салонов, в салон экспортируется множество автомобилей множеством сотрудников).

Классы принадлежности: необязательные (сотрудник может не экспортировать автомобилей, в салон могут не экспортироваться автомобили, автомобиль может быть не экспортирован).

4. Сотрудник продаёт автомобиль клиенту:

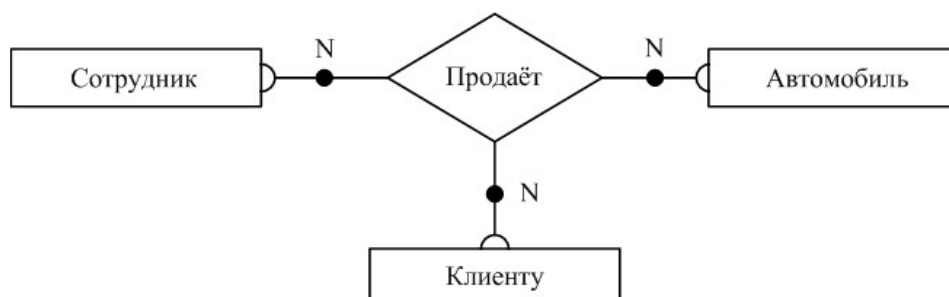


Рисунок 4 – ER-диаграмма связи «Сотрудник продаёт автомобиль клиенту»

Типы связей: многие ко многим (сотрудник продаёт множество автомобилей множеству клиентов, клиент может купить много автомобилей у различных сотрудников).

Классы принадлежности: необязательные (сотрудник может не продавать автомобилей, автомобиль может быть не продан).

5. Сотрудник обслуживает автомобиль:



Рисунок 5 – ER-диаграмма связи «Сотрудник обслуживает автомобиль»

Типы связей: многие ко многим (сотрудник обслуживает множество автомобилей, автомобиль может быть обслужен многими сотрудниками).

Классы принадлежности: необязательные (сотрудник может не обслуживать автомобилей, автомобиль может быть не обслужен).

Построение общей ER-диаграммы:

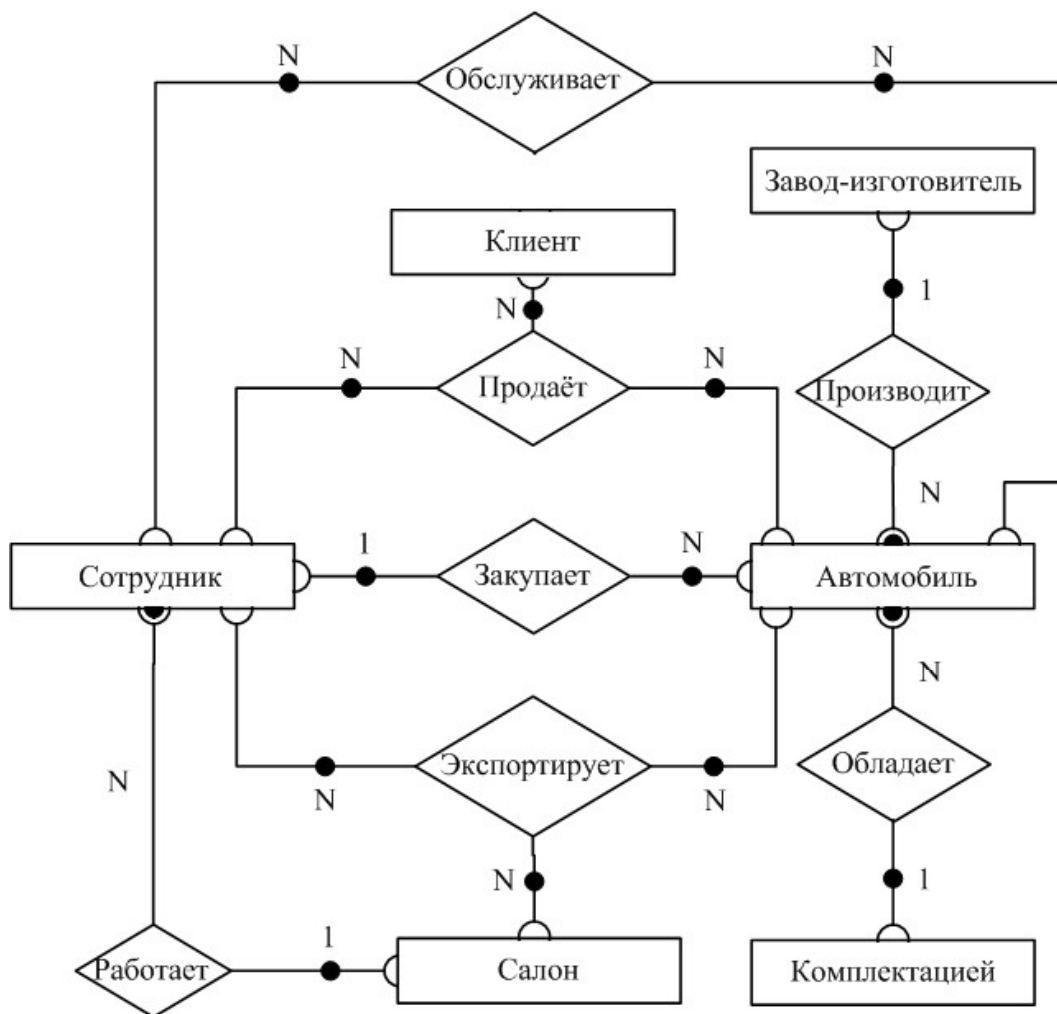


Рисунок 6 – Общая ER-диаграмма

По результатам первых двух этапов проектирования была получена инфологическая модель – полная ER-диаграмма классов, представляющая собой объединение всех построенных ER-диаграмм для отдельных связей.

4.2.3. Формирование предварительных отношений по ER-диаграмме

Третий этап выполняется на основе специальных правил. При переходе от ER-диаграммы к предварительным отношениям используются следующие правила:

Правило 1: 1:1, класс принадлежности обеих сущностей – обязательный. Формируют 1 отношение, ключом является ключ любой из двух сущностей.

Правило 2: 1:1, класс принадлежности одной сущности – обязательный, второй – нет. Формируют 2 отношения по одному для сущности, ключом является ключ соответствующей сущности, при этом ключ сущности с необязательным классом принадлежности добавляется в отношение для сущностей с обязательным классом принадлежности.

Правило 3: 1:1, класс принадлежности обеих сущностей – необязательный. Формируют 3 отношения по одному для каждой сущности и одно для связи; ключом в первых двух отношениях является ключ соответствующей сущности. Отношение для связи должно содержать ключи обеих сущностей. В качестве ключа этого отношения можно принять ключ любой из сущностей.

Правило 4: 1:N, класс принадлежности связи со стороны N – обязательный. Формируется 2 отношения по одному для сущности, ключ – ключ сущности. В отношении со стороны N добавляется ключ односвязной сущности.

Правило 5: 1:N, класс принадлежности связи со стороны N – необязательный. Формируется 3 отношения: по одному для сущности и одно для связи. Ключи первых двух – ключи соответствующих сущностей, в отношении для связи содержатся ключи обеих сущностей, причём первичным ключом является ключ N-связной сущности.

Правило 6: N:N. Формируются три отношения: по одному для сущностей и одно для связи. Первичные ключи первых двух – ключи соответствующих сущностей. В отношении для связи должны содержаться ключи обеих сущностей, которые совместно являются первичным ключом отношения.

Правило 7: при наличии n-арной связи следует использовать n+1 отношение: по одному для каждой сущности (первичным ключом является ключ соответствующей сущности) и одно для связи (содержит ключи всех сущностей, которые совместно образуют первичный ключ).

По описанным выше ER-диаграммам и правилам получаем следующие предварительные отношения:

1. Сотрудник работает в офисе: по правилу 4 формируем 2 отношения: по одному для каждой сущности, где ключами являются ключи соответствующих сущностей:

Сотрудник(ФИОСотрудника, №Офиса),
Офис(№Офиса);

2. Сотрудник закупает автомобиль некоторой комплектации у завода-изготовителя: для связи «Завод-изготовитель производит автомобиль» по правилу 4 формируем 2 отношения: по одному для каждой сущности, где ключами являются ключи соответствующих сущностей:

Завод-изготовитель(№Завода),
Автомобиль(IDАвтомобиля, №Завода);

для связи «Автомобиль обладает комплектацией» по правилу 4 формируем 2 отношения: по одному для каждой сущности, где ключами являются ключи соответствующих сущностей:

Комплектация(№Комплектации),
Автомобиль(IDАвтомобиля, №Завода, №Комплектации);

для связи «Сотрудник закупает автомобиль» по правилу 5 формируем 3 отношения: по одному для каждой сущности, где ключами являются ключи соответствующих сущностей и одно для связи, которое содержит ключи обеих сущностей и где ключом является ключ N-связи:

Сотрудник(ФИОСотрудника, №Офиса),
Автомобиль(IDАвтомобиля, №Завода, №Комплектации),
Закупка(IDАвтомобиля, ФИОСотрудника);

3. Сотрудник экспортирует автомобиль в салон: по правилу 7 формируем 4 отношения: по одному для каждой сущности, где ключами являются ключи соответствующих сущностей и одно для связи, где ключ содержит ключи всех сущностей:

Сотрудник(ФИОСотрудника, №Офиса),
Офис(№Офиса),
Автомобиль(IDАвтомобиля, №Завода, №Комплектации),
Экспорт(IDАвтомобиля, ФИОСотрудника, №Офиса);

4. Сотрудник продаёт автомобиль клиенту: по правилу 7 формируем 4 отношения: по одному для каждой сущности, где ключами являются ключи соответствующих сущностей и одно для связи, где ключ содержит ключи всех сущностей:

Сотрудник(ФИОСотрудника, №Офиса),
Автомобиль(IDАвтомобиля,

№Завода, №Комплектации),
 Клиент(ФИОКлиента),
 Продажа(IDАвтомобиля, ФИОСотрудника, ФИОКлиента);

5. Сотрудник обслуживает автомобиль: по правилу 6 формируем 3 отношения: по одному для каждой сущности, где ключами являются ключи соответствующих сущностей и одно для связи, где ключ содержит ключи всех сущностей:

Сотрудник(ФИОСотрудника, №Офиса),
 Автомобиль(IDАвтомобиля, №Завода, №Комплектации),
 Обслуживание(IDАвтомобиля, ФИОСотрудника).

4.2.4. Подготовка списка атрибутов

На данном этапе подготавливаются атрибуты, которые должны храниться в базе данных. Это все, требуемые для конечного пользователя системы, атрибуты, кроме тех, которые уже были выделены на предыдущих этапах и попали схему БД. Совместим реализацию данного этапа со следующим (распределение атрибутов) по отношениям.

4.2.5. Распределение подготовленных атрибутов по предварительным отношениям

- 1) Офис(№Офиса, ТипОфиса, АдресОфиса, ТелефонОфиса);
- 2) Сотрудник(ФИОСотрудника, №Офиса, Должность, ТелефонСотрудника);
- 3) Завод-изготовитель(№Завода, НазваниеЗавода, АдресЗавода, ТелефонЗавода);
- 4) Комплектация(№Комплектации, НазваниеКомплектации, Цвет, ОбъёмДвигателя, КПП, Электропакет, ЧислоПодушекБезопасности, МузыкальноеОборудование);
- 5) Автомобиль(IDАвтомобиля, №Завода, Модель, №Комплектации);
- 6) Клиент(ФИОКлиента, ТелефонКлиента);
- 7) Закупка(IDАвтомобиля, ФИОСотрудника, ДатаЗакупки, СтоимостьЗакупки);
- 8) Экспорт(IDАвтомобиля, ФИОСотрудника, №Офиса, ДатаЭкспорта);
- 9) Продажа(IDАвтомобиля, ФИОСотрудника, ФИОКлиента, ДатаПродажи, СтоимостьПродажи);
- 10) Обслуживание(IDАвтомобиля, ФИОСотрудника, ДатаОбслуживания, ПричинаОбращения).

Как мы видим, поглощения отношений не произошло, так как каждое из них несёт дополнительную информацию по отношению к другим.

4.2.6. Проверка отношений на соответствие БКНФ

Существует несколько нормальных форм, каждой из которых соответствует определенный набор ограничений, и отношение находится в соответствующей нормальной форме, если оно удовлетворяет свойственному ей набору ограничений.

Выделяют 1НФ, 2НФ, 3НФ, БКНФ, 4НФ, 5НФ. Каждая нормальная форма сохраняет свойства предыдущих нормальных форм. Переход к более высокой НФ выполняется путем декомпозиции исходного отношения на два или более отношений, удовлетворяющих требованиям этой НФ.

Рассмотрим каждую из перечисленных нормальных форм более подробно.

1НФ. При 1НФ все атрибуты отношения должны быть простыми (атомарными, неделимыми) с точки зрения СУБД. Таблица не должна иметь повторяющихся записей, групп полей, необходимо наличие, по крайней мере, одного уникального индекса.

2НФ. Отношение находится в 2НФ, если оно находится в 1НФ и каждый неключевой атрибут функционально-полно зависит от первичного ключа. Из этого следует, что если ключ составной, то любой неключевой атрибут зависит от всего ключа и не зависит от какого-либо из входящих в него атрибутов.

3НФ. При третьей нормальной форме (3НФ) отношение должно находиться в 2НФ, а каждый неключевой атрибут нетранзитивно зависеть от ключа, т.е. отсутствуют функциональные зависимости между неключевыми атрибутами.

БКНФ. Нормальная форма Бойса-Кодда требует, чтобы в отношении были в 3НФ, и в нем отсутствовали зависимости атрибутов первичного ключа от неключевых атрибутов (ситуация, когда отношение имеет два или более возможных ключа, которые являются составными и имеют общий атрибут). В данном курсовом проекте стояло условие необходимости соответствия таблиц БД нормальной форме Бойса-Кодда.

Проверим наши отношения на принадлежность к НФ:

1) Офис:

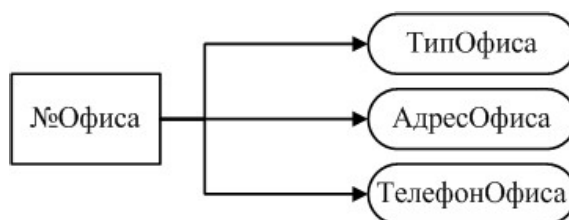


Рисунок 7 – Отношение «Офис»

2) Сотрудник:

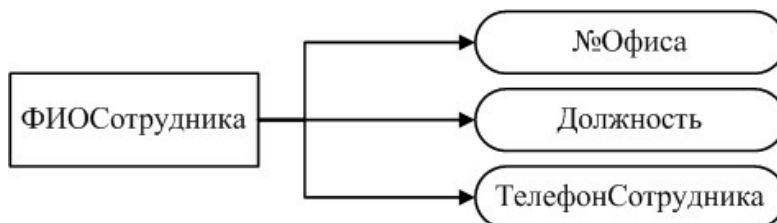


Рисунок 8 – Отношение «Сотрудник»

3) Завод-изготовитель:



Рисунок 9 – Отношение «Завод-изготовитель»

4) Комплектация:

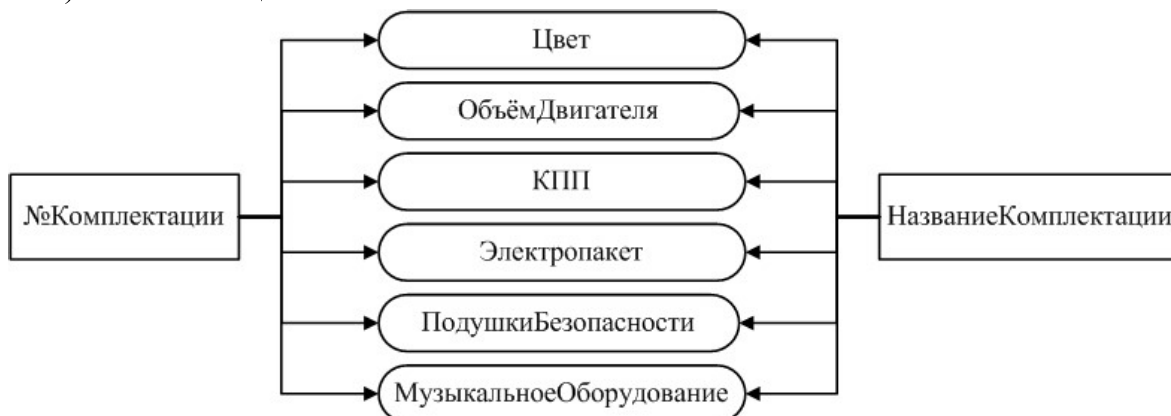


Рисунок 10 – Отношение «Комплектация»

5) Автомобиль:

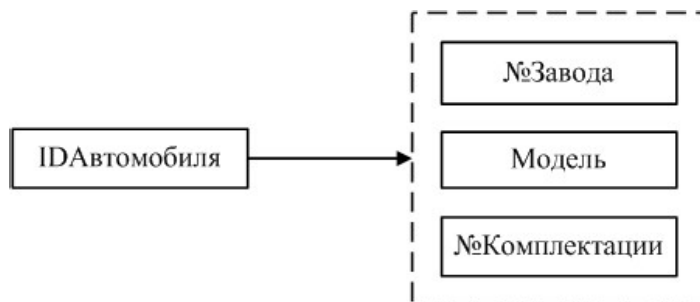


Рисунок 11 – Отношение «Автомобиль»

6) Клиент:



Рисунок 12 – Отношение «Клиент»

7) Закупка

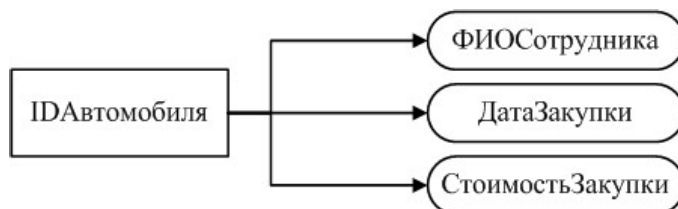


Рисунок 13 – Отношение «Закупка»

8) Экспорт:

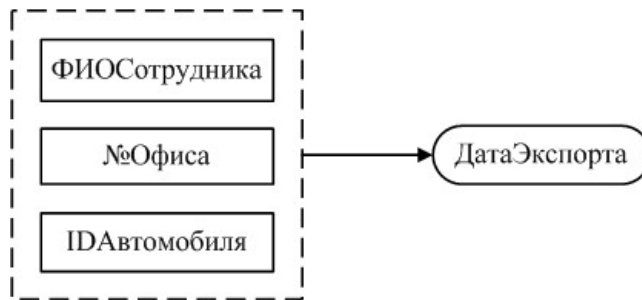


Рисунок 14 – Отношение «Экспорт»

9) Продажа:

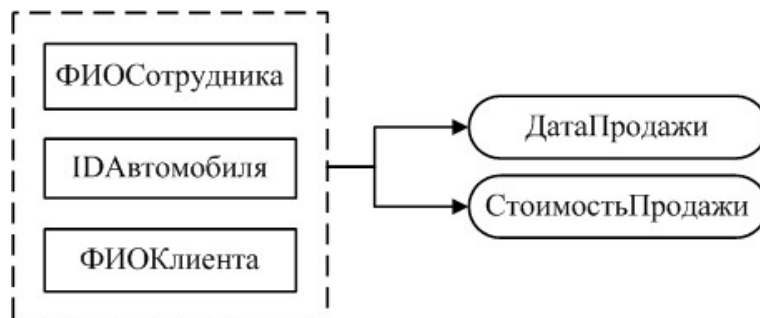


Рисунок 15 – Отношение «Продажа»

10) Обслуживание:

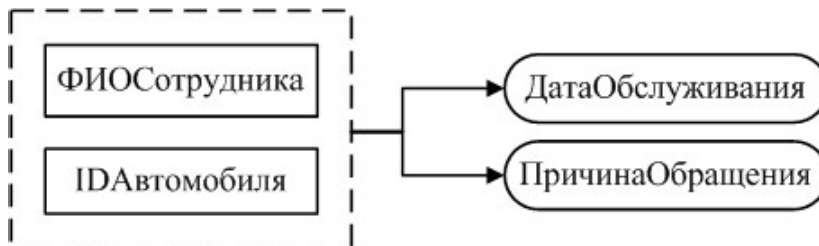


Рисунок 16 – Отношение «Обслуживание»

Все отношения находятся в 1НФ, так как на пересечении любой строки и столбца в отношениях находится ровно одно значение и так как все они имеют ключевые поля, которые по определению являются уникальными индексами.

Все отношения находятся во 2НФ, так как они находятся в 1НФ и каждый неключевой атрибут функционально полно зависит от потенциального ключа.

Все отношения находятся во 3НФ, так как они находятся во 2НФ и в них нет транзитивных зависимостей неключевых атрибутов от потенциального ключа.

Все отношения находятся в БКНФ, так как они находятся в 3НФ и детерминанты всех функциональных зависимостей являются потенциальными ключами.

4.2.7. Пересмотр ER-диаграммы

Пересмотр ER-диаграммы не требуется, так как все отношения находятся в БКНФ и отсутствуют нераспределённые атрибуты.

4.2.8. Построение схемы БД

По результатам предыдущих семи этапов мы получили даталогическую модель нашей базы данных.

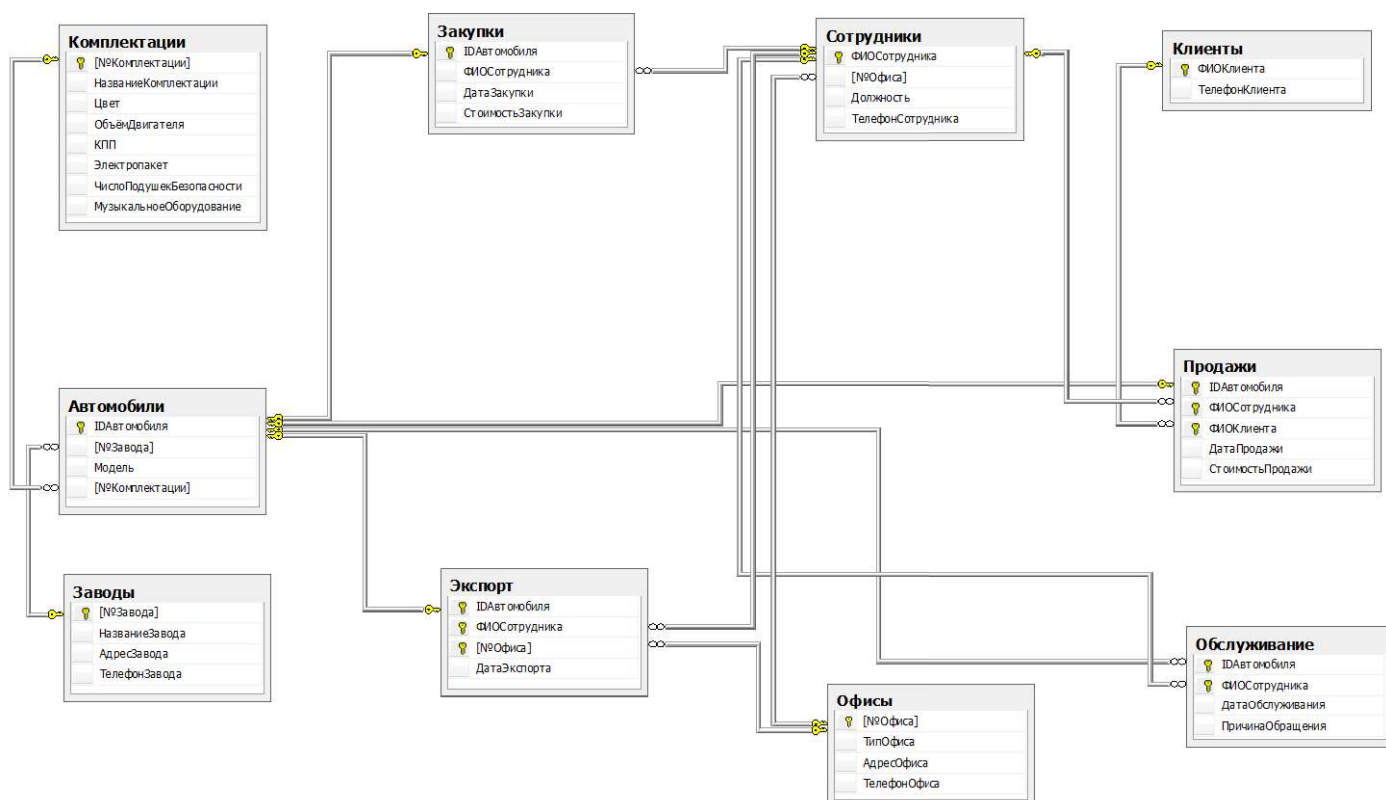


Рисунок 17 – Схема базы данных

4.3. Разработка БД и основных объектов БД.

4.3.1. Разработка БД и ограничений

Приведем пример создания базы данных Avto, которая будет включать ограничения первичных и внешних ключей, а также другие необходимые для пользователя ограничения:

```
create database Avto
on
```



```

(Name='Avto_Data',          --имя файла
Filename='C:\Avto_Data.mdf', --путь к файлу
Size=3,                    --начальный размер
Maxsize=20,                --максимальный размер
Filegrowth=1)              --шаг увеличения размера файла
log on
(Name='Avto_Log',
Filename='C:\Avto_Log.ldf',
Size=3,
Maxsize=20,
Filegrowth=1)
go

```

При создании таблиц в созданной базе данных описываются ограничения первичных и вторичных ключей для обеспечения ссылочной целостности базы данных.

Создание таблицы Офисы, содержащей следующую информацию: номер офиса (первичный ключ, столбец-счётчик), тип офиса (not null), адрес (not null), телефон:

```

create table Офисы([№Офиса] int identity(1, 1) primary key,
ТипОфиса nvarchar(50) not null, АдресОфиса nvarchar(50) not null,
ТелефонОфиса nvarchar(20))

```

Создание таблицы Сотрудники, содержащей следующую информацию: ФИО Сотрудника (первичный ключ), Номер офиса, в котором работает сотрудник (not null), должность (not null), телефон. Создаётся ограничение для связи поля Номер офиса с таблицей Офисы:

```

create table Сотрудники(ФИОСотрудника nvarchar(50) primary key,
[№Офиса] int not null, Должность nvarchar(50) not null,
ТелефонСотрудника nvarchar(20), constraint FK_Сотрудники_Офисы
foreign key ([№Офиса]) references Офисы([№Офиса]))

```

Создание таблицы Заводы, содержащей следующую информацию: номер завода (первичный ключ, столбец-счётчик), название завода (not null и уникально), адрес завода, телефон:

```

create table Заводы([№Завода] int identity(1, 1) primary key,
НазваниеЗавода nvarchar(10) not null unique, АдресЗавода
nvarchar(50), ТелефонЗавода nvarchar(20))

```

Создание таблицы Комплектации, содержащей следующую информацию: Номер комплектации (первичный ключ, столбец-счётчик), название комплектации (not null и уникально), цвет (not null), объём двигателя (not null), коробка передач (not null), электропакет (not null), число подушек безопасности (not null), музыкальное оборудование (not null):

```

create table Комплектации([№Комплектации] int identity(1, 1)
primary key, НазваниеКомплектации nvarchar(20) not null unique, Цвет
nvarchar(15) not null, ОбъёмДвигателя real not null, КПП nvarchar(20)
not null, Электропакет nvarchar(20) not null,
ЧислоПодушекБезопасности int not null, МузыкальноеОборудование
nvarchar(20) not null)

```

Создание таблицы Автомобили, содержащей помимо приведённой ниже информации ограничения для связи полей Номер завода и Номер комплектации с таблицами Заводы и Комплектации:

```

create table Автомобили(IDАвтомобиля int primary key, [№Завода]
int not null, Модель nvarchar(20) not null, [№Комплектации] int not
null constraint FK_Автомобили_Заводы foreign key ([№Завода])
references Заводы([№Завода]),
constraint FK_Автомобили_Комплектации foreign key
([№Комплектации]) references Комплектации([№Комплектации]))

```

Создание таблицы Клиенты:

```

create table Клиенты (ФИОКлиента nvarchar(50) primary key,
ТелефонКлиента nvarchar(20))

```

Создание таблицы Закупки, содержащей ограничения для связи полей IDАвтомобиля, ФИО Сотрудника с базовыми таблицами:

```

create table Закупки(IDАвтомобиля int unique, ФИОСотрудника
nvarchar(50), ДатаЗакупки datetime not null, СтоимостьЗакупки money
not null,
constraint PK_Закупки primary key (IDАвтомобиля),
constraint FK_Закупки_Автомобили foreign key (IDАвтомобиля)
references Автомобили(IDАвтомобиля),
constraint FK_Закупки_Сотрудники foreign key (ФИОСотрудника)
references Сотрудники(ФИОСотрудника))

```

Создание таблицы Экспорт, содержащей ограничения для связи полей IDАвтомобиля, ФИО Сотрудника и Номер офиса с базовыми таблицами:

```

create table Экспорт(IDАвтомобиля int unique, ФИОСотрудника
nvarchar(50), [№Офиса] int not null, ДатаЭкспорта datetime not null,
constraint PK_Экспорт primary key (ФИОСотрудника, [№Офиса],
IDАвтомобиля),
constraint FK_Экспорт_Сотрудники foreign key (ФИОСотрудника)
references Сотрудники(ФИОСотрудника),
constraint FK_Экспорт_Офисы foreign key ([№Офиса]) references
Офисы([№Офиса]),
constraint FK_Экспорт_Автомобили foreign key (IDАвтомобиля)
references Автомобили(IDАвтомобиля))

```

Создание таблицы Продажи, содержащей ограничения для связи полей IDАвтомобиля, ФИО Сотрудника и ФИО клиента с базовыми таблицами:

```

create table Продажи(IDАвтомобиля int unique, ФИОСотрудника
nvarchar(50), ФИОКлиента nvarchar(50), ДатаПродажи datetime not null,
СтоимостьПродажи money not null,
constraint PK_Продажи primary key (ФИОСотрудника, IDАвтомобиля,
ФИОКлиента),
constraint FK_Продажи_Сотрудники foreign key (ФИОСотрудника)
references Сотрудники(ФИОСотрудника),
constraint FK_Продажи_Автомобили foreign key (IDАвтомобиля)
references Автомобили(IDАвтомобиля),

```

```
constraint FK_Продажи_Клиенты foreign key (ФИОКлиента) references
Клиенты(ФИОКлиента)
```

Создание таблицы Обслуживание, содержащей ограничения для связи полей IDАвтомобиля, ФИО Сотрудника с базовыми таблицами:

```
create table Обслуживание(IDАвтомобиля int, ФИОСотрудника
nvarchar(50), ДатаОбслуживания datetime not null, ПричинаОбращения
nvarchar(50),
constraint PK_Обслуживание primary key (ФИОСотрудника,
IDАвтомобиля),
constraint FK_Обслуживание_Сотрудники foreign key (ФИОСотрудника)
references Сотрудники(ФИОСотрудника),
constraint FK_Обслуживание_Автомобили foreign key (IDАвтомобиля)
references Автомобили(IDАвтомобиля))
```

Таким образом, в нашей базе данных содержится десять таблиц, что соответствует требованиям, предъявляемым к данной курсовой работе.

4.3.2. Разработка правил

Для ограничения вводимых пользователем значений было разработано правило «Больше нуля»:

```
create rule БольшеНуля as @x>0
```

Данное правило было установлено для тех полей таблиц базы данных, величина которых не может принимать отрицательные значения (это номер офиса, объём двигателя, число подушек безопасности, номер завода, номер комплектации, стоимость закупки и продажи):

```
exec sp_bindrule 'БольшеНуля', 'Сотрудники.№Офиса'
exec sp_bindrule 'БольшеНуля', 'Комплектации.ОбъёмДвигателя'
exec sp_bindrule 'БольшеНуля',
'Комплектации.ЧислоПодушекБезопасности'
exec sp_bindrule 'БольшеНуля', 'Автомобили.№Завода'
exec sp_bindrule 'БольшеНуля', 'Автомобили.№Комплектации'
exec sp_bindrule 'БольшеНуля', 'Закупки.СтоимостьЗакупки'
exec sp_bindrule 'БольшеНуля', 'Экспорт.№Офиса'
exec sp_bindrule 'БольшеНуля', 'Продажи.СтоимостьПродажи'
```